



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA SOFTWARE

**DESARROLLO DE UNA APLICACIÓN WEB Y APLICACIÓN
MÓVIL PARA LA GESTIÓN DE INFORMACIÓN DE PROCESOS
ADMINISTRATIVOS DE LA ESCUELA DE CONDUCCIÓN
“CONDUESPOCH” APLICANDO EL FRAMEWORK REACT
BAJO UN ENTORNO DE PRUEBAS**

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO EN SISTEMAS INFORMÁTICOS

AUTORES:

WILMER ALEXANDER ALDAS PANIMBOZA

CLAUDIO LENIN PILATAXI YAUCÁN

Riobamba – Ecuador

2021



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA SOFTWARE

**DESARROLLO DE UNA APLICACIÓN WEB Y APLICACIÓN
MÓVIL PARA LA GESTIÓN DE INFORMACIÓN DE PROCESOS
ADMINISTRATIVOS DE LA ESCUELA DE CONDUCCIÓN
“CONDUESPOCH” APLICANDO EL FRAMEWORK REACT
BAJO UN ENTORNO DE PRUEBAS**

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO EN SISTEMAS INFORMÁTICOS

AUTORES: WILMER ALEXANDER ALDAS PANIMBOZA

CLAUDIO LENIN PILATAXI YAUCÁN

DIRECTOR: ING. DANILO MAURICIO PASTOR RAMÍREZ PhD.

Riobamba – Ecuador

2021

© 2021, **Wilmer Alexander Aldas Panimboza & Claudio Lénin Pilataxi Yaucán**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica el documento, siempre y cuando se reconozca el Derecho de Autor.

Nosotros, WILMER ALEXANDER ALDAS PANIMBOZA y CLAUDIO LENIN PILATAXI YAUCÁN, declaramos que el presente Trabajo de Integración Curricular es de nuestra autoría y los resultados de este son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

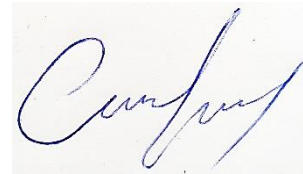
Como autores asumimos la responsabilidad legal y académica de los contenidos de este trabajo de Integración Curricular; El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 05 de noviembre de 2021



Wilmer Alexander Aldas Panimboza

180458328-2



Claudio Lenin Pilataxi Yaucán

060483876-3

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA SOFTWARE

El Tribunal de Trabajo de Integración Curricular certifica que: El trabajo de Integración Curricular: Tipo: Proyecto Técnico, “**DESARROLLO DE UNA APLICACIÓN WEB Y APLICACIÓN MÓVIL PARA LA GESTIÓN DE INFORMACIÓN DE PROCESOS ADMINISTRATIVOS DE LA ESCUELA DE CONDUCCIÓN “CONDUESPOCH” APLICANDO EL FRAMEWORK REACT BAJO UN ENTORNO DE PRUEBAS**”, realizado por los señores: **WILMER ALEXANDER ALDAS PANIMBOZA** y **CLAUDIO LENIN PILATAXI YAUCÁN**, ha sido minuciosamente revisado por los miembros del Tribunal de Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

FIRMA

FECHA

Ing. Ivonne Elizabeth Rodríguez Flores PhD.



Firmado electrónicamente por:
**IVONNE ELIZABETH
RODRIGUEZ FLORES**

PRESIDENTE DEL TRIBUNAL

2021-11-05

Ing. Danilo Mauricio Pastor Ramírez PhD.

Firmado digitalmente
Por **DANILO MAURICIO
PASTOR RAMIREZ**
Fecha: 2021.11.30
12:41:22 -05'00'

**DIRECTOR DE TRABAJO DE
INTEGRACIÓN CURRICULAR**

2021-11-05

Ing. Raúl Hernán Rosero Miranda PhD.



Firmado electrónicamente por:
**RAUL HERNAN
ROSERO
MIRANDA**

MIEMBRO DEL TRIBUNAL

2021-11-05

DEDICATORIA

El presente trabajo va dedicado a Dios por permitirme culminar esta etapa de mi vida, a mi familia por brindarme todo el apoyo requerido en mi carrera estudiantil, a mis maestros por ayudarme a formarme mejor como persona y que aportaron con sus conocimientos para poder lograr mis objetivos académicos.

Claudio.

A Dios por darme la vida y permitirme vivir estas maravillosas experiencias, a mis padres que han sacrificado tanto por mí, a mis maestros que formaron parte de mi vida estudiantil, y que aportaron con sus conocimientos para poder cumplir una etapa de mi vida.

Wilmer.

AGRADECIMIENTO

El más sincero agradecimiento a la Escuela Superior Politécnica de Chimborazo, por darnos la oportunidad de obtener una profesión y ser una ayuda para la sociedad y al Ingeniero Danilo Pastor que nos ha guiado en el transcurso de este trabajo.

Claudio & Wilmer

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS.....	xiii
ÍNDICE DE GRÁFICOS.....	xiii
ÍNDICE DE ANEXOS.....	xv
ÍNDICE DE ABREVIATURAS.....	xvi
RESUMEN.....	xvii
ABSTRACT.....	xviii
INTRODUCCION.....	1

CAPÍTULO I

1.	DIAGNÓSTICO DEL PROBLEMA.....	2
1.1.	Formulación del problema.....	4
1.2.	Sistematización del problema.....	4
1.3.	Justificación.....	4
1.3.1.	<i>Justificación teórica</i>	4
1.3.2.	<i>Justificación aplicativa</i>	5
1.4.	Objetivos.....	7
1.4.1.	<i>Objetivo General</i>	7
1.4.2.	<i>Objetivos Específicos</i>	7

CAPÍTULO II

2.	FUNDAMENTOS TEÓRICOS.....	8
2.1.	Estado del arte.....	8
2.2.	Procesos departamentales de CONDUESPOCH.....	10
2.2.1.	<i>Procesos académicos</i>	10
2.2.2.	<i>Procesos de gestión de recursos para el manejo vehicular</i>	12
2.2.3.	<i>Procesos de desarrollo y mantenimiento de aplicaciones</i>	16
2.2.4.	<i>Proceso de gestión de informes y notificaciones</i>	17
2.2.5.	<i>Procesos de gestión de sílabos, docentes e instructores</i>	19
2.2.6.	<i>Procesos de gestión de ventas y pagos</i>	22

2.3.	Herramientas de desarrollo	27
2.3.1.	<i>Visual Studio Code</i>	27
2.3.2.	<i>Postgresql</i>	28
2.3.3.	<i>Arquitectura Modelo-Vista-Controlador (MVC)</i>	29
2.3.4.	<i>Lenguajes de programación para interfaces web</i>	30
2.3.5.	<i>Frameworks para el desarrollo de interfaces web</i>	31
2.3.6.	<i>Tecnologías para desarrollar servicios web</i>	33
2.3.7.	<i>Framework React JS</i>	34
2.3.8.	<i>Framework Laravel</i>	36
2.3.9.	<i>Tipos de aplicaciones móviles</i>	37
2.3.9.1.	<i>Ventajas y desventajas de los diferentes tipos de aplicaciones</i>	38
2.3.10.	<i>Frameworks de desarrollo de aplicaciones móviles multiplataforma</i>	39
2.3.11.	<i>Framework React Native</i>	40
2.3.12.	<i>Biblioteca para el desarrollo de interfaces</i>	41
2.3.13.	<i>Administrador de paquetes de Node (NPM)</i>	41
2.3.14.	<i>Librería axios</i>	42
2.3.15.	<i>Librería redux</i>	42
2.3.15.1.	<i>Arquitectura de datos de Redux</i>	43
2.3.15.2.	<i>Uso de redux</i>	45
2.3.16.	<i>Librería jest</i>	46
2.4.	Entorno de pruebas de software	48
2.4.1.	<i>Principales beneficios de las pruebas de software</i>	48
2.4.2.	<i>Verificación y validación</i>	49
2.4.3.	<i>Objetivos del testing</i>	49
2.4.4.	<i>Niveles de prueba</i>	50
2.4.5.	<i>Soporte de herramientas para pruebas de ciclo de vida</i>	51
2.4.6.	<i>Metodología de pruebas</i>	53
2.4.7.	<i>Artefactos de prueba para la identificación de casos de prueba</i>	55
2.4.8.	<i>Artefactos de prueba para el diseño y construcción de casos de prueba</i>	57
2.4.9.	<i>Artefactos de prueba para ejecución y comparación de casos de prueba</i>	59
2.4.10.	<i>Pruebas de caja negra</i>	59
2.4.11.	<i>Pruebas de caja blanca</i>	61
2.4.12.	<i>Pruebas unitarias</i>	62
2.4.12.1.	<i>Motivos para realizar un test unitario</i>	62
2.4.12.2.	<i>Organizar-Actuar-Afirmar casos de prueba</i>	63
2.4.12.3.	<i>Cobertura de código</i>	63

2.4.12.4.	<i>Ejecución de las pruebas unitarias</i>	65
2.5.	Metodología de desarrollo ágil SCRUM	65
2.5.1.	<i>Características de Scrum</i>	66
2.5.2.	<i>Ventajas de usar Scrum</i>	66
2.5.3.	<i>Roles Scrum</i>	66
2.5.4.	<i>Artefactos de SCRUM</i>	67
2.6.	Norma ISO/IEC 25010	69

CAPÍTULO III

3.	MARCO METODOLOGICO	70
3.1.	Introducción	70
3.2.	Diseño de la investigación	700
3.2.1.	<i>Población</i>	700
3.2.2.	<i>Muestra</i>	70
3.2.3.	<i>Métodos de investigación</i>	71
3.2.4.	<i>Técnicas de investigación</i>	72
3.2.5.10.	<i>Método utilizado para evaluar la eficiencia de tiempo del sistema</i>	72
3.3.	Aplicación de la metodología Ágil SCRUM	73
3.3.1.	<i>Estudio preliminar</i>	73
3.3.2.	<i>Fase de planificación</i>	73
3.3.2.1.	<i>Planificación</i>	73
3.3.2.2.	<i>Requerimientos (Product backlog)</i>	74
3.3.2.3.	<i>Alcance del proyecto</i>	75
3.3.2.4.	<i>Limitaciones</i>	75
3.3.2.5.	<i>Especificaciones para los requerimientos</i>	76
3.3.2.6.	<i>Personal involucrado y roles establecidos en el proyecto</i>	79
3.3.2.7.	<i>Roles establecidos para el sistema</i>	80
3.3.2.8.	<i>Sprint Backlog</i>	81
3.3.2.9.	<i>Identificación del plan de pruebas de software</i>	84
3.3.3.	<i>Fase de desarrollo</i>	105
3.3.3.1.	<i>Estándar de codificación y diseño de base de datos</i>	105
3.3.3.2.	<i>Arquitectura del sistema web y móvil</i>	106
3.3.3.3.	<i>Diagramas de casos de uso</i>	107
3.3.3.4.	<i>Diseño de la interfaz web del usuario</i>	119
3.3.3.5.	<i>Diseño de la base de datos</i>	124

3.3.3.6.	<i>Categorización de riesgos</i>	127
3.3.3.7.	<i>Historias de usuario</i>	130
3.3.3.8.	<i>Tareas de ingeniería</i>	132
3.3.3.9.	<i>Historias técnicas</i>	136
3.3.3.10.	<i>Diseño y construcción de casos de prueba</i>	137
3.3.4.	<i>Fase de cierre</i>	140
3.3.4.1.	<i>BurnDown Chart</i>	141

CAPÍTULO IV

4.	RESULTADOS	143
4.1.	Ejecución y resultados de casos de prueba	143
4.1.1.	<i>Alcance de las pruebas</i>	143
4.1.2.	<i>Cobertura de código</i>	144
4.1.3.	<i>Resumen de pruebas</i>	149
4.2.	Tiempos de respuesta	150
4.2.1.	<i>Procesos actuales</i>	151
4.2.2.	<i>Procesos automatizados</i>	152
4.2.3.	<i>Comparación de tiempos</i>	153
4.3.	Pruebas estadísticas	153
4.3.1.	<i>Análisis de normalidad</i>	153
4.3.2.	<i>Test T-Student para datos relacionados</i>	154
4.3.3.	<i>Toma de decisiones</i>	156
	CONCLUSIONES	158
	RECOMENDACIONES	160

GLOSARIO

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla 1-2:	Aporte al trabajo de integración curricular.....	10
Tabla 2-2:	Comparación de Frameworks de FrontEnd	32
Tabla 3-2:	Comparación de lenguajes de BackEnd	33
Tabla 4-2:	Ventajas y desventajas de los diferentes tipos de aplicaciones móviles.....	38
Tabla 5-2:	Frameworks de desarrollo de aplicaciones móviles multiplataforma.....	39
Tabla 1-3:	Tareas a evaluar para determinar la eficiencia.....	72
Tabla 2-3:	Determinación del criterio de evaluación	74
Tabla 3-3:	Puntos estimados según tallas de Product Backlog	75
Tabla 4-3:	Historias de usuario del sistema	77
Tabla 5-3:	Historias técnicas del sistema	79
Tabla 6-3:	Personal involucrado y roles establecidos en el desarrollo del proyecto.....	80
Tabla 7-3:	Roles establecidos para el uso del sistema	81
Tabla 8-3:	Planificación para el desarrollo del proyecto	82
Tabla 9-3:	Especificación del alcance de las pruebas	87
Tabla 10-3:	Especificación de los ítems a probar	87
Tabla 11-3:	Definición de casos de prueba por historia de usuario	87
Tabla 12-3:	Estrategia de prueba implementada.....	103
Tabla 13-3:	Calendario para la automatización de los casos de prueba.....	104
Tabla 14-3:	Tabla de mitigación de riesgos de los casos de prueba	105
Tabla 15-3:	Responsable para llevar a cabo las tareas de testing.....	106
Tabla 16-3:	Estándar de codificación y diseño base de datos.....	107
Tabla 17-3:	Probabilidad de ocurrencia de riesgos	128
Tabla 18-3:	Impacto de riesgo	128
Tabla 19-3:	Exposición del riesgo	129
Tabla 20-3:	Priorización de riesgos del proyecto.....	129
Tabla 21-3:	Hoja de gestión de riesgo	130
Tabla 22-3:	Autenticación de usuarios por número de cédula y contraseña.....	131
Tabla 23-3:	Prueba de aceptación PA-01 de HU-01	132
Tabla 24-3:	Prueba de aceptación PA-02 de HU-01	132
Tabla 25-3:	Prueba de aceptación PA-03 de HU-01	132
Tabla 26-3:	Prueba de aceptación PA-04 de HU-01	133
Tabla 27-3:	Crear el controlador de autenticación de usuarios.....	133
Tabla 28-3:	Prueba de aceptación PA-05 de HU-01	134

Tabla 29-3:	Crear el modelo para la autenticación de usuarios	134
Tabla 30-3:	Prueba de aceptación PA-06 de HU-01	134
Tabla 31-3:	Prueba de aceptación PA-07 de HU-01	135
Tabla 32-3:	Crear el API REST de la autenticación de usuarios	135
Tabla 33-3:	Prueba de aceptación PA-08 de HU-01	135
Tabla 34-3:	Prueba de aceptación PA-09 de HU-01	136
Tabla 35-3:	Prueba de aceptación PA-10 de HU-01	136
Tabla 36-3:	Desarrollo del Frontend para la autenticación de usuarios	137
Tabla 37-3:	Prueba de aceptación PA-11 de HU-01	137
Tabla 38-3:	Prueba de aceptación PA-12 de HU-01	137
Tabla 39-3:	Historia técnica	138
Tabla 40-3:	Caso de prueba	139
Tabla 41-3:	Actividades para la conclusión del proyecto	142
Tabla 1-4:	Resultado de cobertura de código por requerimientos	146
Tabla 2-4:	Resultados de casos de prueba	150
Tabla 3-4:	Tiempo requerido para realizar una actividad de forma manual	153
Tabla 4-4:	Tiempo requerido para realizar una actividad de forma automatizada.....	153
Tabla 5-4:	Diferencia de tiempos entre la forma manual y automatizada	154
Tabla 6-4:	Valores para la prueba T-Student	156

ÍNDICE DE FIGURAS

Figura 1-2:	Patrón modelo vista controlador	30
Figura 2-2:	Estructura de una función en React.....	35
Figura 3-2:	Estructura de una clase en React.....	35
Figura 4-2:	Estado de los componentes en React	36
Figura 5-2:	Estado solo de lectura	44
Figura 6-2:	Acciones en React.....	44
Figura 7-2:	Funciones reductoras en React.....	45
Figura 8-2:	Estructura de un script de prueba con Jest	47
Figura 9-2:	Cálculo de cobertura de código.....	63
Figura 10-2:	Ejemplo de un método parcialmente cubierto por una prueba.....	64
Figura 11-2:	Cobertura total de un caso de prueba.....	64
Figura 12-2:	Proceso de la metodología SCRUM	65
Figura 13-2:	Product Backlog.....	68
Figura 1-3:	Planificación para el desarrollo del proyecto.....	75
Figura 2-3:	Arquitectura del sistema web y móvil.....	107
Figura 3-3:	Interfaz de la página principal de la aplicación web	121
Figura 4-3:	Interfaz para la solicitud de inscripciones	122
Figura 5-3:	Interfaz base para las vistas de todos los procesos.....	123
Figura 6-3:	Interfaz principal de la aplicación móvil.....	123
Figura 7-3:	Interfaz de solicitud de examen psicosenométrico	124
Figura 8-3:	Interfaz de solicitudes pendientes de atender.....	124
Figura 9-3:	Diccionario de datos de la tabla tarchivo	127
Figura 1-4:	Estructura de una prueba unitaria.....	145
Figura 2-4:	Cobertura de código de los actions creators del sistema.....	148
Figura 3-4:	Cobertura de código de las funciones reductoras del sistema.....	149
Figura 4-4:	Casos de prueba ejecutados en la aplicación web.....	150
Figura 5-4:	Casos de prueba ejecutados en la aplicación móvil	150
Figura 6-4:	Test Shapiro-Wilk del tiempo sin utilizar el sistema	155
Figura 7-4:	Test Shapiro Wilk del tiempo al utilizar el sistema	155
Figura 8-4:	Prueba T-Student pareado.....	156

ÍNDICE DE GRÁFICOS

Gráfico 1-2:	Proceso de Inscripción.....	11
Gráfico 2-2:	Proceso de Matrícula.....	12
Gráfico 3-2:	Proceso de Movimiento de vehículos de uso administrativo	13
Gráfico 4-2:	Proceso de Movimiento de vehículos de uso institucional.....	14
Gráfico 5-2:	Proceso de Gestión de control vehicular	14
Gráfico 6-2:	Proceso de solicitar requerimientos del departamento de asesoría vial.....	15
Gráfico 7-2:	Proceso de Gestión de rutas.....	16
Gráfico 8-2:	Proceso de Desarrollo e implementación de aplicaciones.....	17
Gráfico 9-2:	Proceso de Mantenimiento del departamento de DTIC.....	17
Gráfico 10-2:	Proceso de Revisión de las aulas de clase	18
Gráfico 11-2:	Proceso previo a los exámenes de grado	18
Gráfico 12-2:	Proceso de gestión de sílabos académicos y planes de estudio de docentes	19
Gráfico 13-2:	Proceso de gestión de sílabos académicos y plan de estudio de instructores ...	20
Gráfico 14-2:	Proceso de capacitación y apoyo a instructores.....	20
Gráfico 15-2:	Proceso de preparación de capacitación y apoyo a docentes.....	21
Gráfico 16-2:	Proceso de control de instructores.....	22
Gráfico 17-2:	Proceso de pago de servicios a docentes	23
Gráfico 18-2:	Proceso de pago de servicios a instructores	24
Gráfico 19-2:	Proceso de pago de mensualidad de estudiantes	25
Gráfico 20-2:	Proceso de venta de solicitudes generales a personas particulares.....	25
Gráfico 21-2:	Proceso de venta de solicitudes generales a estudiantes.....	26
Gráfico 22-2:	Proceso de venta de solicitudes de justificación a estudiantes	27
Gráfico 23-2:	Generalidades de Visual Studio Code	28
Gráfico 24-2:	Arquitectura de datos en Redux.....	43
Gráfico 25-2:	Diagrama de las funciones reductoras.....	45
Gráfico 26-2:	Árbol de componentes de React	46
Gráfico 27-2:	React con Redux.....	46
Gráfico 28-2:	Fases de desarrollo y testing en el modelo V	51
Gráfico 29-2:	Herramientas necesarias en el ciclo de vida de desarrollo software.....	52
Gráfico 30-2:	Actividades del ciclo de vida al desarrollar casos de prueba	54
Gráfico 31-2:	Gráfica de avance de proyectos (Burndown Chart).....	69
Gráfico 1-3:	Casos de uso del usuario administrador	110
Gráfico 2-3:	Casos de uso del directivo de secretaría	111

Gráfico 3-3:	Casos de uso del directivo de contabilidad/tesorería.....	113
Gráfico 4-3:	Casos de uso del directivo de inspección	114
Gráfico 5-3:	Casos de uso del directivo de pedagogía.....	115
Gráfico 6-3:	Casos de uso del directivo de asesoría vial	116
Gráfico 7-3:	Casos de uso del directivo de gerencia.....	116
Gráfico 8-3:	Diagrama de casos de uso del directivo de servicio	117
Gráfico 9-3:	Casos de uso para el practicante de DTIC.....	117
Gráfico 10-3:	Casos de uso del estudiante	118
Gráfico 11-3:	Casos de uso del docente.....	119
Gráfico 12-3:	Casos de uso del instructor	119
Gráfico 13-3:	Casos de uso de solicitud de examen psicosenométrico.....	119
Gráfico 14-3:	Casos de uso para la solicitud del examen psicosenométrico.....	120
Gráfico 15-3:	Diagrama lógico de la base de datos del sistema web.....	126
Gráfico 16-3:	Diagrama lógico de la base de datos del sistema móvil	127
Gráfico 17-3:	Avance del proyecto (BurnDownChart).....	141
Gráfico 1-4:	Cobertura de código por Sprints.....	151
Gráfico 2-4:	Prueba T-Student pareado gráfico	157
Gráfico 3-4:	Tiempos de ejecución de procesos con aplicación y sin aplicación	158

ÍNDICE DE ANEXOS

ANEXO A: HISTORIAS DE USUARIO DEL PROYECTO

ANEXO B: GESTIÓN DE RIESGOS DEL PROYECTO

ANEXO C: MANUAL DE USUARIO

ANEXO D: DICCIONARIO DE DATOS

ANEXO E: HISTORIAS TÉCNICAS DEL PROYECTO

ÍNDICE DE ABREVIATURAS

SQA	Aseguramiento de calidad de software
UML	Lenguaje de modelado unificado
API	Interfaz de Programación de Aplicaciones
IDE	Entorno de desarrollo integrado
BPM	Sistema de Gestión de Procesos
ISO	Organización Internacional de Normalización
IU	Interfaz de Usuario
DOM	Modelo de Objetos del Documento
HTML	Lenguaje de Marcas de Hipertexto
TI	Tecnologías de la Información
SI	Sistemas de Información
CMS	Sistema de Gestión de Contenidos
CSS	Hojas de Estilo en Cascada
ES6+	ECMAScript 6
JSX	JavaScript XML
AJAX	JavaScript Asíncrono y XML
XML	Lenguaje de Etiquetado Extensible
CLI	Interfaz de Línea de Comandos
KB	KiloByte
GUI	Interfaz Gráfica de Usuario
QA	Aseguramiento de la calidad
EP	Empresa Pública
EIS	Escuela de Ingeniería en Sistemas
ESPOCH	Escuela Superior Politécnica de Chimborazo.
MVC	Modelo Vista Controlador.
DTIC	Departamento de Tecnologías de la Información y Comunicación.
GIFS	Formato de Intercambio de Gráficos
ASP	Active Server Pages
PHP	Hypertext Preprocessor
SQL	Lenguaje de Consulta Estructurada
ACID	Atomicidad, Consistencia, Aislamiento, Durabilidad
REST	Transferencia de estado representacional

RESUMEN

El presente trabajo de integración curricular tuvo como objetivo la creación de una aplicación web y móvil para la gestión de información de la escuela de conducción CONDUESPOCH E.P para mejorar la eficiencia de tiempo dentro de dicha institución. Las tecnologías que se utilizaron para la realización del sistema fueron: el framework React JS, React Native, Postgresql, Jest para la automatización de casos de prueba, todo esto siguiendo la metodología SCRUM ya que permite tener una mejor interacción con el cliente durante el proceso de desarrollo, para lo cual se obtuvieron 15 Sprints y un total de 1752 puntos estimados, lo que se traduce en 98 historias de usuario y 11 tareas de ingeniería. Una vez finalizada la aplicación se procedió con la capacitación a los usuarios, así como también se realizó la evaluación de la eficiencia del sistema por medio de la norma ISO-25010 tomando en cuenta los tiempos necesarios para realizar tareas determinadas mediante el uso del método manual y el método automatizado, para esto se utilizó el test de Shapiro Wilk y el test de T-Student, dando como resultado una reducción del tiempo en un 92.07%, es decir el tiempo empleado disminuyó considerablemente. Una vez que se han realizado todos los estudios de eficiencia se concluye que al utilizar el método automatizado los usuarios pueden disponer de más tiempo para realizar otras tareas, tiempo que era ocupado en realizar procesos manuales. Se recomienda utilizar diferentes lenguajes de programación en la parte de servidor como NodeJs para que todo el sistema sea desarrollado completamente en Javascript, además utilizar una metodología de pruebas distinta a la de este proyecto y comparar los resultados de las pruebas.

Palabras clave: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA> <INGENIERÍA DE SOFTWARE> <GESTIÓN DE PROCESOS> <METODOLOGÍA SCRUM> <FRAMEWORK REACT>



2012-DBRA-UPT-2021


2021-10-28

ABSTRACT

The objective of the current curriculum integration work was to create a web and mobile application for information management of the CONDUESPOCH E.P driving school to improve time efficiency within the institution. The technologies that were used for the development of the system were: the React JS, React Native, Postgresql, Jest framework for the automation of test cases, following the SCRUM methodology because it allows to have a better interaction with the client during the process development; thus, 15 Sprints and a total of 1752 estimated points were obtained, which means 98 user stories and 11 engineering tasks. Once the application was completed, we trained the users and, the evaluation of the system efficiency was carried out, utilizing the ISO-25010 standard, taking into consideration the times necessary to perform certain tasks using the manual method. and the automated method; for this we utilized the Shapiro Wilk test and the T-Student test, resulting in a reduction of 92.07% in the time which means that the time spent decreased considerably. Once all the efficiency studies have been carried out, it is concluded that by using the automated method, users can have more time to do other tasks, time that was occupied in carrying out manual processes. It is recommended to use different programming languages on the server such as NodeJs so that the entire system is fully developed in Javascript, besides, it is recommended to utilize a different test methodology from this project and compare the test results.

Key words: <TECHNOLOGY AND ENGINEERING SCIENCE> <SOFTWARE ENGINEERING> <PROJECT MANAGEMENT <SCRUM METHODOLOGY> <REACT FRAMEWORK>.

DIANA
CAROLINA
CAMPAÑA DIAS



Firmado digitalmente
por DIANA CAROLINA
CAMPAÑA DIAS
Fecha: 2021.12.03
09:54:29 -05'00'

INTRODUCCIÓN

La tecnología se ha convertido en la ayuda para el desarrollo de la sociedad, siendo de esta manera que los procesos repetitivos que antes se llevaban a cabo por el personal humano han sido reemplazados por maquinas, los cuales aparte de ser más eficientes en costes de tiempo y dinero, han permitido la creación de nuevas formas de trabajo que antes eran impensables. Una de ellas es la gestión de datos referentes a los procesos que tiene una determinada empresa o institución, los cuales con la ayuda de sistemas automatizados alojados en internet se los puede monitorear desde cualquier parte del mundo. La gran mayoría de los entornos de trabajo en la actualidad ven el uso de las herramientas tecnológicas, principalmente las que se desarrollan a través de internet, como una oportunidad de competencia en una determinada industria, por lo que se ven obligados a transformarse como entes digitales, sin embargo, para que una herramienta sea lo suficientemente útil, debe cumplir con que éste presente características únicas que lo hagan destacar del resto, a esto se le conoce como herramientas hechas a la medida. La funcionalidad más básica de estas herramientas son la recopilación, procesamiento y entrega de datos de una determinada actividad.

Hoy en día las herramientas de procesamiento de datos más utilizados son los sistemas web, que permiten interactuar con muchos usuarios al mismo tiempo a diferencia de otros tipos de sistemas, además el acceso se lo puede realizar desde cualquier dispositivo como por ejemplo computadoras, smartphones, entre otros con acceso a internet. Por otro lado, el desarrollo de estas requiere de mucho tiempo y recursos por lo que es necesario emplear una planificación que permita gestionar desde la propuesta del producto hasta su respectiva etapa de prueba. En el desarrollo de software la etapa que más tiempo y recursos consume es la de las pruebas, ya que los errores que se detecten aquí se podrán solucionar o prevenir antes de ponerlo en producción, sin embargo, esto no significa que no se vaya a encontrar otros más adelante, debido a esto las pruebas exhaustivas en todo el ciclo de vida de un sistema es muy importante.

Este trabajo está enfocado en implementar un sistema de gestión de datos, de los procesos administrativos más importantes que se llevan a cabo en la Escuela de Conducción Profesional “CONDUESPOCH E.P”, para lo cual en cada uno de los capítulos planteados se detallan todo lo referente a su desarrollo. En el Capítulo I se establece el diagnóstico del problema. En el Capítulo II los fundamentos teóricos, en el cual se recopila información necesaria relacionada a temas específicos de este trabajo. En el Capítulo III se presenta la creación, diseño y construcción del sistema, donde se define los diferentes módulos, y finalmente en el Capítulo IV se da a conocer el proceso llevado a cabo en la implantación y evaluación del sistema.

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA

Una de las finalidades de todo sistema automatizado es la de gestionar datos, procesarlos y ofrecer información útil en el menor tiempo posible, siguiendo un determinado proceso de ejecución; esto permite aumentar la productividad de una empresa o institución por lo cual su desarrollo e implementación deben ser cuidadosamente observados.

CONDUESPOCH se dedica a formar conductores profesionales consientes, íntegros, éticos, responsables comprometidos con la sociedad y el medio ambiente, a través de la generación, del conocimiento científico y tecnológico, con personal altamente capacitado buscando así contribuir con el mejoramiento de la seguridad vial del país. CONDUESPOCH forma parte de las más prestigiosas instituciones de conducción de la ciudad de Riobamba ubicado junto a la Facultad de Administración de Empresas de la Escuela Superior Politécnica de Chimborazo, dirección PANAMERICANA SUR Km 11/2.

El servicio que ofrece a la ciudadanía es la licencia profesional tipo C, para lo cual se requiere realizar un curso de 6 meses en los horarios de nocturna, vespertina y fines de semana con un total aproximado de 150 alumnos en cada periodo académico, 25 profesores y 10 miembros de personal administrativo.

La administración de CONDUESPOCH se divide en procesos que se realizan en diferentes departamentos los cuales se clasifican en:

- **Departamento de Secretaría:** Los principales procesos que realizan en este departamento son la inscripción y matrícula de estudiantes. Tanto el procedimiento de inscripción como de matrícula son muy similares, lo cual consiste en solicitud de información, entrega de documentos necesarios, realizar pruebas de aptitud y finalmente generar un informe impreso que son almacenados en archivadores de CONDUESPOCH, seccionados por carpetas relacionados a cada estudiante.
- **Departamento de Asesoría vial:** El departamento de asesoría vial está centrado principalmente en los procesos de asignación de vehículos tanto al personal administrativo como al personal de instrucción, control de instructores, control de vehículos y generación de rutas de recorrido de los vehículos. El proceso principal de este departamento es la asignación de vehículos a los miembros de la CONDUESPOCH, lo cual consiste en solicitar autorización de uso, aceptar aprobación del director de la institución, generar salvoconducto y ruta para el solicitante, asignar responsable de uso vehicular, generar horario de recorrido, generar informe de estado del vehículo y finalmente archivarlos en documentos impresos.

- **Departamento de DTIC:** El proceso principal del departamento de DTIC consiste en la gestión y desarrollo de software, siendo así que consiste en, receptar solicitud de requerimientos de software de los diferentes departamentos de la institución, aprobar la solicitud de requerimiento, recolección y análisis de información, diseño e implementación, instalación y pruebas. Únicamente las aplicaciones software son almacenados y procesados en los servidores de la misma institución el resto como solicitudes y manuales son almacenados en archivadores.
- **Departamento de Inspección:** El departamento de inspección está centrado principalmente en la gestión de las aulas de clases para los estudiantes y docentes. El proceso consiste en la verificación diaria de estas, realizar correcciones en caso de requerir, realizar informes de estado, aplicar medidas de solución dadas por el director institucional en caso de ser requerida.
- **Departamento de Pedagogía:** Este departamento está encargado de la gestión de planificaciones, sílabos académicos, metodologías de estudio y capacitaciones de los docentes. La principal actividad de este departamento consiste en el evaluación y supervisión del cumplimiento de los planes de estudio, lo cual consiste en solicitar información al docente, elaboración del silabo, elaboración de matriz de contenidos, revisión y aprobación del material por parte del director de la institución.
- **Departamento de Contabilidad y Tesorería:** Contabilidad y Tesorería son dos departamentos que se complementan debido a que se centran en administración financiera de la institución, los cuales son la gestión de procesos de compra y venta de solicitudes. Lo cual consiste en el llenado de la solicitud con el número de factura, envío al director de la institución para su debida aprobación y finalmente se archiva para la constancia de compra o venta.

Todos estos procesos de las diferentes áreas se lo realizan con herramientas ofimáticas como Microsoft Word, Excel y Power Point, lo que hace difícil tener un estándar de documentación, además que para su almacenamiento deben ser impresos en papel, resultando mucho más difícil tener un control adecuado de la información que se contiene y es más propenso a perderse, resultando en altos costos económicos y principalmente de tiempo. Otro punto que se destaca es que no se hace uso adecuado de las herramientas informáticas que se tiene, un ejemplo es el desaprovechamiento del servidor de base de datos.

En conclusión, se creará una aplicación web y aplicación móvil para automatizar la gestión de información de procesos en los diferentes departamentos administrativos de la escuela de conducción profesional CONDUESPOCH, permitiendo solventar principalmente el inconveniente de falta de eficiencia en tiempo de respuesta de los trámites que realiza cada

departamento, sirviendo además como una herramienta de mejor aprovechamiento de los recursos con que cuenta la institución y finalmente permitir un adecuado almacenamiento de la información.

1.1. Formulación del problema

¿La aplicación web y aplicación móvil mejorará la eficiencia de tiempo de ejecución de los diferentes procesos llevados a cabo en cada uno de los departamentos administrativos en la escuela de conducción profesional “CONDUESPOCH”?

1.2. Sistematización del problema

¿Cómo se realiza actualmente el proceso de gestión de información de procesos administrativos de la escuela de conducción profesional “CONDUESPOCH”?

¿Cuáles son los aspectos a tomarse en cuenta para determinar las herramientas tecnológicas de desarrollo de la aplicación web y aplicación móvil?

¿Cuál es el contenido a implementar en el desarrollo de la aplicación web y aplicación móvil?

¿Cuáles son los parámetros a evaluar para garantizar la eficiencia de la aplicación web y aplicación móvil?

1.3. Justificación

1.3.1. Justificación teórica

La tecnología se ha convertido en una gran revolución desde sus inicios, influyendo en todas las áreas de la vida diaria, como la salud, educación, entretenimiento, negocios, etc. Cada área es única, por lo tanto, cuentan con herramientas diferentes ya sean de hardware o software. Aunque los dispositivos hardware han disminuido en tamaño, su funcionalidad es la misma, recoger y mostrar datos, no sucede lo mismo con el software, que debido a las necesidades de los usuarios se han creado una infinidad con el objetivo de resolver problemas con un alto índice de rendimiento y velocidad de proceso.

La automatización de gestión de procesos mediante software dentro de una institución permite organizar y manejar datos mucho más rápido comparado con la forma tradicional. Todo esto permite administrar información de una manera más eficiente y adaptadas a las necesidades de los clientes. Por lo general los datos de información que se manejan están relacionados con los suministros, materiales, funcionarios, colaboradores, conocimiento, entre otros, a su vez todo esto permite clasificarlos en distintas áreas de control, los cuales son conocidos como departamentos. En el ámbito empresarial a las tareas de gestión de procesos se les conoce como BPM por sus siglas en inglés Business Process Management o en español Gestión de Procesos de Negocio lo

cual se refiere a un conjunto de elementos y metodologías que tiene por objeto, precisamente, una gestión sistemática e integrada de todos los departamentos de una empresa, así como las funciones y responsabilidades de cada empleado o colaborador dentro de la empresa. (Oliveira, 2017)

Tomando en cuenta todas estas consideraciones, se ha planteado desarrollar una aplicación web y aplicación móvil que funcione como un gestor de información, de acuerdo a los requerimientos del cliente, lo cual también incluye el uso de tecnologías más actuales que aquellas que son usadas en las aplicaciones genéricas. Por lo que la herramienta tecnológica a emplear para la creación del sistema es React JS, el cual es un Framework para la creación de la aplicación web, que se basa principalmente en el lenguaje de programación Javascript, además se utilizará Bootstrap 4 para la estilización de las interfaces de usuario, y HTML para la estructuración de cada componente del sitio web; por otro lado para la creación de la aplicación móvil la herramienta a emplear es React Native, que también es un Framework pero a diferencia de React JS, este permite crear aplicaciones multiplataforma orientadas a dispositivos móviles, es decir aplicaciones que son soportadas por diferentes entornos como Android y iOS, su desarrollo se basa principalmente en el manejo de componentes, con el lenguaje Javascript y una estructura de etiquetas similar a HTML. En cuanto al almacenamiento de datos, se empleará Postgres para la gestión de la base de datos y también el uso de visual studio code como IDE de desarrollo.

1.3.2 Justificación aplicativa

Los datos que son utilizados para la generación de información valiosa son de mucha importancia dentro de una empresa, ya que en su mayoría son utilizados para la toma de decisiones, así pues, los sistemas que realizan este tipo de trabajos también lo son. Sin embargo, los sistemas de mayor rendimiento son aquellos que mejor estructurados se encuentren y además estén automatizados, debido a que esto último permite una mayor competencia en la industria, ahorro de tiempo y recursos.

Con el presente proyecto, se pretende desarrollar un sistema de gestión de procesos administrativos, siendo la misma una aplicación web y aplicación móvil para su debido uso; esto con el fin de satisfacer las necesidades de las distintas áreas de la Escuela de Conducción Profesional CONDUESPOCH.

En términos generales los inconvenientes que se presentan en todas las áreas de la institución son, la deficiente generación de informes con datos históricos, debido a que el actual proceso llevado en hojas de papel es muy propenso a pérdidas, otro punto importante es el tiempo empleado en los distintos procesos, ya que se considera que se podría atender a más personas en el lapso de 8 horas laborales y por último está el uso de recursos, aquí se establece que no se hace uso de las tecnologías de hardware y software que posee en la institución.

Por lo que se propone cubrir dichos inconvenientes, presentados en las distintas áreas de la institución mediante la implementación de la aplicación web/móvil a través de internet, con el cual se espera mejorar el almacenamiento de datos, eficiencia en los procesos y el uso adecuado de sus herramientas tecnológicas.

En el transcurso del desarrollo del sistema se toma en cuenta varios módulos, los cuales detallaremos a continuación:

Para la realización de la aplicación se implementará diferentes módulos tales como:

- **Usuarios**

Módulo de autenticación

Módulo de gestión de roles

- **Departamento de secretaría**

Módulo de gestión de proceso de inscripción de aspirantes.

Módulo de gestión de proceso de matrícula de estudiantes.

- **Departamento de asesoría vial**

Módulo de gestión de movimiento de vehículos de uso administrativo.

Módulo de gestión de movimiento de vehículos de uso institucional.

Módulo de gestión de control de instructores.

Módulo de gestión de control de vehículos.

Módulo de gestión de requerimientos de asesoría vial.

Módulo de gestión de rutas

- **Departamento de DTIC**

Módulo de gestión de desarrollo e implementación de aplicaciones.

Módulo de gestión de soporte técnico preventivo y correctivo (Hardware- Software)

Módulo de gestión de laboratorio de computación.

Módulo de gestión de sistemas afines a la institución.

- **Departamento de inspección**

Módulo de gestión de control de las aulas de clases.

Módulo de gestión de actividades previo a los exámenes de grado.

- **Departamento de pedagogía**

Módulo de gestión de sílabos académicos.

Módulo de gestión de preparación de capacitación y apoyo a docentes.

Módulo de gestión de planificaciones académicas.

Módulo de gestión de docentes

- **Departamento de contabilidad y tesorería**

Módulo de gestión de compras.

Módulo de gestión de ventas.

Las ventajas de la implementación del sistema serán:

- Análisis rápido y preciso en la gestión y búsqueda de información.
- Menor consumo de tiempos y recursos en el Proceso de gestión de la información de cada departamento.
- Obtención de reportes que ayudará al proceso y mejora de la toma de decisiones.

El proyecto se encuentra alineado al proceso de Análisis y Diseño de Software, en lo que se refiere a la construcción del software y con línea transversal en Tecnologías de la Información y Comunicación, ubicando en la investigación de la ESPOCH - EIS, En el Plan Nacional Toda Una Vida cumple con el objetivo 5, política 5.10

1.4. Objetivos

1.4.1. Objetivo General

Desarrollo de una aplicación web y aplicación móvil para la gestión de información de los procesos llevados a cabo en cada uno de los departamentos administrativos de la escuela de conducción “ConduesPOCH” aplicando el Framework React bajo un entorno de pruebas.

1.4.2. Objetivos Específicos

- Analizar todas las actividades administrativas llevadas a cabo en la escuela de conducción profesional “CONDUESPOCH”.
- Analizar las tecnologías necesarias y apropiadas para la realización de la aplicación web y aplicación móvil.
- Desarrollar los módulos de gestión de información e implementar en el sistema.
- Implementar la metodología de pruebas a la aplicación para verificar el correcto funcionamiento de los componentes del sistema en los niveles pertinentes.
- Determinar el nivel de eficiencia del sistema en relación al tiempo empleado en la gestión de procesos administrativos haciendo uso del estándar ISO 25010.

CAPÍTULO II

2. FUNDAMENTOS TEÓRICOS

En este apartado se establece la teoría referente a temas relacionados con los elementos de influencia para el presente trabajo, donde se destaca lo más importante como los temarios tentativos sobre la gestión de información que pueden servir para desarrollar los procesos administrativos en la escuela de conducción CONDUESPOCH E.P, además de las diferentes tecnologías y métodos que se van a utilizar para su creación y desarrollo.

Para la realización de la información detallada a continuación se ha obtenido información de distintas fuentes bibliográficas, en donde varios autores dan su punto de vista sobre temas concretos que se van a utilizar.

2.1. Estado del arte

El trabajo de (Moran, 2012, p.5) titulado “IMPLEMENTACIÓN DE UN SISTEMA PARA AUTOMATIZAR LOS PROCESOS ACADÉMICOS Y ADMINISTRATIVOS DE LA ESCUELA DE CONDUCCIÓN DE CHOFERES PROFESIONALES DE LOS RÍOS MANUEL BHRUNISS VILLACRÉS DE LA CIUDAD DE BABAHOYO”, trata en profundidad el tema de los procesos académicos y administrativos de la escuela de conducción Manuel Bhruniss de los Ríos de la ciudad de Babahoyo con el objetivo de automatizar dichos procesos. La escuela está conformada por departamentos similares a CONDUESPOCH (Secretaría, Dtic y asesoría vial), con el cual se tiene una guía básica para el desarrollo de la aplicación web, a diferencia de la aplicación mencionada, se realiza con diferentes frameworks

Según el trabajo de (Edgar, 2018, p.7), titulado “AUTOMATIZAR LA GESTIÓN DE MATRÍCULA Y NOTAS EN ENTORNO WEB DE LA ESCUELA DE CAPACITACIÓN PARA CONDUCTORES PROFESIONALES DEL SINDICATO DE CHÓFERES DEL CANTÓN MEJÍA PARROQUIA MACHACHI “, se realizó con el objetivo y la necesidad de utilizar un sistema académico automático para el remplazo de los procesos manuales, el cual facilite el almacenamiento y uso de la información que maneja dicha escuela, el sistema desarrollado utiliza una metodología ágil y técnicas de programación javascript, los objetivos de esta aplicación están orientados a mejorar y optimizar los recursos, siendo satisfactorios para el personal de dicha escuela.

El trabajo de (Nuñez, 2017, p.9) con título “APLICACIÓN WEB PARA LA GESTIÓN AUTOMATIZADA DE LOS SERVICIOS QUE PRESTA LA ESCUELA DE CAPACITACIÓN PARA CONDUCTORES DEL SINDICATO DE CHOFERES PROFESIONALES DEL CANTÓN MOCACHE”, se basa en la creación de un software con la finalidad de facilitar al acceso de información que maneja dicha escuela. Para el desarrollo de la aplicación se realizó una recopilación de información en la cual se pudo evidenciar que los procesos administrativos de la escuela de conducción se llevaban de forma manual por lo cual la elaboración de la aplicación web ayudó a la escuela a mejorar el funcionamiento administrativo y servicio al cliente. Dicha aplicación está desarrollada por una arquitectura MVC el cual es la misma a utilizar en el sistema para CONDUESPOCH.

Con la ejecución del trabajo de (Blanco, 2014, p.12), titulado “SOFTWARE PARA LA ESCUELA DE CONDUCCIÓN GENERAL SANTANDER ESCUGESAN” se busca desarrollar e implementar un sistema de información computarizado, que tenga la capacidad de agilizar y reducir el tiempo en el proceso de expedición de certificados y de esta manera lograr mayor calidad en la atención del usuario ya que se cuenta con la tecnología adecuada para el manejo de la información invirtiendo un mínimo de recursos tanto humano como de equipos. El sistema cuenta con los módulos de ingreso alumnos, reportes, consultas, administración, ayuda, copias de seguridad. Los módulos alumnos, reportes, consultas, administración son similares a módulos a desarrollar dentro de la aplicación para CONDUESPOCH.

De acuerdo con (Wyse, 2015, p.2), en el artículo titulado” REACT: A Framework for Rapid Exploration of Approximate Computing Techniques”, React permite el desarrollo de aplicaciones web grandes y complejas que pueden cambiar sus datos sin actualizaciones posteriores de la página. Su objetivo es proporcionar mejores experiencias de usuario y con una velocidad increíblemente rápida y robusta, también se puede integrar con otras bibliotecas o marcos JavaScript en MVC, esta herramienta se utiliza para crear las interfaces para CONDUESPOCH.

Tabla 1-2: Aporte al trabajo de integración curricular

Referencia	Aporte
“IMPLEMENTACIÓN DE UN SISTEMA PARA AUTOMATIZAR LOS PROCESOS ACADÉMICOS Y ADMINISTRATIVOS DE LA ESCUELA DE CONDUCCIÓN DE CHOFERES PROFESIONALES DE LOS RÍOS MANUEL BHRUNISS VILLACRES DE LA CIUDAD DE BABAHOYO”	Los procesos internos realizados dentro de la institución mencionada en la tesis son similares a los procesos de CONDUESPOCH, más específicamente en el departamento académico y DTIC. Con esto podemos tener una guía de cómo realizar estos procesos de automatización, pero se debe implementar los módulos faltantes.
“AUTOMATIZAR LA GESTIÓN DE MATRÍCULA Y NOTAS EN ENTORNO WEB DE LA ESCUELA DE CAPACITACIÓN PARA CONDUCTORES PROFESIONALES DEL SINDICATO DE CHÓFERES DEL CANTÓN MEJÍA PARROQUIA MACHACHI”	De la misma manera la forma de realizar los procesos en el sistema implementado en la Escuela de Capacitación para conductores profesionales del sindicato de choferes del cantón Mejía, es muy similar, específicamente el módulo de matrículas, con ciertas variaciones que deben ser desarrolladas en el presente sistema.
“APLICACIÓN WEB PARA LA GESTIÓN AUTOMATIZADA DE LOS SERVICIOS QUE PRESTA LA ESCUELA DE CAPACITACIÓN PARA CONDUCTORES DEL SINDICATO DE CHOFERES PROFESIONALES DEL CANTÓN MOCACHE”	La aplicación web automatiza los procesos internos con javascript y plantillas de interfaz de usuarios, con el cual se puede desarrollar de forma óptima, con interfaces propias y utilizando el framework REACT
“SOFTWARE PARA LA ESCUELA DE CONDUCCIÓN GENERAL SANTANDER ESCUGESAN”	Este software utiliza javascript para el departamento de dtic de la escuela de conducción, con el cual se puede comparar la forma de desarrollo, y que la aplicación final tenga un mejor desempeño.

Realizado por: Aldas W.; Pilataxi C., 2020

2.2. Procesos departamentales de CONDUESPOCH

El sistema a desarrollar permitirá gestionar la información de cada uno de los departamentos que forman parte de la escuela de conducción profesional “CONDUESPOCH”, facilitando la comunicación y el manejo de los datos que permiten obtener resultados valiosos de los procesos internos.

Los procesos que se realizan en CONDUESPOCH se detallan a continuación:

2.2.1. Procesos académicos

Los procesos académicos llevados a cabo dentro de la institución son la inscripción y matrícula de los estudiantes, mismos que se detallan a continuación:

Inscripción de Estudiantes

El proceso de inscripción en CONDUESPOCH es llevado a cabo por el departamento de Secretaría, Contabilidad y finalmente Gerencia administrativa, el proceso consiste en la

recopilación de información necesaria sobre el nuevo aspirante. A pesar de que existe la intervención de varios departamentos, el que realiza más interacción es el de Secretaría ya que como se muestra en la **Gráfico 1-2**, aquí es donde empieza y termina la inscripción de un estudiante.

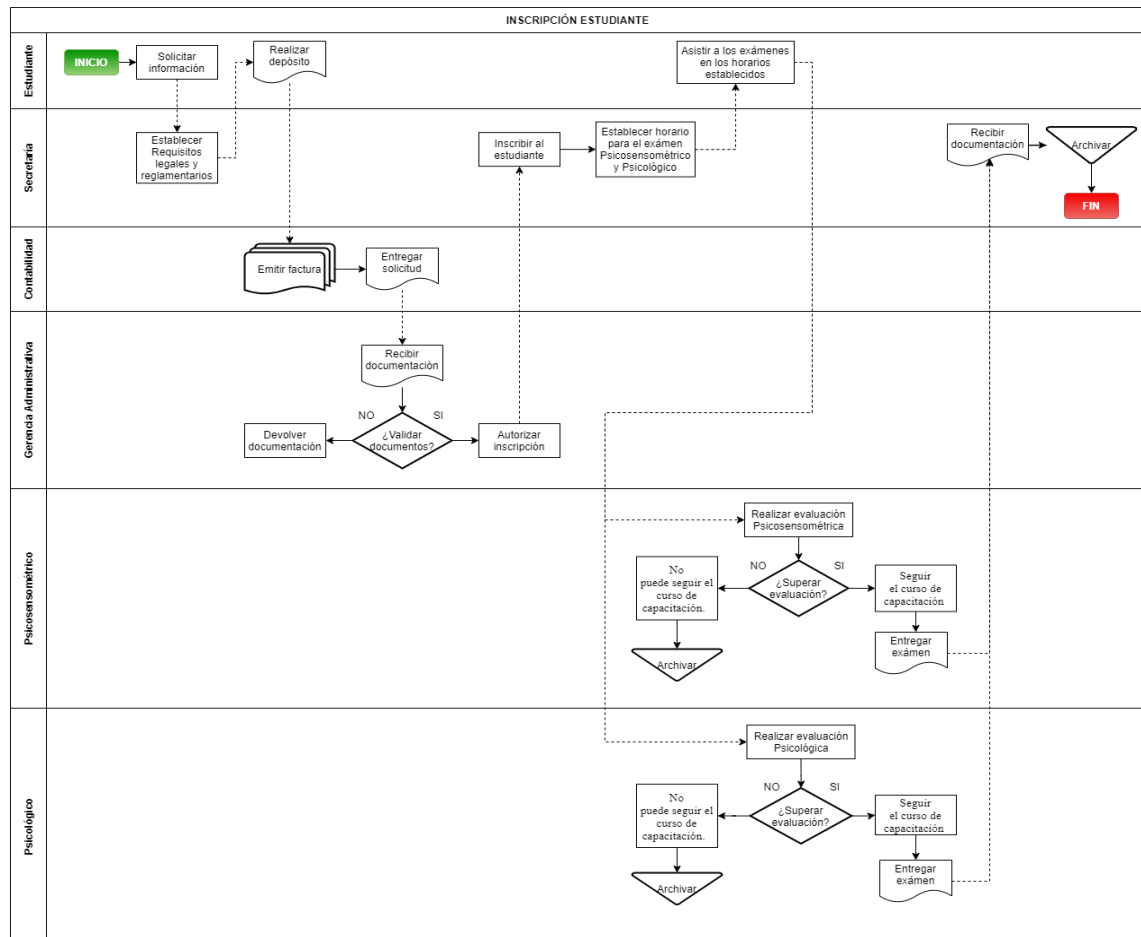


Gráfico 1-2: Proceso de Inscripción

Realizado por: Aldas W.; Pilataxi C., 2020

Matrícula de Estudiantes

Cuando el alumno ha culminado el proceso de inscripción, entonces se habilita el proceso de matrícula, que al igual que la anterior, ésta necesita ser verificada por los departamentos de Secretaría, Contabilidad, Gerencia administrativa y finalmente Pedagogía como se presenta en la **Gráfico 2-2**.

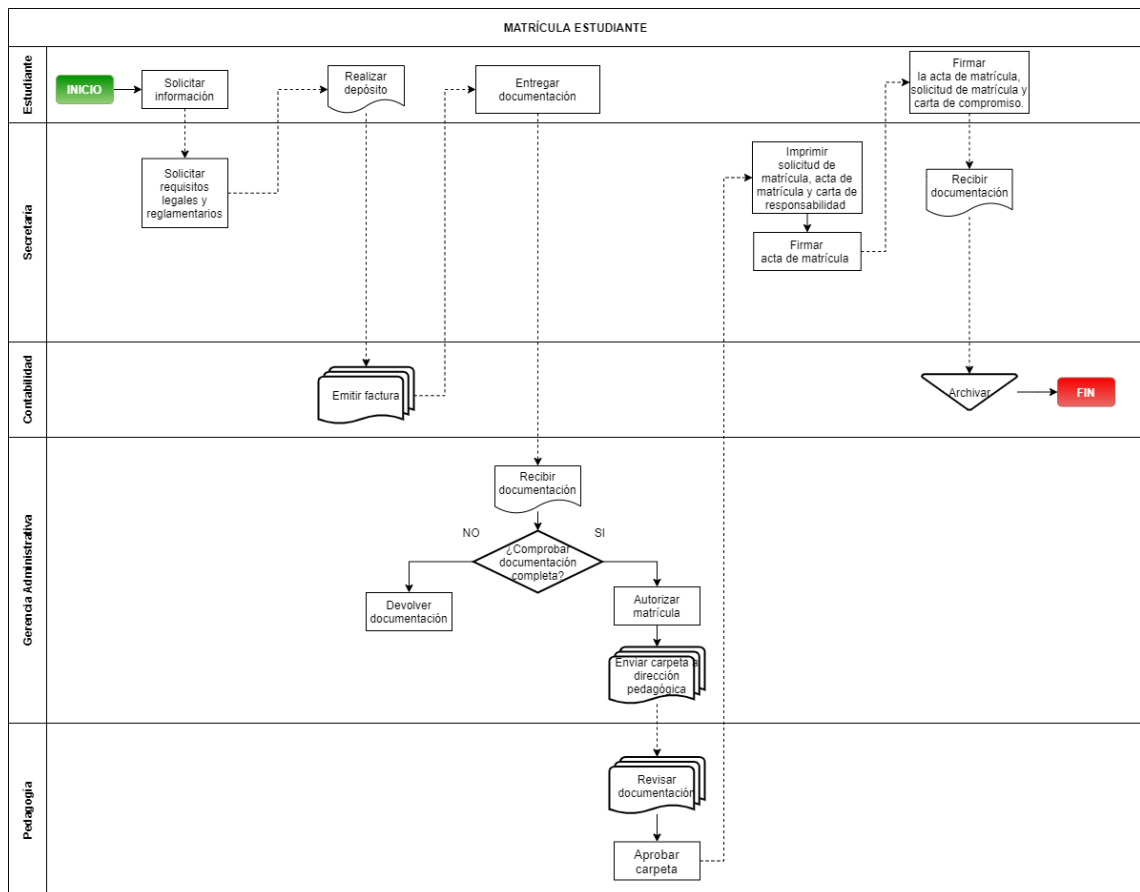


Gráfico 2-2: Proceso de Matrícula

Realizado por: Aldas W.; Pilataxi C., 2020

2.2.2. *Procesos de gestión de recursos para el manejo vehicular*

Entre los procesos de gestión de recursos para el manejo vehicular constan la gestión de movimiento de vehículo administrativo, gestión de movimiento de vehículo institucional, control vehicular, requerimientos de Asesoría Vial, gestión de rutas para el recorrido, los cuales se detallan a continuación:

Movimiento de vehículos de uso administrativo

Uno de los procesos que encabeza el departamento de Asesoría vial es el de gestionar las solicitudes de movimiento de vehículos de uso administrativo, para lo cual los miembros de cada departamento son quienes solicitan un vehículo disponible al departamento Gerencial que posteriormente terminan siendo despachados por el departamento de Asesoría vial, el cual además archiva los documentos recibidos de todo el proceso, los movimientos realizados se detallan en la **Gráfico 3-2**.

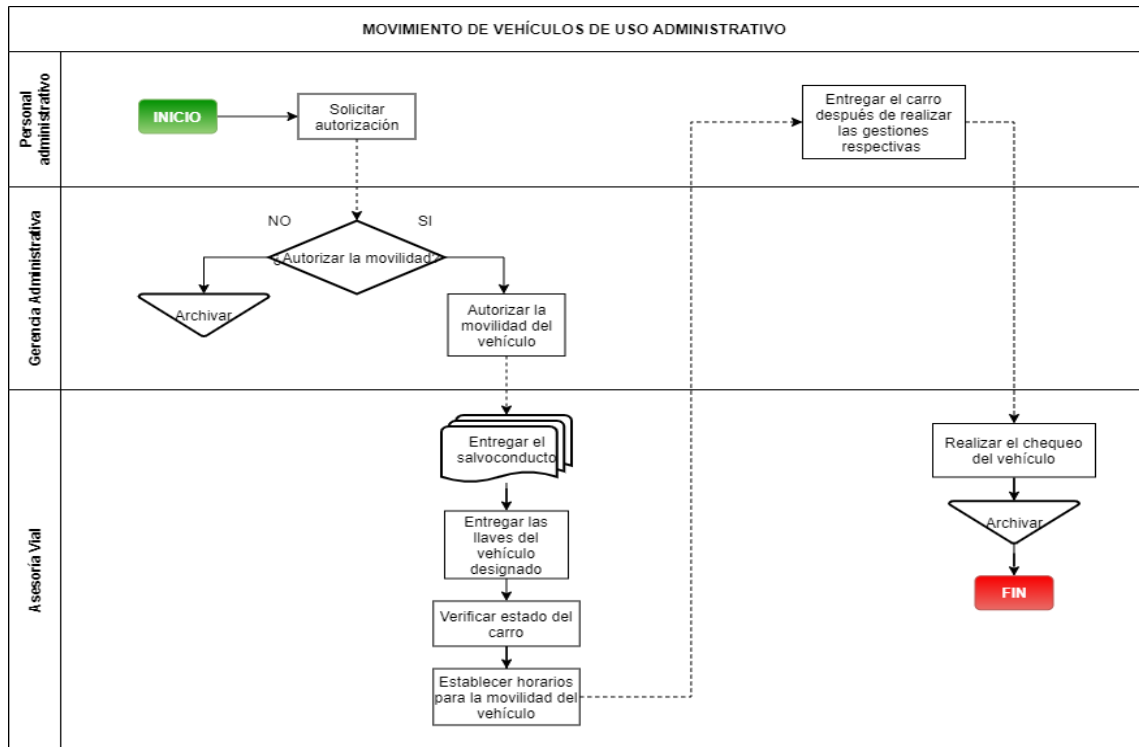


Gráfico 3-2: Proceso de Movimiento de vehículos de uso administrativo

Realizado por: Aldas W.; Pilataxi C., 2020

Movimiento de vehículos de uso institucional

El departamento de Asesoría vial se encarga de gestionar el movimiento de vehículos de uso institucional, para lo cual este departamento debe solicitar el vehículo al departamento Gerencial con todos los documentos necesarios que será asignado a un instructor, mismo que al terminar con la práctica de conducción correspondiente debe entregar todo en el mismo estado en que recibió, este proceso se detalla a continuación en la **Gráfico 4-2**.

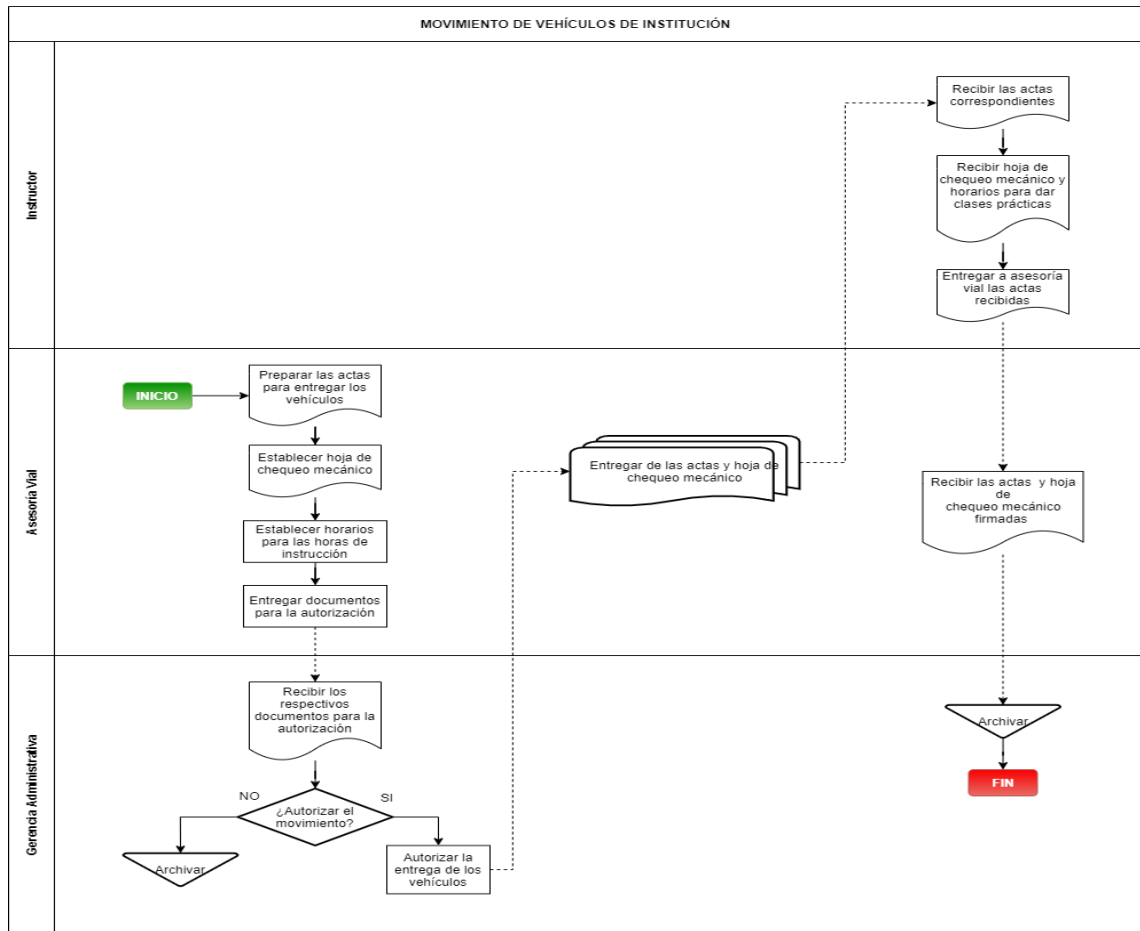


Gráfico 4-2: Proceso de Movimiento de vehículos de uso institucional

Realizado por: Aldas W.; Pilataxi C., 2020

Control Vehicular

El control vehicular es una actividad de registro de los informes de estado de cada vehículo, esto se lo realiza únicamente para llevar un historial de fallas o inconvenientes suscitados con cada uno de los autos de CONDUESPOCH. Este proceso está detallado de mejor manera en la **Gráfico 5-2.**

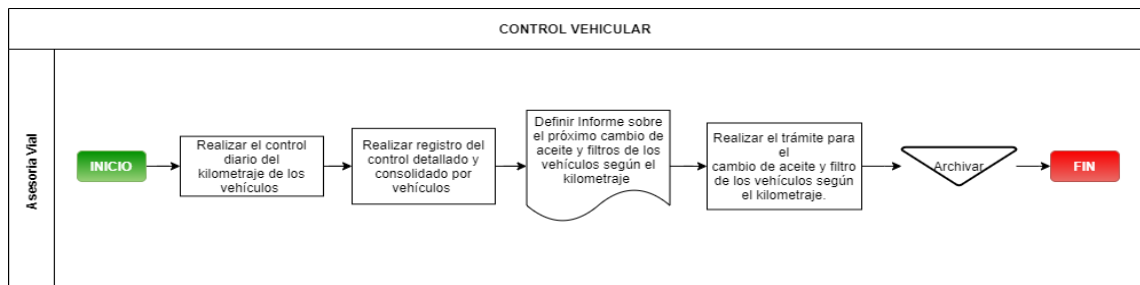


Gráfico 5-2: Proceso de Gestión de control vehicular

Realizado por: Aldas W.; Pilataxi C., 2020

Requerimientos de Asesoría Vial

Los requerimientos realizados por los instructores deben ser revisados y aprobados o rechazados por el departamento de Asesoría vial, para proceder a completarse en el departamento de Gerencia Administrativa y por último Contabilidad. Los requerimientos pueden ser muy variados, yendo desde la solicitud de combustible hasta repuestos para los vehículos, así como se demuestra en la **Gráfico 6-2**.

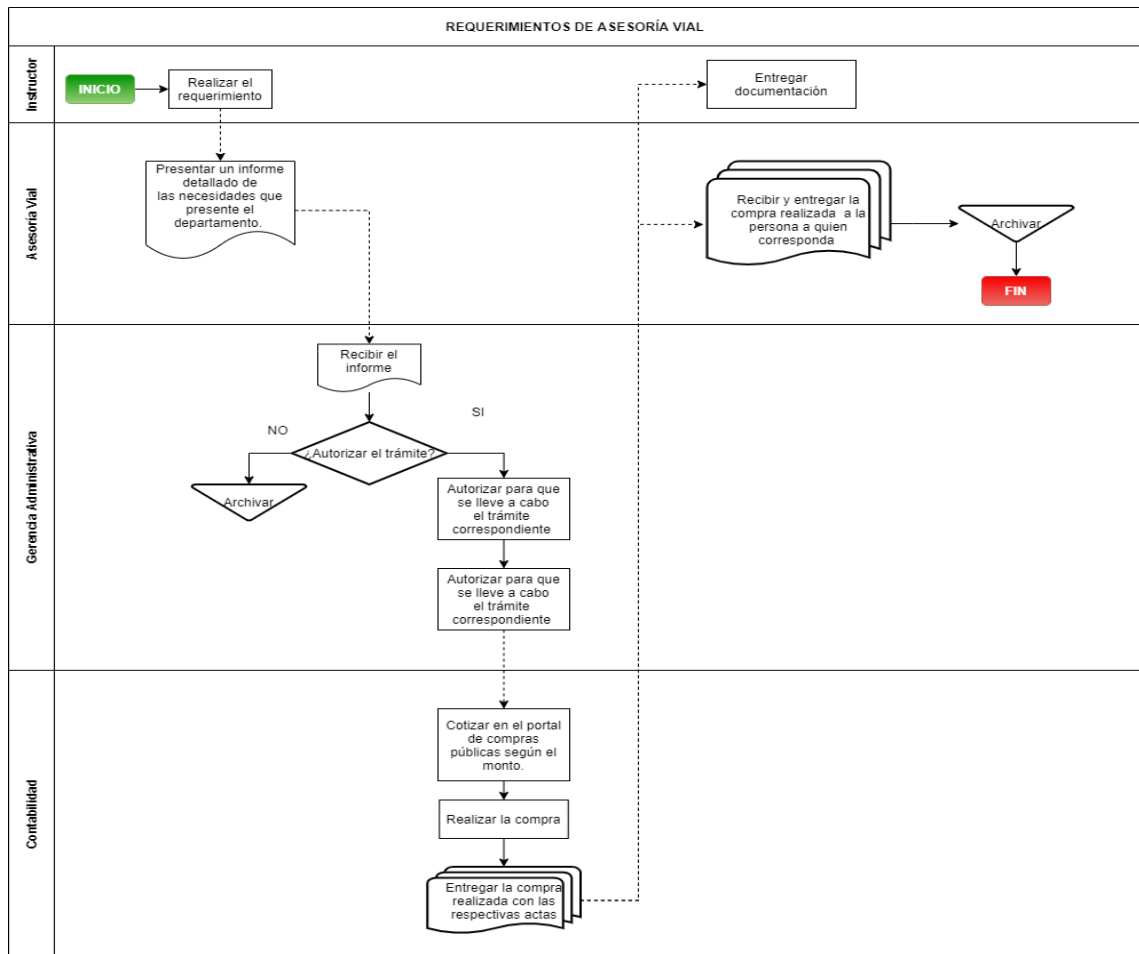


Gráfico 6-2: Proceso de solicitar requerimientos del departamento de asesoría vial

Realizado por: Aldas W.; Pilataxi C., 2020

Rutas

Para la gestión de rutas, el departamento de Asesoría vial necesita realizar los estudios correspondientes de las circunstancias climatológicas, demográficas entre otras características más que se deben tener en cuenta al momento de generar los respectivos informes de rutas para el recorrido de los vehículos en las horas de clases, mismos que serán presentados a los

instructores de turno correspondientes, siendo aprobados previamente por el departamento Gerencial, así como se muestra en la **Gráfico 7-2**.

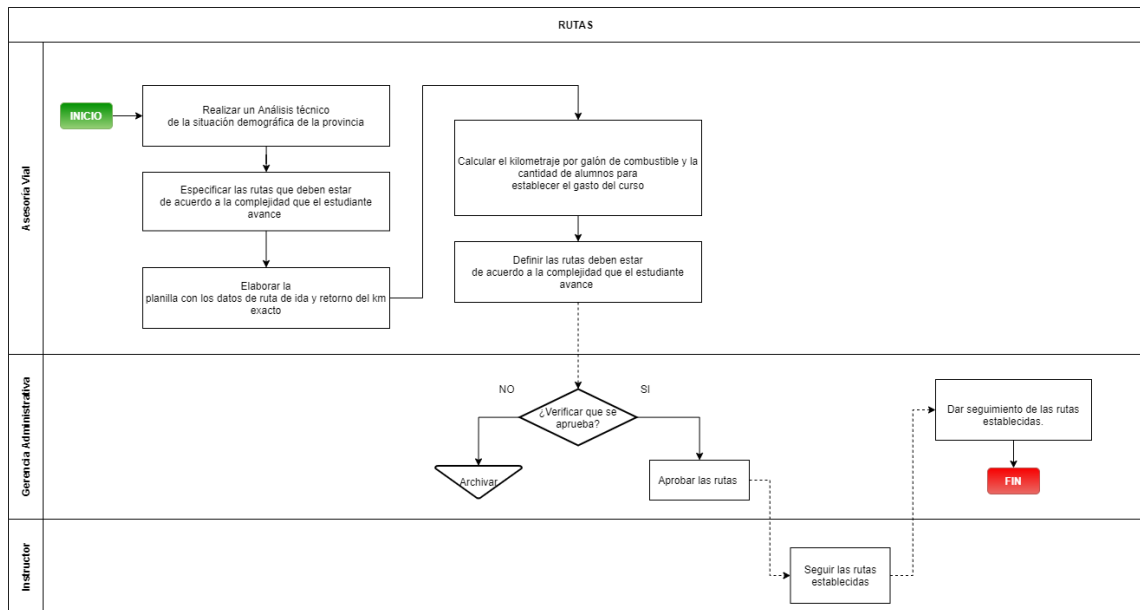


Gráfico 7-2: Proceso de Gestión de rutas

Realizado por: Aldas W.; Pilataxi C., 2020

2.2.3. *Procesos de desarrollo y mantenimiento de aplicaciones*

Los procesos de desarrollo y mantenimiento de aplicaciones consisten en dar seguimiento a cada una de las tareas que se llevadas a cabo dentro del departamento de DTIC, además de las herramientas utilizadas para el desarrollo y mantenimiento de aplicaciones en la institución, estos procesos se describen a continuación:

Desarrollo e implementación de aplicaciones

El proceso de desarrollo de aplicaciones es dirigido al departamento de DTIC, inicia con la petición del requerimiento de parte de los demás departamentos de la institución, luego, esta es aprobada o rechazada por el departamento en cuestión para finalmente enviarse al departamento Gerencial, el cual es el último en decidir si tal requerimiento es viable o no realizarse. Los detalles referentes a todo lo descrito se da a conocer de mejor manera en la **Gráfico 8-2**.

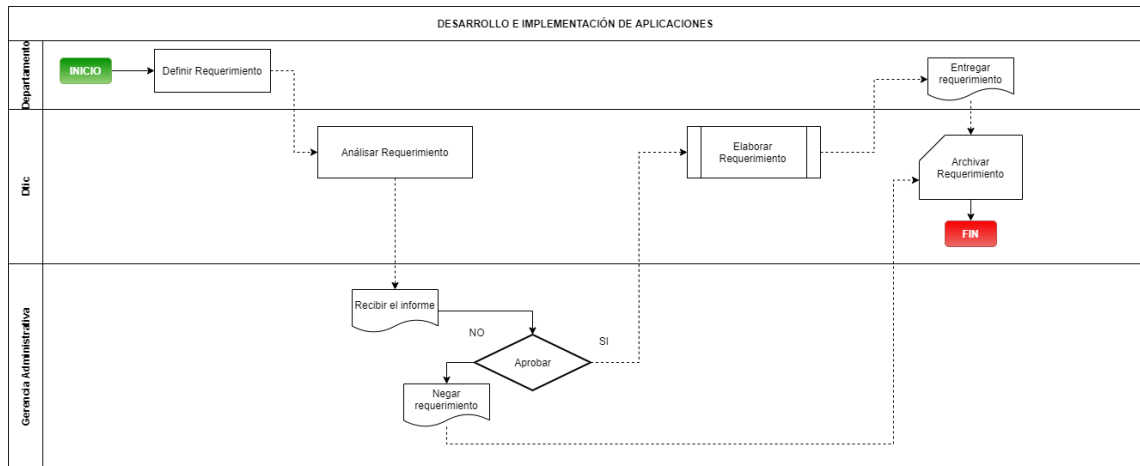


Gráfico 8-2: Proceso de Desarrollo e implementación de aplicaciones

Realizado por: Aldas W.; Pilataxi C., 2020

Tareas de mantenimiento

El proceso de mantenimiento del laboratorio de computación, soporte técnico y manejo del sitio web de la institución consisten en que los departamentos realizan una solicitud en el que se detalle dicha petición que posteriormente es evaluado por el departamento de DTIC y posteriormente puesto en ejecución, finalmente los resultados son archivados para posteriores revisiones, cada uno de los pasos descritos se detallan de mejor forma en la Gráfico 9-2.

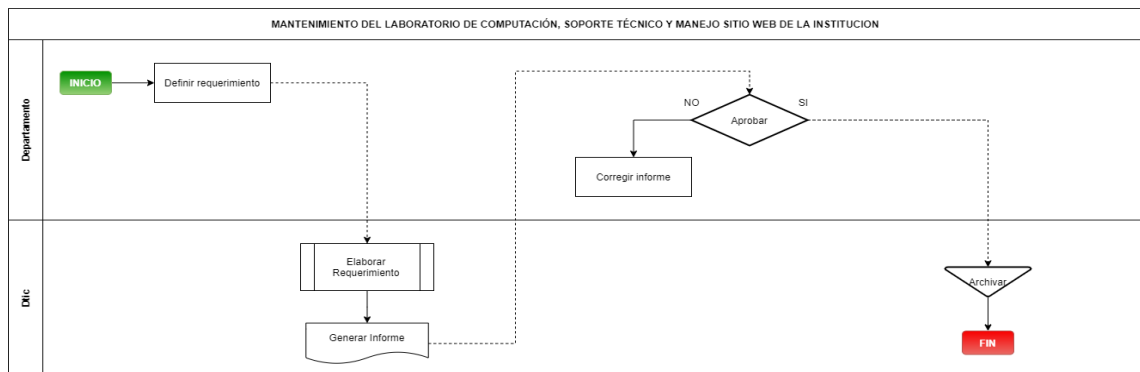


Gráfico 9-2: Proceso de Mantenimiento del departamento de DTIC

Realizado por: Aldas W.; Pilataxi C., 2020

2.2.4. *Procesos de gestión de informes y notificaciones*

Estos procesos están enfocados en generar informes y certificaciones al momento de realizar las tareas de revisión de aulas de clases y desarrollo de actividades previas al examen de grado, estos se detallan a continuación:

Revisión de aulas de clases

El proceso de revisión de las aulas de clase consiste en el registro del estado actual de cada una de las aulas utilizadas para su respectiva cátedra, mismos que a su vez contarán con las herramientas necesarias de apoyo. Este proceso al involucrar solamente un departamento, funciona a modo de historial de los recursos que se posee CONDUESPOCH, en el cual se detalla las características que estas poseen. Las tareas que se llevan a cabo se pueden apreciar en la **Gráfico 10-2**.

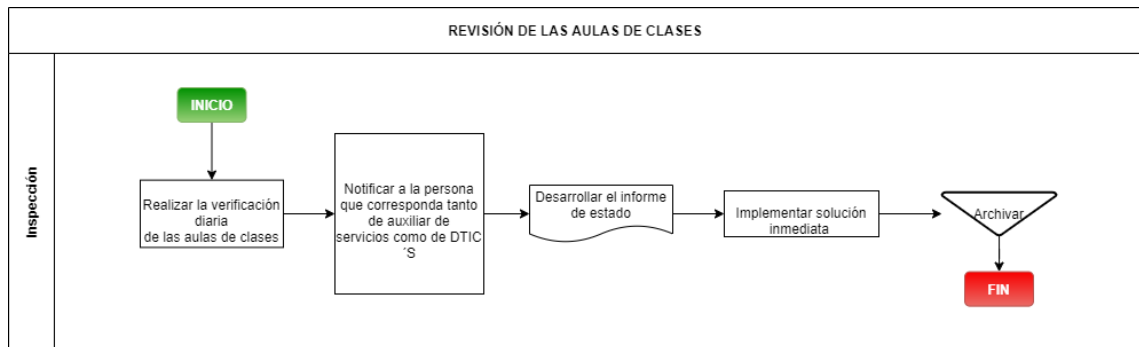


Gráfico 10-2: Proceso de Revisión de las aulas de clase

Realizado por: Aldas W.; Pilataxi C., 2020

Actividades previo a exámenes de grado

La gestión de procesos previos a los exámenes de grado, consiste en que el personal del departamento de Inspección notifique los acontecimientos suscitados dentro de las aulas de clases, es decir, se informa sobre los inconvenientes presentados por los alumnos, docentes o instructores con el fin de dar solución a sus problemas. Las notificaciones son dirigidas al departamento de Pedagogía quien al revisar aplica las medidas pertinentes, así como se detalla en la **Gráfico 11-2**.

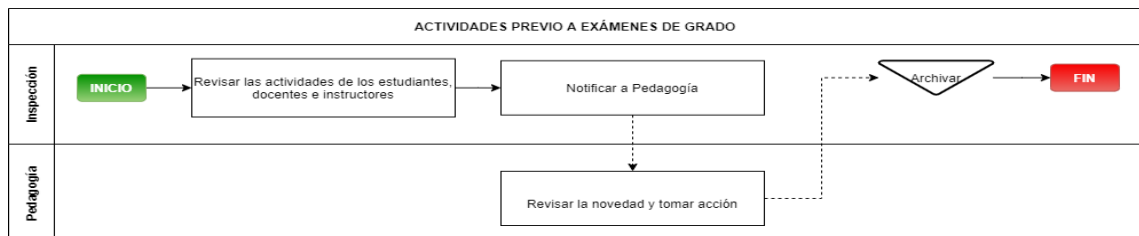


Gráfico 11-2: Proceso previo a los exámenes de grado

Realizado por: Aldas W.; Pilataxi C., 2020

2.25. *Procesos de gestión de sílabos, docentes e instructores*

Los procesos dentro de este grupo están relacionados con la gestión de los materiales de estudio para el dictado de clases, entre estos se encuentran, la gestión de sílabos académicos y planes de estudio tanto de los instructores como de los docentes, la preparación de capacitación y apoyo a instructores o docentes y el control de instructores, mismos que se detallan a continuación:

Gestión de sílabos académicos y planes de estudio de los docentes

El proceso de gestión de sílabos consiste en revisar los sílabos y planes de estudio de todos los docentes de la institución, para lo cual cada uno de los miembros activos deben enviar sus respectivos documentos para ser aprobados, este proceso se basa en la interacción únicamente entre el docente y el departamento de Pedagogía. Cada uno de estos archivos son almacenados en los registros de cada período académico, tal como se describe en la **Gráfico 12-2**.

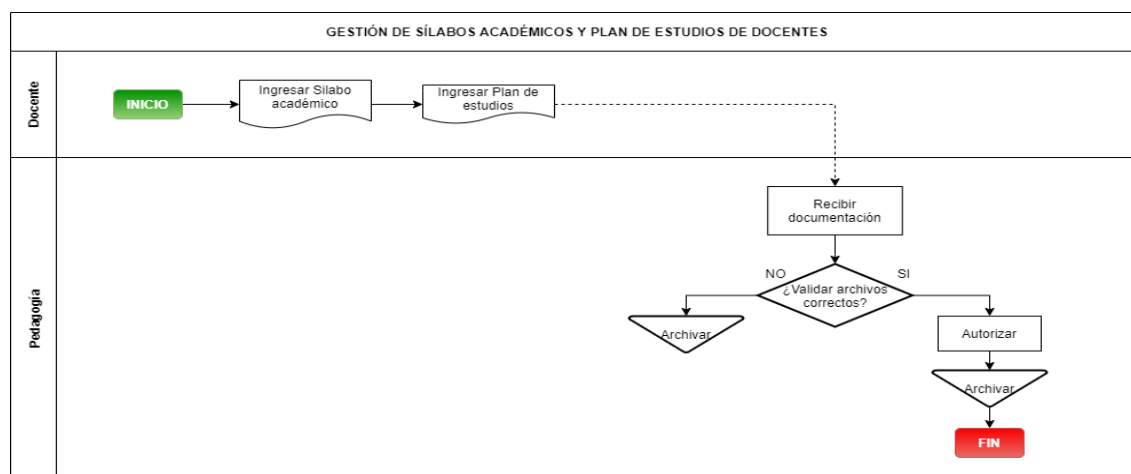


Gráfico 12-2: Proceso de gestión de sílabos académicos y planes de estudio de docentes

Realizado por: Aldas W.; Pilataxi C., 2020

Gestión de sílabos académicos y planes de estudio de los instructores

El proceso de Gestión de sílabos académicos y planes de estudio de los instructores, son gestionados por el departamento de Asesoría vial, para lo cual cada uno de los miembros debe preparar los informes requeridos para ser aprobados y posteriormente entregados a los estudiantes para su respectivo uso, en el caso de que no se aprueben, el instructor deberá realizar otro informe. Los detalles de este proceso se muestran en la **Gráfico 13-2**.

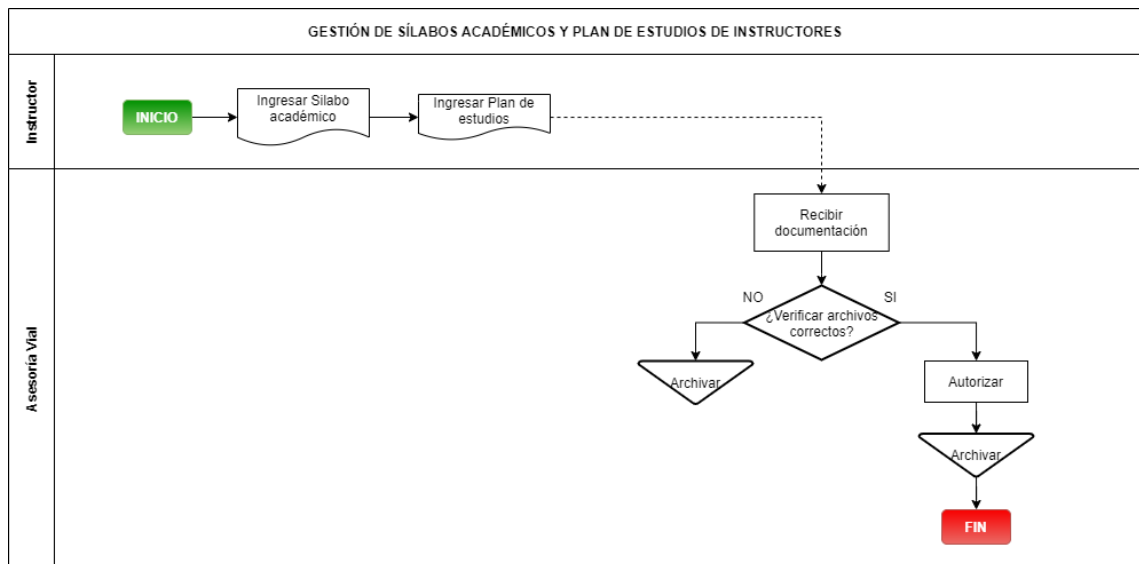


Gráfico 13-2: Proceso de gestión de sílabos académicos y plan de estudio de instructores

Realizado por: Aldas W.; Pilataxi C., 2020

Preparación de capacitación y apoyo a instructores

El personal del departamento de Asesoría vial al ser responsable de los instructores de CONDUESPOCH, tienen la obligación de capacitar y dar apoyo a estos, en el caso de que así lo requieran o por mandato de los altos mandos, para la realización de este proceso el departamento evalúa periódicamente a cada instructor a través de sus estudiantes, que posteriormente se miden los resultados obtenidos y se realiza un informe en el cual se detalle los puntos fuertes o débiles de cada uno de los miembros evaluados. Las etapas de este proceso se detallan en la **Gráfico 14-2**.

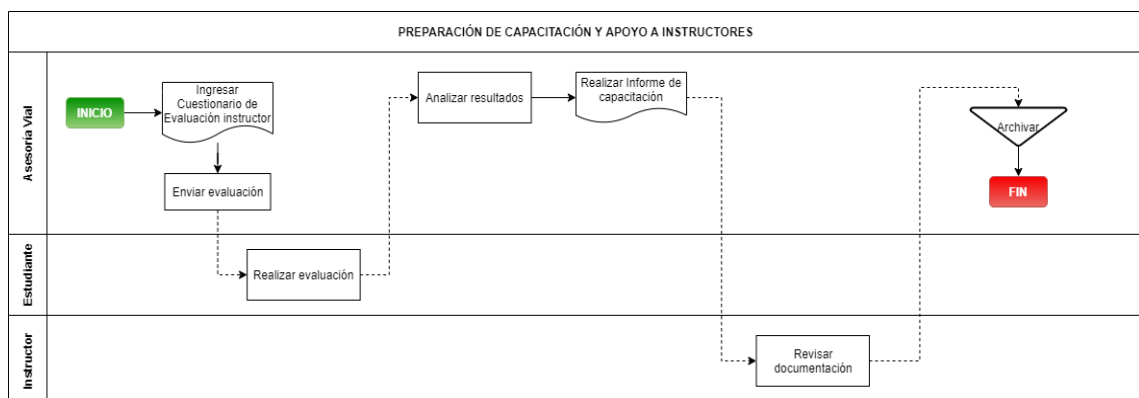


Gráfico 14-2: Proceso de capacitación y apoyo a instructores

Realizado por: Aldas W.; Pilataxi C., 2020

Preparación de capacitación y apoyo a docentes

El proceso de preparación de capacitación y apoyo a docentes es llevado a cabo por el departamento de Pedagogía, éste realiza las respectivas evaluaciones a los docentes que serán desarrolladas por los respectivos estudiantes, estos son quienes determinan el rendimiento de cada uno, a partir de los resultados obtenidos se realiza un informe dirigido al docente en el cual se especifica los inconvenientes presentados y posteriormente se lo capacita para un mejor desempeño académico. Los pasos para realizar este proceso se detallan en la **Gráfico 15-2**.

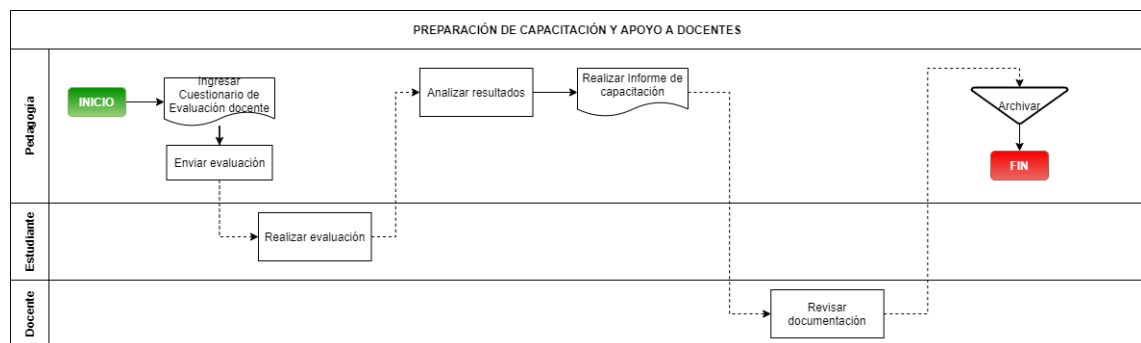


Gráfico 15-2: Proceso de preparación de capacitación y apoyo a docentes

Realizado por: Aldas W.; Pilataxi C., 2020

Control de instructores

Por otro lado, el proceso de gestión de instructores consiste en la recepción y almacenamiento de la información necesaria que sirva de guía para los estudiantes, cada instructor es responsable de presentar los informes solicitados por el departamento de Asesoría vial para su debida aprobación y posterior uso en su respectiva catedra. Los documentos solicitados a cada instructor se detallan en la **Gráfico 16-2**.

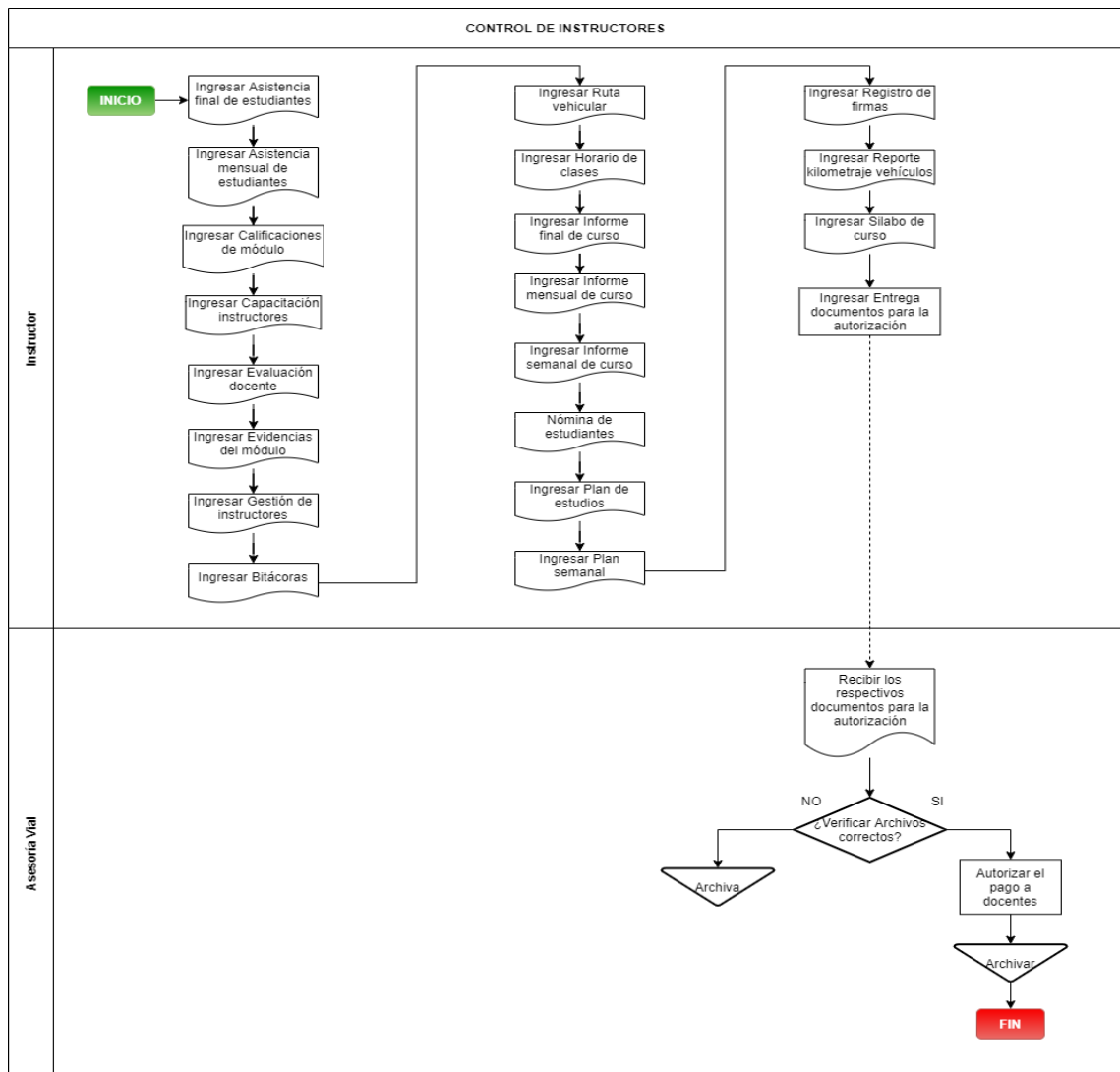


Gráfico 16-2: Proceso de control de instructores

Realizado por: Aldas W.; Pilataxi C., 2020

2.2.6. *Procesos de gestión de ventas y pagos*

Los departamentos de Contabilidad y Tesorería, están constituidos en uno solo debido a que ambos se centran en la gestión financiera de la institución y por esta razón cada vez que un proceso se dirija a uno de estos departamentos, en realidad se estará refiriendo al mismo.

Los procesos categorizados como compras y ventas son, el pago de servicios a docentes, pago servicios a instructores, pago de mensualidades de los estudiantes, venta de solicitudes generales a particulares, venta de solicitudes generales a estudiantes, venta de solicitudes de justificación a estudiantes, los que se detallan a continuación:

Pago servicios a docentes

El proceso de pago de servicios a docentes está considerado como compras de servicios debido a que mediante este proceso se lleva a cabo el desembolso de mensualidades a cada uno del personal académico que haya cumplido con las respectivas disposiciones establecidas por la institución, lo cual consiste en que se deba cumplir con los materiales de estudio, entre otras cosas. El desarrollo de este proceso se presenta en la **Gráfico 17-2**.

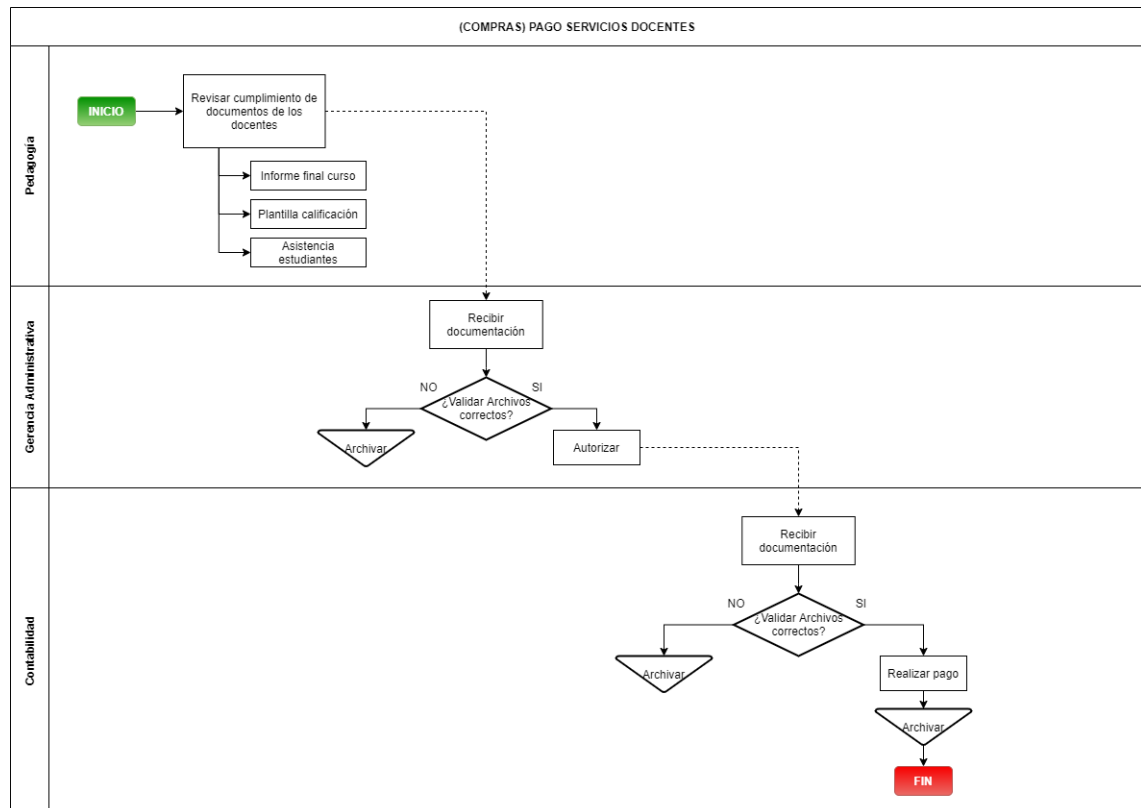


Gráfico 17-2: Proceso de pago de servicios a docentes

Realizado por: Aldas W.; Pilataxi C., 2020

Pago servicios a instructores

El proceso para el pago de servicios a instructores es muy similar al pago de docentes, con la excepción de que se incluye el departamento de Asesoría vial, mismo que inicia el proceso al determinar que algún instructor cumple con los requisitos solicitados para el pago, los requisitos, así como el procedimiento que se lleva a cabo se muestran en la **Gráfico 18-2**.

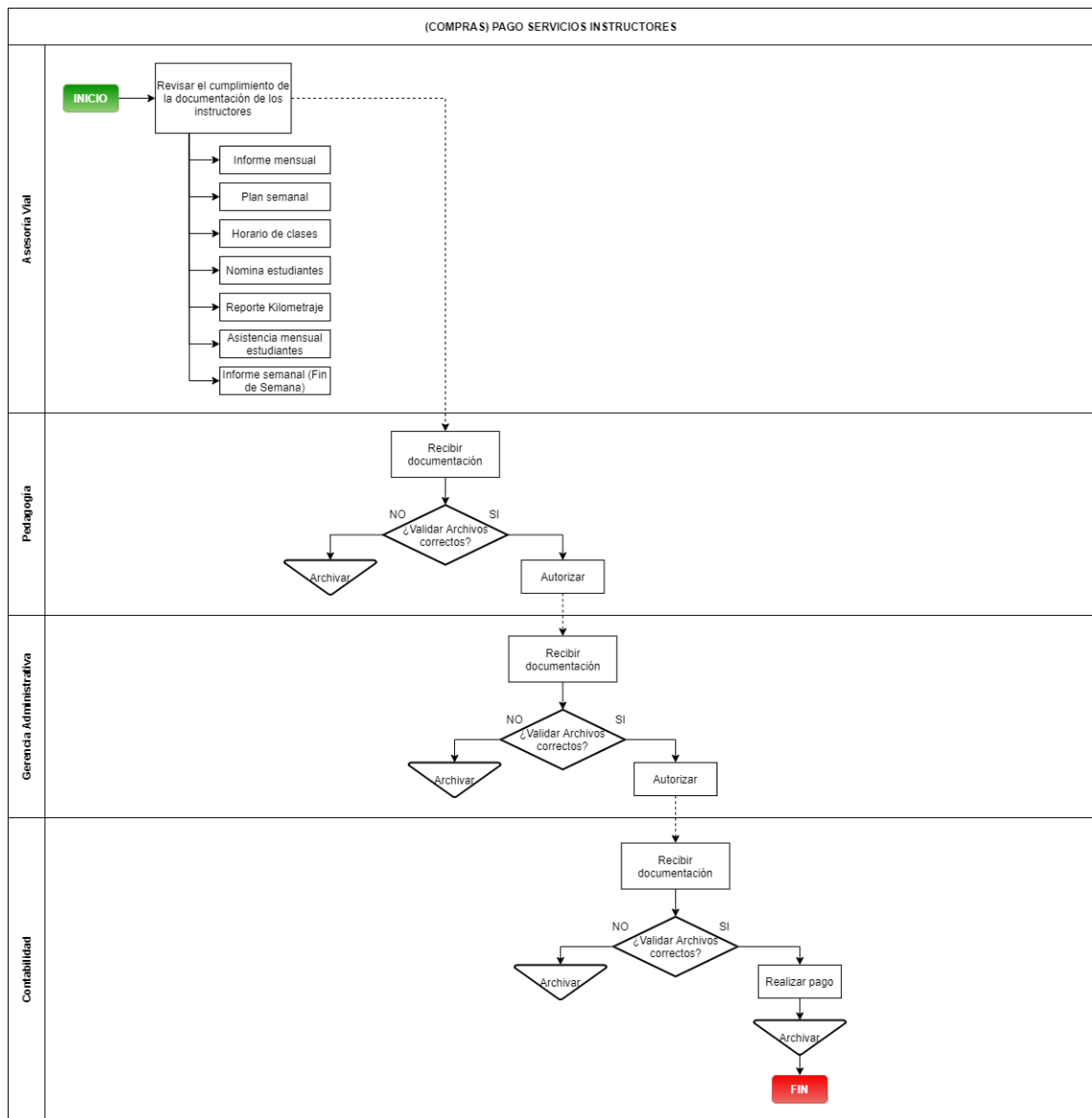


Gráfico 18-2: Proceso de pago de servicios a instructores

Realizado por: Aldas W.; Pilataxi C., 2020

Pago de mensualidades de los estudiantes

Para llevar a cabo el proceso de pagos de mensualidades de los estudiantes, estos deben enviar sus comprobantes de depósito mensual a la cuenta bancaria institucional, lo cual será analizado por el departamento de Contabilidad mismo que determinará la validez del documento presentado, que posteriormente se registrará en cada uno de los archivadores pertenecientes a cada estudiante. El desenlace de este proceso se muestra con mayor detalle en la **Gráfico 19-2**.

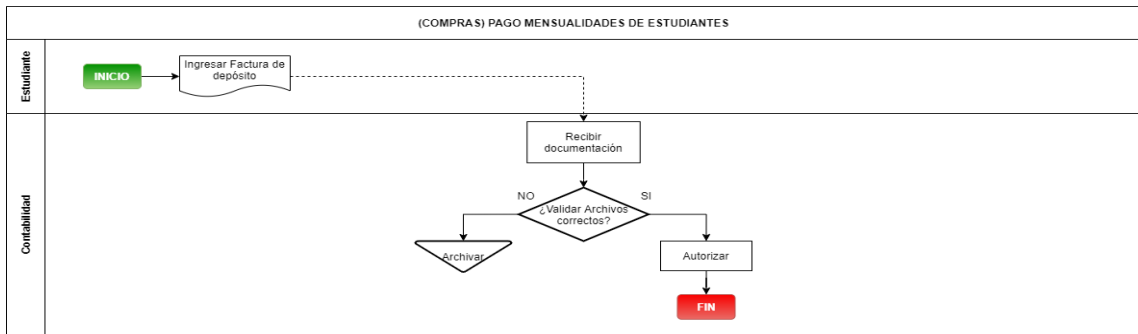


Gráfico 19-2: Proceso de pago de mensualidad de estudiantes

Realizado por: Aldas W.; Pilataxi C., 2020

Venta de solicitudes generales a particulares

El proceso de venta de solicitudes de tipo general que se realicen a personas particulares, consiste en que una persona ajena a la institución requiera de una solicitud ya sea para un examen psicológico o para cualquier otro motivo, este tipo de solicitudes son vendidos en el departamento de Contabilidad que posteriormente es enviado a Gerencia administrativa y finalmente al departamento respectivo de la institución de ser necesario para su debida aprobación. La solicitud al ser aprobada, es archivada para su verificación en posteriores procesos en el que se necesite de su uso, este proceso se puede ver de forma detallada en la **Gráfico 20-2**.

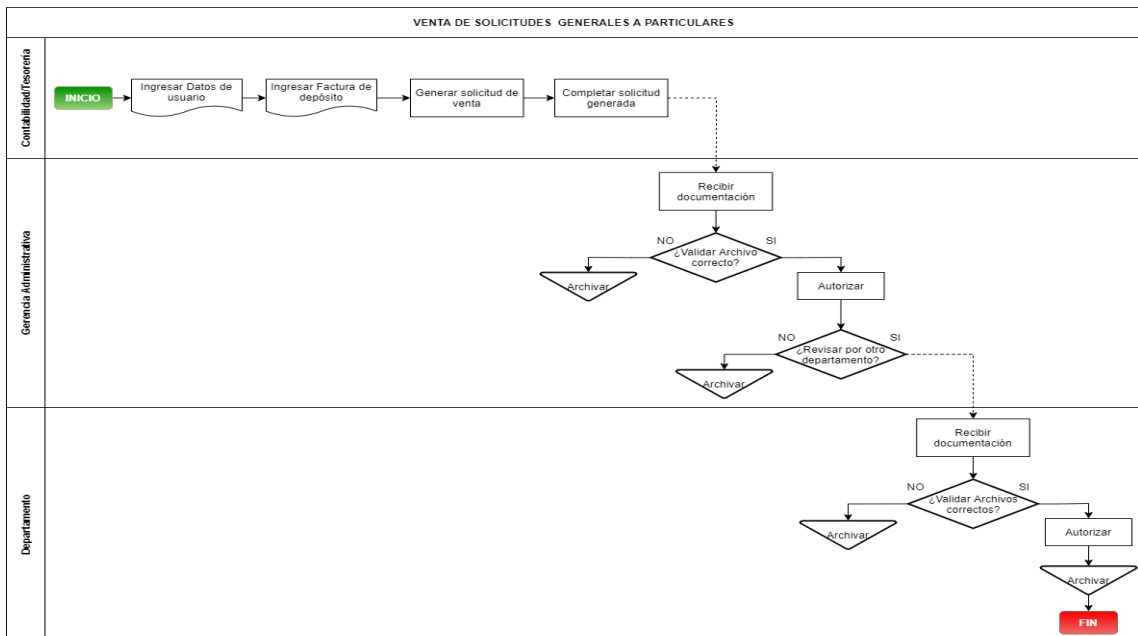


Gráfico 20-2: Proceso de venta de solicitudes generales a personas particulares

Realizado por: Aldas W.; Pilataxi C., 2020

Venta de solicitudes generales a estudiantes

El proceso de venta de solicitudes generales a estudiantes consiste en vender las solicitudes generales a los estudiantes, por lo general son usadas para referir quejas sobre el rendimiento académico de sus docentes, instructores entre otras premisas que se suscitan dentro del aula y en el horario respectivo de clases, para lo cual el alumno acude al departamento de Contabilidad para comprar una de estas solicitudes y posteriormente completar los campos necesarios del documento, para enviárselo al departamento Gerencial, el cual a su vez lo aprueba y reenvía a otro departamento para que se tomen las medidas respectivas sobre el caso presentado. Las referencias sobre el proceso detallado se pueden ver en la **Gráfico 21-2**.

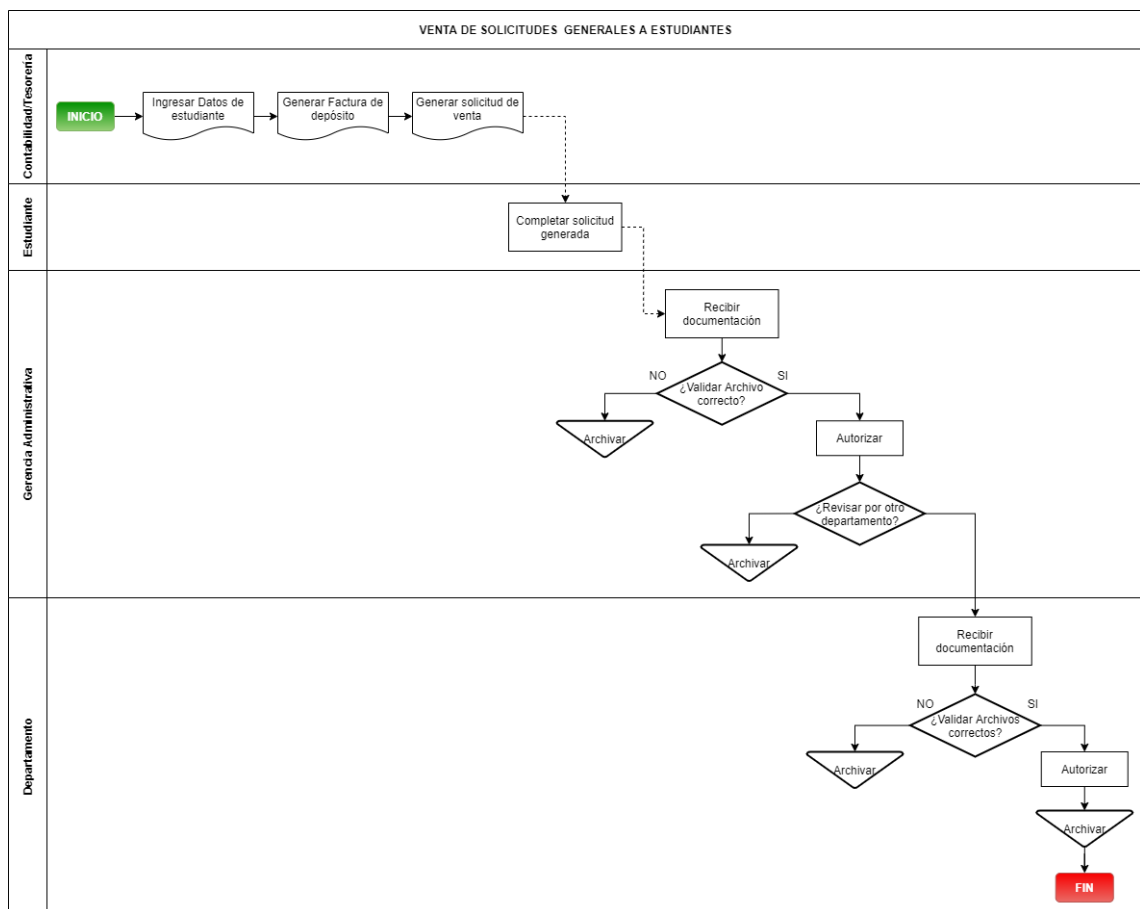


Gráfico 21-2: Proceso de venta de solicitudes generales a estudiantes

Realizado por: Aldas W.; Pilataxi C., 2020

Venta de solicitudes de justificación a estudiantes

El proceso de Venta de solicitudes de justificación consiste la venta de dichas justificaciones, ésta debe ser adquirida en el departamento de Contabilidad de CONDUESPOCH y ser

reportadas a los docentes mediante una solicitud. En el documento se detalla las razones por las que no se ha podido asistir, además se incluye documentos adicionales como por ejemplo certificados médicos. La adquisición y aprobación se detallan de mejor manera en la **Gráfico 22-2**.

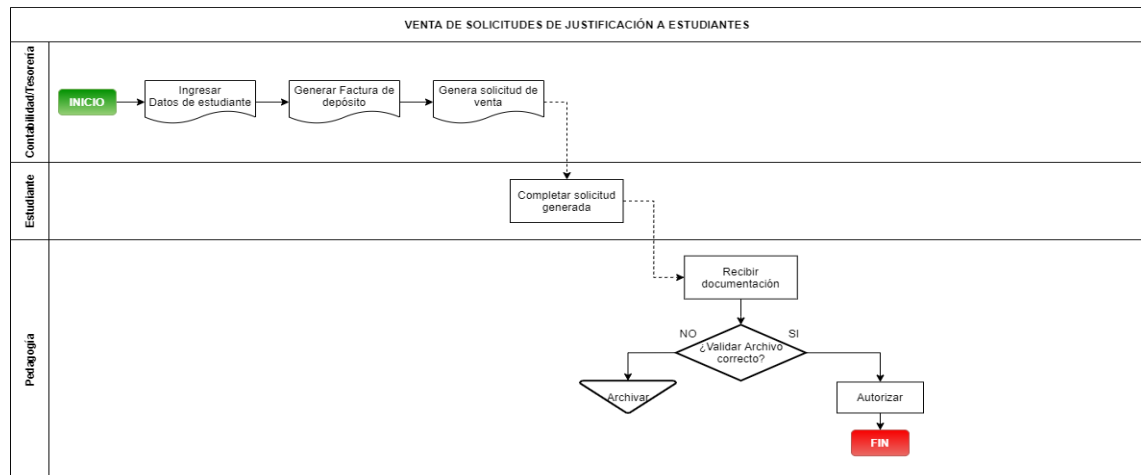


Gráfico 22-2: Proceso de venta de solicitudes de justificación a estudiantes

Realizado por: Aldas W.; Pilataxi C., 2020

2.3. Herramientas de desarrollo

En esta sección se establecerá el análisis de las distintas herramientas para el desarrollo tanto de la aplicación web como móvil, los distintos tipos de pruebas para el desarrollo de un sistema, la metodología para llevar a cabo cada una de las tareas, la normativa para el análisis de la eficiencia en tiempo de cada sistema y entre otras temáticas que serán de mucha importancia para la culminación de este proyecto, a continuación, se detallan cada uno de estos puntos tratados.

2.3.1. Visual Studio Code

El editor de Visual Studio proporciona muchas características que le facilitan la escritura y la administración del código y del texto. Puede expandir y contraer diferentes bloques de código mediante el uso de la esquematización. Puede obtener más información sobre el código mediante IntelliSense, el Examinador de objetos y la jerarquía de llamadas (Microsoft, 2020, párr. 1).

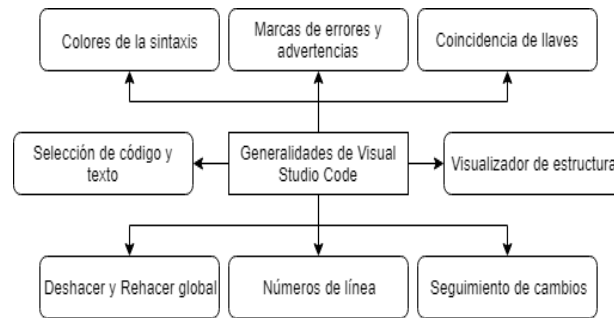


Gráfico 23-2: Generalidades de Visual Studio Code

Realizado por: Microsoft, 2020

23.2. *Postgresql*

PostgreSQL es un gestor de bases de datos relacional y orientado a objetos. Su licencia y desarrollo es de código abierto, siendo mantenida por una comunidad de desarrolladores, colaboradores y organizaciones comerciales de forma libre y desinteresadamente. Es reconocido actualmente como uno de los sistemas gestores de bases de datos relacionales más potentes del mercado. Presenta fácil accesibilidad, es multiplataforma y está disponible para su utilización en casi todos los sistemas operativos utilizados en la actualidad sin disminuir su rendimiento (PostgreSQL, 2017, párr.1).

Las ventajas de PostgreSQL de acuerdo con (PostgreSQL, 2017, párr.12) son las que se detallan a continuación:

- **Instalación y uso gratuito:** PostgreSQL es un gestor de base de datos de código libre y completamente gratuito, por lo que podemos instalarlo y utilizarlo las veces que queramos y en todos los dispositivos que queramos.
- **Sistema disponible Multiplataforma:** Es compatible con prácticamente todas las tecnologías y sistemas operativos de la actualidad.
- **Estabilidad:** PostgreSQL está hace más de 20 años en el mercado con un constante desarrollo y soporte de sus versiones.
- **Escalabilidad y configuración:** Es posible configurar de forma individual PostgreSQL según los recursos de hardware disponibles en nuestro sistema, por lo que podemos ajustar el número de CPU y cantidad de memoria disponible de para un funcionamiento óptimo.
- **Estándar SQL:** Implementa la mayor parte de las funcionalidades principales del estándar SQL, por lo que se puede realizar de forma sencilla el incluir consultas y scripts de otros motores de bases de datos.
- **Herramienta gráfica:** Incorpora una herramienta gráfica para la administración de las bases

de datos de forma fácil e intuitiva, por la cual podemos ejecutar sentencias SQL, realizar copias de seguridad o tareas de mantenimiento.

- **Robustez y fiabilidad:** PostgreSQL cumple con la característica y protocolo ACID, lo que significa Atomicidad, Consistencia, Aislamiento y Durabilidad. Por ello, se garantiza la información de la base de datos y fiabilidad en el sistema.
- **Soporte y ayuda:** A pesar de no contar con soporte telefónico o en línea, existe una infinidad de foros y páginas para nuestra ayuda. Además, la comunidad de PostgreSQL es una de las más activas.

Por otro lado las desventajas de PostgreSQL de acuerdo con (PostgreSQL, 2017, párr.13) son las que se detallan a continuación:

- PostgreSQL está diseñado específicamente para ambientes con alto volumen de datos, por lo que puede parecer lento en implementaciones para bases de datos de pequeño y mediano tamaño.
- No presenta una facilidad en comandos o sintaxis, por lo que se puede dificultar su uso si no poseemos conocimientos medios en lenguaje SQL.

233. *Arquitectura Modelo-Vista-Controlador (MVC)*

De acuerdo con (rjcodeadvance,2018,párr.1), el patrón MVC (Modelo-Vista-Controlador) fue desarrollado para aplicaciones de escritorio en los años 1970, y al pasar los años, ha sido ampliamente adaptado como arquitectura para diseñar e implementar aplicaciones web, este patrón, como su nombre lo indica, utiliza 3 componentes, modelo, vista, y controlador. Lo que hace este patrón es separar los datos y la lógica de negocio de la presentación y el módulo encargado de gestionar los eventos y las comunicaciones. Estos se detallan a continuación:

- **Modelo:** Este componente representa la información con la cual el sistema opera, por lo tanto, gestiona todos los datos, tantas consultas como actualizaciones, implementando también las reglas del negocio (rjcodeadvance,2018,párr.2).
- **Vista:** En este componente solamente está las interfaces de usuario, ya sea formularios o archivos HTML. La vista se encarga de presentar la información del modelo, es decir, mostrar los datos que se solicita al modelo, para luego mostrarlo en pantalla, a este componente se le conoce como salida (rjcodeadvance,2018,párr.3).
- **Controlador:** Se encarga de controlar, usualmente eventos que codifican las pulsaciones de las teclas o botones del mouse, es decir, controla las acciones del usuario, por lo tanto, recibe

las órdenes del usuario y se encarga de solicitar información al modelo y de comunicárselos a la vista(rjcodeadvance,2018,párr.4).

Según (rjcodeadvance.com, 2018, párr.5) algunos de sus principales beneficios del patrón de diseño MVC son:

- Menor acoplamiento
- Mayor cohesión
- Mayor flexibilidad y agilidad
- Más claridad de diseño
- Facilita el mantenimiento
- Mayor escalabilidad.

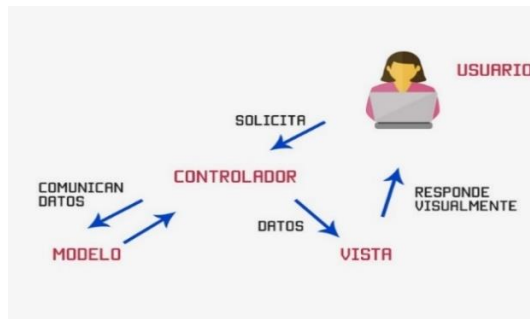


Figura 1-2: Patrón modelo vista controlador

Realizado por: CódigoFacilito, 2017

234. *Lenguajes de programación para interfaces web*

De acuerdo a (Next University, 2016, párr.1), las tecnologías de frontend son aquellas que permiten la interacción con el usuario, es decir las que permiten ver de forma gráfica y animada los datos en una pantalla de un dispositivo hardware. A continuación, se detalla las esenciales para este tipo de desarrollo:

HTML: HTML es el acrónimo en inglés de HyperText Markup Language (en español se traduce como lenguaje de marcado de hipertexto). Es un lenguaje abstracto que usan las aplicaciones para representar documentos y que puede ser transmitido fácilmente por algún medio, como Internet. Este lenguaje está definido por lo que se llama etiquetas, cuyo nombre se delimita usando los símbolos <y>, de la siguiente forma: <etiqueta>. Dichas etiquetas se utilizan para describir algo que se quiere representar en una página web (Next University, 2016,párr.2).

CSS: Las siglas CSS (Cascading Style Sheets) significan «Hojas de estilo en cascada» y parten de un concepto simple pero muy potente: aplicar estilos (colores, formas, márgenes, etc.) a uno o varios documentos (generalmente documentos HTML, páginas webs) de forma masiva. La idea de CSS es la de utilizar el concepto de separación de presentación y contenido, intentando que los documentos HTML incluyan sólo información y datos, relativos al significado de la información a transmitir (el contenido), y todos los aspectos relacionados con el estilo (diseño, colores, formas, etc.) se encuentren en un documento CSS independiente (la presentación) (lenguajecss.com, 2019,párr.5).

Javascript: JavaScript es un lenguaje de programación que permite hacer operaciones complejas en una página web. Muestra actualizaciones de contenido, animaciones gráficas e interactúa con mapas, entre otras acciones. JavaScript sirve para conseguir un mayor índice de interactividad y mejorar la experiencia de usuario. Se emplea, sobre todo, para dotar a la página de más funciones y facilitar la comunicación. (fp.uoc.fje.edu, 2019, párr.1).

Estos lenguajes que se han presentado son lo esencial para creación del frontend en un sistema web, sin embargo, a día de hoy este tipo de tecnologías no son suficientes debido a que, al escribir código de esta manera, se vuelve tedioso y la organización se hace muy difícil, por lo tanto, es necesario la implementación de frameworks, ya que estos a más de permitir la organización de archivos, aumenta la productividad e incrementa la escalabilidad de las aplicaciones.

235. *Frameworks para el desarrollo de interfaces web*

Un framework es un entorno de trabajo que tiene como objetivo facilitar la labor de programación ofreciendo una serie de características y funciones que aceleran el proceso, reducen los errores, favorecen el trabajo colaborativo y consiguen obtener un producto de mayor calidad. (Seoestudios, 2020, párr.3).

Un framework front-end es una herramienta que se integra con nuestro proyecto web para conseguir que el desarrollo sea más fácil, rápido y robusto; convirtiéndolo en una herramienta muy útil para desarrolladores principiantes y programadores con poco tiempo y poca experiencia en ámbitos de diseño. (La Salle, 2020, párr.1).

Debido a la gran cantidad de Frameworks web para el desarrollo de FrontEnd, muchas veces es difícil elegir para las necesidades presentadas, tomando en cuenta este detalle, se ha organizado una lista de tecnologías que actualmente se usan para este tipo de trabajos, en ella se da a conocer las principales fortalezas y limitaciones, posterior a los datos que se presentan en la

Tabla 2-2, se procederá a la selección de acuerdo a nuestra conveniencia.

Tabla 2-2: Comparación de Frameworks de FrontEnd

Framework	Fortalezas	Limitaciones
React JS	<ul style="list-style-type: none"> - Ahorro de tiempo al reutilizar componentes. - Virtual DOM - mejora tanto la experiencia de los usuarios como el trabajo del desarrollador. - Una biblioteca de código abierto con una diversidad de herramientas. - El código fijo es proporcionado por un movimiento de datos en una dirección. - Utilizado para desarrollar la interfaz de usuario, especialmente cuando necesita crear aplicaciones de una sola página. 	<ul style="list-style-type: none"> - Ausencia de documentación debido a un ritmo de desarrollo significativo. - La curva de aprendizaje comparativamente larga. - A los desarrolladores les resulta difícil comprender las complicaciones de JSX. - Cuando no tiene conocimientos prácticos con JavaScript, React no es la alternativa sugerida.
Angular	<ul style="list-style-type: none"> - Facilitar el procedimiento de codificación gracias a sus servicios de refactorización y navegación mejorada. - Gran ecosistema. - Angular Material reorganiza la producción de la interfaz de Material Design. - Alto rendimiento - Amplía la ejecución de aplicaciones basadas en navegador al modernizar vigorosamente los contenidos en menos tiempo, ya que ejerce el enlace de datos bidireccional. 	<ul style="list-style-type: none"> - Complicación angular - Reubicación de esquemas heredados de AngularJS a Angular. - La documentación de la CLI no está bien definida. - El esfuerzo de aprendizaje.
Vue.js	<ul style="list-style-type: none"> - Documentación extensa y detallada. - Sencillez y claridad. - Extensiones de herramientas de desarrollo del navegador - Reutilización de código e integración simple 	<ul style="list-style-type: none"> - Comunidad de desarrolladores reducida. - La flexibilidad conduce a irregularidades en el código
Ember.js	<ul style="list-style-type: none"> - Marco de desarrollo más rápido - Bien gestionado - Documentación apropiada - Enlace de datos bidireccional 	<ul style="list-style-type: none"> - Actualizaciones lentas y sintaxis desafiante - Una comunidad menor y menos popular - Marco sustancial para aplicaciones pequeñas - Curva de aprendizaje difícil
Svelte	<ul style="list-style-type: none"> - Marco escalable - Reactividad mejorada - Es liviano, simple y utiliza las bibliotecas de JavaScript predominantes. - El más nuevo de todos - Más rápido que cualquier otro marco, es decir, React o Angular 	<ul style="list-style-type: none"> - Falta de apoyo - Todavía no es muy popular - Comunidad menor - Falta de herramientas

Fuente: monocubed, 2021

Realizado por: Aldas W.; Pilataxi C., 2020

Teniendo en cuenta las características principales de los frameworks más utilizados para el desarrollo del frontend en aplicaciones web, se ha determinado que la opción adecuada es React JS, debido a la facilidad de uso que presenta por estar basado en Javascript, por otra parte, el uso

de esta herramienta permite la escalabilidad del software desarrollado, ya que la manera de estructurar un sistema es por componentes, es decir que se escribe una vez y se lo utiliza en varias partes del sistema, agilizando el proceso de desarrollo.

23.6. Tecnologías para desarrollar servicios web

Este grupo de tecnologías están encargadas de proporcionar datos de manera ordenada y organizada, ya que estos tienen la responsabilidad de manejar y procesar los datos que tiene un sistema, es decir que aquí se encuentran los algoritmos de los procesos de una empresa o institución. Las tecnologías backend suelen recurrir a los frameworks para que los usuarios puedan interactuar y desempeñar ciertas tareas o generar respuestas (sumatd, 2019, párr.1).

En la **Tabla 3-2** se detalla las fortalezas y debilidades de los principales lenguajes de programación para el desarrollo del backend, lo cual servirá como referencia para seleccionar el que mejor se adapte a nuestra conveniencia.

Tabla 3-2: Comparación de lenguajes de BackEnd

Lenguajes	Fortalezas	Limitaciones
PHP	<ul style="list-style-type: none"> - Orientado a objetos - Fácil de automatizar - Comunidad amplia - Documentación necesaria - Soporte de bases de datos más utilizadas. - Fácil de usar - De código abierto - Versátil - Seguro 	<ul style="list-style-type: none"> - Popularidad en descenso. - Baja competencia con actuales lenguajes de backend - Se le da usos incorrectos. - Necesita un servidor para su funcionamiento.
Javascript	<ul style="list-style-type: none"> - Desarrollo rápido. - Tecnología de backend indulgente, es decir que se puede usar cualquier tipo de middleware. - Desarrollo rentable, ya que funciona como backend y frontend. - De código abierto. - Menos gastos generales de secuencias de comandos. 	<ul style="list-style-type: none"> - Las funciones controladas por eventos son muy complicadas. - La mayoría de los programadores que utilizan JavaScript no pueden comprender el middleware que se utiliza con él. - JavaScript requiere MySQL para servicios de bases de datos que son muy complejos y desactualizados. - La mayoría de los desarrolladores no prefieren la libertad que ofrece el marco de backend de JavaScript.
Python	<ul style="list-style-type: none"> - Fácil de aprender - Hay muchas bibliotecas disponibles - Funciones de IoT - Códigos incrustados - Desarrollo rentable por la gran 	<ul style="list-style-type: none"> - La capa de acceso a la base de datos no está tan desarrollada como otros lenguajes. - Los programas desarrollados en Python requieren muchas pruebas y

Lenguajes	Fortalezas	Limitaciones
	cantidad de librerías y bibliotecas disponibles.	depuración. - Python depende en gran medida de bibliotecas y marcos de trabajo de terceros. - Si se interrumpe la aplicación, el tiempo de ejecución se vuelve muy lento.
Java	<ul style="list-style-type: none"> - Escalable. - Fácil de entender. - Compatible con subprocesos múltiples, es decir que permite varias peticiones simultáneamente. - Bibliotecas de código abierto. - Opciones de seguridad mejoradas. 	<ul style="list-style-type: none"> - El lenguaje de programación Java requiere mucho tiempo. - No se realiza ninguna programación de bajo nivel en Java. - No hay comandos para la recolección de datos basura. - Java es un lenguaje que se ejecuta mejor en sistemas costosos o de alta gama. Hace que el desarrollo de Java sea muy caro. - En la sección GUI de las aplicaciones de desarrollo de Java, faltan muchas cosas necesarias, como herramientas y objetos de interfaz de moda.

Fuente: Back4app, 2021

Realizado por: Aldas W.; Pilataxi C., 2021

Con el análisis realizado previamente, se ha determinado que el lenguaje a utilizar como soporte en el backend es php, principalmente por conveniencias de facilidad de uso y además por la gran comunidad de desarrolladores que han hecho de este lenguaje el más popular, extendiendo así una amplia gama de documentación. Aparte de la facilidad de uso, se debe destacar la amplia existencia de servidores que soporten este lenguaje a diferencia de otros lenguajes de nueva generación.

23.7. *Framework React JS*

React es una biblioteca escrita en JavaScript, desarrollada en Facebook para facilitar la creación de componentes interactivos, reutilizables, para interfaces de usuario. Se utiliza en Facebook para la producción de componentes, e Instagram está escrito enteramente en React. (Next Universty, 2016, párr.2).

React.js está construido en torno a hacer funciones, que toman las actualizaciones de estado de la página y que se traduzcan en una representación virtual de la página resultante. Siempre que React es informado de un cambio de estado, vuelve a ejecutar esas funciones para determinar una nueva representación virtual de la página, a continuación, se traduce automáticamente ese resultado en los cambios del DOM necesarios para reflejar la nueva presentación de la página. (Next Universty, 2016, párr.3).

Según (freecodecamp, 2020, párr.47) las características principales de React JS son que se describen a continuación:

- **Lenguaje:** React se basa en JavaScript ES6+ combinado con JSX script. JSX es una extensión para la sintaxis, lo que hace que un código JavaScript se parezca a un código escrito en HTML. Esto hace que el código sea más fácil de entender y los errores tipográficos son más fáciles de detectar. Para que el código JSX se compile en un navegador, React se complementa con Babel, una herramienta de traducción de código.
- **Desempeño:** El rendimiento de React ha mejorado enormemente con la introducción del DOM virtual. Dado que todos los árboles DOM virtuales son ligeros y están integrados en el servidor, se reduce la carga en el navegador.
- **Estructura de la aplicación:** La estructura de React ofrece a los desarrolladores la libertad de elegir. No existe “una única estructura correcta” para una aplicación React.

La arquitectura de una aplicación React está basada en componentes. El código está hecho de componentes React, que se renderizan con la biblioteca React DOM y se dirigen de dos maneras: funcional y basado en clases. (freecodecamp, 2020)

```
function Hola(props){  
  return <div>Hola {props.nombre}</div>  
}
```

Figura 24-2: Estructura de una función en React

Realizado por: monocubed, 2021

```
class Hola extends React.Component {  
  render() {  
    return <h1>Hola, {this.props.nombre}</h1>;  
  }  
}
```

Figura 3-2: Estructura de una clase en React

Realizado por: monocubed, 2021

- **Gestión de Estado:** En React, cada componente tiene su propio estado. Un desarrollador de React puede crear componentes especiales para mantener el estado de toda la aplicación o una parte en concreto de ella. La principal desventaja aquí consiste en el hecho de que el estado global debe almacenarse en múltiples partes diferentes de la aplicación y los datos se pasan manualmente a los diferentes niveles del árbol de componentes.

```
class Reloj extends React.Component {
  constructor(props) {
    super(props);
    this.state = {date: new Date()};
  }
  render() {
    return (
      <div>
        <h1>¡Hola Mundo!</h1>
        <h2>Ahora es {this.state.date.toLocaleTimeString()}.</h2>
      </div>
    );
  }
}
```

Figura 25-2: Estado de los componentes en React

Realizado por: monocubed, 2021

Para resolver este problema, existe una librería de gestión especial del estado: Redux. La idea de está es que el estado global se representa como un único objeto con estado, que se modifica en diferentes partes de la aplicación con la ayuda de reductores: funciones especiales de Redux.

- **Herramientas:** React es compatible con varios editores de código. Por ejemplo, el código en React se puede editar con Sublime Text, Visual Studio y Atom. Para iniciar un proyecto, puede usar la herramienta Create React App (CLI). A su vez, la renderización del lado del servidor se completa con el uso del framework Next.js.

238. *Framework Laravel*

El objetivo principal que persiguen este tipo de frameworks, es que los procesos de desarrollo estén más acelerados, que se puedan volver a utilizar los fragmentos de códigos que ya existen y que se realicen buenas prácticas mediante el uso de distintos patrones establecidos. Esto quiere decir que tendrá una mayor productividad en todos los ámbitos en los que sea utilizado, y se establecerá un coste mínimo en el desarrollo de aplicaciones. (Morelli, 2019, párr..5)

Laravel es el nombre de uno de los frameworks de desarrollo web PHP más populares. Se destaca por ser simple, intuitivo y poseer una sintaxis elegante. El cuidado con la calidad del código es uno de los principales pilares de su arquitectura. Esto es incluso lo que lo hace equivalente al trabajo de un artesano por su comunidad de usuarios. De ahí viene el eslogan “el framework PHP para artesanos de la web”, muy utilizado para describirlo (Hostgator, 2021, párr.1).

A pesar de las bondades ofrecidas por el lenguaje php, la manera más adecuada de desarrollar servicios del lado del servidor es mediante un framework, que ofrezca la mayor facilidad de cambios, soporte, etc., razón por la cual también se ha optado por el uso del Framework Laravel para la creación de APIs Rest.

239. Tipos de aplicaciones móviles

De acuerdo a (crehana, 2020, párr.8), “App” es como generalmente se les dice a los diferentes tipos de aplicaciones, es una abreviatura de la voz inglesa “application” y es utilizada para hacer referencia a las aplicaciones informáticas que se crean para móviles, tabletas y hasta relojes. Por lo que, app, aplicación o aplicaciones móviles se refiere a un software creado y pensado para los dispositivos electrónicos que las personas pueden descargar y acceder directamente. Los diferentes tipos de aplicaciones móviles se detallan a continuación:

Aplicaciones Nativas

Son aquellas aplicaciones diseñadas para una serie determinada de dispositivos (tales como: teléfono o tableta) que se instalan en los dispositivos y que se encuentran en las diferentes plataformas digitales de aplicaciones móviles (como, por ejemplo: Play Store o App Store). Es decir que son diseñadas para dispositivos junto con sistemas operativos específicos, Android y iOS son dos de los más comunes. (crehana, 2020,párr.17).

Aplicaciones Híbridas o Multiplataforma

También llamadas aplicaciones multiplataforma son aquellas desarrolladas con lenguajes de programación distintos a los lenguajes nativos de cada plataforma. La mayoría de los videojuegos móviles que tienen gráficos renderizados 2D o 3D son aplicaciones híbridas, ya que existen muchas frameworks multiplataforma que ayudan en su desarrollo a este tipo de aplicación móvil (crehana, 2020, párr.12).

Aplicaciones móviles web o Progresivas

Este tipo de aplicación tiene cierta similitud con las aplicaciones nativas; la principal diferencia radica en que, para acceder a ellas, es necesario un navegador web, por ejemplo; Google Chrome, Opera o Mozilla Firefox. (crehana, 2020, párr.21)

A diferencia de una aplicación móvil nativa, estas no requieren espacio dentro del teléfono ya que no requieren una previa instalación para poder hacer uso de ellas. Estas están diseñadas para adaptarse al usuario adaptando su código e interfaz al dispositivo cualquiera que sea su sistema operativo. (crehana, 2020, párr.22)

2.3.9.1. Ventajas y desventajas de los diferentes tipos de aplicaciones

Para destacar las características de desarrollo de los diferentes tipos de aplicaciones móviles, se ha definido una tabla comparativa en la que se establece las principales ventajas y desventajas de cada uno de estos, mismos que se detallan en la **Tabla 4-2**.

Tabla 4-2: Ventajas y desventajas de los diferentes tipos de aplicaciones móviles

Tipo	Ventajas	Desventajas
Híbridas	<ul style="list-style-type: none">- Se puede distribuir en la App Store y Google Play.- Está construida bajo HTML, CSS y JavaScript.- Tiene un código base multiplataforma.- Tiene acceso al hardware del dispositivo.	<ul style="list-style-type: none">- Su experiencia de usuario es básica.- El diseño no siempre está adaptado al tipo de dispositivo.
Nativas	<ul style="list-style-type: none">- Tiene acceso al dispositivo.- Ofrece una mejor experiencia de usuario.- Se puede distribuir en la App Store.- Envía notificaciones push.- Tiene actualizaciones constantes.	<ul style="list-style-type: none">- Diferentes herramientas para cada plataforma.- Requieren una gran inversión para desarrollarlas.- Su código no se puede utilizar en diferentes plataformas.
Progresivas	<ul style="list-style-type: none">- Su proceso de desarrollo es más sencillo y hasta económico.- Su código base se puede utilizar en múltiples plataformas.- No necesitan ningún permiso para distribución, ya que se ejecutan en el navegador.- Siempre se encuentra actualizada.- Está desarrollado para adaptarse a cualquier dispositivo.	<ul style="list-style-type: none">- Requieren de conexión a internet para funcionar.- No poseen acceso por completo al hardware del dispositivo.- La experiencia de usuario, navegación o interacción es básica y a veces nula.

Fuente: crehana, 2020

Realizado por: Aldas W.; Pilataxi C., 2020

23.10. Frameworks de desarrollo de aplicaciones móviles multiplataforma

Actualmente existen muchos frameworks de desarrollo de aplicaciones móviles, por lo cual se vuelve difícil elegir el más apropiado, dado los variados servicios que cada uno ofrece al programador, de este modo en la **Tabla 5-2**, se establece los más destacados, en el cual se detalla las características esenciales, sus ventajas y desventajas.

Tabla 5-2: Frameworks de desarrollo de aplicaciones móviles multiplataforma

Framework	Características	Ventajas	Desventajas
React Native	<ul style="list-style-type: none"> - El propósito de React Native es implementar apps nativas adecuadas, en vez de crear aplicaciones híbridas que solamente se ejecuten en un navegador. - El desarrollo se realiza a través de Javascript y, lógicamente, React. - Hace poco, este framework anunció que soportaba Android, así que podrás desarrollar también aplicaciones multiplataforma. 	<ul style="list-style-type: none"> - Aplicaciones nativas - Gran comunidad 	<ul style="list-style-type: none"> - Curva de aprendizaje pronunciada
Ionic	<ul style="list-style-type: none"> - Permite crear increíbles diseños con Css - Esta sincronizado con Angular JS. - Emuladores integrados y una app packager basada en Cordova 	<ul style="list-style-type: none"> - Funciona con elementos predefinidos - Gran comunidad - Interfaz de línea de comandos con increíbles características 	<ul style="list-style-type: none"> - Necesitas conocer Angular JS para desarrollar aplicaciones complejas
jQuery Mobile	<ul style="list-style-type: none"> - Su propósito es ayudar a las aplicaciones web a que funcionen correctamente en todos los navegadores móviles (incluidos los dinosaurios de Windows Phone, Blackberry y Symbian). - Es muy ligero, solo depende de jQuery y, además, es muy sencillo de aprender. - Cuenta con soporte para el táctil y también para Phonegap y Cordova. 	<ul style="list-style-type: none"> - Soporte para todos los navegadores - Simple de utilizar 	<ul style="list-style-type: none"> - Estilo que no recuerdan a iOS y Android - Builder de Phonegap/Cordova no incluido, pero soportado.
Native Script	<ul style="list-style-type: none"> - Permite escribir la funcionalidad de tu aplicación en JavaScript - Este framework requiere algunos conocimientos de programación, pero lo compensa con una extensa documentación en profundidad. 	<ul style="list-style-type: none"> - El enfoque: escríbelo una vez y úsalo donde quieras. - Gran documentación. 	<ul style="list-style-type: none"> - Curva de aprendizaje pronunciada - Comunidad pequeña.

Fuente: Programación en Castellano, 2019

Realizado por: Aldas W.; Pilataxi C., 2020

Al realizar un análisis de los frameworks más importantes de desarrollo de aplicaciones multiplataforma, se ha llegado a la conclusión de que la herramienta que ofrece mayor capacidad de escalabilidad, mantenimiento y fácil aprendizaje es React Native, el scripting de desarrollo de esta es muy similar a React JS que a su vez utiliza el lenguaje Javascript como su principal dependencia, lo cual ha permitido forjarse una amplia comunidad de desarrolladores y documentación a la par.

23.II. Framework React Native

React Native es un framework JavaScript para crear aplicaciones reales nativas para iOS y Android, basado en la librería de JavaScript React para la creación de componentes visuales, cambiando el propósito de los mismos para, en lugar de ser ejecutados en navegador, correr directamente sobre las plataformas móviles nativas, en este caso iOS y Android. Es decir, en lugar de desarrollar una aplicación web híbrida o en HTML5, lo que obtienes al final como resultado es una aplicación real nativa, indistinguible de la que podrías desarrollar con código en Objective-C o Java (Deloitte, 2020, párr.1).

Con esta idea de construcción de aplicaciones React Native de acuerdo con (Deloitte, 2020) nos proporciona las siguientes funcionalidades:

- **Compatibilidad Cross-Platform:** ya que la mayoría de las APIs de React Native lo son de por sí, lo cual ayuda a los propios desarrolladores a crear aplicaciones que puede ser ejecutados tanto en iOS como Android simultáneamente con el mismo código base.
- **Funcionalidad nativa:** las aplicaciones creadas mediante React Native funcionan de la misma manera que una aplicación nativa real creada para cada uno de los sistemas usando su lenguaje nativo propio.
- **Actualizaciones instantáneas (para desarrollo y/o test):** con la extensión de JavaScript, los desarrolladores tienen la flexibilidad de subir los cambios contenidos en la actualización directamente al dispositivo del usuario sin tener que pasar por las tiendas de aplicaciones propias de cada sistema y sus tediosos ciclos de procesos obligatorios previos.
- **Sencilla curva de aprendizaje:** React Native es extremadamente fácil de leer y sencillo de aprender ya que se basa en los conceptos fundamentales del lenguaje JavaScript, siendo especialmente intuitivo tanto para los ya expertos en dicho lenguaje o incluso para las personas sin experiencia en él, ya que nos provee de un rango muy amplio de componentes.
- **Experiencia positiva para el desarrollador:** si bien la curva de aprendizaje hemos dicho que es sencilla, también el propio lenguaje nos motiva y ayuda a la hora de la evolución según aumentamos nuestro conocimiento y dominio del mismo.

Para la implementación de React Native se debe conocer los principios de funcionamiento de React Native que son prácticamente idénticos a React, excepto que React Native no manipula el DOM a través del DOM virtual. Se ejecuta en un proceso en segundo plano (que interpreta el JavaScript escrito por los desarrolladores) directamente en el dispositivo final y se comunica con la plataforma nativa a través de un puente serializado, asíncrono y por lotes (Deloitte, 2020, párr.10).

23.12. Biblioteca para el desarrollo de interfaces

Material-UI es una biblioteca de código abierto que implementa el lenguaje visual de «materiales» de Google en sus componentes React. Ofrece la capacidad de combinar su biblioteca de interfaz de usuario, con el marco front-end de React.js. (Tajamar 2020, párr.1).

Con Material UI se pueden crear diseños profesionales modernos con personalización completa para aplicaciones sin las limitaciones de tiempo de iniciar CSS completo desde cero. De acuerdo con (Tajamar, 2020, párr.2) los pros destacados de Material UI son:

- Implementa material design en sus componentes
- Agiliza el desarrollo del front-end
- Implementa buenas prácticas como el uso de hooks
- Documentación completa
- Es customizable

23.13. Administrador de paquetes de Node (NPM)

NPM son las siglas de Node Package Manager, es decir, gestor de paquetes de NodeJS, un entorno de ejecución multiplataforma para ejecutar Javascript no sólo en un navegador web (como se concibió originalmente) sino fuera de él, y poder utilizarlo en sistemas de escritorio o servidores web. Este gestor de paquetes (muy similar al concepto de apt-get en GNU/Linux), nos permitirá instalar de forma muy sencilla y automática paquetes Javascript (tanto de Node como Javascript para el navegador) y utilizarlo para nuestros fines (lenguajejs, 2018, párr.1).

Según (lenguajejs, 2018, párr.12) las 2 principales modalidades para utilizar NPM son:

- **A nivel de proyecto:** Probablemente la modalidad más utilizada es la de usar NPM como un gestor de dependencias de un proyecto, esto es, un sistema con el que controlamos que paquetes o librerías Javascript están instalados (y que versión), de modo que quedan

asociados al proyecto en sí. Esto facilita que, si un usuario diferente se descarga el proyecto, pueda gestionarlo fácil y rápidamente.

- **A nivel global:** Existen algunas situaciones específicas, en las que los paquetes son realmente utilidades que no se utilizan en proyectos, muy común en aplicaciones de línea de comandos (CLI) que usamos desde terminal. En esta modalidad, los paquetes se instalan a nivel del sistema, por lo que están disponibles siempre que el usuario quiera utilizarlos, sin necesidad de tenerlo en cada proyecto.

23.14 Librería axios

Axios es una librería JavaScript que puede ejecutarse en el navegador y que nos permite hacer sencillas las operaciones como cliente HTTP, por lo que podremos configurar y realizar solicitudes a un servidor y recibiremos respuestas fáciles de procesar. Aunque podría resultarnos muy útil en infinidad de situaciones, deberíamos antes analizarla las casuísticas con cuidado para determinar si es la mejor solución (García, 2019, párr.1).

De acuerdo con (García, 2019, párr.3) axios es una alternativa que nos brinda multitud de ventajas:

- La API es unificada para las solicitudes Ajax.
- Está optimizado para facilitar el consumo de servicios web, API REST y que devuelvan datos JSON.
- De fácil utilización y como complemento perfecto para las páginas convencionales.
- Pesa poco, apenas 13KB minimizado. Menos aún si se envía comprimido al servidor.
- Compatibilidad con todos los navegadores en sus versiones actuales.

23.15 Librería redux

Redux es un patrón de arquitectura de datos que permite manejar el estado de la aplicación de una manera predecible. Está pensado para reducir el número de relaciones entre componentes de la aplicación y mantener un flujo de datos sencillo (desarrolloweb, 2019, párr.1).

Según (desarrolloweb, 2019, párr.4) estos son los beneficios que aporta el patrón Redux a la aplicación:

- Arquitectura escalable de datos
- Mayor control sobre el flujo de datos y el estado de la aplicación
- Estado global e inmutable

2.3.15.1. Arquitectura de datos de Redux

Redux al ser una nueva herramienta de datos para el desarrollo de aplicaciones web, su implementación se ha visto muy limitada debido a la gran complejidad de aprendizaje, siendo así que muchos desarrolladores lo ignoran y prefieren organizar los datos de manera tradicional, lo cual muchas veces se vuelve difícil de manejar, sin mencionar que la cantidad de líneas de código se ve incrementado, reflejando el mal diseño de desarrollo, por lo cual en la **Gráfico 24-2**, se da a conocer de manera gráfica la simplicidad de esta arquitectura de datos, esperando que sea de mayor comprensión al momento de implementarlo (desarrolloweb, 2019, párr.11).

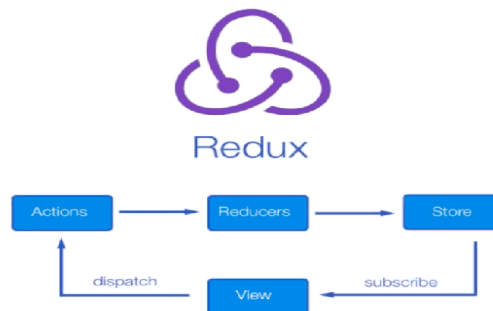


Gráfico 24-2: Arquitectura de datos en Redux

Realizado por: desarrolloweb, 2019

A continuación se muestran los principios básicos de Redux según (desarrolloweb, 2019, párr.15)

- Store

El store es el estado al que todos atienden. El único estado de la aplicación válido está en el store. Los componentes que necesiten conocer al estado acudirán al store para recuperarlo.

Los datos del store viajan a los componentes en una única dirección. Esto se traduce en que el data-binding que usaremos para enviar al store hacia los componentes será one-way, de una única dirección (desarrolloweb, 2019, párr.17).

- El estado (State) es de solo lectura

El estado en Redux es un único objeto Javascript, organizado en modo de árbol (como JSON), que contiene todos los datos que la aplicación va a manejar (desarrolloweb, 2019, párr.19).

El estado de una aplicación podría parecerse a algo como:

```
{
  hero: "Super Yo",
  supervillanosPreferidos: ["Joker", "Magneto"],
  vidas: 5,
  nivel: 1
}
```

Figura 5-2: Estado solo de lectura

Realizado por: desarrolloweb, 2019

Los componentes que manejan los datos del estado solo leen los datos, no los manipulan. En el caso que ocurra una manipulación permanece sólo en ese componente y no se transfiere al resto de la aplicación. Para poder alterar el estado de la aplicación se usarán acciones. Las acciones (actions) representan la única vía de producirse una modificación en el estado global de la aplicación (desarrolloweb, 2019, párr.21).

Esta es la forma que podrían tener las acciones:

```
{ type: "INCREMENTAR_NIVEL" }
```

```
{ type: "DECREMENTAR_VIDA" }
```

```
{ type: "ANADIR_VILLANO", nombre: "Dr. Doom" }
```

Figura 26-2: Acciones en React

Realizado por: desarrolloweb, 2019

Como hemos dicho, las dos primeras acciones no requieren ningún dato adicional, porque se entiende que tienen que sumar o restar una unidad, pero en la acción de añadir un villano necesitamos indicar el nombre (desarrolloweb.com, 2019, párr.22).

- Los cambios se hacen por medio funciones puras

Una vez emitidas las acciones, estas se procesan por medio de funciones puras, a las que se denominan “reducers”. El reducer siempre recibe dos parámetros: una acción y el estado anterior. Contiene la lógica para procesar tales acciones y como consecuencia de ellas puede modificar al estado. Una vez ejecutada la lógica, el reducer devuelve el nuevo estado. Básicamente nuestros reducers se podrían parecer a un código como este, en el que tenemos un switch con diferentes case para cada acción. Aunque lógicamente, cuando la aplicación se hace grande se podría organizar el código de otra manera, por ejemplo, haciendo una función que se encargue de mantener cada parte del estado (desarrolloweb, 2019, párr.23).


```

function reducers(state, action) {
  switch (action.type) {
    case 'INCREMENTAR_NIVEL':
      // Ejecutar la lógica
      // Devolver un nuevo estado
    case 'DECREMENTAR_NIVEL':
      // ...
    default:
      return state
  }
}

```

Figura 27-2: Funciones reductoras en React

Realizado por: desarrolloweb, 2019

El concepto de reducers como “funciones puras” quiere decir que éstas no provocan ningún tipo de efecto colateral. Se encargan de modificar al estado, ellas mismas, y nada más. Esto quiere decir que no modifican directamente partes de la interfaz, no almacenan en base de datos, no producen solicitudes Ajax a servicios web, no llaman a otras funciones para realizar su trabajo, etc (desarrolloweb, 2019, párr.25).

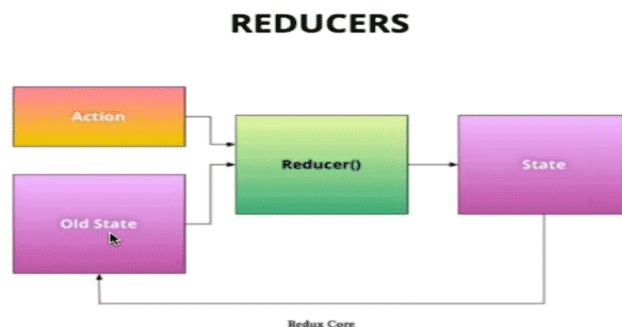


Gráfico 28-2: Diagrama de las funciones reductoras

Realizado por: desarrolloweb, 2019

2.3.15.2. *Uso de redux*

Cuando creamos aplicaciones en React, éstas se organizan como una serie de componentes anidados, su naturaleza es funcional. En otras palabras, reciben información a través de sus argumentos (props) y pasan la información a través de sus valores de retorno y a esto se le llama one-way binding, los datos sólo se transmiten de los componentes a sus hijos (Tijero, 2019, párr.3).

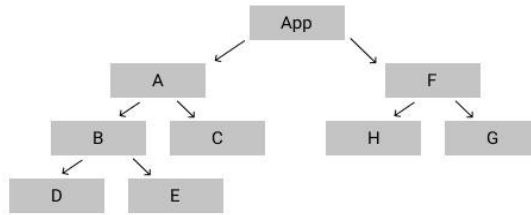


Gráfico 26-2: Árbol de componentes de React

Realizado por: Tijero, 2019

Entonces, necesitamos tener toda esa información en un solo lugar, para que pueda estar disponible en todos nuestros componentes, algo como nuestra única fuente de la verdad: (Tijero, 2019, párr.4)

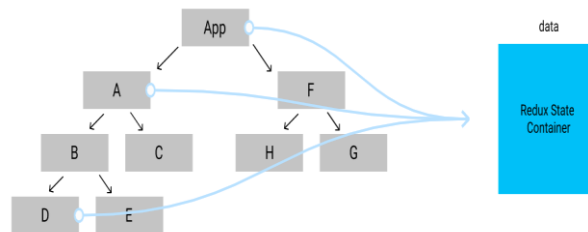


Gráfico 29-2: React con Redux

Realizado por: Tijero, 2019

Teniendo toda esa información en nuestros estados Redux todos nuestros componentes tendrán acceso a la misma información en todo momento (Tijero, 2019, párr.5).

23.16 Librería jest

Jest es una librería abierta para pruebas en JavaScript desarrollada por Facebook. Su eslogan es “Pruebas de JavaScript deliciosas”. Si bien Jest se puede usar para probar cualquier librería de JavaScript, brilla cuando se trata de React y React Native. (tutsplus, 2020, párr.1).

De acuerdo con (Accomazo,2017, p.294) establece que las bibliotecas de prueba tienen por lo general tres componentes principales que son:

- El corredor de pruebas. Esto es lo que ejecuta en la línea de comandos. El ejecutor de pruebas es responsable de encontrar sus pruebas, ejecutarlas e informarle los resultados en la consola.
- Un lenguaje específico de dominio para organizar sus pruebas. Como veremos, estas funciones nos ayudan a realizar tareas comunes como organizar la configuración y el

desmontaje antes y después de la ejecución de las pruebas.

- Una biblioteca de afirmaciones. Las funciones de aserción proporcionadas por estas bibliotecas nos ayudan a realizar fácilmente aserciones complejas, como verificar la igualdad entre los objetos de JavaScript o la presencia de ciertos elementos en una matriz.

A continuación, se muestra la forma de estructurar una prueba en jest:

El script de prueba es un archivo que contiene la suite de pruebas que a su vez esta contiene a los casos de prueba. Se puede tener varias suites de pruebas dentro de un mismo script, estas representan el conjunto de condiciones a probar de una funcionalidad en específico. En la **Figura 8-2**, se detalla cada uno de los elementos descritos anteriormente (paradigmadigital, 2018, parr.6).

```
//SCRIPT DE PRUEBA DE MYOBJECT
import MyObject from "./MyObject";

describe("Suite de pruebas de MiComponente", () => {

  beforeEach(() => {
    //Pre-condiciones
    //Inicialización de datos de prueba
  });
  afterEach(() => {
    //Post-condiciones
    //Destrucción de datos de prueba
  });

  it("Prueba de unidad que debe pasar", () => {
    expect(MyObject.contain(['Condición de la prueba'])).matcher('Resultado esperado');
  });
});
```

Figura 30-2: Estructura de un script de prueba con Jest

Realizado por: Aldas W.; Pilataxi C., 2020

De acuerdo con (paradigmadigital, 2018, párr.8) las partes de un script, con una o varias suites y uno o varios casos de prueba se deben estructurar mediante las siguientes partes:

- *MyObject*: Representa la importación del elemento a ser testeado, este elemento por lo general representa una clase de JavaScript, al cual se le aplica una suite de pruebas.

Una suite de pruebas en jest está representado por la función describe() que a su vez es el conjunto de pruebas unitarias referentes a un elemento en particular.

- *describe(['group description'], () => ...)*: Nos permite agrupar nuestros bloques de test en un mismo fichero de manera separada y así tener más organizado nuestro bloque de test. Si queremos probar diferentes partes de un mismo método que contenga “if” o “switch”.

- *beforeEach(() => ...)*: Se escribe dentro de nuestra agrupación (describe). Ejecuta un código antes de cada uno de los test, por ejemplo, para reinicializar una clase, componente, método...
- *afterEach(() => ...)*: Se escribe dentro de nuestra agrupación (describe) y ejecuta un código después de cada uno de los test. Nos sirve para reinicializar una clase, componente, método
- *it('[test description]', () => ...)*: Dentro de cada uno se escriben nuestros test y los agrupamos mediante el describe.
- *expect()*: Se escribe dentro del test (it) para decir qué esperamos que ocurra con lo que hemos escrito.
- *matchers()*: Jest usa “matchers” para permitirle probar valores de diferentes maneras.

En las pruebas, a veces es necesario distinguir entre undefined, null, y false, pero a veces no desea tratarlos de manera diferente. Jest contiene helpers que permiten ser explícito sobre lo que se quiere (paradigmadigital, 2018, párr.12).

A continuación, de acuerdo con (jestjs, 2019) se presentan algunos ejemplos de helpers:

- *toBeNull* coincide solo con *null*
- *toBeUndefined* coincide solo con *undefined*
- *toBeDefined* es el opuesto a *toBeUndefined*
- *toBeTruthy* coincide con lo que sea que el condicional if devuelva como true
- *toBeFalsy* coincide con lo que sea que el condicional if devuelva como false

2.4. Entorno de pruebas de software

Las pruebas de software son procesos cíclicos que permiten a los desarrolladores verificar si un requerimiento o cambio en el sistema es correcto o no. Adicionalmente permite verificar que una versión específica del software tenga todos los requerimientos solicitados. También pueden ayudar a verificar que el software puede funcionar correctamente en diferentes medios o con diferentes integraciones, como por ejemplo en diferentes sistemas operativos, o en diferentes versiones de software base (Andrader, 2021,párr.5).

2.4.1. Principales beneficios de las pruebas de software

Los beneficios del desarrollo de pruebas según (Andrader, 2021, párr.11), son las siguientes:

- **Asegura funcionalidad completa:** Asegura que todos los requerimientos desarrollados estén presentes en el producto final.
- **Alerta temprana:** Permite saber muy temprano en el desarrollo de software acerca de defectos que pueden afectar negativamente al usuario.
- **Reconstrucción de código:** Si tenemos un conjunto de pruebas muy amplio, es mucho más fácil hacer reconstrucciones internas del código para mejorar arquitectura sin afectar al usuario final.

242. Verificación y validación

La verificación es un tipo de actividad que nos ayuda a evaluar un sistema de software al determinar si el producto de una determinada fase de desarrollo satisface los requisitos establecidos antes del inicio de esa fase. Se puede observar que un producto puede ser un producto intermedio, como la especificación de requisitos, la especificación de diseño, el código, el manual del usuario o incluso el producto final. Las actividades que comprueban la exactitud de una fase de desarrollo se denominan actividades de verificación (Kshirasagar,2008, p.8).

La validación es un tipo de actividades que nos ayudan a confirmar que un producto cumple con su uso previsto. Las actividades de validación tienen como objetivo confirmar que un producto cumple con las expectativas de sus clientes (Kshirasagar,2008, p.8).

243. Objetivos del testing

Según (Kshirasagar,2008, p.10), diferentes partes interesadas ven un proceso de prueba desde diferentes perspectivas, como se explica a continuación:

- **Funciona:** mientras implementa una unidad de programa, el programador puede querer probar si la unidad funciona o no en circunstancias normales. El programador obtiene mucha confianza si la unidad funciona a su satisfacción. La misma idea se aplica también a todo un sistema: una vez que se ha integrado un sistema, los desarrolladores pueden querer probar si el sistema realiza o no las funciones básicas.
- **No funciona:** una vez que el programador (o el equipo de desarrollo) está satisfecho de que una unidad (o el sistema) funciona hasta cierto punto, se realizan más pruebas con el objetivo de encontrar fallas en la unidad (o el sistema). Aquí, la idea es intentar hacer que la unidad (o el sistema) falle.

- **Reduzca el riesgo de fallas:** la mayoría de los sistemas de software complejos contienen fallas que hacen que el sistema falle de vez en cuando. Este concepto de “fallar de vez en cuando” da lugar a la noción de tasa de fallas. A medida que se descubren y corrigen las fallas mientras se realizan más y más pruebas, la tasa de fallas de un sistema generalmente disminuye. Por lo tanto, un objetivo de nivel más alto de realizar pruebas es reducir el riesgo de fallar a un nivel aceptable.
- **Reducir el costo de las pruebas:** los diferentes tipos de costos asociados con un proceso de prueba incluyen: el costo de diseñar, mantener y ejecutar casos de prueba, el costo de analizar el resultado de ejecutar cada caso de prueba, el costo de documentar los casos de prueba, y el costo de ejecutar realmente el sistema y documentarlo.
Por lo tanto, cuanto menor sea el número de casos de prueba diseñados, menor será el costo asociado de las pruebas. Sin embargo, producir una pequeña cantidad de casos de prueba arbitrarios no es una buena forma de ahorrar costos. El nivel más alto de objetivo de realizar pruebas es producir software de bajo riesgo con un menor número de casos de prueba. Esta idea nos lleva al concepto de efectividad de los casos de prueba. Por lo tanto, los ingenieros de pruebas deben seleccionar juiciosamente menos casos de prueba efectivos (Kshirasagar,2008, p.10).

2.4.4. Niveles de prueba

Las pruebas se realizan en diferentes niveles que involucran el sistema completo o partes de él a lo largo del ciclo de vida de un producto de software. Un sistema de software pasa por cuatro etapas de prueba antes de implementarse. Estas cuatro etapas se conocen como pruebas de nivel de unidad, integración, sistema y aceptación. Los primeros tres niveles de prueba son realizados por una serie de diferentes partes interesadas en la organización de desarrollo, mientras que los clientes realizan las pruebas de aceptación. Las cuatro etapas de la prueba se han ilustrado en la forma de lo que se denomina modelo V clásico en la **Gráfico 28-2**. (Kshirasagar,2008, p.10).

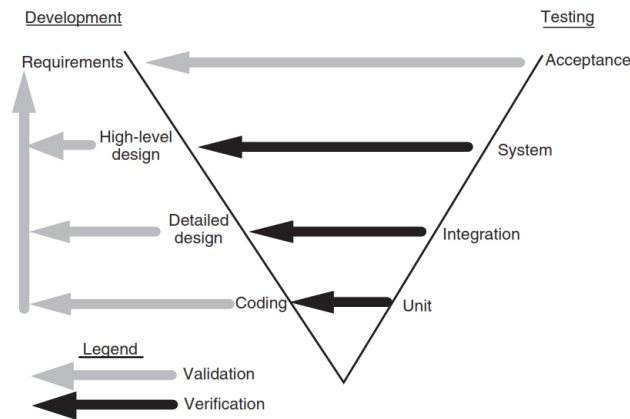


Gráfico 31-2: Fases de desarrollo y testing en el modelo V

Realizado por: Kshirasagar,2008

En las pruebas unitarias, los programadores prueban unidades de programas individuales, como procedimientos, funciones, métodos o clases, de forma aislada. Después de asegurarse de que las unidades individuales funcionen de manera satisfactoria, los módulos se ensamblan para construir subsistemas más grandes siguiendo las técnicas de prueba de integración. Los desarrolladores de software y los ingenieros de pruebas de integración realizan conjuntamente las pruebas de integración. El objetivo de las pruebas de integración es construir un sistema razonablemente estable que pueda soportar el rigor de las pruebas a nivel del sistema. Las pruebas a nivel de sistema incluyen un amplio espectro de pruebas, como pruebas de funcionalidad, pruebas de seguridad, pruebas de robustez, pruebas de carga, pruebas de estabilidad, pruebas de estrés, pruebas de rendimiento y pruebas de confiabilidad. La prueba del sistema es una fase crítica en un proceso de desarrollo de software debido a la necesidad de cumplir con un cronograma ajustado cerca de la fecha de entrega, para descubrir la mayoría de las fallas y para verificar que las reparaciones estén funcionando y no hayan dado como resultado nuevas fallas. La prueba del sistema comprende una serie de actividades distintas: crear un plan de prueba, diseñar un conjunto de pruebas, preparar entornos de prueba, ejecutar las pruebas siguiendo una estrategia clara y monitorear el proceso de ejecución de la prueba (Kshirasagar,2008, p.16).

245. Soporte de herramientas para pruebas de ciclo de vida

El soporte de herramientas está disponible para probar en cada etapa del ciclo de vida del desarrollo de software; Los diferentes tipos de herramientas y su posición dentro del ciclo de vida se muestran en la **Gráfico 29-2**, utilizando nombres alternativos para las etapas de desarrollo. (Kshirasagar,2008, p.16).

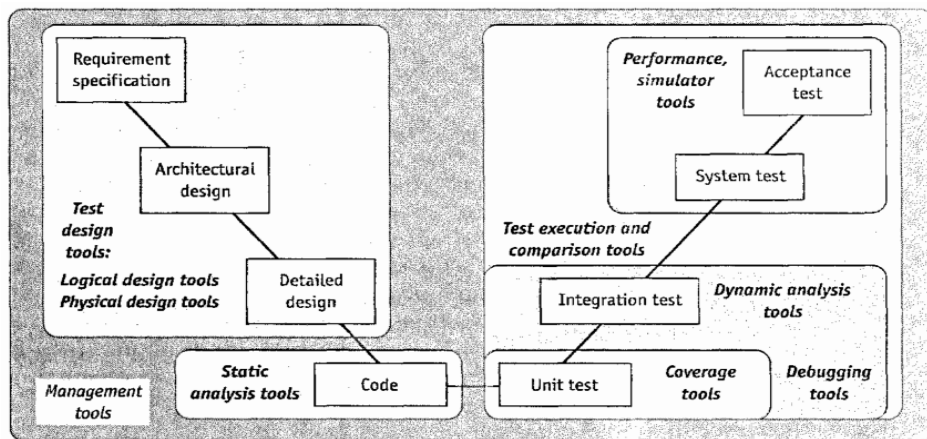


Gráfico 32-2: Herramientas necesarias en el ciclo de vida de desarrollo software

Realizado por: Kshirasagar,2008

Las herramientas definidas para el control de pruebas de software según Mark Fewster (Fewster, 1999, p.7), redacta en su libro las siguientes:

- Las herramientas de diseño lógico funcionan a partir de la lógica de una especificación, una interfaz o un código y, a veces, se denominan generadores de casos de prueba.
- Las herramientas de diseño físico manipulan datos existentes o generan datos de prueba. Por ejemplo, una herramienta que puede extraer registros aleatorios de una base de datos sería una herramienta de diseño físico. Una herramienta que pueda derivar entradas de prueba a partir de una especificación sería una herramienta de diseño lógico.
- Las herramientas de gestión de pruebas incluyen herramientas para ayudar en la planificación de pruebas, realizar un seguimiento de las pruebas que se han ejecutado, etc. Esta categoría también incluye herramientas para ayudar a la trazabilidad de las pruebas a los requisitos, diseños y código, así como herramientas de seguimiento de defectos.
- Las herramientas de análisis estático analizan el código sin ejecutarlo. Este tipo de herramienta detecta determinados tipos de defectos de forma mucho más eficaz y económica de lo que se puede conseguir por cualquier otro medio. Dichas herramientas también calculan varias métricas para el código, como la complejidad ciclomática de McCabe, métricas de Halstead y muchas más.
- Las herramientas de cobertura evalúan qué parte del software sometido a prueba se ha utilizado mediante un conjunto de pruebas. Se utilizan con frecuencia a nivel de prueba unitaria. Por ejemplo, la cobertura de ramas es un requisito para probar sistemas críticos para la seguridad o relacionados con la seguridad. Las herramientas de cobertura también pueden medir la cobertura de construcciones de nivel de diseño como árboles de llamadas.

- Las herramientas de depuración no son realmente herramientas de prueba, ya que la depuración no forma parte de las pruebas. (Las pruebas identifican defectos, la depuración los elimina y, por lo tanto, es una actividad de desarrollo, no una actividad de prueba). Sin embargo, las herramientas de depuración se utilizan a menudo en las pruebas, especialmente cuando se intenta aislar un defecto de bajo nivel. Las herramientas de depuración permiten al desarrollador recorrer el código ejecutando una instrucción a la vez y mirando el contenido de las ubicaciones de los datos.
- Las herramientas de análisis dinámico evalúan el sistema mientras se ejecuta el software. Por ejemplo, las herramientas que pueden detectar pérdidas de memoria son herramientas de análisis dinámico. Se produce una pérdida de memoria si un programa no libera bloques de memoria cuando debería, por lo que el bloque se ha "filtrado" del grupo de bloques de memoria disponibles para todos los programas. Eventualmente, el programa defectuoso terminará "poseyendo" toda la memoria; nada se puede ejecutar, el sistema 'cuelga' y debe reiniciarse (en un sistema operativo en modo no protegido).
- Los simuladores son herramientas que permiten probar partes de un sistema de formas que no serían posibles en el mundo real. Por ejemplo, los procedimientos de fusión de una central nuclear se pueden probar en un simulador.

2.4.6. Metodología de pruebas

En esta sección, describimos las actividades de prueba, ya que estas son las actividades que tal vez queramos automatizar. Existe una gran variación en la forma en que estas actividades se llevan a cabo en diferentes organizaciones. Algunas organizaciones realizarán todas estas actividades formalmente. Otros pueden ser informales hasta el punto de ser casi caóticos. En cualquier caso, las actividades centrales aún se llevan a cabo más o menos en la secuencia descrita en la **Gráfico 30-2**. (Fewster, 1999, p.7).

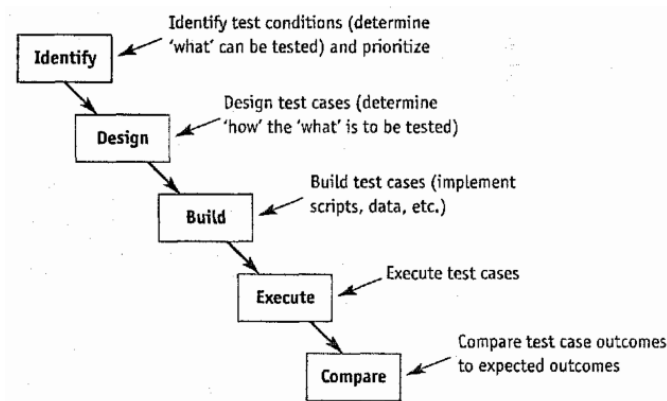


Gráfico 33-2: Actividades del ciclo de vida al desarrollar casos de prueba

Realizado por: Fewster, 1999

Respecto al proceso a tener en cuenta para el desarrollo de tareas de prueba Mark Fewer (Fewster, 1999, p.15) define los siguientes:

- **Identificar condiciones de prueba:** La primera actividad es determinar "qué" se puede probar e idealmente priorizar estas condiciones de prueba. Una condición de prueba es un elemento o evento que podría verificarse mediante una prueba. Habrá muchas condiciones de prueba diferentes para un sistema y para diferentes categorías de prueba, como pruebas de funcionalidad, pruebas de rendimiento, pruebas de seguridad, etc.
- **Diseño de casos de prueba:** El diseño del caso de prueba determina cómo se probará el "qué". Un caso de prueba es un conjunto de pruebas realizadas en una secuencia y relacionadas con un objetivo de prueba, es decir, el motivo o el propósito de las pruebas. El diseño de casos de prueba producirá una serie de pruebas que comprenden valores de entrada específicos, resultados esperados y cualquier otra información necesaria para que se ejecute la prueba, como los requisitos previos del entorno.
- **Construcción de los casos de prueba:** Los casos de prueba se implementan mediante la preparación de scripts de prueba, entradas de prueba, datos de prueba y resultados esperados. Un script de prueba son los datos y/o instrucciones con una sintaxis formal, utilizados por una herramienta de automatización de ejecución de pruebas, que normalmente se encuentran en un archivo. Un script de prueba puede implementar uno o más casos de prueba, navegación, procedimientos de configuración o limpieza, o verificación.
Las condiciones previas para los casos de prueba deben implementarse para que las pruebas se puedan ejecutar. Por ejemplo, si una prueba usa algunos datos de un archivo o base de datos, ese archivo o base de datos debe inicializarse para contener la información que la prueba espera encontrar.

- **Ejecutar los casos de prueba:** El software bajo prueba se ejecuta utilizando los casos de prueba. Para la prueba manual, esto puede consistir en que los probadores se sienten y posiblemente sigan un procedimiento manual impreso. Ingresarían entradas, observarían resultados y tomarían notas sobre cualquier problema a medida que avanzaban. Para las pruebas automatizadas, esto puede implicar iniciar la herramienta de prueba y decirle qué casos de prueba ejecutar.

La ejecución de la prueba solo se puede realizar después de que exista el software, es decir, es una actividad en el lado derecho del modelo V.

- **Compare los resultados de las pruebas con los resultados esperados:** Se debe investigar el resultado real de cada prueba para ver si el software que se está probando funcionó correctamente. Esto puede ser una confirmación informal de lo que el evaluador espera ver, o puede ser una comparación rigurosa y detallada de los resultados reales exactos con los resultados esperados. La comparación de algunos resultados, como los mensajes enviados a una pantalla, se puede realizar mientras se ejecuta la prueba. Otros resultados, como un cambio en los registros de la base de datos, solo se pueden comparar después de que se haya completado la ejecución del caso de prueba. Es posible que una prueba automatizada deba utilizar una combinación de estos dos enfoques.

La suposición es que, si los resultados reales y esperados son los mismos, entonces el software ha pasado la prueba; si son diferentes, el software no pasó la prueba.

Existe una diferencia entre comparar y verificar; una herramienta puede comparar, pero no verificar. Una herramienta puede comparar un conjunto de resultados de prueba con otro y puede marcar cualquier diferencia entre los dos. Pero la herramienta no puede decir si los resultados son correctos o no; esto es verificación y normalmente lo hacen los probadores. Son los probadores quienes confirman o aseguran que los resultados que se comparan sean de hecho correctos. En algunas circunstancias especiales, es posible automatizar la generación de resultados esperados, pero este no es el caso de la mayoría de las pruebas industriales realizadas con herramientas comerciales de ejecución de pruebas.

2.4.7. Artefactos de prueba para la identificación de casos de prueba

Los artefactos de prueba son elementos que permiten llevar a cabo las tareas de testing en un determinado proyecto. El artefacto principal para esta etapa de desarrollo es el plan de pruebas, mediante el cual se establece las condiciones iniciales bajo lo cual se llevará a cabo las tareas de testing durante todo el proyecto. (pmoinformatica, 2016, párr.1). Este plan de pruebas se detalla a continuación:

El plan de pruebas de software se elabora para atender los objetivos de calidad en un desarrollo de sistemas, encargándose de definir aspectos como por ejemplo los módulos o funcionalidades sujeto de verificación, tipos de pruebas, entornos, recursos asignados, entre otros aspectos.

Para elaborar un plan de pruebas de software lo primero que debes hacer es entender los requerimientos de usuario que componen la iteración o proyecto, que son el sujeto de la verificación de calidad que se va a realizar (pmoinformatica, 2016, párr.2).

De acuerdo con (UTPL, 2018, p.7) el propósito del plan de pruebas es explicitar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de pruebas. Un plan de pruebas incluye:

- **Alcance:** Indica el tipo de prueba y las propiedades/elementos del software a ser probado.
- **Items a probar:** Indica la configuración a probar y las condiciones mínimas que debe cumplir para comenzar a aplicarle el plan. Por un lado, es difícil y riesgoso probar una configuración que aún reporta fallas; por otro lado, si esperamos a que todos los módulos estén perfectos, puede que detectemos fallas graves demasiado tarde.
- **Estrategia:** Describe la técnica, patrón y/o herramientas a utilizarse en el diseño de los casos de prueba. Por ejemplo, en el caso de pruebas unitarias de un procedimiento, esta sección podría indicar: “Se aplicará la estrategia caja-negra de fronteras de la precondition” o “Ejercicio de los caminos ciclo-máticos válidos”. En lo posible la estrategia debe precisar el número mínimo de casos de prueba a diseñar, por ej. 100% de las fronteras, 60% de los caminos ciclo-máticos... La estrategia también explicita el grado de automatización que se exigirá, tanto para la generación de casos de prueba como para su ejecución.
- **Categorización de la configuración:** Explicita las condiciones bajo las cuales, el plan debe ser: Suspendido, Repetido; Culminado. En algunas circunstancias (las cuales deben ser explicitadas) el proceso de prueba debe suspenderse en vista de los defectos o fallas que se han detectado. Al corregirse los defectos, el proceso de prueba previsto por el plan puede continuar, pero debe explicitarse a partir de qué punto, ya que puede ser necesario repetir algunas pruebas. Los criterios de culminación pueden ser tan simples como aprobar el número mínimo de casos de prueba diseñados o tan complejo como tomar en cuenta no sólo el número mínimo, sino también el tiempo previsto para las pruebas y la tasa de detección de fallas.
- **Tangibles:** Explicita los documentos a entregarse al culminar el proceso previsto por el plan p. ej. sub-planes, especificación de pruebas, casos de prueba, resumen gerencial del proceso y bitácora de pruebas.
- **Recursos:** Especifica las propiedades necesarias y deseables del ambiente de prueba,

incluyendo las características del hardware, el software de sistemas (p. ej. el sistema de operación), cualquier otro software necesario para llevar a cabo las pruebas, así como la colocación específica del software a probar (p. ej. qué módulos se colocan en qué máquinas de una red local) y la configuración del software de apoyo.

La sección incluye un estimado de los recursos humanos necesarios para el proceso. También se indican cualquier requerimiento especial del proceso: actualización de licencias, espacio de oficina, tiempo en la máquina de producción, seguridad.

- **Calendario:** Esta sección describe los hitos del proceso de prueba y el grafo de dependencia en el tiempo de las tareas a realizar.
- **Manejo de riesgos:** Explicita los riesgos del plan, las acciones mitigantes y de contingencia.
- **Responsables:** Especifica quién es el responsable de cada una de las tareas previstas en el plan

248. *Artefactos de Prueba para el diseño y construcción de casos de prueba*

Según (TestingColombia, 2016, párr.12), en esta fase se hacen presentes distintos términos conocidos dentro del mundo del testing como son los casos de prueba, script de prueba y suite de prueba, mismos que se detallan a continuación:

Caso de prueba

Un conjunto de valores de entrada, precondiciones de ejecución, resultados esperados y post-condiciones de ejecución, desarrollados para un objetivo particular de condición de prueba, tal como para ejercer una ruta de un programa en particular o para verificar el cumplimiento de un requisito específico (TestingColombia, 2016, párr..13).

De acuerdo con (TestingColombia, 2016, párr.15) hay muchos formatos con diferentes tipos de datos, pero en realidad no existe un formato único para diseñar casos de prueba, dado que dependiendo el negocio o escenario será necesario personalizar los datos para ajustarlo a lo que se requiere probar, sin embargo hay campos mínimos que se deben contemplar en todos los casos, entonces se debe cumplir con la regla de oro “el caso de prueba debe ser claro y entendible” para otra persona lograr reproducirlo sin mucho esfuerzo.

- **Identificador:** identifica el caso de prueba, puede ser numérico o alfanumérico, la idea es que un caso de prueba se diferencie de otro caso a través de este indicador.
- **Nombre del Caso de Prueba:** es un nombre descriptivo del caso de prueba, en algunos procesos de calidad se hace necesario cumplir una nomenclatura clara y definida.

- Precondición/es: hace referencia a lo que se debe tener listo para la ejecución del caso de prueba, pueden ser la ejecución de otros casos de pruebas, un archivo, la creación de un dato, entre otros.
- Pasos: Define las acciones de usuario expresadas en términos de negocio y del aplicativo para la ejecución del caso de prueba, como por ejemplo ingresar el nombre en el campo “Nombre usuario” o hacer clic en el botón “Enviar”.
- Resultado esperado: Este apartado es muy importante, porque es el que determina si la ejecución del caso va siendo exitosa por cada paso, en algunos pasos de prueba no es necesario tener siempre un resultado esperado, se recomienda que se utilice en los pasos de mayor importancia para el negocio, como lo puede ser al momento de crear un usuario y se genera una ventana de confirmación y se puede apoyar también en una imagen que haga referencia al resultado deseado.
- Dato de Prueba: los pasos de pruebas se apoyan en datos, es por esto que por cada paso de prueba se puede hacer necesario especificar cuál es el dato a usar. Como lo puede ser un nombre de usuario, un password, etc.
- Resultado Real: como se busca que los casos de pruebas sean reproducibles las veces que sean necesarios, esta opción permite al analista estar registrando los sucesos de cada paso (Donde sea necesario, no implica uno a uno de los pasos).

Script de prueba

Cada caso de prueba por lo general está asociado a un script de prueba; sin embargo, puede ejecutar una prueba sin ningún script de prueba asociado.

Un script de prueba es un script manual o automático que contiene instrucciones para implementar un caso de prueba. Puede escribir scripts de prueba manuales para ser ejecutados por un verificador humano o bien puede automatizar algunas o todas las instrucciones del script de prueba. Puede asociar scripts de prueba funcional, scripts de prueba de rendimiento y scripts de prueba de seguridad automatizados a un caso de prueba (IBM, 2019, párr.6).

Suite de prueba

Una suite de pruebas es una colección de casos de prueba que se han agrupado para la ejecución de pruebas. Puede ejecutar una suite de pruebas abriendo la suite de pruebas o abriendo la ventana Examinar suites de prueba. Puede optar por ejecutar los casos de prueba en su suite de pruebas en secuencia o en paralelo. Si ejecuta una suite de pruebas en secuencia, un caso de

prueba se ejecuta sólo después de que haya finalizado el caso de prueba anterior en la secuencia de ejecución. Cuando ejecuta una suite de pruebas en paralelo, puede ejecutar todos los casos de prueba al mismo tiempo (IBM,2018, párr.8).

Una vez iniciada la prueba, pueden pausar la ejecución de la suite de prueba en la consola de Ejecución de suite de pruebas. Puede poner en pausa las ejecuciones de la suite de pruebas cuando el sistema que está probando necesite algunos ligeros ajustes antes de que puedan continuarse las pruebas. Cuando se pone en pausa la ejecución de una suite de pruebas, también se pone en pausa cualquier ejecución de prueba manual que forme parte de la suite. Si una prueba se pone en pausa mientras se ejecuta una prueba automatizada, dicha prueba se completa. Sin embargo, las pruebas automatizadas que no se han iniciado todavía no se ejecutan. En cualquiera de los casos, la suite de pruebas continúa en un estado de pausa. Reanude las suites de pruebas que están en pausa desde la consola de Ejecución de suite de pruebas, la vista de lista Registros de ejecución de suite de pruebas o el editor de Registro de ejecución de suite de pruebas (IBM,2018, párr.9).

Puede detener las suites de prueba desde la consola de Ejecución de suite de pruebas. Los casos de prueba sin finalizar dentro en dicha suite se marcan como incompletos. Las suites de prueba detenidas no se pueden reanudar (IBM,2018, párr.10).

2.4.9. Artefactos de prueba para ejecución y comparación de casos de prueba

En esta última etapa de desarrollo de pruebas se toma en consideración el informe de resultados obtenidos tras la ejecución de todas las tareas planteadas previamente. Cuando ejecutamos proyectos de desarrollo de software, la fase de pruebas suele ser crítica, y es un momento en el cual diversos interesados requieren información al minuto sobre el estado de la calidad del software que se está desarrollando. Para ello, se suele manejar un informe de avance de cómo van las pruebas, el cual según la criticidad del proyecto puede ser solicitado una o varias veces al día. La intención es comunicar a todos los involucrados de las áreas de pruebas, desarrollo, funcionales y área de negocio cual es la situación de las pruebas, que defectos críticos se están reportando y cuantos casos faltan por ejecutar (pmoinformatica,2015, párr.16).

2.4.10. Pruebas de caja negra

Las pruebas de caja negra, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o

escenarios de ejecución internos en el software. En las pruebas de caja negra, nos enfocamos solamente en las entradas y salidas del sistema, sin preocuparnos en tener conocimiento de la estructura interna del programa de software. Para obtener el detalle de cuáles deben ser esas entradas y salidas, nos basamos únicamente en los requerimientos de software y especificaciones funcionales (pmoinformatica.com, 2018, párr.12).

De acuerdo con (itpedia,2019, párr.7) podemos aplicar el método de prueba Black Box a los siguientes niveles de prueba de software:

- Pruebas de integración.
- Pruebas del sistema.
- Pruebas de aceptación.

Los beneficios de usar pruebas de caja negra según (itpedia,2019, párr.11), son las siguientes:

- Realizamos estas pruebas desde la perspectiva del usuario y ayudamos a identificar diferencias en las especificaciones.
- Los probadores no necesitan saber lenguajes de programación ni saber nada sobre la implementación del software.
- Podemos realizar las pruebas independientemente de los consultores o desarrolladores del proveedor. Esto da como resultado un informe de prueba objetivo. Los desarrolladores conocen su propio software, lo que puede provocar cierta ceguera ante los errores de software.
- Casos de prueba Se puede hacer una vez que las especificaciones estén completas.

Las desventajas de usar pruebas de caja negra según (itpedia,2019, párr.13), son las siguientes:

- Solo podemos probar una pequeña cantidad de casos de prueba posibles para que muchas rutas de programas permanezcan sin probar.
- En muchos casos, los requisitos o especificaciones no son claros, lo que dificulta la obtención de casos de prueba.
- Si un consultor o desarrollador del proveedor ya ha realizado un determinado caso de prueba, a veces se vuelve a ejecutar innecesariamente. Sin embargo, el cliente es responsable de la aceptación y debe determinar por sí mismo que el software funciona correctamente.
- En la prueba de caja negra, nunca estamos seguros de haber probado todos los rincones relevantes del software.

2.4.11. Pruebas de caja blanca

La prueba de caja blanca es una metodología utilizada para garantizar y validar el marco interno, los mecanismos, los objetos y los componentes de una aplicación de software. Las pruebas de caja blanca verifican el código de acuerdo con las especificaciones de diseño y descubre las vulnerabilidades de la aplicación. Durante las pruebas de caja blanca, el código se ejecuta con valores de entrada preseleccionados para la validación de los valores de salida preseleccionados. Las pruebas de caja blanca a menudo implican escribir códigos de código de software y controladores (continuosdev,2021, párr.1).

Por otro lado, las pruebas de caja blanca según (itpedia,2018, párr.2) son los siguientes:

- Prueba unitaria: Suele ser el primer tipo de prueba que realizamos en una aplicación. Realizamos la prueba unitaria en cada módulo o bloque de código de programa mientras lo desarrollamos. Es principalmente el desarrollador quien realiza las pruebas unitarias. El desarrollador de software desarrolla algunas líneas de código, una sola función o un objeto y lo prueba para asegurarse de que funciona antes de continuar. La mayoría de los errores se encuentran al principio del ciclo de vida del desarrollo de software. Si encontramos errores en esta etapa inicial, será más barato y fácil de solucionar.
- Prueba de fugas de memoria: Las pérdidas de memoria son las principales causas de que las aplicaciones sean más lentas. Un especialista en control de calidad con experiencia en la detección de fugas de memoria es esencial cuando la aplicación se ejecuta lentamente.

Los beneficios de este tipo de pruebas de acuerdo con (itpedia,2018, párr.5), son las siguientes:

- Optimización de código al encontrar errores ocultos.
- También podemos automatizar fácilmente los casos de prueba.
- La prueba es más exhaustiva, porque generalmente verificamos todas las rutas de código.
- Las pruebas pueden comenzar temprano en el proyecto, incluso si la GUI no está disponible.

Las desventajas según (itpedia,2018, párr.6), son las siguientes:

- Las pruebas de caja blanca pueden ser complejas y costosas.
- Los desarrolladores que suelen ejecutar casos de prueba de caja blanca lo detestan. Por lo tanto, los desarrolladores no prueban la caja blanca en detalle, lo que puede provocar errores de producción.
- Las pruebas de caja blanca también requieren herramientas profesionales y conocimientos profundos de programación e implementación.

- La prueba de caja blanca lleva mucho tiempo, lleva tiempo probar completamente las aplicaciones más grandes.

2.4.12. Pruebas unitarias

Las pruebas unitarias o unit testing son una forma de comprobar que un fragmento de código funciona correctamente. Es un procedimiento más de los que se llevan a cabo dentro de una metodología ágil de trabajo (Yeeply, 2020, párr.1).

Las pruebas unitarias consisten en aislar una parte del código y comprobar que funciona a la perfección. Son pequeños tests que validan el comportamiento de un objeto y la lógica.

El unit testing suele realizarse durante la fase de desarrollo de aplicaciones de software o móviles. Normalmente las llevan a cabo los desarrolladores, aunque en la práctica, también pueden realizarlas los responsables de QA (Yeeply, 2020, párr.2).

2.4.12.1. Motivos para realizar un test unitario

Los motivos de implementar test unitarios según (Yeeply, 2020, párr.7), presentan muchas ventajas que se describen a continuación:

- Las pruebas unitarias demuestran que la lógica del código está en buen estado y que funcionará en todos los casos.
- Aumentan la legibilidad del código y ayudan a los desarrolladores a entender el código base, lo que facilita hacer cambios más rápidamente.
- Los test unitarios bien realizados sirven como documentación del proyecto.
- Se realizan en pocos milisegundos, por lo que podrás realizar cientos de ellas en muy poco tiempo.
- Las unit testing permiten al desarrollador re-factorizar el código más adelante y tener la garantía de que el módulo sigue funcionando correctamente. Para ello se escriben casos de prueba para todas las funciones y métodos, para que cada vez que un cambio provoque un error, sea posible identificarlo y repararlo rápidamente.
- La calidad final del código mejorará ya que, al estar realizando pruebas de manera continua, al finalizar el código será limpio y de calidad.
- Como las pruebas unitarias dividen el código en pequeños fragmentos, es posible probar distintas partes del proyecto sin tener que esperar a que otras estén completadas.

2.4.12.2. Organizar-Actuar-Afirmar casos de prueba

Para hacer pruebas unitarias de calidad, es preciso que éstas estén estructuradas siguiendo las tres A's del Unit Testing (ceroideas, 2020, párr.6).

Estamos ante un concepto fundamental para el desarrollo de este tipo de pruebas, que describe un proceso que consta de tres pasos los cuales según (ceroideas, 2020) se describen a continuación:

- Arrange(Organizar): Este es el primer paso que se debe llevar a cabo, aquí es donde se definen los requisitos que debe cumplir el código.
- Act(Actuar): Se trata de un paso intermedio en la realización de pruebas, en este momento se ejecuta el test que va a dar lugar a los resultados que el desarrollador deberá analizar.
- Assert(Afirmar): Aquí es donde se comprueba si los resultados obtenidos son los esperados, de ser así se valida y se continúa, si no es así, se corrige hasta que el error desaparezca.

2.4.12.3. Cobertura de código

Una métrica de cobertura muestra la cantidad de código fuente que ejecuta un conjunto de pruebas, desde ninguno hasta el 100%. Existen diferentes tipos de métricas de cobertura y, a menudo, se utilizan para evaluar la calidad de un conjunto de pruebas. La creencia común es que cuanto mayor sea el número de cobertura, mejor (Khorikov,2020, p.8).

Desafortunadamente, no es tan simple y las métricas de cobertura, si bien brindan comentarios valiosos, no se pueden usar para medir de manera efectiva la calidad de un conjunto de pruebas. Es la misma situación que con la capacidad de realizar pruebas unitarias del código: las métricas de cobertura son un buen indicador negativo, pero un mal positivo (Khorikov,2020, p.8).

Si una métrica muestra que hay muy poca cobertura en su base de código, digamos, solo el 10%, es una buena indicación de que no está probando lo suficiente. Pero lo contrario no es cierto: ni siquiera una cobertura del 100% garantiza que tenga un conjunto de pruebas de buena calidad. Un conjunto de pruebas que proporciona una alta cobertura puede ser de mala calidad. La primera métrica de cobertura y la más utilizada es la cobertura de código, también conocida como cobertura de prueba; ver **Figura 9-2**. Esta métrica muestra la relación entre el número de líneas de código ejecutadas por al menos una prueba y el número total de líneas en la base del código de producción (Khorikov,2020, p.9).

$$\text{Code coverage (test coverage)} = \frac{\text{Lines of code executed}}{\text{Total number of lines}}$$

Figura 34-2: Cálculo de cobertura de código

Realizado por: itpedia,2018

Veamos un ejemplo para comprender mejor cómo funciona esto. La **Figura 10-2** muestra un método `IsStringLong` y una prueba que lo cubre. El método determina si una cadena que se le proporciona como parámetro de entrada es larga (aquí, la definición de larga es cualquier cadena con una longitud superior a cinco caracteres). La prueba ejercita el método usando "abc" y verifica que esta cadena no se considere larga (Khorikov,2020, p.9).

```
public static bool IsStringLong(string input)
{
    if (input.Length > 5)
        return true;
    return false;
}

public void Test()
{
    bool result = IsStringLong("abc");
    Assert.Equal(false, result);
}
```

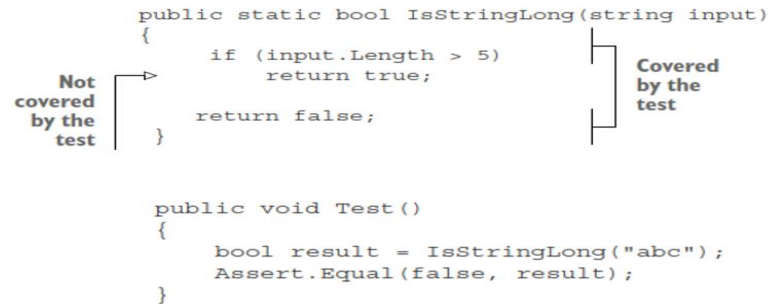


Figura 35-2: Ejemplo de un método parcialmente cubierto por una prueba

Realizado por: itpedia,2018

Es fácil calcular la cobertura del código aquí. El número total de líneas en el método es cinco (las llaves también cuentan). El número de líneas ejecutadas por la prueba es cuatro: la prueba pasa por todas las líneas de código excepto por el retorno verdadero. Esto nos da $4/5 = 0,8 = 80\%$ de cobertura de código (Khorikov,2020, p.9).

Ahora, al re-factorizar el código como se muestra en la **Figura 11-2**, el resultado es otro.

```
public static bool IsStringLong(string input)
{
    return input.Length > 5;
}

public void Test()
{
    bool result = IsStringLong("abc");
    Assert.Equal(false, result);
}
```

Figura 36-2: Cobertura total de un caso de prueba

Realizado por: itpedia,2018

¿Cambia el número de cobertura del código? Sí, lo hace. Debido a que la prueba ahora ejercita las tres líneas de código (la declaración de retorno más dos llaves), la cobertura del código aumenta al 100%. Este sencillo ejemplo muestra lo fácil que es jugar con los números de cobertura. Cuanto más compacto sea su código, mejor será la métrica de cobertura de prueba, ya que solo tiene en cuenta los números de línea sin procesar (Khorikov,2020, p.10).

2.4.12.4. Ejecución de las pruebas unitarias

De acuerdo a (Yeeply,2020, párr.7), hay muchas opciones disponibles, que varían en función del lenguaje de programación que se esté utilizando. Estos son algunos ejemplos de este tipo de herramientas que te ayudarán con las pruebas.

- **xUnit:** se trata de una herramienta de pruebas unitarias para el framework .NET.
- **Junit:** es un conjunto de bibliotecas para realizar pruebas unitarias de aplicaciones Java.
- **NUnit:** inicialmente portado desde JUnit, NUnit 3 se ha reescrito por completo para dotarlo de nuevas características y soporte para una amplia gama de plataformas .NET.
- **PHPUnit:** entorno de pruebas unitarias en el lenguaje de programación PHP.

Al utilizar estas herramientas, se codifican los criterios en la prueba que verificarán si el código es o no correcto. Durante la fase de ejecución, la herramienta puede detectar las pruebas con errores. Si alguno de estos errores es grave, puede detener pruebas posteriores que iban a realizarse a continuación (Yeeply, 2020, párr.9).

2.5. Metodología de desarrollo ágil SCRUM

Fue desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle, define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos años, es una metodología ágil para gerenciar y controlar el desarrollo de software de un producto en forma iterativa e incremental (Trigas, 2019, p.10).

En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente (Scrum, 2020, párr.3).

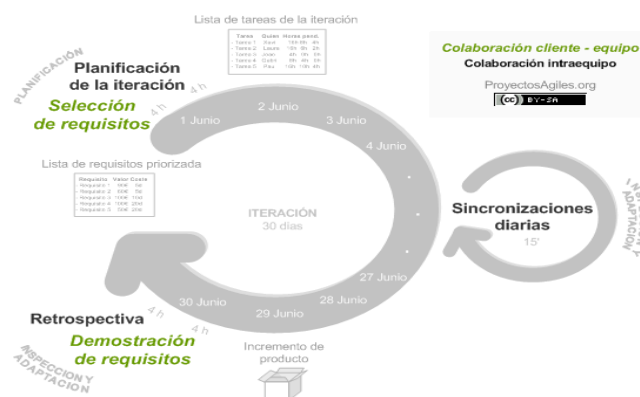


Figura 37-2: Proceso de la metodología SCRUM

Realizado por: proyectosagiles, 2017

25.1. Características de Scrum

Según (Scrum, 2020, párr.6) su prioridad es la satisfacción del cliente, que se da con la continua interacción entre éste y el equipo de desarrolladores.

- Se aceptan requisitos cambiantes.
- Enfocado a conseguir pequeños incrementos de software completamente funcionales.
- Es un modo de desarrollo adaptable, antes que predictivo.
- Orientado a las personas, más que a los procesos.
- Emplea el modelo de construcción incremental basado en iteraciones y revisiones.
- Alta flexibilidad.

25.2. Ventajas de usar Scrum

Las ventajas planteadas según (Scrum, 2020, párr.7) son las siguientes:

- Incremento en la productividad.
- Mejoras constantes.
- El producto total se convierte en una serie de pequeños pedazos manejables.
- Existe un progreso, inclusive si los requerimientos no están bien definidos.
- Todo es visible para todos.
- Existe una gran comunicación en el equipo.
- El equipo comparte los éxitos desde el principio hasta el final.
- El cliente se mantiene informado en cada mejora del producto
- Entrega de un producto funcional (jugable) al finalizar cada sprint.

25.3. Roles Scrum

Según (Scrum, 2020, párr.11), con la metodología Scrum, el equipo tiene como foco entregar valor y ofrecer resultados de calidad que permitan cumplir los objetivos de negocio del cliente. Para ello, los equipos de Scrum son auto-organizados y multifuncionales. Es decir, cada uno es responsable de unas tareas determinadas y de terminarlas en los tiempos acordados. En Scrum existen 3 roles muy importantes. Estos roles se detallan a continuación:

Product Owner: Él es responsable de decir lo que se debe desarrollar y el orden de los elementos que se deben cumplir. Puede considerarlo como la única autoridad que le diría al resto del equipo lo que necesitan crear y qué características deberían ser lo primero. En

resumen, es él quien les dice a los demás miembros del equipo lo que deberían proponer (Vardy,2016, p.20).

ScrumMaster: ScrumMaster se encarga de la orientación del equipo en el desarrollo del producto, así como de seguir el proceso basado en Scrum. Él es quien hace que todos comprendan las prácticas, principios y valores que todos deben cumplir para lograr el éxito del proyecto. En el equipo de Scrum, el ScrumMaster sirve como entrenador que proporciona indicaciones sobre cómo optimizar el rendimiento proporcionando el liderazgo de proceso necesario (Vardy,2016, p.23).

Development team: El equipo de desarrollo determina los métodos para entregar lo que el maestro de producto ha solicitado. Este equipo está compuesto por diferentes personas con diferentes descripciones de trabajo, como diseñador, probador y administrador de base de datos y arquitecto, lo que les permite realizar funciones cruzadas y volverse dinámicos a la hora de diseñar, probar y construir el producto requerido por el propietario del producto (Vardy,2016, p.23).

2.5.4 Artefactos de SCRUM

Los artefactos de Scrum representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la información clave, que es necesaria para asegurar que todos tengan el mismo entendimiento del artefacto. (ScrumGuide, 2019, párr.3). Estos artefactos se detallan a continuación:

Product Backlog: La cartera de productos es la 'lista de deseos' del propietario del producto. Cualquier cosa y todo lo que creen que podrían desear en el producto se incluye en esta lista. Podría ser infinito, ya que siempre hay nuevas ideas sobre cómo ampliar las características de un producto. El propietario del producto mantiene la acumulación del producto, aunque otras partes interesadas deben tener visibilidad y la capacidad de sugerir nuevos elementos para la lista (Viscardi,2013, p.23).

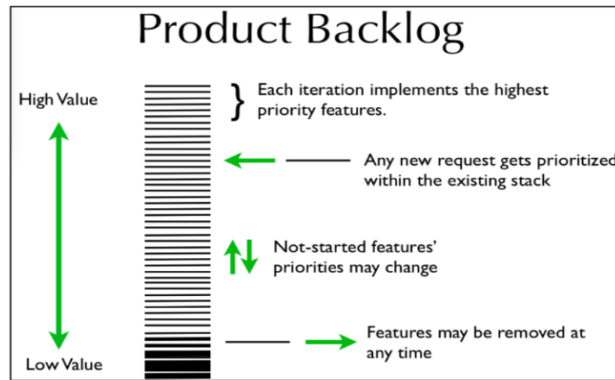


Figura 38-2: Product Backlog

Realizado por: Viscardi Stacia, 2013

Sprint backlog: Propiedad del equipo, la cartera de pedidos del sprint refleja los elementos de la cartera de productos con los que el equipo se comprometió en la planificación del sprint, así como las tareas y recordatorios posteriores. Los miembros del equipo lo actualizan todos los días para reflejar cuántas horas quedan en sus tareas; Los miembros del equipo también pueden eliminar tareas, agregar tareas o cambiar tareas a medida que el sprint está en marcha (Viscardi,2013, p.24).

El incremento de producto: El incremento de producto es un conjunto de características, historias de usuario u otros entregables completados por el equipo en el sprint. El incremento de producto debería poder enviarse potencialmente, es decir, de calidad lo suficientemente alta como para entregárselo a los usuarios. El propietario del producto es responsable de aceptar el incremento de producto durante cada sprint, de acuerdo con la Definición de Terminado y los criterios de aceptación acordados para cada entrega de sprint. Sin un incremento de producto, el propietario del producto y otras partes interesadas no tienen forma de inspeccionar y adaptar el producto (Viscardi,2013, p.24).

Progreso visible: Un equipo debe mantener visible su progreso en todo momento. Creará muchos artefactos adicionales para garantizar la visibilidad. Algunas herramientas de visibilidad habituales son los gráficos de evolución de la versión y del sprint (Viscardi,2013, p.24).

Burn-down chart: Un gráfico de evolución es una representación gráfica de la rapidez con la que trabaja el equipo a través de las historias de usuario de un cliente, una herramienta ágil que se utiliza para capturar una descripción de una función desde la perspectiva del usuario final. El gráfico de evolución muestra el esfuerzo total frente a la cantidad de trabajo de cada iteración (ProjectManager 2019, párr.7).

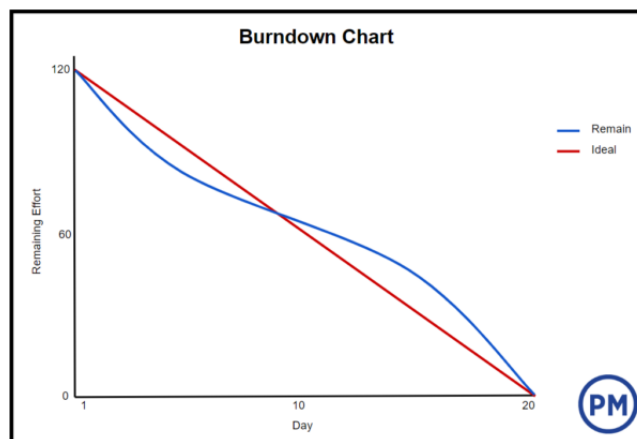


Gráfico 39-2: Gráfica de avance de proyectos (Burndown Chart)

Realizado por: ProjectManager 2019

El punto de inicio del proyecto es el punto más alejado a la izquierda del gráfico y ocurre el día cero del proyecto o iteración. El punto final del proyecto está más a la derecha y marca el último día del proyecto o iteración (ProjectManager 2019, párr.9).

2.6. Norma ISO/IEC 25010

ISO/IEC 25010 - *System and software quality models*: describe el modelo de calidad para el producto software y para la calidad en uso. Esta Norma presenta las características y subcaracterísticas de calidad frente a las cuales evaluar el producto software (iso25000,2019, párr.1).

Los beneficios del uso de esta norma según (iso25000,2019, párr.2) se detallan a continuación:

- El modelo busca proporcionar una metodología para evaluar de un producto de software con el objetivo de obtener la calidad esperada.
- Permite identificar los criterios que debe tener un producto software respecto a la calidad interna y externa que espera un usuario.
- Permite que las empresas o personas que están desarrollando un software evalúen su producto buscando llevar al usuario un producto de alta calidad.
- Mejora la calidad del producto

Las normas ISO/IEC 25010 se definen como las características que debe tener un software para cumplir con los estándares establecidos buscando satisfacer las necesidades del usuario. La ISO dice que un modelo de calidad puede ser concebido como el conjunto de factores de calidad, y de las relaciones entre ellos. (iso25000,2019, párr.3)

CAPÍTULO III

3. MARCO METODOLOGICO

3.1. Introducción

En este capítulo se describe todo lo relacionado a la metodología, la técnica de investigación, variables e indicadores de estudio que se han utilizado en el desarrollo de este proyecto.

La metodología de investigación que se emplea es el descriptivo, analítico y deductivo, mediante las técnicas de la entrevista y exploración de documentación relacionada con el problema. En cuanto a la metodología de programación se ha empleado SCRUM, debido a sus características de gestión ágil de proyectos y la entrega por iteraciones al cliente.

3.2. Diseño de la investigación

El presente trabajo de integración curricular se ha establecido como una investigación de tipo aplicada y además se ha empleado todos los conocimientos que se han adquirido a lo largo de la carrera de formación académica, por lo tanto, esto ha intervenido en el desarrollo de la solución a la problemática de gestión de procesos administrativos de la escuela de conducción CONDUESPOCH mismos que han servido para definir la población y muestra que se detalla a continuación.

3.2.1. Población

Para medir la eficiencia del sistema de gestión de información de procesos administrativos de la escuela de conducción profesional “CONDUESPOCH” se tomaron como población los 98 requerimientos funcionales, los cuales fueron definidos como historias de usuario al inicio de la planificación del proyecto, para su debido desarrollo.

3.2.2. Muestra

El tipo de muestra que se ha escogido pertenece al muestreo no probabilístico por conveniencia, debido a la facilidad de estudio y obtención de datos.

Para elegir la muestra se tomó en cuenta los procesos que se realizan con más frecuencia y por tanto tienen prioridad sobre otros dentro de la institución ya que son necesarios para llevar a cabo otros procesos posteriores, estos se muestran en la **Tabla 1-3**.

Tabla 1-3: Tareas a evaluar para determinar la eficiencia

Módulo	Tarea
Gestión de proceso de inscripción de aspirantes.	<ul style="list-style-type: none">- Registro de datos de inscripción del estudiante- Buscar datos de inscripción por cédula de identidad o nombres del usuario
Autenticación	<ul style="list-style-type: none">- Registrar un usuario- Permitir que los usuarios(clientes) puedan registrar un horario para el examen psicosenométrico.
Gestión de movimiento de vehículos de uso administrativo	<ul style="list-style-type: none">- Ingresar una solicitud al departamento Gerencial
Gestión de desarrollo e implementación de aplicaciones	<ul style="list-style-type: none">- Ingreso de solicitud al departamento de DTIC- Ingresar un nuevo proyecto de DTIC
Gestión de actividades previo a los exámenes de grado	<ul style="list-style-type: none">- Ingresar comunicado
Gestión de ventas	<ul style="list-style-type: none">- Ingreso de solicitud de tipo general o justificación
Gestión de compras	<ul style="list-style-type: none">- Ingreso pago de mensualidad de los estudiantes

Realizado por: Aldas W.; Pilataxi C., 2020

3.2.3. *Métodos de investigación*

Analítico

El método analítico es un modelo de estudio que se basa en la experimentación directa y la lógica empírica, analiza el fenómeno que estudia, en otras palabras, lo descompone en elementos simples y más básicos. Este método consiste en la aplicación de la experiencia directa y nos permite estudiar cada uno de los procesos administrativos que se llevan a cabo en CONDUESPOCH.

Deductivo

Mediante este método podemos establecer el funcionamiento del sistema de manera lógica, encaminada al funcionamiento de cada proceso administrativo. Es decir que podemos segmentar cada proceso en múltiples subprocesos de manera que el problema se resuelve con facilidad.

Descriptivo

Con este método podemos representar los datos y características que se requieren en el proceso de desarrollo del sistema, esto a su vez ayuda en el análisis estadístico de resultados de su implantación.

3.2.4. *Técnicas de investigación*

Entrevista

Mediante esta técnica implementada al director de DTIC de la institución, se ha podido determinar el funcionamiento de cada uno de los procesos administrativos que se llevan a cabo y además los requerimientos necesarios a implementar en el sistema.

Análisis de documentación

Esta técnica se ha implementado para la realización documental de este trabajo de integración curricular, lo más utilizado en esta técnica son artículos, libros y diferentes manuales de las herramientas que se han utilizado.

3.2.5. *Método utilizado para evaluar la eficiencia de tiempo del sistema*

Para evaluar la eficiencia del presente proyecto, se realizó una comparación entre el tiempo que se emplea en llevar a cabo los procesos administrativos de la institución por medio del método manual y el automatizado, de esta manera se seleccionaron 10 procesos, ya que son los más empleados dentro de la institución, a cada uno de dichos procesos se les ha tomado 3 mediciones en un ambiente real, las mismas que se representan en minutos y que fueron recopilados por medio de un cronómetro.

La toma de tiempo empleado en la ejecución de cada una de las tareas seleccionadas para el análisis se las realizó mediante los distintos usuarios de la institución, quienes al interactuar con el sistema arrojaron datos relativos a cada tarea, mismos que se registraron con una diferencia de 48 horas para tener un tiempo prudente entre cada medición.

Determinación de criterios de evaluación

Con el fin de evaluar la eficiencia de la aplicación web que permite la gestión de procesos administrativos en la escuela de conducción “CONDUESPOCH E.P”, se establece indicadores en base a la norma ISO/IEC 25010 la cual evalúa la calidad del software.

En la **Tabla 2-3** se puede observar la variable dependiente identificada para este trabajo de integración curricular:

Tabla 2-3: Determinación del criterio de evaluación

VARIABLE	TIPO	INDICADOR	DESCRIPCION	INTERPRETACION	TECNICA	HERRAMIENTA
Eficiencia	Dependiente	Tiempo de respuesta	¿Qué tan rápido responde el sistema?	Entre menos tiempo es mejor	Observación directa	Cronometro

Realizado por: Aldas W.; Pilataxi C., 2020

3.3. Aplicación de la metodología Ágil SCRUM

3.3.1. Estudio preliminar

La Escuela de Conducción Profesional CONDUESPOCH E.P está ubicado en la ciudad de Riobamba junto a la Facultad de Administración de Empresas de la Escuela Superior Politécnica de Chimborazo, dirección Panamericana sur Km 1 ½, la cual se dedica a la formación de conductores profesionales con licencia tipo C, en un periodo semestral, la persona encargada de la gestión de los sistemas virtuales de la institución es el Ing. Carlos Guffante.

Actualmente CONDUESPOCH cuenta con siete departamentos administrativos los cuales son, Secretaría, Asesoría vial, DTIC, Inspección, Pedagogía, Contabilidad y Tesorería, a su vez cada uno posee sus propios procesos de gestión. La institución no posee un sistema que les permita archivar los registros de cada proceso, lo cual causa pérdidas de estos, que a su vez provoca que haya contratiempos al momento de las revisiones periódicas de constancia. Para resolver este inconveniente, agilizar y optimizar los procesos administrativos se ha establecido en automatizar las actividades más importantes llevadas a cabo en CONDUESPOCH.

Para la realización del presente proyecto se ha definido las fases de planificación, desarrollo y cierre, permitiéndonos llevar a cabo distintas tareas guiadas por la metodología ágil SCRUM, las cuales se detallan a continuación.

3.3.2. Fase de planificación

3.3.2.1. Planificación

Para llevar a cabo el desarrollo de la aplicación web y aplicación móvil para la gestión de información de procesos administrativos de la escuela de conducción profesional CONDUESPOCH E.P bajo un entorno de pruebas, se ha tenido que establecer una proyección

del tiempo de entrega de cada tarea por lo que se ha utilizado la herramienta Gantt Project para tal actividad.

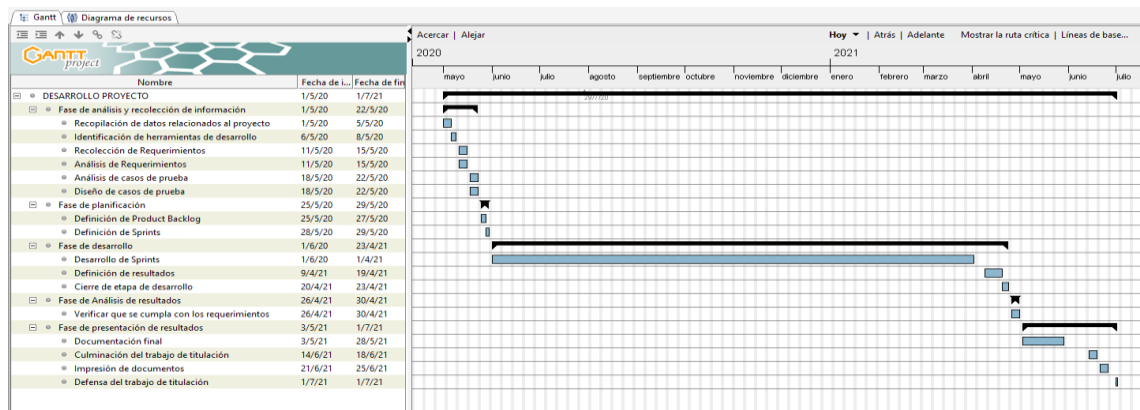


Figura 1-3: Planificación para el desarrollo del proyecto

Realizado por: Aldas W.; Pilataxi C., 2020

En la **Figura 1-3** se muestra la cantidad de actividades por realizar dentro del proyecto, a su vez estas están ordenadas por fases, las cuales van desde la fase de análisis y recolección de información, fase de planificación, fase de desarrollo, fase de análisis de resultados y fase de presentación de resultados. Además, para que todo el proceso sea más fácil de manejar se ha implementado el uso de la metodología ágil SCRUM.

3.3.2.2. *Requerimientos (Product backlog)*

Los requerimientos del usuario son establecidos como historias de usuario y las rutinas de funcionamiento del sistema que no están especificadas se denominan historias técnicas, las cuales son útiles para la creación del producto backlog, el cual está compuesto por una cantidad determinada de entregables o Sprints.

Al crear el producto backlog, cada tarea a realizar necesita un puntaje de estimación, para lo cual se ha escogido el método de la talla de la camiseta o T-shirt. Las unidades estimadas son XXS, XS, S, M, L, XL, a continuación, se presenta en la **Tabla 3-3**

Tabla 3-3: Puntos estimados según tallas de Product Backlog

Tallas	Puntos Estimados
XXS	8
XS	16
S	24
M	40
L	80
XL	120

Realizado por: Aldas W.; Pilataxi C., 2020

El tiempo de desarrollo diario es de 8 horas dentro de un lapso semanal de 5 días laborales. Cada punto estimado representa una hora laboral, por lo que la talla XXS que tiene un estimado de 8 puntos su valor en tiempo laboral es de 8 horas, es decir un día laboral, para la talla XS de 16 puntos estimados su tiempo laboral es de 2 días, para la talla S de 24 puntos estimados su tiempo laboral es de 3 días, para la talla M de 40 puntos estimados su tiempo laboral es de 5 días es decir una semana de trabajo, para la talla L de 80 puntos estimados su tiempo laboral es de 10 días es decir dos semanas de trabajo, para la talla XL de 120 puntos estimados su tiempo laboral es de 15 días es decir tres semanas de trabajo.

De acuerdo con la metodología Scrum todas las iteraciones o Sprints deben tener un mismo número estimado de puntos, en este caso se toma que cada Sprint es representado con 120 puntos lo cual equivale a 120 horas laborales es decir 3 semanas, mediante esta aclarativa se seleccionan las historias técnicas o de usuario que sumados den un resultado que no sobrepase la cantidad de puntos de cada Sprint, esto de acuerdo con la prioridad que haya establecido el cliente.

3.3.2.3. *Alcance del proyecto*

La aplicación web permanecerá alojada en un hosting empleado por la institución, con acceso a través de internet, a su vez este consumirá datos almacenados en el servidor de base de datos de la misma institución. En cuanto a la aplicación móvil, debido al bajo presupuesto económico de la institución, permanecerá alojado en el mismo hosting de la aplicación web, lugar del cual será accedido y descargado mediante la aplicación web.

3.3.2.4. *Limitaciones*

El uso de la aplicación web, está restringido únicamente al personal administrativo u otros usuarios a los que se les haya dado permiso de acceso, lo cual será verificado mediante sus credenciales de autenticación, cada vez que quieran interactuar con el sistema. Además, solo el administrador tendrá acceso a todas las funcionalidades del sistema, mientras que el resto de los usuarios estarán limitados a un grupo de funcionalidades de acuerdo con sus roles.

La aplicación móvil ha sido desarrollada como una aplicación multiplataforma es decir que puede ser utilizada por más de un sistema operativo, sin embargo, debido a la gran cantidad de usuarios de dispositivos smartphones con sistema Android, se ha optado únicamente en poner en producción para este sistema.

3.3.2.5. Especificaciones para los requerimientos

En esta sección se establecen todos los requerimientos necesarios para el funcionamiento del sistema que permitirá la gestión de información de los procesos administrativos de la escuela de conducción CONDUESPOCH E.P, mismos que son especificaciones de las funcionalidades de cada uno de los departamentos.

Para establecer los requerimientos del sistema se ha tenido que realizar una entrevista al director de DTIC de la escuela de conducción CONDUESPOCH E.P, para lo cual se han establecido 98 requerimientos, los mismos que serán rotulados como historias de usuario (HU), además se ha visto oportuno la realización de 18 historias técnicas (HT) para la elaboración de actividades que sirven de apoyo al funcionamiento del sistema.

El desarrollo del sistema se establece en las siguientes historias de usuario detalladas en la **Tabla 4-3**, con su respectiva cantidad de esfuerzo, mismas que representan la cantidad de horas empleadas en el desarrollo de cada tarea, con el fin de encajar el desarrollo del sistema en un total de 10 meses, desde el proceso de recolección de cada uno de los requerimientos hasta la implementación en el entorno de producción.

A continuación, se muestran las historias de usuario que son necesarias para el desarrollo de este proyecto:

Tabla 4-3: Historias de usuario del Sistema

ID	TAREA	PRIORIDAD	ESTIMACIÓN
HU-01	Autenticación de los usuarios por número de cédula y contraseña	Alta	24
HU-02	Modificar datos personales	Baja	24
HU-03	Registrar un usuario	Alta	16
HU-04	Modificar datos de un usuario	Media	16
HU-05	Eliminar datos de un usuario.	Media	8
HU-06	Ver listado de usuarios por roles	Media	8
HU-07	Buscar usuarios por número de cédula o nombres	Baja	16
HU-08	Migrar datos de docentes del sistema de calificaciones de la institución	Media	16
HU-09	Cambiar de departamento a un usuario con rol de directivo	Baja	16
HU-10	Registro de datos de inscripción del estudiante	Alta	16
HU-11	Listar las inscripciones recibidas en su respectivo departamento.	Alta	24
HU-12	Buscar datos de inscripción por cédula de identidad o nombres del usuario	Media	16
HU-13	Ingresar fecha de inicio y fin de inscripción o matrícula	Alta	24
HU-14	Listar fechas de inscripción y matrícula	Alta	8
HU-15	Eliminar la fecha de inscripción o matrícula	Alta	8
HU-16	Modificar las fechas de inscripción o matrícula	Alta	16
HU-17	Registrar datos de matrícula de un estudiante	Alta	16

ID	TAREA	PRIORIDAD	ESTIMACIÓN
HU-18	Listar las matrículas recibidas en su respectivo departamento.	Alta	24
HU-19	Buscar datos de matrícula por cédula de identidad o nombres del usuario	Media	16
HU-20	Ingresar una solicitud al departamento de Asesoría Vial	Media	16
HU-21	Modificar una solicitud al departamento de Asesoría Vial	Media	8
HU-22	Eliminar una solicitud al departamento de Asesoría Vial	Media	8
HU-23	Listar las solicitudes al departamento de Asesoría Vial	Media	8
HU-24	Buscar una solicitud del departamento de Asesoría Vial	Media	16
HU-25	Listar los vehículos de la institución	Baja	8
HU-26	Ingresar un informe de gestión vehicular	Baja	16
HU-27	Listar informes de gestión vehicular	Baja	8
HU-28	Eliminar un informe de gestión vehicular	Baja	8
HU-29	Ingreso de solicitud al departamento de DTIC	Media	16
HU-30	Listar solicitudes del departamento de DTIC	Media	8
HU-31	Modificar de solicitud al departamento de DTIC	Media	16
HU-32	Eliminar una solicitud al departamento de DTIC	Media	8
HU-33	Buscar una solicitud del departamento de DTIC	Media	8
HU-34	Ingresar un nuevo proyecto de DTIC	Alta	24
HU-35	Listar los proyectos de DTIC	Alta	16
HU-36	Modificar un proyecto de DTIC	Media	16
HU-37	Eliminar un proyecto de DTIC	Media	8
HU-38	Ingresar tareas por practicante	Alta	16
HU-39	Asignar un practicante a una tarea	Alta	8
HU-40	Eliminar un practicante de una tarea	Alta	8
HU-41	Modificar una tarea	Media	16
HU-42	Eliminar una tarea	Media	8
HU-43	Listar las tareas asignadas a un practicante	Alta	8
HU-44	Ingresar una solicitud al departamento Gerencial	Media	24
HU-45	Listado de solicitudes del departamento Gerencial	Media	24
HU-46	Eliminar una solicitud del departamento Gerencial	Media	8
HU-47	Ingreso de novedades de los docentes, instructores y estudiantes	Media	16
HU-48	Modificar novedades de los docentes, instructores y estudiantes	Media	8
HU-49	Eliminar novedades de los docentes, instructores y estudiantes	Media	16
HU-50	Listar novedades de los docentes, instructores y estudiantes	Media	8
HU-51	Ingresar un comunicado	Alta	8
HU-52	Modificar un comunicado	Media	8
HU-53	Eliminar un comunicado	Media	8
HU-54	Listar comunicados	Alta	8
HU-55	Registro de aulas	Media	16
HU-56	Listado de aulas	Media	8
HU-57	Modificar datos de las aulas	Media	24
HU-58	Eliminar datos de las aulas	Media	8
HU-59	Ingresar recursos de un aula	Media	8
HU-60	Listar los recursos de un aula	Media	8
HU-61	Modificar los recursos de un aula	Media	24
HU-62	Eliminar los recursos de un aula	Media	8
HU-63	Ingresar datos de un silabo académico	Alta	24
HU-64	Modificar datos de un silabo académico	Media	40
HU-65	Ingresar evaluación al docente o instructor	Alta	40
HU-66	Modificar evaluación al docente o instructor	Media	16
HU-67	Eliminar evaluación al docente o instructor	Alta	8
HU-68	Listar evaluaciones docentes o instructores	Alta	8
HU-69	Ingresar preguntas por evaluación docente o instructor	Alta	40
HU-70	Modificar preguntas por evaluación docente o instructor	Media	16
HU-71	Eliminar preguntas por evaluación docente o instructor	Alta	8
HU-72	Listar preguntas por evaluación docente o instructor	Alta	8
HU-73	Listar resultados de evaluación	Alta	16

ID	TAREA	PRIORIDAD	ESTIMACIÓN
HU-74	Ingresar datos de una planificación	Alta	24
HU-75	Modificar datos de una planificación	Media	40
HU-76	Ingresar materiales de estudio de los docentes o instructores	Alta	24
HU-77	Aprobar materiales de estudio de los docentes o instructores	Alta	24
HU-78	Eliminar materiales de estudio de los docentes o instructores	Media	16
HU-79	Ingresar observación en un material de estudio de docente o instructor	Media	8
HU-80	Ingreso de recursos para el pago a docentes o instructores	Media	24
HU-81	Listado de recursos para los pagos de docentes o instructores	Media	8
HU-82	Eliminar los recursos para el pago de docentes o instructores	Media	8
HU-83	Ingreso de solicitud de tipo general o justificación	Alta	24
HU-84	Eliminar solicitud de tipo general o justificación	Media	8
HU-85	Modificar solicitud de tipo general o justificación	Media	16
HU-86	Mostrar detalles de solicitud de tipo general o justificación	Alta	8
HU-87	Listar solicitudes de tipo general o justificación	Alta	16
HU-88	Aprobar solicitud de tipo general o justificación	Alta	24
HU-89	Ingreso pago de mensualidad de los estudiantes	Media	16
HU-90	Aprobar pago de mensualidad de los estudiantes	Media	16
HU-91	Eliminar pago de mensualidad de los estudiantes	Media	16
HU-92	Ingreso de observación al pago de mensualidad de un estudiante	Media	16
HU-93	Realizar la interfaz de usuario de la aplicación móvil	Alta	24
HU-94	Autenticar administradores desde la aplicación móvil	Alta	24
HU-95	Registrar solicitudes desde la aplicación móvil	Alta	16
HU-96	Permitir el acceso a los usuarios del sistema a su respectiva cuenta.	Media	24
HU-97	Permitir que los usuarios(clientes) puedan registrar un horario para el examen psicosenométrico.	Alta	24
HU-98	Permitir que los administradores puedan ver los horarios de examen psicosenométrico.	Alta	16

Realizado por: Aldas W.; Pilataxi C., 2020

Mediante las historias técnicas descritas en la **Tabla 5-3**, se pretende llevar a cabo el desarrollo de aquellas tareas que no se han especificado de manera explícita en los requerimientos del usuario, pero que sin embargo el equipo de desarrollo lo deben tomar en cuenta para el funcionamiento esperado del sistema.

Tabla 5-3: Historias técnicas del sistema

ID	TAREA	PRIORIDAD	ESTIMACIÓN
HT-01	Establecer el estándar de codificación	Media	16
HT-02	Establecer la arquitectura del sistema	Alta	40
HT-03	Diseño de estándar de interfaz	Alta	24
HT-04	Desarrollo de la base de datos	Alta	24
HT-05	Análisis y gestión de riesgos	Baja	16
HT-06	Implementación de pruebas automatizadas Sprint 2	Media	8
HT-07	Implementación de pruebas automatizadas Sprint 3	Media	8
HT-08	Implementación de pruebas automatizadas Sprint 4	Media	8
HT-09	Implementación de pruebas automatizadas Sprint 5	Media	8
HT-10	Implementación de pruebas automatizadas Sprint 6	Media	8
HT-11	Implementación de pruebas automatizadas Sprint 7	Media	8
HT-12	Implementación de pruebas automatizadas Sprint 8	Media	8

ID	TAREA	PRIORIDAD	ESTIMACIÓN
HT-13	Implementación de pruebas automatizadas Sprint 9	Media	8
HT-14	Implementación de pruebas automatizadas Sprint 10	Media	8
HT-15	Implementación de pruebas automatizadas Sprint 11	Media	8
HT-16	Implementación de pruebas automatizadas Sprint 12	Media	8
HT-17	Implementación de pruebas automatizadas Sprint 13	Media	8
HT-18	Implementación de pruebas automatizadas Sprint 14	Media	8
HT-19	Implementación de pruebas automatizadas Sprint 15	Media	8

Realizado por: Aldas W.; Pilataxi C., 2020

Los campos que se han tomado en consideración tanto en las historias de usuario como en las historias técnicas son los siguientes:

- **ID:** Es el identificador de la historia en cuestión, para una historia de usuario este campo se lo completa con HU, mientras que para una historia técnica su terminología es HT, seguidos por un número el cual corresponde al orden en que se han ido desarrollando.
- **TAREA:** Es el nombre propiamente dicho de la tarea que se ha realizado, es decir que es lo que el cliente ha solicitado para solucionar un problema.
- **PRIORIDAD:** Es una determinación a la importancia que se le da a una historia ya sea de usuario o técnica, lo cual se ha tomado en cuenta en el desarrollo de cada Sprint. Los valores varía de Baja, Media y Alta.
- **ESTIMACIÓN:** Es la cantidad de puntos de esfuerzo establecidos por el equipo de desarrollo del sistema, tal cual lo indica la metodología ágil scrum.

3.3.2.6. Personal involucrado y roles establecidos en el proyecto

La metodología ágil Scrum plantea que, para cada proyecto de software debe existir un grupo mínimo de personas interesadas en su desarrollo, los cuales estarán a cargo de diferentes actividades. Este grupo de personas debe estar integrada por al menos un director de proyecto (Scrum Master), un propietario del producto (Product Owner) y el personal de desarrollo (Development Team), se puede implementar más roles dependiendo del tipo de la problemática a resolver, sin embargo, para este trabajo se ha tomado como referencia únicamente aquellos descritos anteriormente, lo cual se especifica en la **Tabla 6-3**.

Tabla 6-3: Personal involucrado y roles establecidos en el desarrollo del proyecto

PERSONA	ROL	CONTACTO	INSTITUCIÓN
Ing. Danilo Pastor	Scrum Master	0992741298 danilo.pastor@epoch.edu.ec	ESPOCH-FIE
Ing. Carlos Guffante	Product Owner	0996383423 carlos.guffante@epoch.edu.ec	CONDUESPOCH E.P
Wilmer Aldas	Development Team	0983005547 wilmer.aldas@epoch.edu.ec	ESPOCH-FIE

PERSONA	ROL	CONTACTO	INSTITUCIÓN
Claudio Pilataxi	Development Team	0989884666 claudio.pilataxi@epoch.edu.ec	ESPOCH-FIE

Realizado por: Aldas W.; Pilataxi C., 2020

3.3.2.7. Roles establecidos para el sistema

Debido a las múltiples funcionalidades planteadas por el cliente para el desarrollo del sistema, se ha determinado que los roles apropiados para su debido funcionamiento son los detallados en la **Tabla 7-3**. Tanto el entorno como las tareas a realizar por cada uno de los usuarios autorizados, varían considerablemente, pero que se desenvuelven en conjunto para llevar a cabo los procedimientos definidos.

Tabla 7-3: Roles establecidos para el uso del sistema

USUARIO	DESCRIPCIÓN	ROL	RESPONSABLE
Administrador	Responsable de administrar todo el sistema.	Control de todo el sistema es decir de todos los departamentos administrativos.	Ing. Carlos Guffante
Directivo de Secretaría	Usuario que accede a la sección correspondiente de su departamento administrativo	Gestión de usuarios, comunicados, solicitudes, solicitudes generales recibidas, inscripciones y matrículas.	Ing. Bethzabe Borja
Directivo de Asesoría Vial	Usuario que accede a la sección correspondiente de su departamento administrativo	Gestión de comunicados, solicitudes, solicitudes generales recibidas, cursos de los instructores a cargo, pago de instructores y vehículos.	Ing. David Yupanqui
Directivo de Inspección	Usuario que accede a la sección correspondiente de su departamento administrativo	Gestión de usuarios, comunicados, solicitudes, solicitudes generales recibidas, solicitudes de justificaciones, novedades presentadas por los docentes, instructores o estudiantes, y gestión de aulas.	Ing. David Yupanqui
Directivo de Pedagogía	Usuario que accede a la sección correspondiente de su departamento administrativo	Gestión de usuarios, comunicados, solicitudes, solicitudes generales recibidas, solicitudes de justificaciones, matrículas, cursos de los docentes a cargo, pagos docentes, pago instructores y novedades presentadas por los docentes, instructores o estudiantes.	Ing. Carlos Guffante
Directivo de Contabilidad/Tesorería	Usuario que accede a la sección correspondiente de su departamento administrativo	Gestión de comunicados, solicitudes, solicitudes generales, inscripciones, matrículas, pago a docentes e instructores, venta de solicitudes y pago de mensualidad de los estudiantes.	Ing. Evelyn Delgado
Directivo de Gerencia	Usuario que accede a la sección correspondiente de su departamento administrativo	Gestión de comunicados, solicitudes, solicitudes generales recibidas, inscripciones, matrículas y pago a docentes e instructores	Ing. Ronald Rivera
Directivo de Servicio	Usuario que accede a la sección correspondiente de su departamento administrativo	Gestión de comunicados y solicitudes generales recibidas	Sr. Edison Serrano
Docente	Usuario que accede al sistema, pero	Gestión de comunicados, solicitudes y cursos asignados	Personal administrativo

USUARIO	DESCRIPCIÓN	ROL	RESPONSABLE
	únicamente a la sección asignada al docente		
Instructor	Usuario que accede al sistema, pero únicamente a la sección asignada al instructor	Gestión de comunicados, solicitudes y cursos asignados	Personal administrativo
Estudiante	Usuario que accede al sistema, pero únicamente a la sección asignada al estudiante	Gestión de comunicados, inscripción, matrícula, sílabos, planificaciones, evaluaciones a los docentes, pago de mensualidad, solicitudes generales y solicitudes de justificación	Personal administrativo
Practicante de DTIC	Usuario que accede al sistema, pero únicamente a la sección asignada al practicante de Dtic	Gestión de comunicados, solicitudes y proyectos	Personal administrativo

Realizado por: Aldas W.; Pilataxi C., 2020

3.3.2.8. *Sprint Backlog*

Para el desarrollo del sistema se ha establecido un total de 15 Sprints entregables, lo cual representa un avance periódico del proyecto, en cada uno de estos se integra un conjunto de varias actividades como las historias de usuario y las historias técnicas. La fecha de inicio de desarrollo del sistema es el 01/06/2020 y finaliza el 01/04/2021, siendo así que el proyecto tendrá una duración de 11 meses. Además, se debe tener en cuenta que cada Sprint entregable tiene un límite de tiempo de 15 días laborales.

En la **Tabla 8-3** se detalla el orden de desarrollo de las historias técnicas y las historias de usuario, que han sido distribuidos a lo largo de los Sprints planteados, además se ha agregado el esfuerzo que requieren cada uno de ellos, el tipo de actividad, la fecha de inicio y finalización.

Tabla 8-3: Planificación para el desarrollo del proyecto

N° Sprint	ID	Esfuerzo	Esfuerzo Total	Tipo	Fecha de Inicio	Fecha de Finalización
1	HT-01	16	120	Análisis	01/06/20	02/06/20
	HT-02	40		Análisis	03/06/20	09/06/20
	HT-03	24		Diseño	10/06/20	12/06/20
	HT-04	24		Desarrollo	15/06/20	17/06/20
	HT-05	16		Análisis	18/06/20	19/06/20
2	HU-01	24	120	Desarrollo	22/06/20	24/06/20
	HU-02	24		Desarrollo	25/06/20	29/06/20
	HU-03	16		Desarrollo	30/06/20	01/07/20
	HU-04	16		Desarrollo	02/07/20	03/07/20
	HU-05	8		Desarrollo	06/07/20	06/07/20
	HU-06	8		Desarrollo	07/07/20	07/07/20
	HU-07	16		Desarrollo	08/07/20	09/07/20
3	HT-06	8	120	Pruebas	10/07/20	10/07/20
	HU-08	16		Desarrollo	13/07/20	14/07/20
	HU-09	16		Desarrollo	15/07/20	16/07/20

N° Sprint	ID	Esfuerzo	Esfuerzo Total	Tipo	Fecha de Inicio	Fecha de Finalización
	HU-10	16		Desarrollo	17/07/20	20/07/20
	HU-11	24		Desarrollo	21/07/20	23/07/20
	HU-12	16		Desarrollo	24/07/20	27/07/20
	HU-13	24		Desarrollo	28/07/20	30/07/20
	HT-07	8		Pruebas	31/07/20	31/07/20
4	HU-14	8	120	Desarrollo	03/08/20	03/08/20
	HU-15	8		Desarrollo	04/08/20	04/08/20
	HU-16	16		Desarrollo	05/08/20	06/08/20
	HU-17	16		Desarrollo	07/08/20	10/08/20
	HU-18	24		Desarrollo	11/08/20	13/08/20
	HU-19	16		Desarrollo	14/08/20	17/08/20
	HU-20	16		Desarrollo	18/08/20	19/08/20
	HU-21	8		Desarrollo	20/08/20	20/08/20
	HT-08	8		Pruebas	21/08/20	21/08/20
5	HU-22	8	120	Desarrollo	24/08/20	24/08/20
	HU-23	8		Desarrollo	25/08/20	25/08/20
	HU-24	16		Desarrollo	26/08/20	27/08/20
	HU-25	8		Desarrollo	28/08/20	28/08/20
	HU-26	16		Desarrollo	31/08/20	01/09/20
	HU-27	8		Desarrollo	02/09/20	02/09/20
	HU-28	8		Desarrollo	03/09/20	03/09/20
	HU-29	16		Desarrollo	04/09/20	07/09/20
	HU-30	8		Desarrollo	08/09/20	08/09/20
	HU-31	16		Desarrollo	09/09/20	10/09/20
	HT-09	8		Pruebas	11/09/20	11/09/20
	6	HU-32		8	120	Desarrollo
HU-33		8	Desarrollo	15/09/20		15/09/20
HU-34		24	Desarrollo	16/09/20		18/09/20
HU-35		16	Desarrollo	21/09/20		22/09/20
HU-36		16	Desarrollo	23/09/20		24/09/20
HU-37		8	Desarrollo	25/09/20		25/09/20
HU-38		16	Desarrollo	28/09/20		29/09/20
HU-39		8	Desarrollo	30/09/20		30/09/20
HU-40		8	Desarrollo	01/10/20		01/10/20
HT-10		8	Pruebas	02/10/20		02/10/20
7		HU-41	16	120		Desarrollo
	HU-42	8	Desarrollo		07/10/20	07/10/20
	HU-43	8	Desarrollo		08/10/20	08/10/20
	HU-44	24	Desarrollo		09/10/20	13/10/20
	HU-45	24	Desarrollo		14/10/20	16/10/20
	HU-46	8	Desarrollo		19/10/20	19/10/20
	HU-47	16	Desarrollo		20/10/20	21/10/20
	HU-48	8	Desarrollo		22/10/20	22/10/20
	HT-11	8	Pruebas		23/10/20	23/10/20
8	HU-49	16	120	Desarrollo	26/10/20	27/10/20
	HU-50	8		Desarrollo	28/10/20	28/10/20
	HU-51	8		Desarrollo	29/10/20	29/10/20
	HU-52	8		Desarrollo	30/10/20	30/10/20
	HU-53	8		Desarrollo	02/11/20	02/11/20
	HU-54	8		Desarrollo	03/11/20	03/11/20
	HU-55	16		Desarrollo	04/11/20	05/11/20
	HU-56	8		Desarrollo	06/11/20	06/11/20

N° Sprint	ID	Esfuerzo	Esfuerzo Total	Tipo	Fecha de Inicio	Fecha de Finalización
	HU-57	24		Desarrollo	09/11/20	11/11/20
	HU-58	8		Desarrollo	12/11/20	12/11/20
	HT-12	8		Pruebas	13/11/20	13/11/20
9	HU-59	8	120	Desarrollo	16/11/20	16/11/20
	HU-60	8		Desarrollo	17/11/20	17/11/20
	HU-61	24		Desarrollo	18/11/20	20/11/20
	HU-62	8		Desarrollo	23/11/20	23/11/20
	HU-63	24		Desarrollo	24/11/20	26/11/20
	HU-64	40		Desarrollo	27/11/20	03/12/20
	HT-13	8		Pruebas	04/12/20	04/12/20
10	HU-65	40	120	Desarrollo	07/12/20	11/12/20
	HU-66	16		Desarrollo	14/12/20	15/12/20
	HU-67	8		Desarrollo	16/12/20	16/12/20
	HU-68	8		Desarrollo	17/12/20	17/12/20
	HU-69	40		Desarrollo	18/12/20	24/12/20
	HT-14	8		Pruebas	25/12/20	25/12/20
11	HU-70	16	120	Desarrollo	28/12/20	29/12/20
	HU-71	8		Desarrollo	30/12/20	30/12/20
	HU-72	8		Desarrollo	31/12/20	31/12/20
	HU-73	16		Desarrollo	01/01/21	04/01/21
	HU-74	24		Desarrollo	05/01/21	07/01/21
	HU-75	40		Desarrollo	08/01/21	14/01/21
	HT-15	8		Pruebas	15/01/21	15/01/21
12	HU-76	24	120	Desarrollo	18/01/21	20/01/21
	HU-77	24		Desarrollo	21/01/21	25/01/21
	HU-78	16		Desarrollo	26/01/21	27/01/21
	HU-79	8		Desarrollo	28/01/21	28/01/21
	HU-80	24		Desarrollo	29/01/21	02/02/21
	HU-81	8		Desarrollo	03/02/21	03/02/21
	HU-82	8		Desarrollo	04/02/21	04/02/21
	HT-16	8		Pruebas	05/02/21	05/02/21
13	HU-83	24	120	Desarrollo	08/02/21	10/02/21
	HU-84	8		Desarrollo	11/02/21	11/02/21
	HU-85	16		Desarrollo	12/02/21	15/02/21
	HU-86	8		Desarrollo	16/02/21	16/02/21
	HU-87	16		Desarrollo	17/02/21	18/02/21
	HU-88	24		Desarrollo	19/02/21	23/02/21
	HU-89	16		Desarrollo	24/02/21	25/02/21
	HT-17	8		Pruebas	26/02/21	26/02/21
14	HU-90	16	120	Desarrollo	01/03/21	02/03/21
	HU-91	16		Desarrollo	03/03/21	04/03/21
	HU-92	16		Desarrollo	05/03/21	08/03/21
	HU-93	24		Desarrollo	09/03/21	11/03/21
	HU-94	24		Desarrollo	12/03/21	16/03/21
	HU-95	16		Desarrollo	17/03/21	18/03/21
	HT-18	8		Pruebas	19/03/21	19/03/21
15	HU-96	24	72	Desarrollo	22/03/21	24/03/21
	HU-97	24		Desarrollo	25/03/21	29/03/21
	HU-98	16		Desarrollo	30/03/21	31/03/21
	HT-19	8		Pruebas	01/04/21	01/04/21
Puntos totales: 1752				Planificación: Del 01/06/20 Al 01/04/21		

Realizado por: Aldas W.; Pilataxi C., 2020

En la **Tabla 8-3** se estable Sprint Backlog del sistema, por lo que para clasificarlos se ha considerado los siguientes campos:

- **N° Sprint:** Indica la cantidad de sprints que se han establecido para el cumplimiento del proyecto, en este caso van de 1 a 15.
- **ID:** Representa el identificador de la historia de usuario o técnica con su respectivo número, para cada sprint.
- **Esfuerzo:** Es la cantidad de puntos individuales por historia de usuario o técnica que previamente se ha establecido en el producto backlog.
- **Esfuerzo total:** Es la suma total de los puntos individuales de cada una de las historias, por cada sprint, en total cada sprint tiene un valor de 120 puntos, que representa 15 días laborales, a excepción del sprint 15 el cual únicamente cuenta con 72 puntos.
- **Fecha de inicio:** Representa la fecha de inicialización de cada uno de los sprints.
- **Fecha de finalización:** Representa la fecha de finalización de cada uno de los sprints.

3.3.2.9. *Identificación del plan de pruebas de software*

Una metodología de pruebas ofrece los pasos a seguir para probar la funcionalidad de un determinado sistema, sin embargo, no existe una rutina específica a la cual debemos regirnos, debido a que cada proyecto tiene sus propios objetivos a cubrir. A pesar de esto, la gran mayoría de metodologías coinciden en que las etapas a considerar como esenciales son la identificación, diseño, construcción, ejecución y comparación de resultados de prueba, todo esto independientemente de que tipo de prueba se vaya a llevar a cabo, esta serie de pasos deben ser ejecutados de manera secuencial es decir una a continuación de la otra y en orden. Una vez que se tiene claro la secuencia de tareas por hacer, lo siguiente es identificar los denominados artefactos de prueba, que son herramientas que permiten ejecutar todo el proceso planteado.

Los artefactos al igual que las metodologías pueden variar de proyecto a proyecto, sin embargo, los que se consideran indispensables son, el plan de pruebas, los casos de prueba, los scripts de prueba, las suites de prueba, los datos de prueba y el informe de resultados.

Partiendo de estas especificaciones, lo siguiente es establecer el tipo de pruebas que se quieran realizar y de acuerdo al modelo V, el cual es el más utilizado dentro del ámbito del testing, existen las pruebas de unidad, integración, sistema y aceptación, los cuales a su vez se corresponden a las distintas etapas de la metodología de desarrollo de software.

Con la finalidad de demostrar el uso de la metodología con los respectivos artefactos de prueba

anteriormente descritos y tomando como base el modelo V, se ha establecido realizar las pruebas unitarias al sistema de gestión de información de los procesos administrativos de la escuela de conducción profesional CONDUESPOCH E.P., este tipo de pruebas ha sido escogido debido a su facilidad y rapidez de desarrollo, ejecución y documentación, en comparación a otros tipos, que aparte de eso necesitan recursos tecnológicos más específicos de lo cual no disponemos.

Los artefactos establecidos a realizar son, el plan de pruebas en cual se detallan de manera descriptiva sobre las características a cumplir, lo siguiente son los casos de prueba que son denominados conjunto de condiciones a superar por una funcionalidad, el script de prueba que es la cantidad de instrucciones para probar un recurso en específico, la suite de prueba que es el total de varios casos de prueba, los datos de prueba que son introducidos a cada caso de prueba con el fin de evaluar la funcionalidad y finalmente el informe de resultados obtenidos.

Los casos de prueba planteadas pertenecen únicamente a los requerimientos funcionales del sistema, los cuales en su mayoría son de tipo ingresar, listar, modificar y eliminar datos dentro de todo el sistema. Por otro lado, la manera más simple de verificar el funcionamiento de una prueba unitaria es midiendo la cobertura de código, el cual es una técnica que consiste en analizar la cantidad de líneas de código que se está y no se está probando dentro de una funcionalidad. La cobertura de código por lo general se expresa en un porcentaje que va de 0 a 100, sin embargo, que una prueba cubra el 100% de las líneas de código no significa que no tenga errores, pero es una forma rápida de obtener un informe detallado de la situación.

Para la realización de las pruebas unitarias se emplea el Framework Jest, el cual nos permite escribir, ejecutar y documentar cada una de las pruebas, esto incluye la obtención de la cobertura de código de cada una de las pruebas realizadas.

Alcance

El alcance de las pruebas consiste en especificar el tipo de pruebas a implementar en el desarrollo de un sistema, además también se excluye aquellas pruebas que no se van a realizar.

Las pruebas planteadas para este trabajo se detallan a continuación en la **Tabla 9-3**:

Tabla 9-3: Especificación del alcance de las pruebas

#	Especificación
1	Este plan aplica para probar el desarrollo del sistema de gestión de información administrativa de la escuela de conducción profesional CONDUESPOCH E.P en su primera versión.
2	Este plan aplica únicamente para el desarrollo de las funcionalidades planteadas por el usuario, los cuales por lo general son ingresar, listar, actualizar y eliminar datos.
3	Las pruebas establecidas dentro de este plan, son las pruebas unitarias, las cuales son un tipo de prueba de caja blanca.
4	No se realizará ningún otro tipo de pruebas ya sean de caja blanca, caja negra o caja gris, debido a que no se cuenta con los recursos necesarios para llevarlos a cabo.

Realizado por: Aldas W.; Pilataxi C., 2020

Ítems a probar

Las pruebas unitarias al ser un tipo de pruebas de caja blanca, necesitan ser ejecutadas en un ambiente previamente configurado para dicho propósito. La configuración requerida se detalla a continuación en la **Tabla 10-3:**

Tabla 10-3: Especificación de los ítems a probar

#	Especificación
1	Tener una versión del sistema cargada y levantada en un servidor de manera local con las respectivas pruebas a desarrollar.
2	Haber instalado previamente el entorno de desarrollo Node js, el cual incluye todas las dependencias necesarias para el soporte de React js.
3	Haber instalado previamente la librería Jest en el entorno local, ya sea por medio del archivo de dependencias del mismo proyecto o de manera remota.
4	Disponibilidad de una consola de comandos que permita iniciar el proceso de pruebas automatizadas mediante el comando correspondiente.

Realizado por: Aldas W.; Pilataxi C., 2020

Cada caso de prueba planteado en este trabajo tiene su propia pre-condición, post-condición y el resultado esperado. Además, se debe tomar en cuenta que para el desarrollo de las pruebas se ha hecho uso de mocks para simular el ingreso de datos en su correspondiente evaluación, básicamente los mocks representan datos falsos, es decir que no alteran los registros reales del sistema. A continuación, en la **Tabla 11-3**, se detallan los casos de pruebas identificados por cada historia de usuario, los cuales se han llevado a cabo durante los Sprints planteados en la metodología Scrum.

Tabla 11-3: Definición de casos de prueba por historia de usuario

Historias de usuario	#	Casos de prueba
HU-01 Autenticación de los usuarios por número de cédula y contraseña. Sprint 2	1	Verificar que se ejecute la acción de envío del formulario de autenticación
	2	Verificar que se ejecute la función reductora de autenticación de usuarios
	3	Verificar que el token devuelto se aloje en el localStorage del navegador
	4	Verificar que los datos devueltos de usuario no sean nulos
	5	Verificar que las credenciales de usuario no sean nulas

Historias de usuario	#	Casos de prueba
	6	Verificar que los datos de la foto de usuario no sean nulos
	7	Verificar que los datos de usuario del localStorage asignado al sistema se actualice
	8	Verificar que las credenciales de usuario del localStorage asignado al sistema se actualice
	9	Verificar que la foto de usuario del localStorage asignado al sistema se actualice
HU-02 Modificar datos personales. Sprint 2	1	Verificar que se ejecuta la acción de modificar los datos del usuario
	2	Verificar que la información del localStorage se ha actualizado
	3	Verificar que se devuelva el estado de modificación
	4	Verificar que se actualice las credenciales del usuario
	5	Verificar que las credenciales del usuario se almacenen en el listado de usuarios correspondiente
	6	Verificar que se devuelve el estado de actualización de las credenciales del usuario
	7	Verificar que se ejecuta la acción de modificar la foto de usuario
	8	Verificar que se actualice la foto de perfil del usuario en el localStorage
	9	Verificar que se devuelve el mensaje de estado de actualización de la foto de usuario
HU-03 Registrar un usuario. Sprint 2	1	Verificar que se ejecuta la acción de ingresar un usuario administrador
	2	Verificar que los datos de registro de administrador sean válidos
	3	Verificar que se envíe un solo registro de administrador
	4	Verificar que no se envíen datos nulos en el registro de administrador.
	5	Verificar que se devuelve el estado de ingreso de administrador
	6	Verificar que se ejecuta la acción de ingresar un usuario directivo
	7	Verificar que los datos de registro de directivo sean válidos
	8	Verificar que se envíe un solo registro de directivo
	9	Verificar que no se envíen datos nulos en el registro de directivo.
	10	Verificar que se devuelve el estado de ingreso de directivo
	11	Verificar que se ejecuta la acción de ingresar un usuario docente
	12	Verificar que los datos de registro de docente sean válidos
	13	Verificar que se envíe un solo registro de docente
	14	Verificar que no se envíen datos nulos en el registro de docente.
	15	Verificar que se devuelve el estado de ingreso de docente
	16	Verificar que se ejecuta la acción de ingresar un usuario estudiante
	17	Verificar que los datos de registro de estudiante sean válidos
	18	Verificar que se envíe un solo registro de estudiante
	19	Verificar que no se envíen datos nulos en el registro de estudiante.
	20	Verificar que se devuelve el estado de ingreso de estudiante
	21	Verificar que se ejecuta la acción de ingresar un usuario practicante
	22	Verificar que los datos de registro de practicante sean válidos
	23	Verificar que se envíe un solo registro de practicante
	24	Verificar que no se envíen datos nulos en el registro de instructor.
	25	Verificar que se devuelve el estado de ingreso de instructor
	26	Verificar que se ejecuta la acción de ingresar un usuario instructor
	27	Verificar que los datos de registro de instructor sean válidos
	28	Verificar que se envíe un solo registro de instructor
	29	Verificar que no se envíen datos nulos en el registro de instructor.
	30	Verificar que se devuelve el estado de ingreso de instructor
	31	Verificar que se ha ejecutado la función reductora para agregar un nuevo usuario administrador
	32	Verificar que se ha ejecutado la función reductora para agregar un nuevo usuario directivo
	33	Verificar que se ha ejecutado la función reductora para agregar un nuevo usuario docente
	34	Verificar que se ha ejecutado la función reductora para agregar un nuevo usuario estudiante
	35	Verificar que se ha ejecutado la función reductora para agregar un

Historias de usuario	#	Casos de prueba
		nuevo usuario instructor
	36	Verificar que se ha ejecutado la función reductora para agregar un nuevo usuario practicante
HU-04 Modificar datos de un usuario. Sprint 2	1	Verificar que se ejecute las acciones de modificar los datos de un administrador
	2	Verificar que se ejecute la función de modificar los datos de un administrador
	3	Verificar que se retorne el estado de actualización de un administrador
	4	Verificar que se ejecute las acciones de modificar los datos de un directivo
	5	Verificar que se ejecute la función de modificar los datos de un directivo
	6	Verificar que se retorne el estado de actualización de un directivo
	7	Verificar que se ejecute las acciones de modificar los datos de un docente
	8	Verificar que se ejecute la función de modificar los datos de un docente
	9	Verificar que se retorne el estado de actualización de un docente
	10	Verificar que se ejecute las acciones de modificar los datos de un estudiante
	11	Verificar que se ejecute la función de modificar los datos de un estudiante
	12	Verificar que se retorne el estado de actualización de un estudiante
	13	Verificar que se ejecute las acciones de modificar los datos de un practicante
	14	Verificar que se ejecute la función de modificar los datos de un practicante
	15	Verificar que se retorne el estado de actualización de un practicante
	16	Verificar que se ejecute las acciones de modificar los datos de un instructor
	17	Verificar que se ejecute la función de modificar los datos de un instructor
	18	Verificar que se retorne el estado de actualización de un instructor
	19	Verificar que se ejecuta función reductora para agregar un nuevo administrador
	20	Verificar que se ejecuta función reductora para agregar un nuevo directivo
	21	Verificar que se ejecuta función reductora para agregar un nuevo docente
	22	Verificar que se ejecuta función reductora para agregar un nuevo instructor
	23	Verificar que se ejecuta función reductora para agregar un nuevo practicante
HU-05 Eliminar datos de un usuario. Sprint 2	1	Verificar que se ejecuten las acciones de eliminar un usuario administrador
	2	Verificar que se ejecute la acción de eliminar un usuario administrador
	3	Verificar que se devuelva el estado de eliminar un usuario administrador
	4	Verificar que se ejecuten las acciones de eliminar un usuario directivo
	5	Verificar que se ejecute la acción de eliminar un usuario directivo
	6	Verificar que se devuelva el estado de eliminar un usuario directivo
	7	Verificar que se ejecuten las acciones de eliminar un usuario docente
	8	Verificar que se ejecute la acción de eliminar un usuario docente
	9	Verificar que se devuelva el estado de eliminar un usuario docente
	10	Verificar que se ejecuten las acciones de eliminar un usuario estudiante
	11	Verificar que se ejecute la acción de eliminar un usuario estudiante
	12	Verificar que se devuelva el estado de eliminar un usuario

Historias de usuario	#	Casos de prueba
		estudiante
	13	Verificar que se ejecuten las acciones de eliminar un usuario practicante
	14	Verificar que se ejecute la acción de eliminar un usuario practicante
	15	Verificar que se devuelva el estado de eliminar un usuario practicante
	16	Verificar que se ejecuten las acciones de eliminar un usuario instructor
	17	Verificar que se ejecute la acción de eliminar un usuario instructor
	18	Verificar que se devuelva el estado de eliminar un usuario instructor
	19	Verificar que se ejecuta la función reductora de eliminar un usuario administrador
	20	Verificar que se ejecuta la función reductora de eliminar un usuario directivo
	21	Verificar que se ejecuta la función reductora de eliminar un usuario docente
	22	Verificar que se ejecuta la función reductora de eliminar un usuario estudiante
	23	Verificar que se ejecuta la función reductora de eliminar un usuario instructor
	24	Verificar que se ejecuta la función reductora de eliminar un usuario practicante
HU-06 Ver listado de usuarios por roles. Sprint 2.	1	Verificar la cantidad de usuarios registrados administradores
HU-07 Buscar usuarios por número de cédula o nombres. Sprint 2	2	Verificar que hay un listado de usuarios administradores
	3	Verificar que se ejecute la acción de obtener usuarios administradores
	4	Verificar la cantidad de usuarios registrado directivos
	5	Verificar que hay un listado de usuarios directivos
	6	Verificar que se ejecute la acción de obtener usuarios directivos
	7	Verificar la cantidad de usuarios registrados estudiantes
	8	Verificar que hay un listado de usuarios estudiantes
	9	Verificar que se ejecute la acción de obtener usuarios estudiantes
	10	Verificar la cantidad de usuarios registrados docentes
	11	Verificar que hay un listado de usuarios docentes
	12	Verificar que se ejecute la acción de obtener usuarios docentes
	13	Verificar la cantidad de usuarios registrados instructores
	14	Verificar que hay un listado de usuarios instructores
	15	Verificar que se ejecute la acción de obtener usuarios instructores
	16	Verificar la cantidad de usuarios registrados practicantes
	17	Verificar que hay un listado de usuarios practicantes
	18	Verificar que se ejecute la acción de obtener usuarios practicantes
	19	Verificar que se devuelve el valor por defecto de la función reductora de administradores
	20	Verificar que se obtiene un listado vacío de administradores
	21	Verificar que se obtiene un listado de datos de administradores
	22	Verificar que se devuelve el valor por defecto de la función reductora de directivos
	23	Verificar que se obtiene un listado vacío de directivos
	24	Verificar que se obtiene un listado de datos de directivos
	25	Verificar que se devuelve el valor por defecto de la función reductora de docentes
	26	Verificar que se obtiene un listado vacío de docentes
	27	Verificar que se obtiene un listado de datos de docentes
	28	Verificar que se devuelve el valor por defecto de la función reductora de estudiantes
	29	Verificar que se obtiene un listado vacío de estudiantes
	30	Verificar que se obtiene un listado de datos de estudiantes
	31	Verificar que se devuelve el valor por defecto de la función reductora de instructores
	32	Verificar que se obtiene un listado vacío de instructores
	33	Verificar que se obtiene un listado de datos de instructores

Historias de usuario	#	Casos de prueba
	34	Verificar que se devuelve el valor por defecto de la función reductora de practicantes
	35	Verificar que se obtiene un listado vacío de practicantes
	36	Verificar que se obtiene un listado de datos de practicantes
HU-08 Migrar datos de docentes del sistema de calificaciones de la institución. Sprint 3	1	Verificar que haya un listado de docentes
	2	Verificar que se ejecute la acción de obtener los datos de los docentes a migrar
	3	Verificar que se ejecuten las acciones necesarias para la obtención de datos
	4	Verificar que se han migrado los docentes seleccionados
	5	Verificar que se ejecute la acción de migrar docentes
	6	Verificar que se ejecuten las acciones necesarias para el envío de datos
HU-09 Cambiar de departamento a un usuario con rol de directivo. Sprint 3	1	Verificar que el listado de roles de usuario
	2	Verificar que se ejecute la acción de obtener los roles de usuario
	3	Verificar que se ejecuten las acciones necesarias para la obtención de roles
	4	Verificar el listado de usuarios directivos por roles
	5	Verificar que se ejecute la acción de actualizar el rol de usuario directivo
	6	Verificar que se han ejecutado las acciones necesarias para la actualización de roles.
	7	Verificar que se obtiene el valor por defecto de la función reductora
	8	Verificar el array vacío de roles de usuario
	9	Verificar el listado de roles de usuario
HU-10 Registro de datos de inscripción del estudiante. Sprint 3	1	Verificar que se obtiene el valor por defecto de la función reductora
	2	Verificar los datos vacíos de inscripción del estudiante
	3	Verificar los datos almacenados de inscripción del estudiante
HU-11 Listar las inscripciones recibidas en su respectivo departamento. Sprint 3. HU-12 Buscar datos de inscripción por cédula de identidad o nombres del usuario. Sprint 3.	1	Verificar el listado de inscripciones por departamento
	2	Verificar que al menos se tengo una inscripción registrada
	3	Verificar que se ejecuten las acciones necesarias para la obtención de inscripciones por departamento.
	4	Verificar que se ejecuta la acción de aprobar una inscripción
	5	Verificar que se ejecuten la cantidad de acciones necesarias para la autorización de inscripciones
	6	Verificar que se ejecute la acción de eliminar una autorización
	7	Verificar que se ejecuten la cantidad de acciones necesarias para eliminar la autorización de una inscripción.
	8	Verificar que se ejecute la acción de eliminar una inscripción
	9	Verificar que se envía el código de eliminar una inscripción
	10	Verificar que se ejecuten la cantidad de acciones para eliminar una inscripción.
	11	Verificar que la función devuelva el valor por defecto
	12	Verificar que se obtiene un listado vacío de inscripciones
	13	Verificar que se obtiene un listado de datos de inscripciones
	14	Verificar que se ejecuta la función reductora de actualizar el estado de una inscripción
	15	Verificar que se ejecuta la función reductora de eliminar una inscripción
HU-13 Ingresar fecha de inicio y fin de inscripción o matrícula. Sprint 3	1	Verificar que se ejecuten las acciones necesarias para el ingreso de fechas de inscripción o matrícula
	2	Verificar que el ingreso de una fecha de inscripción o matrícula
	3	Verificar que se envía un registro de fecha de inscripción o matrícula
	5	Verificar que no se envían datos nulos
	5	Verificar que se recibe el estado de registro de la fecha de inscripción o matrícula
	6	Verificar que se ejecute la función reductora de obtención de datos de las fechas de inscripción o matrícula
HU-14 Listar fechas de inscripción y matrícula. Sprint 4	1	Verificar el listado de inscripciones y matriculas
	2	Verificar la última fecha de inscripción
	3	Verificar la última fecha de matrícula

Historias de usuario	#	Casos de prueba	
	4	Verificar la cantidad de acciones que se ejecutan por cada acción	
	5	Verificar que la función reductora retorna el valor por defecto del listado de fechas	
	6	Verificar el listado vacío de fechas	
	7	Verificar el listado de fechas de inscripción y matrícula	
	8	Verificar que la función reductora retorna el valor por defecto de una fecha en específico	
	9	Verificar que se ha obtenido un arreglo vacío de una fecha en específico	
	10	Verificar que se ha obtenido un arreglo con datos de una fecha en específico	
	11	Verificar que la función reductora retorna el valor por defecto de la última fecha de inscripción	
	12	Verificar que se ha obtenido un arreglo vacío de la última fecha de inscripción	
	13	Verificar que se ha obtenido un arreglo con datos de la última fecha de inscripción	
	14	Verificar que la función reductora retorna el valor por defecto de la última fecha de matrícula	
	15	Verificar que se ha obtenido un arreglo vacío de la última fecha de matrícula	
	16	Verificar que se ha obtenido un arreglo con datos de la última fecha de matrícula	
	HU-15 Eliminar la fecha de inscripción o matrícula. Sprint 4	1	Verificar que se ejecute la función de eliminar una fecha de inscripción o matrícula
	HU-16 Modificar las fechas de inscripción o matrícula. Sprint 4	1	Verificar la cantidad de acciones que se ejecutan al momento de modificar una fecha de inscripción o matrícula
	HU-17 Registrar datos de matrícula de un estudiante. Sprint 4	1	Verificar que se envía el formulario de registro de matrícula por estudiante
2		Verificar que se obtiene la matrícula registrada	
3		Verificar que se obtiene el valor por defecto de la función reductora	
4		Verificar el arreglo vacío de la matrícula del estudiante	
5		Verificar el arreglo de datos de la matrícula del estudiante	
HU-18 Listar las matrículas recibidas en su respectivo departamento. Sprint 4 HU-19: Buscar datos de matrícula por cédula de identidad o nombres del usuario. Sprint 4	1	Verificar el listado de matrículas por departamento	
	2	Verificar que se actualiza el estado del registro de matrícula	
	3	Verificar que se elimina el estado de la matrícula	
	4	Verificar que se elimina los datos de una matrícula	
	5	Verificar que se devuelva el valor por defecto de la función reductora	
	6	Verificar el array vacío de matrículas en un determinado departamento	
	7	Verificar el listado de matrículas en un determinado departamento	
	8	Verificar que se ejecute la función reductora de actualizar datos de una matrícula	
	9	Verificar que se ejecute la función reductora de eliminar datos de una matrícula	
HU-20 Ingresar una solicitud al departamento de Asesoría Vial. Sprint 4	1	Verificar que se envíe el formulario sin datos nulos	
	2	Verificar que se ejecute la acción de envío de formulario de solicitud	
	3	Verificar que se devuelva el estado de envío de la solicitud	
HU-21 Modificar una solicitud al departamento de Asesoría Vial. Sprint 4	1	Verificar que se ejecutan las acciones correspondiente para la actualización de los datos de la solicitud	
	2	Verificar que se actualiza los datos de la solicitud	
HU-22 Eliminar una solicitud al departamento de Asesoría Vial. Sprint 5	1	Verificar la cantidad de acciones que se ejecutan al momento de eliminar una solicitud	
	2	Verificar que se ejecute la acción de eliminar una solicitud	
	3	Verificar que se devuelva el estado de eliminar una solicitud	
HU-23 Listar las solicitudes al departamento de Asesoría Vial. Sprint 5 HU-24 Buscar una solicitud del departamento de Asesoría Vial.	1	Verificar la cantidad de acciones que se ejecutan	
	2	Verificar el listado de solicitudes enviadas	
	3	Verificar el estado por defecto de la función reductora	
	4	Verificar el arreglo vacío de las solicitudes enviadas	
	5	Verificar el listado de solicitudes enviadas	

Historias de usuario	#	Casos de prueba
Sprint 5	6	Verificar que se ejecute la acción de eliminar una solicitud enviada
	7	Verificar que se ejecute la acción de modificar una solicitud enviada
HU-25 Listar los vehículos de la institución. Sprint 5	1	Verificar la cantidad de vehículos que posee la institución
	2	Verificar el listado de vehículos de la institución
	3	Verificar la cantidad de acciones que se ejecuta para el listado de vehículos
HU-26 Ingresar un informe de gestión vehicular. Sprint 5	1	Verificar que se envíe un informe válido del estado vehicular
	2	Verificar la acción que se ejecuta para el envío del informe
	3	Verificar la cantidad de acciones que se ejecutan
HU-27 Listar informes de gestión vehicular. Sprint 5	1	Verificar el listado de informes de estado vehicular
	2	Verificar que se ejecute la acción de listado de informes de estado vehicular
	3	Verificar la cantidad de acciones que se ejecutan para la obtención de informes de estado vehicular
HU-28 Eliminar un informe de gestión vehicular. Sprint 5	1	Verificar que se ejecute la acción de eliminar un informe
	2	Verificar la cantidad de acciones que se ejecutan para eliminar un informe
	3	Verificar el valor por defecto de la función reductora de listado de vehículos
	4	Verificar el arreglo vacío de vehículos
	5	Verificar el listado de vehículos de la institución
	6	Verificar el valor por defecto de la función reductora del estado de un vehículo
	7	Verificar el arreglo vacío del estado del vehículo
	8	Verificar el listado de datos de estado del vehículo
	9	Verificar si se agrega un nuevo estado del vehículo
	10	Verificar si se elimina un estado del vehículo
HU-29 Ingreso de solicitud al departamento de DTIC. Sprint 5	1	Verificar el ingreso de datos correctos en la solicitud
HU-30 Listar solicitudes del departamento de DTIC. Sprint 5	1	Verificar que se ejecuten la cantidad de acciones necesarias
	2	Verificar el listado de solicitudes recibidas
	3	Verificar que se ejecute la acción de listar solicitudes
HU-31 Modificar solicitud al departamento de DTIC. Sprint 5	1	Verificar la cantidad necesaria de acciones ejecutadas
	2	Verificar que se actualice los datos de la solicitud
HU-32 Eliminar una solicitud al departamento de DTIC. Sprint 6	1	Verificar que se ejecuten la cantidad de acciones necesarias
	2	Verificar que se elimina un registro de solicitud
	3	Verificar el estado devuelto de eliminar una solicitud
HU-33 Buscar una solicitud del departamento de DTIC. Sprint 6	1	Verificar que se ejecute la cantidad de acciones correspondientes a la búsqueda de una solicitud
	2	Verificar el estado de búsqueda de datos
HU-34 Ingresar un nuevo proyecto de DTIC. Sprint 6	1	Verificar que se envíe datos correctos en el formulario
	2	Verificar que se ejecute la cantidad de acciones correspondientes
	3	Verificar que envíe el formulario de registro
	4	Verificar que se envíe únicamente un formulario a la vez
HU-35 Listar los proyectos de DTIC. Sprint 6	1	Verificar el listado de solicitudes aprobadas
	2	Verificar que se ejecute la acción que permite listar las solicitudes aprobadas
	3	Verificar la cantidad de proyectos aprobados
	4	Verificar el listado de proyectos por solicitud aprobada
	5	Verificar que se ejecuten la cantidad de acciones necesarias
	6	Verificar el resultado por defecto de la función reductora
	7	Verificar el array vacío de las solicitudes aprobadas
	8	Verificar el listado de solicitudes aprobadas devueltas por la función reductora
HU-36 Modificar un proyecto de DTIC. Sprint 6	1	Verificar que se ejecuten las acciones necesarias para modificar los datos de un proyecto
	2	Verificar que se actualice los datos de un proyecto
HU-37 Eliminar un proyecto de DTIC. Sprint 6	1	Verificar que se ejecuten las acciones necesarias para eliminar un proyecto
	2	Verificar que se elimine los datos de un proyecto
HU-38 Ingresar tareas por proyecto.	1	Verificar que no se envíe datos nulos en el formulario de tarea

Historias de usuario	#	Casos de prueba
Sprint 6	2	Verificar que se ejecuten la cantidad de acciones necesarias para el ingreso de tareas.
	3	Verificar que se agregue una nueva tarea a un proyecto
	4	Verificar que se obtenga el valor por defecto de la función reductora de ingreso de tarea
	5	Verificar que se ha obtenido un array vacío de tareas
	6	Verificar que se ha obtenido un listado de tareas de un proyecto por practicante
	7	Verificar que se ha actualizado el estado de una tarea mediante la función reductora correspondiente
	8	Verificar que se ha eliminado la tarea de un proyecto mediante la función reductora correspondiente
	9	Verificar que se ha ingresado un nuevo proyecto mediante la función reductora correspondiente
	10	Verificar que se ha ingresado una nueva tarea mediante la función reductora correspondiente
	11	Verificar que se ha ingresado un nuevo avance de tarea mediante la función reductora correspondiente
	12	Verificar que se ha eliminado un archivo de tarea mediante la función reductora correspondiente
	HU-39 Asignar un practicante a una tarea. Sprint 6 HU-40 Eliminar un practicante de una tarea. Sprint 6	1
2		Verificar que se agregue un nuevo practicante a una tarea
3		Verificar el listado de practicantes por tareas
HU-41 Modificar una tarea. Sprint 7	1	Verificar que se ejecuten las acciones correspondientes para modificar una tarea
	2	Verificar que se actualiza la tarea de un proyecto
	3	Verificar que se envía un formulario de actualización
	4	Verificar que se ejecuten las acciones correspondientes para modificar el estado de una tarea
	5	Verificar que se actualiza el estado de una tarea
	6	Verificar que se ejecuten las acciones correspondientes para eliminar el estado de una tarea
	7	Verificar que sea eliminado una tarea
HU-42 Eliminar una tarea. Sprint 7	1	Verificar que se ejecuten las acciones correspondientes para eliminar una tarea
	2	Verificar que se ha eliminado la tarea de un proyecto
HU-43 Listar las tareas asignadas a un practicante. Sprint 7	1	Verificar la cantidad de tareas por de los practicantes
	2	Verificar que se ha obtenido el listado de tareas de los practicantes
	3	Verificar que se ejecuten la cantidad de acciones ejecutadas para el listado de tareas
	4	Verificarla cantidad de tareas de un practicante
	5	Verificar el listado de un practicante
	6	Verificar la cantidad de acciones que se ejecutan al listar las tareas de un practicante
	7	Verificar el listado de tareas de un practicante
	8	Verificar que se agrega un nuevo practicante a la tarea de un proyecto
	9	Verificar que se ha removido un practicante de la tarea de un proyecto
	10	Verificar que se devuelve el estado por defecto de la función reductora
	11	Verificar que se obtiene un arreglo vacío de proyectos
	12	Verificar que se obtiene un listado de proyectos
	13	Verificar que se ha eliminado un proyecto
	14	Verificar que se ha actualizado los datos de un proyecto
	15	Verificar que se ha almacenado un nuevo proyecto por solicitud aprobada
	16	Verificar que se ha actualizado el estado de un practicante
	17	Verificar que se ha eliminado un practicante de la tarea de un proyecto
	18	Verificar que se ha modificado las tareas de un proyecto

Historias de usuario	#	Casos de prueba
HU-44 Ingresar una solicitud al departamento Gerencial. Sprint 7	1	Verificar que se envíe datos de formulario validos
	2	Verificar que se ejecuten la cantidad de acciones necesarias
	3	Verificar que se agrega una nueva solicitud
HU-45 Listado de solicitudes del departamento Gerencial. Sprint 7	1	Verificar que se ejecuten la cantidad de acciones necesarias
	2	Verificar el listado de solicitudes enviadas
	3	Verificar que se ejecute la acción de listar solicitudes enviadas
HU-46 Eliminar una solicitud del departamento Gerencial. Sprint 7	1	Verificar que se ejecuten la cantidad de acciones necesarias para eliminar una solicitud
	2	Verificar que se ejecute la acción de eliminar una solicitud enviada
	3	Verificar que se muestre el estado de eliminar una solicitud enviada
	4	Verificar el listado de solicitudes enviadas
	5	Verificar que se ejecuta la acción de eliminar una solicitud
	6	Verificar que se actualiza los datos de una solicitud enviada
	7	Verificar que se elimina la autorización de una solicitud
	8	Verificar que se ha obtenido el valor por defecto de la función reductora
	9	Verificar que se ha obtenido un arreglo vacío de datos
	10	Verificar que se ha obtenido el listado de solicitudes enviadas
	11	Verificar que se ha eliminado los datos de una solicitud enviada
	12	Verificar que se ha modificado los datos de una solicitud enviada
HU-47 Ingreso de novedades de los docentes, instructores y estudiantes. Sprint 7	1	Verificar que se envíe datos correctos
	2	Verificar que se ejecuten la cantidad de acciones necesarias
	3	Verificar que se ingresa una novedad
	4	Verificar que se ha devuelto el estado de registro de una novedad
HU-48 Modificar novedades de los docentes, instructores y estudiantes. Sprint 7	1	Verificar que se ejecuten la cantidad de acciones necesarias para modificar una novedad de usuario
	2	Verificar que se ha modificado una novedad de usuario
	3	Verificar que se ha devuelto el estado de actualizar una novedad
HU-49 Eliminar novedades de los docentes, instructores y estudiantes. Sprint 8	1	Verificar que se ejecuten la cantidad de acciones necesarias para eliminar una novedad
	2	Verificar que se elimine una novedad
	3	Verificar que se ha devuelto el estado de eliminar una novedad
HU-50 Listar novedades de los docentes, instructores y estudiantes. Sprint 8	1	Verificar la cantidad de novedades recibidas
	2	Verificar el listado de novedades ingresadas
	3	Verificar que se ejecuten la cantidad de acciones necesarias para listar novedades
HU-51 Ingresar un comunicado. Sprint 8	1	Verificar que se envía el formulario con datos validos
	2	Verificar que se ejecuten la cantidad de acciones necesarias para el ingreso de un comunicado
	3	Verificar que se devuelve el estado de ingreso de comunicado
HU-52 Modificar un comunicado. Sprint 8	1	Verificar que se ejecuten la cantidad de acciones necesarias para el modificar un comunicado
	2	Verificar que se ha modificado un comunicado enviado
HU-53 Eliminar un comunicado. Sprint 8	1	Verificar que se ejecuten la cantidad de acciones necesarias para el eliminar un comunicado
	2	Verificar que se ha eliminado un comunicado enviado
	3	Verificar que se ha devuelto el estado de eliminar un comunicado
HU-54 Listar comunicados. Sprint 8	1	Verificar la cantidad de acciones ejecutadas para el listado de comunicados
	2	Verificar el listado de comunicados enviados
	3	Verificar que se ha devuelto el estado de listar comunicados
	4	Verificar que la función reductora ha devuelto el estado por defecto del listado de comunicados
	5	Verificar que se ha obtenido un arreglo vacío del listado de comunicados
	6	Verificar que se ha obtenido el listado de comunicados
	7	Verificar que se ha eliminado un comunicado
	8	Verificar que se ha modificado un comunicado
HU-55 Registro de aulas. Sprint 8	1	Verificar que se envía un formulario válido
	2	Verificar que se ejecuten la cantidad de acciones necesarias para el registro de un aula

Historias de usuario	#	Casos de prueba
	3	Verificar que se ha devuelto el estado de registro de un aula
HU-56 Listado de aulas. Sprint 8	1	Verificar la cantidad de acciones ejecutadas para el listado de aulas
	2	Verificar el listado de las aulas
	3	Verificar que se ejecuta la acción de listado de las aulas
HU-57 Modificar datos de las aulas. Sprint 8	1	Verificar la cantidad de acciones ejecutadas para modificar el aula
	2	Verificar que se actualiza los datos del aula
	3	Verificar que se envíe un registro de modificar un aula
HU-58 Eliminar datos de las aulas. Sprint 8	1	Verificar la cantidad de acciones que se ejecutan para eliminar los datos de un aula
	2	Verificar que se ejecute la acción de eliminar un aula
	3	Verificar que se ha devuelto el estado de eliminar un aula
	4	Verificar que se devuelva el estado por defecto por la función reductora
	5	Verificar que se ha devuelto un arreglo vacío del listado de aulas
	6	Verificar que se ha obtenido el listado de aulas
	7	Verificar que se ha ingresado una nueva aula
	8	Verificar que se ha eliminado un aula
	9	Verificar que se ha actualizado los datos de un aula
HU-59 Ingresar recursos de un aula. Sprint 9	1	Verificar que se ha enviado un formulario de datos válido
	2	Verificar que se ejecuten la cantidad de acciones necesarias para el ingreso del recurso de un aula
	3	Verificar que se ha ingresado un nuevo recurso de aula
HU-60 Listar los recursos de un aula. Sprint 9	1	Verificar que se ha obtenido un listado de recursos de un aula
	2	Verificar que se ha ejecutado la acción de obtener el listado de recursos de un aula
	3	Verificar que se han ejecutado la cantidad de acciones necesarias para el listado de recursos de un aula
HU-61 Modificar los recursos de un aula. Sprint 9	1	Verificar que se ejecuta la acción de modificar el recurso de un aula
	2	Verificar que se ejecuten la cantidad de acciones necesarias para modificar los recursos de un aula
HU-62 Eliminar los recursos de un aula. Sprint 9	1	Verificar que se ejecuten la cantidad de acciones necesarias para eliminar los recursos de un aula
	2	Verificar que se elimine el recurso de un aula
	3	Verificar que se ha obtenido el estado de eliminar el recurso de un aula
	4	Verificar que se ha obtenido el estado por defecto de la función reductora
	5	Verificar que se ha obtenido un arreglo vacío de los recursos de un aula
	6	Verificar que se ha obtenido el listado de recurso de un aula
	7	Verificar que se ha registrado un nuevo recurso
	8	Verificar que se ha eliminado un recurso de aula
	9	Verificar que se ha actualizado un recurso de aula
HU-63 Ingresar datos de un silabo académico. Sprint 9 HU-64 Modificar datos de un silabo académico. Sprint 9	1	Verificar que se ha obtenido los recursos de un silabo por asignatura
	2	Verificar que se ha ingresado el número de créditos de un silabo
	3	Verificar que se ha ingresado la descripción de un silabo
	4	Verificar que se ha ingresado los requisitos de un silabo
	5	Verificar que se ha ingresado las referencias a libros de un silabo
	6	Verificar que se ha ingresado las referencias de sitios web de un silabo
	7	Verificar que se ha ingresado los objetivos de un silabo
	8	Verificar que se ha ingresado el contenido de un silabo
	9	Verificar que se ha ingresado el número de créditos de un silabo
	10	Verificar la cantidad de acciones que se ejecutan al ingresar el contenido de un silabo
	11	Verificar que se ha ingresado el horario de clases prácticas de un silabo
	12	Verificar que se ha ingresado la contribución de la asignatura
	13	Verificar que se ha ingresado el número de créditos de un silabo
	14	Verificar que se ha ingresado los resultados a obtener de la asignatura

Historias de usuario	#	Casos de prueba
	15	Verificar que se ha ingresado el tipo de evaluación de un silabo
	16	Verificar que se ha ingresado el responsable de ingreso de un silabo
	17	Verificar que se ha ingresado la autorización de un silabo
	18	Verificar que se ha ingresado el informe de un silabo
	19	Verificar que se ha eliminado el informe de un silabo
	20	Verificar que se ha obtenido el estado por defecto de la función reductora de registro de los datos de un silabo
	21	Verificar que se ha seleccionado el período
	22	Verificar que se ha seleccionado la sección
	23	Verificar que se ha seleccionado el paralelo
	24	Verificar que se ha seleccionado la asignatura para el ingreso del silabo
	25	Verificar se ha obtenido el valor por defecto de la obtención de los datos de un silabo
HU-65 Ingresar evaluación al docente o instructor. Sprint 10	1	Verificar que se ha enviado un formulario valido
	2	Verificar que se ejecutan la cantidad de acciones necesarias para el ingreso de evaluaciones
	3	Verificar que se ejecuta la acción de ingresar una evaluación
	4	Verificar que se ha devuelto el estado de registro de una evaluación
	5	Verificar que se envíe un solo formulario de evaluación
	6	Verificar que el formulario que se envía no contenga datos nulos
	7	Verificar que se ejecute solo una acción para el registro de una evaluación
	8	Verificar que se ha ejecutado solo una acción de estado del registro de una evaluación
HU-66 Modificar evaluación al docente o instructor. Sprint 10	1	Verificar que se ejecuten la cantidad de acciones necesarias para modificar los datos de una evaluación
	2	Verificar que se actualiza la evaluación al docente o instructor
	3	Verificar que se envíe un solo formulario de actualización
HU-67 Eliminar evaluación al docente o instructor. Sprint 10	1	Verificar que se ejecuten la cantidad de acciones necesarias para eliminar la evaluación
	2	Verificar que se ha eliminado una evaluación
	3	Verificar que se ha obtenido el estado de eliminar una evaluación
HU-68 Listar evaluaciones docentes o instructores. Sprint 10	1	Verificar la cantidad de evaluaciones obtenidas
	2	Verificar que se ejecuta la acción de obtener el listado de evaluaciones
	3	Verificar que se ejecuten la cantidad de acciones necesarias para la obtención de evaluaciones
	4	Verificar que se ha listado las evaluaciones de los estudiantes
	5	Verificar que se han ejecutado la cantidad de acciones necesarias para la obtención de evaluaciones de los estudiantes
	6	Verificar que se ha obtenido el array por defecto de la función reductora de listar evaluaciones de los docentes
	7	Verificar que se ha obtenido un array vacío de evaluaciones docente
	8	Verificar que se ha obtenido el listado de evaluaciones docente
	9	Verificar que se ha ingresado una nueva evaluación docente
	10	Verificar que se ha eliminado una evaluación docente
	11	Verificar que se ha modificado una evaluación docente
	12	Verificar que se ha obtenido el array por defecto del listado de evaluaciones estudiante
	13	Verificar que se ha obtenido un array vacío de evaluaciones estudiante
	14	Verificar que se ha obtenido el listado de evaluaciones estudiante
	15	Verificar que se ha enviado los datos de evaluación del docente
HU-69 Ingresar preguntas por evaluación docente o instructor. Sprint 10	1	Verificar que se ha enviado datos correctos
	2	Verificar que se hayan ejecutado las acciones necesarias para ingresar una nueva pregunta
	3	Verificar que se haya almacenado una nueva pregunta
HU-70 Modificar preguntas por	1	Verificar que se ejecuten la cantidad de acciones necesarias para

Historias de usuario	#	Casos de prueba
evaluación docente o instructor. Sprint 11		modificar la pregunta de una evaluación
	2	Verificar que se modifique la pregunta correspondiente a una evaluación
HU-71 Eliminar preguntas por evaluación docente o instructor. Sprint 11	1	Verificar que se ejecuten la cantidad de acciones necesarias para eliminar una pregunta de evaluación
	2	Verificar que se ha eliminado la pregunta de una evaluación
	3	Verificar que se ha devuelto el estado de eliminar la pregunta de una evaluación
HU-72 Listar preguntas por evaluación docente o instructor. Sprint 11	1	Verificar la cantidad de preguntas obtenidas por evaluación
	2	Verificar que se ha obtenido un listado de preguntas
	3	Verificar que se ejecutan la cantidad de acciones necesarias para listar preguntas
	4	Verificar que se devuelve el estado de listar preguntas por evaluación
	5	Verificar que se devuelve el estado por defecto de la función reductora de listar las preguntas por evaluación
	6	Verificar que se devuelve un array vacío de preguntas
	7	Verificar que se devuelve un listado de preguntas
	8	Verificar que se ha añadido una nueva pregunta
	9	Verificar que se ha eliminado una pregunta
	10	Verificar que se ha actualizado una pregunta
HU-73 Listar resultados de evaluación. Sprint 11	1	Verificar que se obtiene la cantidad de preguntas por evaluación
	2	Verificar la cantidad de acciones que se ejecutan al listar los resultados de evaluación
	3	Verificar que se devuelve el estado de listar los resultados de una evaluación
HU-74 Ingresar datos de una planificación. Sprint 11 HU-75 Modificar datos de una planificación. Sprint 11	1	Verificar que se ha obtenido la planificación de una asignatura
	2	Verificar que se envíe la información general de la asignatura
	3	Verificar que se envíe los objetivos de la planificación
	4	Verificar que se envíe el programa resumido de la planificación
	5	Verificar que se envíe el programa detallado de la planificación
	6	Verificar que se envíe el texto guía de la planificación
	7	Verificar que se elimine el texto guía de la planificación
	8	Verificar que se envíe las referencias de libros de la planificación
	9	Verificar que se envíe las referencias de sitios web de la planificación
	10	Verificar que se envíe la autorización de la planificación
	11	Verificar que se envíe el informe de planificación
	12	Verificar que se devuelve el estado por defecto de la función reductora
	13	Verificar que se ha obtenido un array vacío de datos de la planificación
	14	Verificar que se ha obtenido los datos de una planificación
HU-76 Ingresar materiales de estudio de los docentes o instructores. Sprint 12	1	Verificar que se ha obtenido un período
	2	Verificar que se ha obtenido el período actual
	3	Verificar que se ejecuten las acciones necesarias para obtener el período actual
	4	Verificar la cantidad de períodos obtenidos
	5	Verificar que se ha obtenido el listado de períodos
	6	Verificar que se ejecuten las acciones necesarias para obtener el listado de períodos
	7	Verificar la cantidad de secciones obtenidas
	8	Verificar que se ha obtenido el listado de secciones
	9	Verificar que se ejecuten las acciones necesarias para obtener el listado de secciones
	10	Verificar que se ha ejecutado la acción de obtener las secciones
	11	Verificar la cantidad de paralelos obtenidos
	12	Verificar que se ha ejecutado la acción de obtener los paralelos
	13	Verificar que se ejecuten las acciones necesarias para obtener el listado de paralelos
	14	Verificar que se ha obtenido el listado de paralelos

Historias de usuario	#	Casos de prueba
	15	Verificar el listado de asignaturas por docente
	16	Verificar el listado de asignaturas de la institución
	17	Verificar la cantidad de acciones que se ejecuten para obtener el listado de asignaturas
	18	Verificar el listado de las asignaturas de docentes que cumplen con los requisitos necesarios para el pago
	19	Verificar el listado de las asignaturas de instructores que cumplen con los requisitos necesarios para el pago
	20	Verificar el listado de asignaturas asignadas a los estudiantes
	21	Verificar que se envíe datos correctos del material de estudio
	22	Verificar que se ejecuten las acciones necesarias para el registro de material de estudio
	23	Verificar que se almacena el material de estudio correspondiente
	24	Verificar que se ha obtenido el listado de material de estudio de la asignatura
	25	Verificar que se ejecuten las acciones necesarias para obtener el listado de material de estudio de la asignatura
	26	Verificar que se ejecuten las acciones necesarias para eliminar el material de estudio de la asignatura
	27	Verificar que se ejecute la acción de eliminar un material de estudio
	28	Verificar que se ha eliminado un material de estudio
	29	Verificar que se envíen datos correctos para modificar un material de estudio
	30	Verificar que se ejecuten las acciones necesarias para modificar un material de estudio
	31	Verificar que se ha actualizado un material de estudio
	32	Verificar que se obtiene el array por defecto de la función reductora de obtener el listado de períodos
	33	Verificar que se ha obtenido un arreglo vacío de períodos
	34	Verificar que se ha obtenido un listado de períodos
	35	Verificar que se obtiene el array por defecto de la función reductora de obtener el período actual
	36	Verificar que se ha obtenido un período
	37	Verificar que se ha obtenido el período actual
	38	Verificar que se obtiene el array por defecto de la función reductora de obtener el listado de secciones
	39	Verificar que se ha obtenido un arreglo vacío de secciones
	40	Verificar que se ha obtenido un listado de secciones
	41	Verificar que se obtiene el array por defecto de la función reductora de obtener el listado de paralelos
	42	Verificar que se ha obtenido un arreglo vacío de paralelos
	43	Verificar que se ha obtenido un listado de paralelos
	44	Verificar que se obtiene el array por defecto de la función reductora de obtener el listado de asignaturas por docente
	45	Verificar que se ha obtenido un arreglo vacío de asignaturas
	46	Verificar que se ha obtenido un listado de asignaturas
	47	Verificar que se ha eliminado una asignatura
	48	Verificar que se obtiene el array por defecto de la función reductora de obtener el listado de asignaturas por estudiante
	49	Verificar que se ha obtenido un arreglo vacío de asignaturas por estudiante
	50	Verificar que se ha obtenido un listado de asignaturas por estudiante
	51	Verificar que se obtiene el array por defecto de la función reductora de material de estudio
	52	Verificar que se ha obtenido un arreglo vacío de materiales de estudio
	53	Verificar que se ha obtenido un listado de materiales de estudio
	54	Verificar que se ha ejecutado la función reductora de agregar un nuevo material de estudio
	55	Verificar que se ha ejecutado la función reductora de eliminar un material de estudio
	56	Verificar que se ha ejecutado la función reductora de modificar un

Historias de usuario	#	Casos de prueba
		material de estudio
	57	Verificar que se ha ejecutado la función reductora de modificar el estado de un material de estudio
	58	Verificar que se ha ejecutado la función reductora de obtener el listado de etiquetas de materiales de estudio.
HU-77 Aprobar materiales de estudio de los docentes o instructores. Sprint 12	1	Verificar que se ha obtenido un listado de materiales de estudio
	2	Verificar que se ejecuta la acción de obtener materiales de estudio
	3	Verificar que se ejecuten la cantidad de acciones necesarias para la aprobación de los materiales de estudio
HU-78 Eliminar materiales de estudio de los docentes o instructores. Sprint 12	1	Verificar que se ejecuten la cantidad de acciones necesarias para eliminar los materiales de estudio
	2	Verificar que se ejecute la acción de eliminar un material de estudio
	3	Verificar que ha eliminado un material de estudio
	4	Verificar que se obtiene el array por defecto de la función reductora de eliminar un material de estudio
	5	Verificar que se ha obtenido un arreglo vacío de materiales de estudio
	6	Verificar que se ha obtenido un listado de materiales de estudio
	7	Verificar que se ha ingresado un nuevo material de estudio
	8	Verificar que se ha eliminado un nuevo material de estudio
	9	Verificar que se ha actualizado un nuevo material de estudio
HU-79 Ingresar observación en un material de estudio de docente o instructor. Sprint 12	1	Verificar que se envía un formulario valido de observaciones
	2	Verificar que se ejecuten la cantidad de acciones necesarias para el ingreso de una observación
	3	Verificar que se ha ingresado la observación a un material de estudio
HU-80 Ingreso de pago de docentes o instructores o instructores. Sprint 12	1	Verificar que se ejecuten la cantidad de acciones necesarias para el ingreso de pago a docentes
	2	Verificar que se envía el pago de docentes
	3	Verificar que se envía un formulario válido del pago a docentes
HU-81 Listado de recursos para los pagos de docentes o instructores. Sprint 12	1	Verificar que se ejecute la acción de obtener el listado de recursos para el pago a docentes
	2	Verificar que se ha devuelto el estado de la petición
	3	Verificar que se ejecuten la cantidad de acciones necesarias para la obtención de recursos de pago
	4	Verificar que se ha obtenido los detalles de un pago a docente o instructor
	5	Verificar que se ha eliminado un recurso de pago
HU-82 Eliminar los recursos para el pago de docentes o instructores. Sprint 12	1	Verificar que se ejecutan la cantidad de acciones necesarias para eliminar un recurso de pago
	2	Verificar que se ejecuta la acción de obtener el listado de pagos
	3	Verificar que se ha obtenido el estado de eliminar el recurso de un pago
	4	Verificar que se ha eliminado el recurso de pago
	5	Verificar que se ejecuta la acción de obtener los detalles de un pago a docente o instructor
	6	Verificar que se ha obtenido el estado de la petición de detalles de un pago
	7	Verificar que se han ejecutado la cantidad de acciones necesarias para obtener los detalles de un pago
	8	Verificar que se ejecute que se ha obtenido los detalles de pago por departamento
	9	Verificar que se ha devuelto el estado de obtener los detalles de pago por departamento
	10	Verificar que se ha ejecutado la cantidad de acciones necesarias para la obtención de detalles de pago
	11	Verificar que se ejecute la acción de actualizar los datos de pago a docentes
	12	Verificar que se ha obtenido la acción de actualizar los datos de pago a docentes
	13	Verificar que se han ejecutado la cantidad de acciones necesarias para actualizar los datos de pago

Historias de usuario	#	Casos de prueba
	14	Verificar que se ha obtenido el valor por defecto de la función reductora de obtener los pagos
	15	Verificar que se ha obtenido un arreglo vacío de pagos
	16	Verificar que se ha obtenido un listado de pagos
	17	Verificar que se ha actualizado los datos de pago
HU-83 Ingreso de solicitud de tipo general o justificación. Sprint 13	1	Verificar que se ejecuten la cantidad de acciones necesarias para el ingreso de una solicitud
	2	Verificar que se ejecute la acción de envío de una solicitud
	3	Verificar que ha obtenido el estado de ingreso de una solicitud
	4	Verificar que se ha enviado un formulario valido
HU-84 Eliminar solicitud de tipo general o justificación. Sprint 13	1	Verificar que se ejecuten la cantidad de acciones necesarias para eliminar una solicitud
	2	Verificar el listado de solicitudes registradas
	3	Verificar que se ha obtenido el estado de eliminar una solicitud
	4	Verificar que se ha eliminado una solicitud
HU-85 Modificar solicitud de tipo general o justificación. Sprint 13	1	Verificar que se ejecuten la cantidad de acciones necesarias para modificar una solicitud
	2	Verificar que se ha actualizado los datos de una solicitud
	3	Verificar que se ha obtenido el estado de modificar los datos de una solicitud
	4	Verificar que se ha enviado un formulario de datos válidos
HU-86 Mostrar detalles de solicitud de tipo general o justificación. Sprint 13	1	Verificar listado de detalles de una solicitud
	2	Verificar que se ha ejecutado la acción de obtener los detalles de una solicitud
HU-87 Listar solicitudes de tipo general o justificación. Sprint 13 HU-88 Aprobar solicitud de tipo general o justificación. Sprint 13	1	Verificar que se ejecute la acción de listar las solicitudes
	2	Verificar que se ha obtenido el estado de listar las solicitudes
	3	Verificar que se han ejecutado la cantidad de acciones necesarias para listar las solicitudes
	4	Verificar que se ha actualizado el estado de una solicitud
	5	Verificar que se ha devuelto el estado de actualizar el estado de una solicitud
	6	Verificar que se han ejecutado la cantidad de acciones necesarias para actualizar el estado de una solicitud
	7	Verifica listado de justificaciones de los estudiantes
	8	Verificar que se ha obtenido el estado de listar justificaciones
	9	Verificar que se han ejecutado la cantidad de acciones necesarias para listar las justificaciones de estudiantes
	10	Verificar el listado de justificaciones por departamento
	11	Verificar que se ha obtenido el estado de listar justificaciones por departamento
	12	Verificar que se ejecuten la cantidad de acciones necesarias para obtener el listado de justificaciones por departamento
	13	Verificar que se actualice los datos de una solicitud
	14	Verificar que se ha obtenido el estado de ejecutar la actualización de una solicitud
	15	Verificar que se ejecuten la cantidad de acciones necesarias para actualizar los datos de una solicitud
	16	Verificar que se ha obtenido el valor por defecto de listado de solicitudes devuelto por la función reductora
	17	Verificar que se ha obtenido un arreglo vacío de solicitudes devuelto por la función reductora
	18	Verificar que se ha obtenido un listado de solicitudes devuelto por la función reductora
	19	Verificar que se ha obtenido el valor por defecto de listado de autorizaciones de una solicitud devuelto por la función reductora
	20	Verificar que se ha obtenido un arreglo vacío de autorizaciones de una solicitud devuelto por la función reductora
	21	Verificar que se ha obtenido un listado de autorizaciones de una solicitud devuelto por la función reductora
HU-89 Ingreso pago de mensualidad de los estudiantes. Sprint 13	1	Verificar que se ha enviado datos correctos del pago de mensualidad
	2	Verificar que se han ejecutado la cantidad de acciones necesarias para el ingreso del pago de mensualidad

Historias de usuario	#	Casos de prueba
	3	Verificar que se ha almacenado el pago de mensualidad
	4	Verificar el listado de pagos de mensualidad
HU-90 Aprobar pago de mensualidad de los estudiantes. Sprint 14	1	Verificar el listado de pagos de mensualidad
	2	Verificar el listado de pagos de mensualidad por estudiante
	3	Verificar que se han ejecutado la cantidad de acciones para aprobar un pago de mensualidad
HU-91 Eliminar pago de mensualidad de los estudiantes. Sprint 14	1	Verificar que se han ejecutado la cantidad de acciones necesarias para eliminar un pago de mensualidad
	2	Verificar que se ha eliminado un pago de mensualidad
	3	Verificar el listado de archivos de pago de mensualidad
	4	Verificar que se obtenido el valor por defecto devuelto por la función reductora
	5	Verificar que se ha obtenido un arreglo vacío de los pagos de mensualidad
	6	Verificar el listado de pagos de mensualidad
	7	Verificar que se ha ingresado un nuevo archivo de pago de mensualidad
	8	Verificar que se ha eliminado un archivo de pago de mensualidad
	9	Verificar que se ha modificado un archivo de pago de mensualidad
HU-92 Ingreso de observación al pago de mensualidad de un estudiante. Sprint 14	1	Verificar que se ha enviado un formulario de datos válidos
	2	Verificar que no se han enviado datos nulos en la observación
	3	Verificar que se han ejecutado la cantidad de acciones necesarias para el ingreso de una observación
	4	Verificar que se ha actualizado los datos de un archivo
HU-94 Autenticar administradores desde la aplicación móvil. Sprint 14	1	Verificar que se ha mostrado la interfaz móvil
	2	Verificar que se ha mostrado el calendario de fechas reservadas
	3	Verificar el listado de usuarios que han reservado exámenes psicosenométricos
	4	Verificar que se ha mostrado los datos del usuario
HU-95 Registrar solicitudes desde la aplicación móvil. Sprint 14	1	Verificar que se ha mostrado la interfaz de inicio
	2	Verificar que se ha enviado una solicitud valida
	3	Verificar que se haya guardado la solicitud
HU-96 Permitir el acceso a los usuarios del sistema a su respectiva cuenta. Sprint 15	1	Verificar que se haya mostrado la interfaz de inicio
	2	Verificar que los campos ingresados son correctos
	3	Verificar que los datos enviados sean validos
	4	Verificar el acceso a la interfaz de usuario.
HU-97 Permitir que los usuarios(clientes) puedan registrar un horario para el examen psicosenométrico. Sprint 15	1	Verificar que se ha mostrado la interfaz de inicio
	2	Verificar que se muestre el formulario de ingreso de solicitud
	3	Verificar que se haya mostrado el campo de calendario
	4	Verificar que se hayan enviado datos correctos
HU-98 Permitir que los administradores puedan ver los horarios de examen psicosenométrico. Sprint 15	1	Verificar que se haya mostrado la interfaz de inicio
	2	Verificar que se haya mostrado la interfaz de usuario
	3	Verificar que se muestren los horarios de exámenes
	4	Verificar que se puedan mostrar los detalles del examen.
14 Sprints involucrados	Total de casos de prueba: 658	

Realizado por: Aldas W.; Pilataxi C., 2020

Estrategia

Se aplicará la estrategia de caja-blanca, mediante el ingreso de datos válidos y no válidos para obtener una determinada salida, lo cual debe ser comparado con el resultado esperado, de esta manera se puede verificar si la funcionalidad es adecuada o no.

Por lo general las pruebas unitarias son desarrolladas después de haber terminado la funcionalidad, tal es el caso de este trabajo.

A continuación, en la **Tabla 12-3**, se establece la estrategia planteada para llevar a cabo el

desarrollo de pruebas unitarias al sistema:

Tabla 12-3: Estrategia de prueba implementada

Pruebas funcionales de caja blanca	
Objetivo	Aplicar pruebas unitarias a los requerimientos definidos por el cliente, para poder determinar el correcto funcionamiento en cuanto al ingreso y salida de datos
	Ejecutar cada caso de prueba con los respectivos datos válidos y no válidos para verificar lo siguiente: <ul style="list-style-type: none">- Se obtendrá los resultados esperados cuando se ingrese datos válidos- Se muestra la advertencia de que se ha producido un error después de ejecutarse la prueba o cuando no se puede- Las reglas de negocio se han aplicado de forma correcta
Criterio de completitud	<ul style="list-style-type: none">- Se ha ejecutado todos los casos de prueba desarrollados- Se ha reportado todos los defectos presentados

Realizado por: Aldas W.; Pilataxi C., 2020

Categorización de la configuración

Bajo esta premisa se da a conocer las condiciones por las cuales se debe suspender la ejecución de las pruebas unitarias, así como también de reanudar con las tareas previstas. Las condiciones para la suspensión de ejecución son las siguientes:

- El resultado obtenido por la prueba no cumpla con las especificaciones establecidas previamente en la funcionalidad del sistema
- Se presente inestabilidad en el entorno de pruebas y/o retorne resultados poco confiables
- Errores que impidan la ejecución de pruebas

El criterio de reanudación se lo realizará únicamente después de haber corregido los errores presentados y finalmente el criterio de culminación se establecerá cuando se haya superado al menos la mitad del total de pruebas.

Tangibles

Al finalizar todo el proceso de pruebas establecidas al sistema de gestión de información de los procesos administrativos de la escuela de conducción profesional CONDUESPOCH E.P en su primera versión, se presentará un informe de resultados, en el que se detallen las pruebas planificadas, las pruebas ejecutadas, las pruebas exitosas, las pruebas fallidas con su respectivo listado de defectos, la cobertura de código de las pruebas unitarias y las conclusiones con las gráficas de los resultados finales.

Recursos

La ejecución y seguimiento de pruebas automatizadas de tipo caja blanca, son en su mayoría maniobradas por personal técnico del área informática o personal de aseguramiento de calidad de software (SQA por sus siglas en ingles de Software quality assurance), sin embargo, para poner todo en contexto, se puede decir que para poder ejecutar las pruebas unitarias desarrolladas en el presente proyecto, se deberá contar con el entorno de desarrollo Visual Studio Code, para poder monitorear las métricas establecidas en esta primera versión del sistema. Para asegurarse del normal comportamiento de todo el proceso se recomienda el uso del sistema operativo Windows 10, navegador Google Chrome para observar la ejecución de la cobertura de código de cada una de las funcionalidades.

El sistema debe estar debidamente alojada en un servidor de manera local que soporte el lenguaje PHP, HTML, Postgresql y demás utilitarios referentes a tecnologías del desarrollo web. Finalmente se aclara que todas las herramientas tecnológicas utilizadas son libres de licencias por lo que no hay inconvenientes por la obtención de ninguna clase de licencias.

Calendario

Para la implementación de los casos de prueba es necesario poseer una planificación en el cual se detalle a breves rasgos las fechas de inicio y finalización, ya que a través de esto podemos controlar las actividades presentadas para la conclusión del proyecto, además sirve como base para la solicitud de tiempo adicional a las personas encargadas de dirigir todo el trayecto de desarrollo, es así que para este trabajo se ha definido de manera breve las posibles fechas a tener en cuenta, lo cual se detalla en la **Tabla 13-3**.

Tabla 13-3: Calendario para la automatización de los casos de prueba

Hito	Funcionalidades a probar	Fecha de inicio planificada	Fecha fin de planificación planificada
Diseño y ejecución de las pruebas de Sprint 2	HU-01, HU-02, HU-03, HU-04, HU-05, HU-06, HU-07	10/07/20	10/07/20
Diseño y ejecución de las pruebas de Sprint 3	HU-08, HU-09, HU-10, HU-11, HU-12, HU-13	31/07/20	31/07/20
Diseño y ejecución de las pruebas de Sprint 4	HU-14, HU-15, HU-16, HU-17, HU-18, HU-19, HU-20, HU-21	21/08/20	21/08/20
Diseño y ejecución de las pruebas de Sprint 5	HU-22, HU-23, HU-24, HU-25, HU-26, HU-27, HU-28, HU-29, HU-30, HU-31	11/09/20	11/09/20
Diseño y ejecución de las pruebas de Sprint 6	HU-32, HU-33, HU-34, HU-35, HU-36, HU-37, HU-38, HU-39, HU-40	02/10/20	02/10/20

Hito	Funcionalidades a probar	Fecha de inicio planificada	Fecha fin de planificación planificada
Diseño y ejecución de las pruebas de Sprint 7	HU-41, HU-42, HU-43, HU-44, HU-45, HU-46, HU-47, HU-48	23/10/20	23/10/20
Diseño y ejecución de las pruebas de Sprint 8	HU-49, HU-50, HU-51, HU-52, HU-53, HU-54, HU-55, HU-56, HU-57, HU-58	13/11/20	13/11/20
Diseño y ejecución de las pruebas de Sprint 9	HU-59, HU-60, HU-61, HU-62, HU-63, HU-64	04/12/20	04/12/20
Diseño y ejecución de las pruebas de Sprint 10	HU-65, HU-66, HU-67, HU-68, HU-69	25/12/20	25/12/20
Diseño y ejecución de las pruebas de Sprint 11	HU-70, HU-71, HU-72, HU-73, HU-74, HU-75	15/01/21	15/01/21
Diseño y ejecución de las pruebas de Sprint 12	HU-76, HU-77, HU-78, HU-79, HU-80, HU-81, HU-82	05/02/21	05/02/21
Diseño y ejecución de las pruebas de Sprint 13	HU-83, HU-84, HU-85, HU-86, HU-87, HU-88, HU-89	26/02/21	26/02/21
Diseño y ejecución de las pruebas de Sprint 14	HU-90, HU-91, HU-92, HU-94, HU-95	19/03/21	19/03/21
Diseño y ejecución de las pruebas de Sprint 15	HU-96, HU-97, HU-98	01/04/21	01/04/21
Entrega de informe de resultados de prueba		03/05/21	28/05/21

Realizado por: Aldas W.; Pilataxi C., 2020

Manejo de riesgos

Los riesgos que se deben tomar en cuenta en esta sección se diferencian de los riesgos de desarrollo del sistema, debido a que los que aquí se definen están enfocados únicamente a solventar los inconvenientes que se presenten únicamente en el desarrollo de los casos de prueba, además las medidas que se aplican son distintas y no requieren de mucho análisis.

En la **Tabla 14-3**, se ha establecido los posibles riesgos con sus respectivas medidas a tomar en cuenta para su resolución.

Tabla 14-3: Tabla de mitigación de riesgos de los casos de prueba

N	Riesgo	Estrategia de mitigación
1	Los miembros involucrados en el desarrollo de pruebas no conocen cómo está constituido el sistema	Implementar entrenamiento a las personas encargadas de cada tarea para poder dar a entender sobre las expectativas del negocio.
2	No se cuenta con la documentación necesaria	
3	La realización de los casos de prueba requiere demasiado tiempo, lo cual sobrepasa al de desarrollo de las funcionalidades del sistema	Re-planificar las tareas a desarrollar tomando en cuenta que se lleve a cabo las pruebas de alto riesgo e importancia.

Realizado por: Aldas W.; Pilataxi C., 2020

Responsables

A continuación, en la **Tabla 15-3**, se especifica sobre los roles necesarios para el desarrollo de este trabajo, en el cual además se define las responsabilidades a los que están sujetos cada uno

de estos, se debe tener en cuenta que se puede agregar más de los que se detallan aquí, sin embargo, dado el alcance planteado en este caso, no será necesario de más miembros.

Tabla 15-3: Responsable para llevar a cabo las tareas de testing

Recursos Humanos		
Rol	Recursos mínimos recomendados	Responsabilidades específicas
Analista de pruebas	2	<ul style="list-style-type: none"> - Definir los casos de prueba necesarios - Documentar detalladamente los casos de prueba - Definir los resultados esperados para los casos de prueba
Tester	2	<ul style="list-style-type: none"> - Escribir el código necesario para cada una de las pruebas - Ejecutar las pruebas realizadas - Registrar resultados obtenidos tras la ejecución - Analizar y reparar fallas en caso de ser necesario - Reporta los defectos encontrados
Diseñador de pruebas	2	<ul style="list-style-type: none"> - Establece el enfoque de cada una de las pruebas planteadas - Revisión constante de la eficiencia de la técnica de pruebas.

Realizado por: Aldas W.; Pilataxi C., 2020

3.3.3. Fase de desarrollo

En esta fase se implementan cada una de las tareas planteadas previamente en la planificación, por lo que se realizan reuniones entre los desarrolladores del sistema y el cliente, para una retroalimentación del estado del producto. (Trigas, 2019)

3.3.3.1. Estándar de codificación y diseño de base de datos

El estándar de codificación consiste en definir la forma en cómo se va a escribir el código del sistema, facilitando la lectura y comprensión para cualquier persona que se involucre con el desarrollo.

Para seleccionar la manera más adecuada de escribir código, se ha analizado las nomenclaturas definidas por CamelCase, el cual establece dos tipos principales de estilo de escritura los cuales son, el lowerCamelCase y el UpperCamelCase, ambos resultan importantes al momento de desarrollar cualquier sistema ya que se los puede combinar de acuerdo con las necesidades presentadas. A continuación, en la **Tabla 16-3**, se detalla la manera en cómo se ha llevado a cabo la escritura del código de este proyecto.

Para el estándar de diseño de la base de datos se debe utilizar un prefijo para cada tabla, en este caso se usará el prefijo “t” que corresponde a la abreviación de “tabla”, es decir cada tabla tendrá este prefijo antes del nombre que le corresponda. Para las filas se usará nombres en minúsculas y sin espacios.

Tabla 16-3: Estándar de codificación y diseño base de datos

Estándar de codificación y diseño de BD		
Elemento	Estándar	Ejemplo
Clases y Funciones	uperCase	Users
Métodos	lowerCase	postArchivo()
Variables	lowerCase	param
Nombres Tablas	lowerCase	tusuario
Filas Tablas	lowerCase	codusuario

Realizado por: Aldas W.; Pilataxi C., 2020

3.3.3.2. *Arquitectura del sistema web y móvil*

El diseño de la arquitectura de un sistema es una de las tareas más importantes debido a que en ella se establecen los elementos necesarios para el funcionamiento del producto final. La arquitectura de un sistema muestra a breves rasgos el entorno de alojamiento de cada uno de los componentes desarrollados.

Para llevar a cabo esta tarea, existen los denominados patrones de diseño, los cuales dan las pautas a seguir cuando no se está familiarizado con la construcción de un sistema. A pesar de la existencia de muchos de estos patrones, el que mayormente se utiliza en el desarrollo de aplicaciones web es el patrón Modelo Vista Controlador (MVC), debido a que es fácil de entender y manejar.

Partiendo de lo anterior, en la **Figura 2-3**, se ha establecido la arquitectura correspondiente a este trabajo el cual está formado a partir del modelo MVC, tanto para la aplicación web como también para la aplicación móvil.

Para empezar, el usuario accede a través de una computadora en el caso de la aplicación web y a través de un dispositivo móvil en el caso de que se trate de la aplicación desarrollada para este entorno. Cada vez que el usuario interactúa con una de las dos aplicaciones mediante una de las Api Rest, estas se comunican con el controlador del sistema, el cual es una serie de scripts que manejan la lógica de negocio, que al ser invocados analizan si lo que se requiere es el llamado a una vista gráfica o ejecutar una consulta a la base de datos mediante el modelo al servidor.

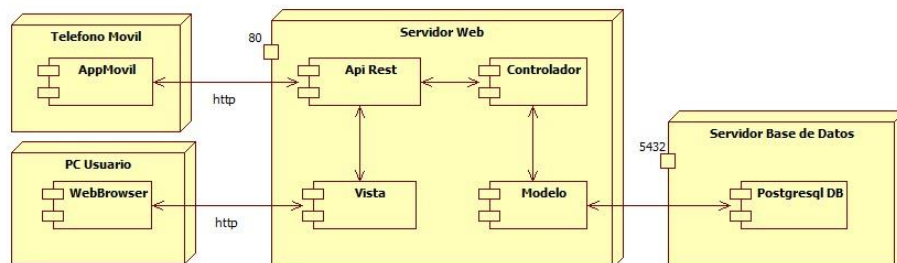


Figura 2-3: Arquitectura del sistema web y móvil

Realizado por: Aldas W.; Pilataxi C., 2020

React JS es un framework principalmente de representación de datos, por lo que en el patrón MVC este viene a ser la parte de la vista. En cuanto el modelo y controlador, estas están desarrolladas en el lenguaje PHP, debido a su fácil manejo de las APIs, programación orientada a objetos y conexión a la base de datos.

- **Modelo:** Representa las entidades de la base de datos con la cual trabaja el sistema, de esta manera resulta más fácil realizar cualquier acción de lectura o escritura sin perjudicar los datos originales, ya que permite manejar a las tablas como objetos.
- **Vista:** Es la parte que permite interactuar con el usuario mediante interfaces gráficas, permitiendo que el uso del sistema sea fácil, ya que permite representar los datos deseados de una manera ordenada.
- **Controlador:** Su principal uso es el de capturar las acciones del usuario, procesarlas y enviar a donde sea requerido, esto puede ser a la base de datos o a la vista del sistema. En pocas palabras sirve de intermediario entre el sistema y el usuario.

Se ha escogido el framework React JS como herramienta de generación de vistas del sistema, por la facilidad de uso e implementación con cualquier lenguaje de backend, además de que la base de desarrollo es mediante la atomicidad de componentes, lo cual permite escribir código una sola vez y utilizarlo en donde se desee dentro del sistema, esto permite que su escalabilidad sea mucho mayor en comparación con otros frameworks. En cuanto al lenguaje de backend, se ha seleccionado PHP, debido a la facilidad de aprendizaje, manejo de programación orientado a objetos y además por la gran cantidad de servidores de alojamiento disponibles en internet, a diferencia de otros lenguajes que han aparecido en los últimos años como es el caso de NodeJS, el cual requiere un hosting que interprete Javascript, razón por la que es muy difícil encontrar uno.

3.3.3.3. *Diagramas de casos de uso*

Los diagramas de caso de uso permiten comprender de manera fácil y sencilla las funcionalidades de un sistema y la manera en cómo estos están ligados a los actores encargados de llevarlos a cabo.

Una de las herramientas más extendidas en este ámbito es StarUML, que como su nombre lo indica permite generar diagramas UML necesarios para los usos requeridos. Esta herramienta tiene una curva de aprendizaje bastante moderada, llegando a ser muy intuitiva y rápida de ejecutar.

Los diagramas correspondientes a este trabajo, presentan detalles muy generalizados, es decir que no se ahonda a profundidad sobre la exactitud de las tareas a realizar, esto corresponde a

obviar terminologías como ingresar, modificar, listar, eliminar, etc., debido a que, si se toma en cuenta estos detalles, el resultado puede ser muy agotador de observar.

Casos de uso para el usuario administrador

La **Gráfico 1-3**, refleja los módulos que están a cargo de este usuario que a grandes rasgos es un breve resumen de lo que éste puede hacer dentro del sistema, y lo cual se detalla a continuación:

- **Autenticar:** Este es un proceso que debe ser cumplido por todos los usuarios antes de tener el control de las demás funcionalidades, ya que de esta depende el acceso al sistema mismo. Para completar esta etapa lo único que se necesita hacer es ingresar sus credenciales que previamente debieron haber sido asignados.
- **Gestión de Perfil:** Al igual que el módulo de autenticación, este también es lo mismo para todos los usuarios, es decir que el proceso no cambia en lo absoluto. Este proceso está diseñado para modificar los datos personales de quien esté autenticado en el momento, los datos permitidos a modificar son los datos personales, la contraseña y la foto de perfil.
- **Gestión de usuarios:** Este módulo permite ingresar, modificar, eliminar y listar los datos de los usuarios que pertenecen a la institución, esto de manera manual ya que también a petición del cliente hay la posibilidad de migrar datos únicamente de los docentes desde otro sistema, y por último también se permite cambiar de departamentos al personal administrativo, debido a que muchas veces es necesario cambiarlos de funcionalidades.
- **Gestión calendario:** Consiste en ingresar, modificar, eliminar y listar las fechas para llevar a cabo los procesos de inscripción y matrícula de los estudiantes, por ejemplo, para que un nuevo usuario ingrese al sistema necesita registrar su inscripción, el cual únicamente estará habilitada dentro de una fecha inicial a una fecha final y lo mismo ocurre con la matrícula, aunque de manera distinta.
- **Gestión comunicados:** Cada vez que se quiera ejecutar una actividad dentro o fuera de la institución, el personal administrativo está en la obligación de comunicar de los hechos a todos los usuarios del sistema, ya sean docentes, instructores, estudiantes o entre los mismos administrativos, es aquí donde esta funcionalidad cobra valor, ya que permite ingresar, modificar, eliminar y listar comunicados ya sean enviados a los posibles interesados o recibidos.
- **Gestión solicitudes:** En este módulo se gestionan aquellas solicitudes internas desarrolladas por los distintos departamentos administrativos, por lo tanto, las tareas que aquí se realizan son ingresar, modificar, eliminar, listar, agregar archivos y aprobar solicitudes

departamentales ya sean enviadas o recibidas con un determinado propósito.

- **Gestión solicitudes generales:** A diferencia del tipo de solicitudes anteriores, estas pueden ser dirigidas tanto a personas externas como también a los alumnos de la institución, estas solicitudes son utilizadas principalmente para presentar quejas o para algún trámite interno. Las actividades comprendidas en esta sección son las de listar y aprobar las solicitudes que hayan llegado.
- **Gestión proyectos:** La gestión de proyectos consiste en la verificación del listado de solicitudes aprobadas relacionadas con el desarrollo de actividades dirigidas al departamento de DTIC, a continuación, lo que el módulo permite es ingresar, modificar, eliminar, listar proyectos con sus respectivas tareas y asignación de practicantes para las respectivas tareas.
- **Gestión actividades departamentos directivos:** En este módulo el usuario administrador tendrá la opción de gestionar las tareas principales de todos los departamentos administrativos.

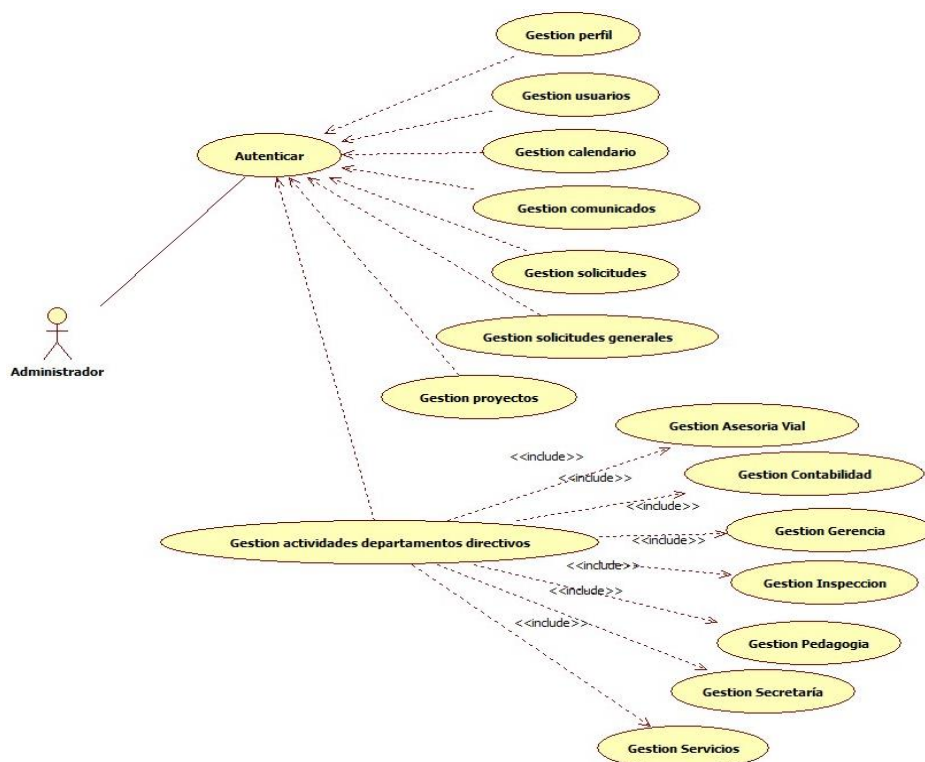


Gráfico 1-3: Casos de uso del usuario administrador

Realizado por: Aldas W.; Pilataxi C., 2020

Casos de uso para el usuario directivo de Secretaría

Los usuarios con rol directivo son aquellas personas que conforman todos los departamentos administrativos, por tal razón, en cada uno de los departamentos se tienen múltiples módulos

dependiendo de las actividades asignadas a realizar.

Muchos de los módulos asignados a este departamento son parecidos a lo del administrador del sistema, razón por la cual únicamente se detallarán aquellos que no han sido presentados todavía.

Los módulos asignados al departamento de Secretaría se detallan en la **Gráfico 2-3**, sin embargo, los módulos presentados muestran de manera resumida todo aquello a realizar, debido a esto se especifican cada uno de estos a continuación:

- **Gestión inscripciones:** El proceso de inscripción dentro de este departamento consiste en verificar el listado de las solicitudes realizadas por parte de los estudiantes, en el que cada uno presenta los datos respectivos, posteriormente el departamento de Secretaría debe aprobar o rechazar, además que también se pueden anexar archivos en formato pdf.
- **Gestión matrículas:** Las matrículas al igual que las inscripciones necesitan de la aprobación de este departamento, para lo cual los estudiantes envían los datos solicitados y al ser aprobados son enviados al siguiente departamento hasta que se complete con el proceso.

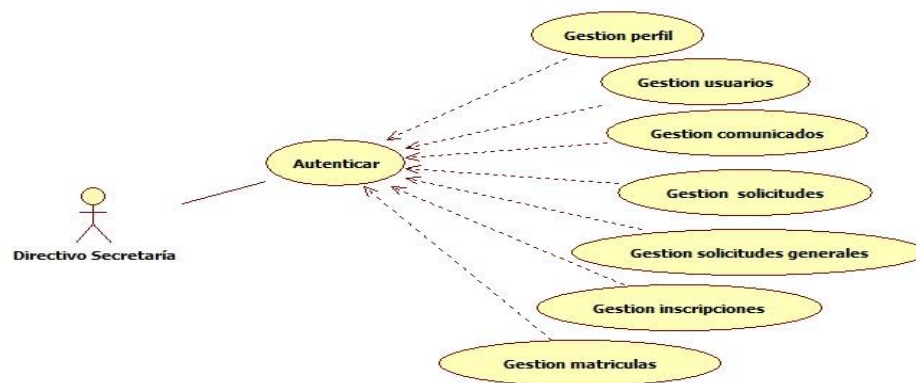


Gráfico 2-3: Casos de uso del directivo de secretaría

Realizado por: Aldas W.; Pilataxi C., 2020

Casos de uso para el usuario directivo de Contabilidad/Tesorería

Tanto el usuario directivo de contabilidad como de tesorería tienen las mismas funcionalidades, por lo tanto, es innecesario gastar esfuerzos en el diseño como también en el desarrollo independiente de estos, así que se define como uno solo. Los módulos a cargo de este departamento se detallan en la **Gráfico 3-3**, que aparte de poseer aquellos que son comunes entre todos los usuarios, también posee algunas que son propias de este departamento, mismos que se detallan a continuación:

- **Gestión de solicitudes generales:** Para este departamento, este módulo es distinto a los demás, ya que a diferencia de los otros departamentos en los que únicamente deben aprobar este tipo de solicitudes, aquí es donde se realiza las actividades de ingresar, modificar, eliminar y listar cada uno de estos, para lo cual existen de dos tipos, mismos que son, de tipo general tanto para los estudiantes como también para personas particulares, y por otro lado están los de tipo justificación únicamente para los estudiantes de la institución. Sin importar el tipo de solicitud que sea, en ambos se deben detallar los motivos del porque se está enviado un documento de este tipo.
- **Gestión inscripciones:** El departamento de contabilidad recibe las inscripciones de parte de secretaría para su respectiva revisión, aprobación y envío al siguiente departamento.
- **Gestión matrículas:** Lo mismo sucede con este módulo, ya que, en este se espera las solicitudes de matrícula de parte de secretaría para su revisión, aprobación y envío al siguiente departamento.
- **Gestión pago docentes:** En este módulo se listan los datos correspondientes a los diferentes docentes o instructores de la institución, quienes ya cuentan con la documentación necesaria para que se les deposite el pago mensual en sus cuentas bancarias, por lo tanto, las tareas que aquí se realizan son las de listar, aprobar, ingresar informes y enviar al departamento Gerencial para su respectiva aprobación.
- **Gestión venta solicitudes:** Este módulo va de la mano con el módulo de gestión de solicitudes generales, ya que cada solicitud creada en aquel módulo viaja a través de los departamentos a los que hayan sido reenviados para su respectiva aprobación, así que en este módulo se listan las solicitudes que se hayan vuelto a enviar a este departamento para su aprobación.
- **Gestión pago mensualidad:** Los pagos de mensualidad los realizan cada uno de los estudiantes, lo cual consiste en ingresar el comprobante de pago a la cuenta bancaria de la institución para su almacenamiento, verificación y aprobación por parte del departamento de contabilidad.

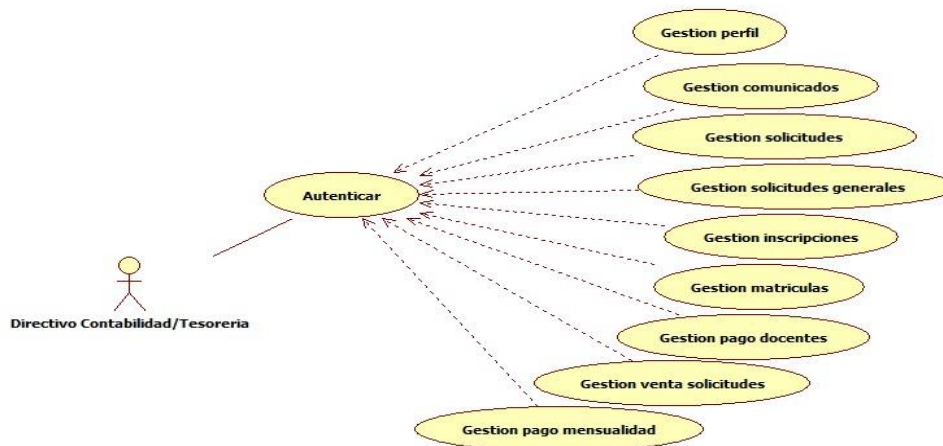


Gráfico 3-3: Casos de uso del directivo de contabilidad/tesorería

Realizado por: Aldas W.; Pilataxi C., 2020

Casos de uso para el usuario directivo de Inspección

Los módulos presentes para este departamento, aparte de ser los que se establecen para todos los usuarios, también posee ciertos procesos a llevar a cabo que son propios de este, mismos que se muestran en la **Gráfico 4-3** y se detallan a continuación:

- **Gestión de solicitudes generales:** En este módulo se muestra el listado de las solicitudes generales dirigidas a este departamento para su correspondiente aprobación y reenvío a otro departamento de ser necesario, además que también se puede ingresar algún informe en el caso de necesitarlo.
- **Gestión justificaciones:** Se listan las solicitudes de justificación que han sido comprados en el departamento de contabilidad y posteriormente enviados por los estudiantes para su correspondiente aprobación.
- **Gestión novedades:** Este módulo permite al usuario de este departamento, ingresar, modificar, eliminar y listar las novedades presentadas ya sea por los estudiantes, instructores o docentes de la institución, en el cual se detallan los acontecimientos para tomar las medidas pertinentes.
- **Gestión aulas:** Este módulo consiste en ingresar, modificar, eliminar y listar las aulas de clases, así como también los recursos que cada uno posee para gestionar el estado de cada uno de ellos.

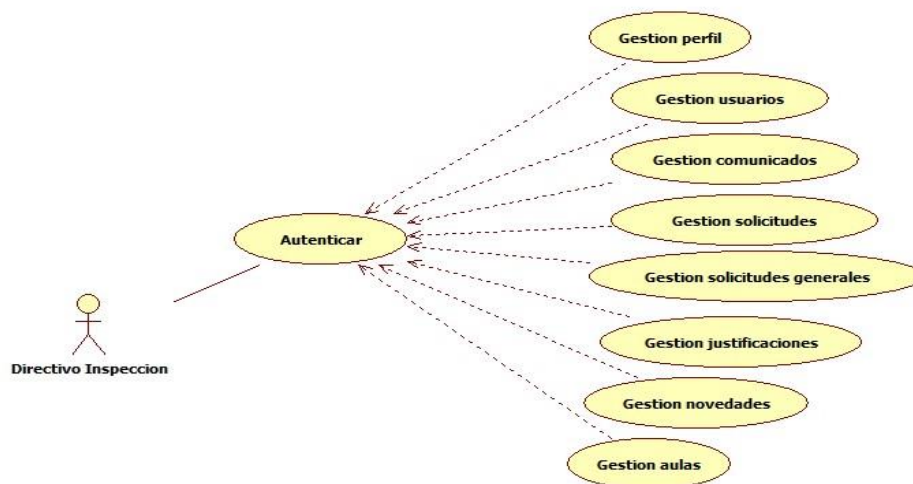


Gráfico 4-3: Casos de uso del directivo de inspección

Realizado por: Aldas W.; Pilataxi C., 2020

Casos de uso para el usuario directivo de Pedagogía

Este departamento tiene a su disposición el uso de múltiples módulos, los cuales muchos de ellos son muy similares a los descritos en los casos anteriores, razón por la cual aquí solo se detallan aquellos que son nuevos o que, a pesar de parecer similares, cumplen algo adicional, de este modo la **Gráfico 5-3**, refleja todo lo establecido, además se describen a continuación:

- **Gestión justificaciones:** En este módulo se listan las solicitudes enviadas por los estudiantes, en el que se detallan las razones de la inasistencia.
- **Gestión matrículas:** En este módulo se recibe los datos previstos para esta actividad con la finalidad de ser aprobadas para completar con el proceso respectivo.
- **Gestión cursos:** Este es uno de los módulos más importantes para este departamento, ya que aquí se encuentra todo lo relacionado al material de estudio a presentar por los docentes para su correspondiente cátedra, para llevar a cabo dicha actividad, el docente es quien envía todos los datos solicitados como pueden ser sílabos, planificaciones, asistencias de estudiantes, etc., todo esto con la finalidad de verificar y aprobar. También se puede realizar las evaluaciones a docentes que son enviados a los estudiantes, lo cual consiste en ingresar, modificar, eliminar, listar resultados de cada uno de estos y realizar los mismos procesos para las preguntas que cada uno contiene.
- **Gestión pago docentes:** Para este módulo el personal administrativo debe verificar que se cumplan con algunos de los datos solicitados en la parte de gestión de cursos, ya que, para realizar el registro de pago a docentes, se necesitará comprobará que hayan sido validados.

En este módulo se permite ingresar, modificar, eliminar y listar los datos correspondientes a la realización de esta actividad.

- **Gestión pago instructores:** Lo único que se hace en este módulo es listar los pagos realizados a los instructores en el departamento de Asesoría vial para su respectiva aprobación.
- **Gestión novedades:** Consiste únicamente en listar las novedades registradas por el departamento de Inspección.

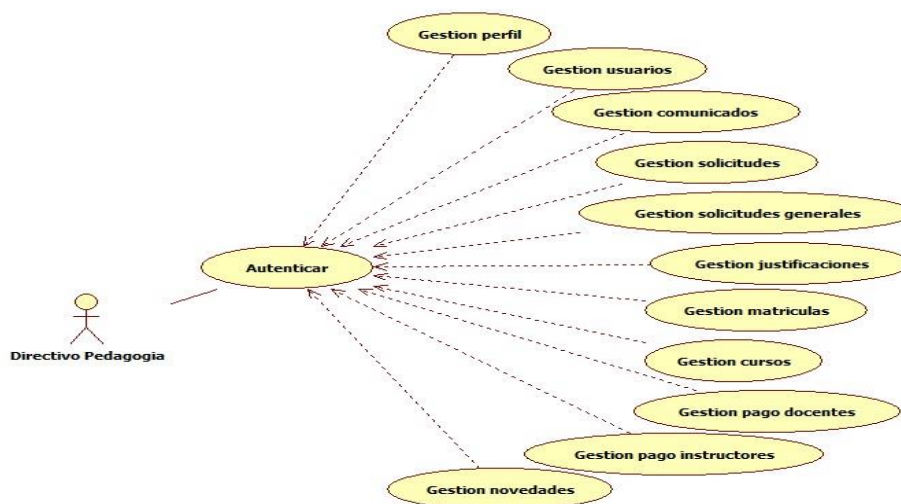


Gráfico 5-3: Casos de uso del directivo de pedagogía

Realizado por: Aldas W.; Pilataxi C., 2020

Casos de uso para el usuario directivo de Asesoría Vial

Los módulos característicos de este departamento se detallan en la **Gráfico 6-3**, que además se detallan a continuación:

- **Gestión instructores:** Consiste en gestionar el material de estudio necesarios para la catedra de conducción por parte de los instructores, por lo que cada instructor envía los datos solicitados por la institución los cuales van desde el silabo, planificación, etc., para su respectiva aprobación. Además, que también se puede ingresar, modificar, eliminar y listar las evaluaciones a los instructores que son desarrollados mediante los estudiantes.
- **Gestión vehículos:** Este módulo consiste en ingresar, eliminar y listar los informes de estado de cada uno de los vehículos disponibles en la institución.

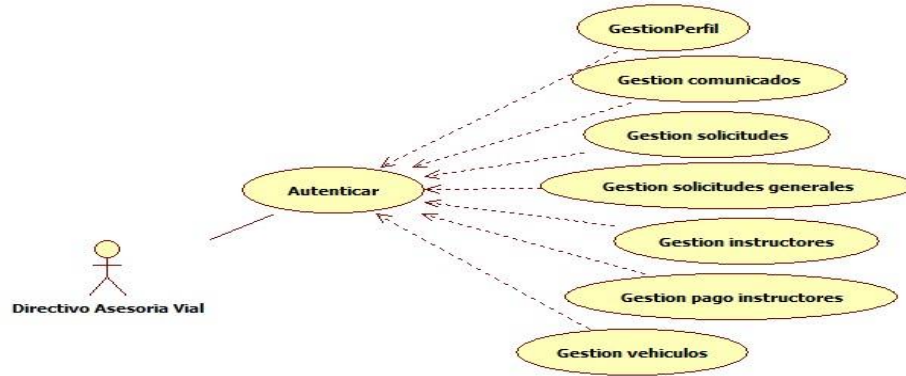


Gráfico 6-3: Casos de uso del directivo de asesoría vial

Realizado por: Aldas W.; Pilataxi C., 2020

Casos de uso para el usuario directivo de Gerencia

Los módulos establecidos en este departamento se muestran en la **Gráfico 7-3**, para lo cual se detalla que todos los procesos llevados a cabo en esta sección son simplemente el listado de cada uno de estos y su respectiva aprobación, debido a que los demás departamentos han realizado la mayor parte de cada uno de los procesos.

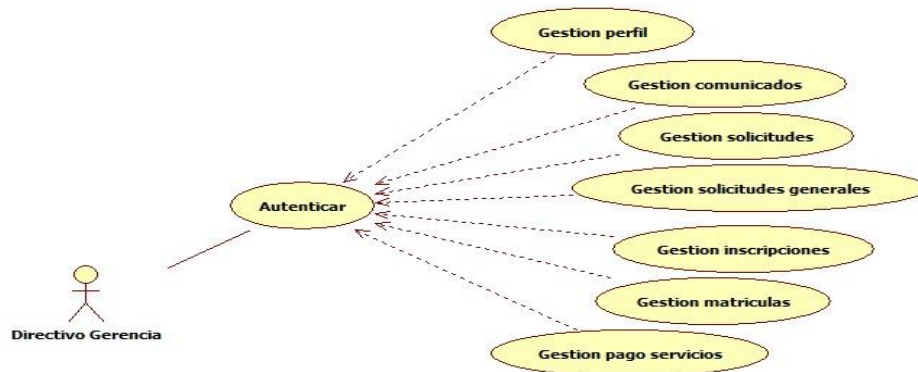


Gráfico 7-3: Casos de uso del directivo de gerencia

Realizado por: Aldas W.; Pilataxi C., 2020

Casos de uso para el usuario directivo de Servicio

Este departamento tiene las funciones básicas de planteadas por el sistema, sin embargo, el módulo a destacar en la **Gráfico 8-3**, es el de gestión de solicitudes generales, en el que simplemente se listan las solicitudes enviadas a este departamento para su debida aprobación.

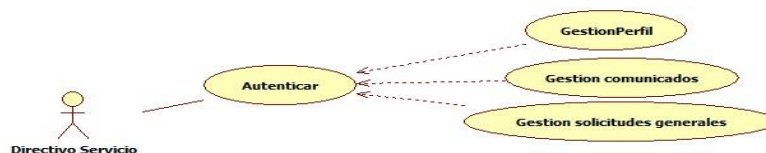


Gráfico 3-3: Diagrama de casos de uso del directivo de servicio

Realizado por: Aldas W.; Pilataxi C., 2020

Casos de uso para el usuario Practicante DTIC

Los módulos disponibles para el practicante se muestran en la **Gráfico 9-3**, son similares a los descritos anteriormente como es el caso de la autenticación y la gestión de perfil, sin embargo, el resto varía en cuanto a funcionalidad, lo cual se detalla a continuación:

- **Gestión Comunicados:** En este módulo el practicante puede ingresar, modificar, eliminar y listar los comunicados, pero con la diferencia de que lo realiza a un grupo limitado de usuarios, lo cual no incluye a estudiantes, docentes o instructores.
- **Gestión proyectos:** Cada practicante tiene bajo su mando las acciones de ingresar, eliminar, listar y modificar el estado de las tareas asignadas.

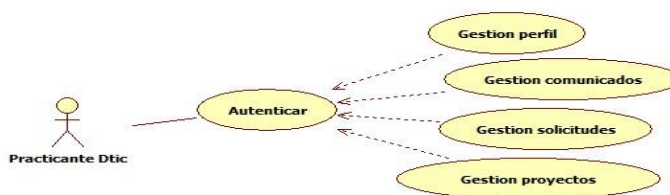


Gráfico 4-3: Casos de uso para el practicante de DTIC

Realizado por: Aldas W.; Pilataxi C., 2020

Casos de uso para el usuario Estudiante

Para acceder al sistema el estudiante, necesita registrar la solicitud de inscripción, mismo que servirá para el registro en el sistema, esta acción solo se podrá hacer una vez, ya que el sistema verifica que su cédula de identidad no se haya registrado previamente, al completar con este proceso, podrá acceder al sistema con sus credenciales asignados por defecto a todos los usuarios lo cual es el número de cédula tanto para el usuario como para la contraseña.

Todos los módulos llevados a cabo por este usuario se pueden ver en la **Gráfico 10-3**, que además se detallan a continuación algunos de ellos:

- **Solicitar inscripción:** Para que un usuario sea parte del sistema como estudiante necesita registrar sus datos en el formulario correspondiente en la fecha prevista para tal acción, cuando se realizase esta acción, obtendrá una cuenta de acceso.
- **Gestión comunicados:** Consiste en el listado de comunicados que han sido enviados por alguno de los departamentos administrativos.
- **Gestión solicitudes generales:** Cuando el estudiante haya comprado este tipo de solicitud en el departamento de contabilidad, entonces se listará en este módulo para ingresar los datos solicitados y posteriormente enviar al departamento Gerencial para su respectiva aprobación.
- **Gestión justificaciones:** Si el estudiante tiene faltas por asistencia a clases, entonces debe ir al departamento de contabilidad a comprar este tipo de solicitud, que posteriormente se le enviará y se listará en este módulo para que se pueda ingresar los datos necesarios y enviar al departamento de pedagogía.
- **Gestión inscripción:** Mediante este módulo el estudiante podrá dar seguimiento a su solicitud de inscripción previamente realizadas, además que podrá ingresar los datos requeridos de ser necesario.
- **Gestión matrícula:** En este módulo el estudiante envía los datos solicitados al departamento de secretaría para su correspondiente aprobación.
- **Gestión cursos:** El módulo permite listar las asignaturas de cada estudiante por fases, y que dentro de cada una de ellas se podrá listar los recursos de evaluación a docente, sílabos, planificaciones para su correspondiente análisis y toma de acción de ser necesario.
- **Gestión pago mensualidad:** A través de este módulo el estudiante podrá ingresar los comprobantes de pago mensual solicitados por la institución en la cuenta bancaria correspondiente.

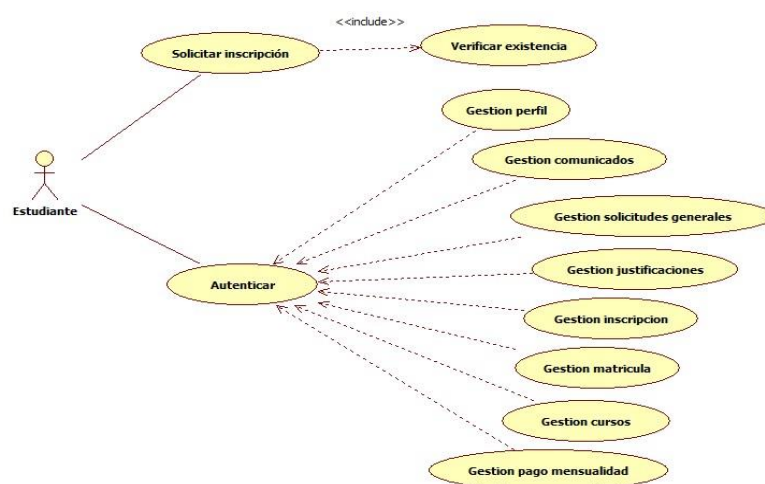


Gráfico 5-3: Casos de uso del estudiante

Realizado por: Aldas W.; Pilataxi C., 2020

Casos de uso para el usuario Docente e Instructor

Tanto el usuario docente como el instructor tienen como proceso principal la gestión de los cursos asignados por sección y paralelo tal como se muestra en la **Gráfico 11-3** y **Gráfico 12-3** respectivamente, mismo que consiste en la administración del material de estudio para su respectiva revisión en cada departamento administrativo, el resto de módulos ya se han descrito en los usuarios anteriores por lo cual no es necesario dar mayor explicación.

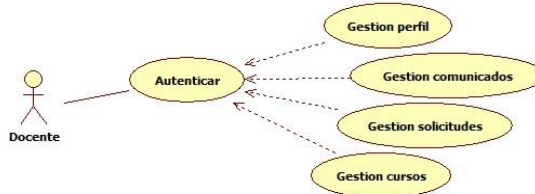


Gráfico 6-3: Casos de uso del docente

Realizado por: Aldas W.; Pilataxi C., 2020

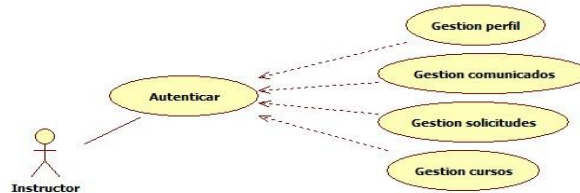


Gráfico 7-3: Casos de uso del instructor

Realizado por: Aldas W.; Pilataxi C., 2020

Casos de uso para la aplicación móvil

Para realizar una solicitud de examen psicosenométrico el usuario(Cliente) de la aplicación móvil, ingresa los datos necesarios y posteriormente son comprobados e ingresados en la base de datos.

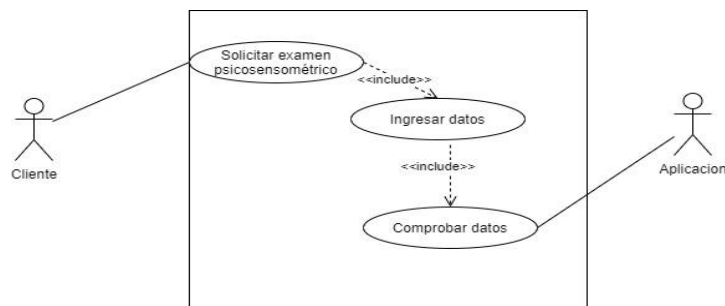


Gráfico 8-3: Casos de uso de solicitud de exámen psicosenométrico

Realizado por: Aldas W.; Pilataxi C., 2020

Para atender una solicitud de examen psicosenométrico el administrador del sistema móvil, es necesario la autenticación del administrador, posteriormente atenderá la solicitud.

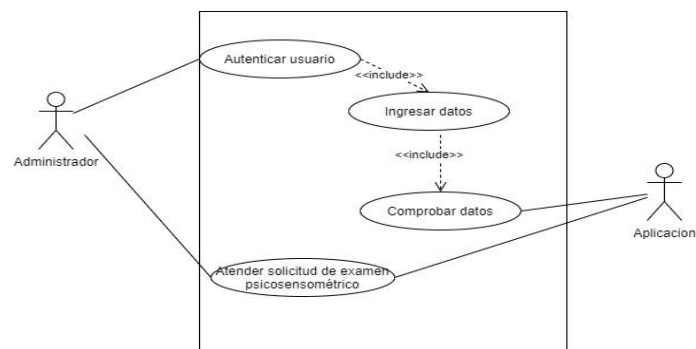


Gráfico 9-3: Casos de uso para la solicitud del examen psicosenométrico

Realizado por: Aldas W.; Pilataxi C., 2020

3.3.3.4. *Diseño de la interfaz web del usuario*

La creación de las interfaces de usuario se ha llevado a cabo con la herramienta Balsamiq, el cual permite crear prototipos dinámicos en etapas tempranas de desarrollo de sistemas virtuales, esto permite tener una retroalimentación crítica del cliente antes de malgastar esfuerzos en escribir código para dar forma a lo que se solicite, ya que ayuda mucho a detectar errores con facilidad.

Interfaces de aplicación web

Para diseñar de manera rápida y con el mínimo detalle posible, se ha utilizado Blasamiq cloud, el cual es una herramienta para el desarrollo temprano de interfaces de usuario intuitivas, que pueden ser utilizadas para representar el resultado final. Al definir las interfaces de usuario para la aplicación web, se ha tomado en cuenta aquellas funcionalidades en la que se requiere de interacción con el usuario, de esta manera en la **Figura 17-3** vemos que para el ingreso al sistema se requiere de una interfaz en el que se solicite los datos de acceso, lo cual consiste en la cédula de identidad y la contraseña.

Al momento de ingresar los datos solicitados el usuario deberá pulsar en el botón “Acceder”, el cual verificará el registro del usuario para posteriormente redirigirlo a su respectiva cuenta de usuario. Por otro lado, tenemos el botón “Inscripción”, el cual, al pulsarlo, redirige al usuario a otra interfaz en el que se podrá registrar. Además de las opciones descritas, también está la opción “Exámen Psicosenométrico”, el cual permite descargar la aplicación móvil para agendar un examen psicosenométrico.



Figura 10-3: Interfaz de la página principal de la aplicación web

Realizado por: Aldas W.; Pilataxi C., 2020

La interfaz de registro de inscripciones es válido únicamente para los usuarios que vayan a ingresar a la institución en calidad de estudiantes, por lo que los datos a ingresar para realizar dicha actividad son, los nombres, apellidos, cédula de identidad, correo electrónico, teléfono de contacto, dirección, sección en el cual se va a registrar, y un archivo con la copia de la cédula de identidad y la papeleta de votación, que posteriormente se enviará al departamento de Secretaría para su debida aprobación.

Este formulario planteado en la **Figura 4-3**, cumple dos funciones, la primer consiste en notificar al departamento de Secretaría sobre el registro de una nueva inscripción y por otro lado también crea la cuenta del usuario, en este caso un estudiante. Esta acción se realiza a través de la opción "Enviar" el cual automáticamente realiza las cuestiones planteadas y redirige hacia otra interfaz el cual permite navegar entre las distintas funcionalidades planteadas para cada caso.

Solicitud de inscripción

https://administrativo.conduespoch.com/inscripcion-estudiante

logo

ESCUELA DE CONDUCCION ESPOCH CONDUESPOCH EP
LICENCIA PROFESIONAL TIPO C

Estimados estudiantes, mediante este formulario ustedes podrán iniciar su proceso de inscripción en el nuevo periodo académico, ...
BIENVENIDOS A LA MEJOR ESCUELA DE CONDUCCIÓN DEL CENTRO DEL PAÍS

Nombres

Apellidos

Cédula

Correo electrónico

Teléfono celular o convencional

Dirección

Seleccione Sección y Horario

--Seleccione opción--

- Nocturna (Lunes a viernes de 18:30 a 21:30 pm)
- Vespertina (Lunes a viernes de 15:00 a 18:00 pm)
- Fin de Semana (Sábado y domingo de 07:30 a 15:45 pm)

Copia de la cédula de identidad y papeleta de votación. Copia del Título de Bachiller o de nivel Superior y Copia del deposito o transferencia bancaria. Tamaño de archivo máximo 3Mb.

Seleccionar archivo Ningún archivo seleccionado

Enviar Cancelar

Figura 11-3: Interfaz para la solicitud de inscripciones

Realizado por: Aldas W.; Pilataxi C., 2020

El hecho de tener múltiples roles de usuario nos da la posibilidad de distribuir distintas funcionalidades apropiadas para cada uno de ellos en distintos grupos de interfaces, sin embargo, todos concuerdan en que poseen una barra lateral de navegación ubicada en la parte izquierda de la pantalla del dispositivo en el que se esté ejecutando. En la parte superior se tiene el encabezado de la aplicación, el cual contiene el nombre del sistema, nombre y foto de perfil de usuario que, al dar clic sobre este, despliega el menú en el que se dispone la opción de abandonar el sistema.

En cuanto al cuerpo de la aplicación, disponemos de una etiqueta en la parte superior, el cual identifica el nombre del módulo al que el usuario ha accedido, siguiendo hacia abajo, se ubican los ítems de los distintos subniveles a desarrollarse dentro de ella. En cada ítem se ubica desde la parte superior todo lo referente a una funcionalidad, empezando por los datos de la institución, un formulario de registro y en la parte inferior el listado de todo lo ingresado en el formulario, cada uno de los registros que se muestran en el formulario poseen las opciones de eliminar, modificar, mostrar, dependiendo de las necesidades presentes.

Los detalles de una interfaz típica utilizada para este trabajo se presentan en la **Figura 5-3**.

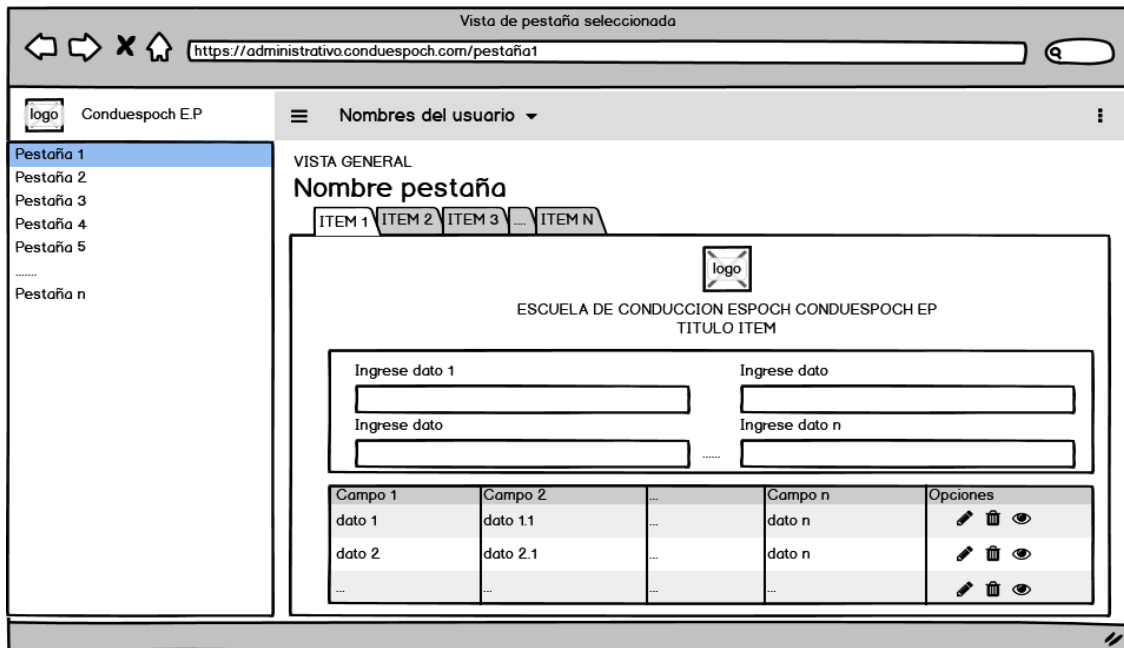


Figura 12-3: Interfaz base para las vistas de todos los procesos

Realizado por: Aldas W.; Pilataxi C., 2020

Interfaces de aplicación móvil

Para diseñar los prototipos de las vistas de la aplicación móvil de la misma manera se utilizó Balsamiq Cloud. Como se muestra en la **Figura 6-3** se puede identificar la pantalla principal de la aplicación web, en donde se tiene el logo de la empresa, además de dos campos (usuario y contraseña), un botón de ingreso, y un enlace para los clientes donde pueden solicitar el examen psicosenométrico.

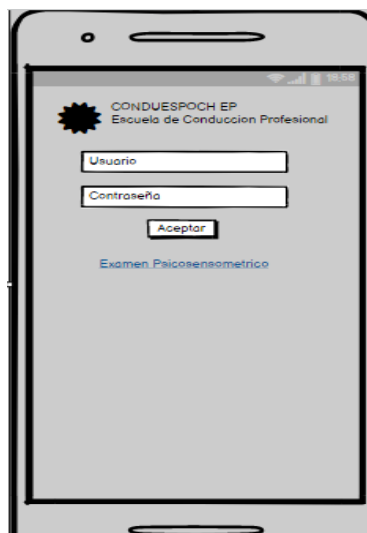


Figura 13-3: Interfaz principal de la aplicación móvil

Realizado por: Aldas W.; Pilataxi C., 2020

Para realizar la solicitud de examen psicosenométrico por parte del cliente se puede observar en la **Figura 7-3** que se tiene un campo de correo, cedula, nombre, apellido, celular, dirección, fecha y hora, y un botón de aceptar.

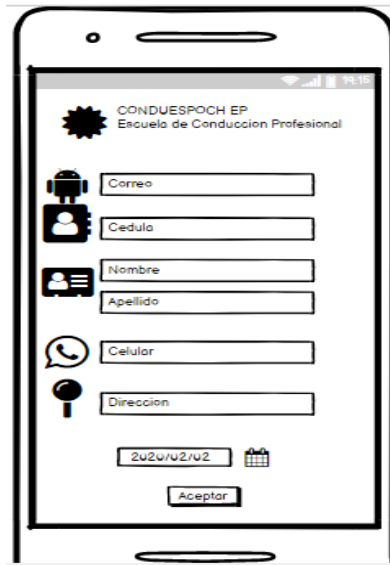


Figura 7-3: Interfaz de solicitud de examen psicosenométrico

Realizado por: Aldas W.; Pilataxi C., 2020

Para que el Administrador pueda atender las solicitudes de los clientes, en el diseño de la interfaz se listan las solicitudes con los datos que fueron ingresados por los clientes como se muestra en la **Figura 8-3**.

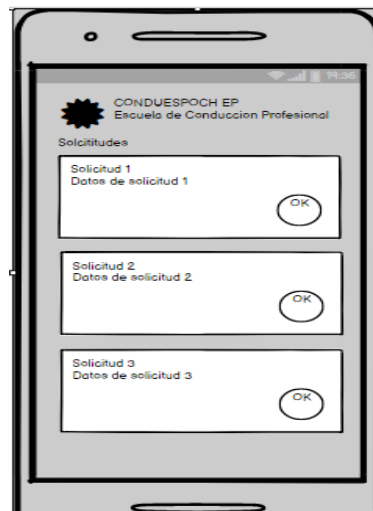


Figura 14-3: Interfaz de solicitudes pendientes de atender

Realizado por: Aldas W.; Pilataxi C., 2020

Los diseños finales de las interfaces de usuario para cada uno de los procesos, se detallan de mejor manera en los manuales de usuario ubicados en el **ANEXO C**.

3.3.3.5. *Diseño de la base de datos*

El diseño de la base de datos al ser uno de los primeros artefactos en ser desarrollado, necesita de una buena representación de las expectativas del cliente, ya que de este dependerá el desempeño de los sistemas que lo vayan a utilizar. El gestor de base de datos utilizado para el desarrollo de este trabajo ha sido Postgresql, el cual permite la realización de ingeniería inversa mediante la herramienta de modelado Power designer, este a su vez ha permitido obtener el modelo lógico que muestra las tablas con sus relaciones.

Los tipos de datos utilizados en todo el diseño de la base de datos son los siguientes:

- **Serial:** Almacena datos de tipo enteros auto genéricos, principalmente usados para la generación automática de las claves primarias de cada registro (Ejemplo: codarchivo)
- **Text:** Almacena cadenas de caracteres sin límite de extensión. (Ejemplo: contenido)
- **Varchar:** Almacena cadenas de texto con un límite de espacio (Ejemplo: cedula)
- **Date:** Almacena fechas en formato yy-mm-dd (Ejemplo: fechaejecucion)
- **Timestamp:** Almacena fechas en formato yy-mm-dd 00:00:00 (Ejemplo: fechacreacion)
- **Time:** Almacena hora, minutos y segundos en formato 00:00:00 (Ejemplo: horainicio)

A continuación, se muestran los diagramas de la base de datos:

Diagrama Lógico de la Base de Datos para la aplicación web

El resultado final es una base de datos relacional con 43 tablas, definidas en la **Gráfico 15-3**, que se aloja en el servidor de base de datos de la misma institución.

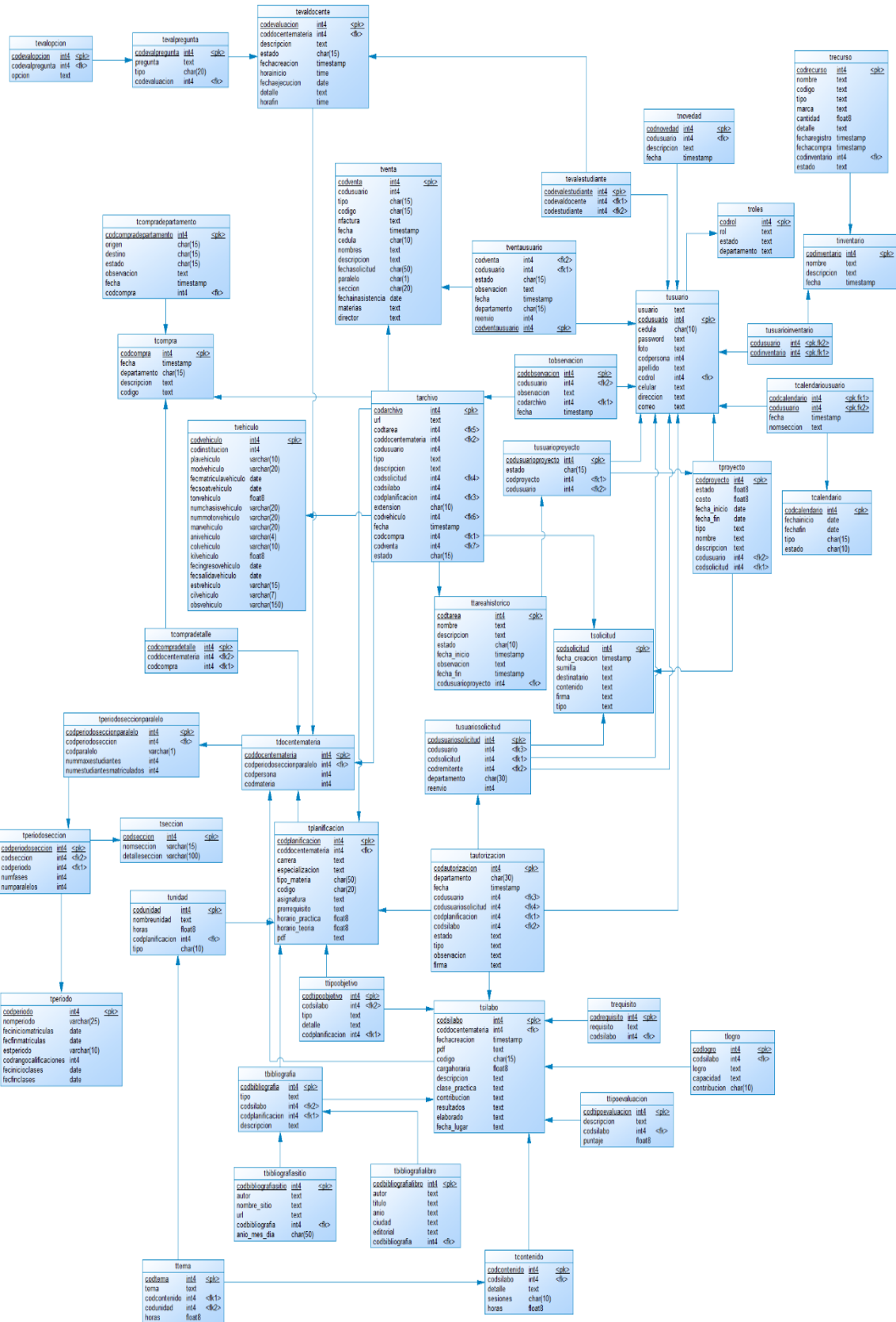


Gráfico 15-3: Diagrama lógico de la base de datos del sistema web

Realizado por: Aldas W.; Pilataxi C., 2020

Diagrama Lógico de la Base de Datos para la aplicación móvil



Gráfico 16-3: Diagrama lógico de la base de datos del sistema móvil

Realizado por: Aldas W.; Pilataxi C., 2020

Una vez que se ha diseñado la base de datos se muestra cómo se encuentran estructuradas las tablas con sus respectivos tipos de datos y sus características.

Diccionario de datos

El diccionario de datos consiste en definir el tipo de datos y sus características de las tablas de una base de datos, los cuales serán utilizados en el desarrollo de un sistema.

Para obtener el diccionario de datos se ha empleado pgAdmin 3, el cual es una herramienta de gestión y administración de bases de bases de datos, que a su vez ofrece una amplia cantidad de opciones extendidas para múltiples tareas, tal es el caso de la generación del diccionario de datos.

Name	Data type	Not Null?	Primary key?	Default	Comment
codarchivo	integer	Yes	Yes	nextval('tarchivo_codarchivo_seq'::regclass)	
url	text	No	No		
codtarea	integer	No	No		
coddocentemateria	integer	No	No		
codusuario	integer	No	No		
tipo	text	No	No		
descripcion	text	No	No		
codsolicitud	integer	No	No		
codsilabo	integer	No	No		
codplanificacion	integer	No	No		
extension	character(10)	No	No		
codvehiculo	integer	No	No		
fecha	timestamp without time zone	No	No		
codcompra	integer	No	No		
codventa	integer	No	No		
estado	character(15)	No	No		

Figura 17-3: Diccionario de datos de la tabla tarchivo

Realizado por: Aldas W.; Pilataxi C., 2020

En la **Figura 9-3** se aprecia el ejemplo de una tabla, la cual tiene el nombre de cada campo, el tipo de dato, campos obligatorios, clave primaria, algoritmo para la creación de la clave por

defecto y un campo de comentario, el resto de las tablas se lo encuentra en el **ANEXO D**.

3.3.3.6. *Categorización de riesgos*

Con la finalidad de prevenir inconvenientes en el desarrollo del proyecto se ha visto la necesidad de establecer medidas preventivas para mitigar los posibles riesgos.

Una vez identificados los riesgos más probables se ha tenido que clasificarlos de acuerdo con los criterios de probabilidad de ocurrencia, impacto y exposición, todo esto se detalla a continuación para una mejor comprensión:

Probabilidad de ocurrencia de los riesgos

Para llevar a cabo esta tarea se han establecido los parámetros que se detallan en la **Tabla 17-3**.

Tabla 17-3: Probabilidad de ocurrencia de riesgos

Rango de probabilidades	Descripción	Valor
1%-33%	Baja	1
34%-67%	Media	2
68%-99%	Alta	3

Realizado por: Aldas W.; Pilataxi C., 2020

Impacto de Riesgo

El impacto del riesgo ha sido determinado utilizando los parámetros que se exponen en la **Tabla 18-3**:

Tabla 18-3: Impacto de riesgo

Rango de probabilidades	Descripción	Valor
1%-33%	Baja	1
34%-67%	Media	2
68%-99%	Alta	3

Realizado por: Aldas W.; Pilataxi C., 2020

Exposición del riesgo

El intervalo de riesgo del proyecto ha sido analizado mediante los parámetros establecidos en la **Tabla 19-3**:

Tabla 19-3: Exposición del riesgo

Rango de probabilidades	Valor	Color
Alta	>=6	Rojo
Media	3-5	Amarillo
Baja	1-2	Verde

Realizado por: Aldas W.; Pilataxi C., 2020

Tabla de priorización de riesgos del proyecto

La priorización de riesgos se ha establecido con el fin de tener claro las posibilidades de ocurrencia de uno u otro riesgo dentro del proyecto, y el nivel de impacto que este pueda causar, se ha establecido un listado de los riesgos más propensos a ocurrir, mismos que han sido categorizado desde los más probables hasta los de menor probabilidad.

Cada uno de los riesgos presentados en la **Tabla 20-3** se han clasificado de manera que se puede apreciar la probabilidad de ocurrencia, impacto y exposición, cada uno con un color distinto que va desde el color rojo, amarillo y verde, dependiendo de la implicación sobre el desarrollo del proyecto.

Tabla 20-3: Priorización de riesgos del proyecto

Id_riesgo	Probabilidad			Impacto		Exposición	
	%	Prob.	Valor	Impacto	Valor	Exposición	Valor
R1	70	Alta	3	Alta	3	Alta	8
R2	80	Alta	3	Alta	3	Alta	6
R3	70	Media	2	Alta	3	Media	4
R4	70	Media	2	Alta	3	Alta	8
R5	50	Media	2	Media	2	Baja	2
R6	40	Media	2	Media	2	Baja	1
R7	40	Media	2	Alta	3	Media	4
R8	30	Baja	1	Alta	3	Media	4
R9	30	Baja	1	Alta	3	Media	4

Realizado por: Aldas W.; Pilataxi C., 2020

Hoja de gestión de riesgos

La hoja de gestión de riesgo es una representación de las medidas a tomar en caso de ocurrencia de alguno de los riesgos definidos previamente, para lo cual se ha tomado a cada uno de estos y se los ha detallado a profundidad sobre las posibles causas, la manera en cómo se podrá cubrir los inconvenientes, entre otros datos que son relevantes para la gestión de desarrollo del proyecto.

Todas las hojas de gestión de riesgo que fueron consideradas para el desarrollo del sistema se encuentran detalladas en el **ANEXO B**.

Tabla 21-3: Hoja de gestión de riesgo

HOJA DE GESTIÓN DE RIESGO			
ID. DEL RIESGO: R1		FECHA: 19/06/2020	
Probabilidad: Alta Valor: 3	Impacto: Alta Valor: 3	Exposición: Alta Valor: 8	Prioridad: Alta Valor: 6
Descripción: Análisis incorrecto de los requerimientos funcionales			
REFINAMIENTO:			
Causas:			
<ul style="list-style-type: none"> - Requisitos con información incompleta. - Falta de comprensión del propósito del negocio. - El cliente tiene poco compromiso al proporcionar la información. 			
Consecuencias:			
<ul style="list-style-type: none"> - Replanificación constante - El diseño del proyecto no se adapta a la visión del cliente - No se cubren todas las funcionalidades 			
REDUCCIÓN:			
<ul style="list-style-type: none"> - Reuniones más prolongadas con el dueño del sistema - Requerimientos en constante revisión - Definir el alcance de los requisitos con documentación de respaldo. 			
SUPERVISIÓN:			
<ul style="list-style-type: none"> - Cumplir con lo planificado para la reducción del riesgo. - Llevar un control periódico de los requerimientos. 			
GESTIÓN:			
<ul style="list-style-type: none"> - Reuniones continuas con los stakeholders del proyecto, para las correspondientes revisiones de los avances del sistema. 			
RESPONSABLE:			
Claudio Pilataxi			

Realizado por: Aldas W.; Pilataxi C., 2020

La **Tabla 21-3** es un ejemplo de la hoja de gestión de riesgos, de cada uno de los que se ha considerado para este proyecto, además a continuación se especifica los campos a considerar para el análisis de un riesgo:

- **Id del riesgo:** Es una abreviación para establecer el número del riesgo que se va a analizar, en este proyecto se han establecido 9 riesgos por lo que el ID empieza desde R1 hasta R9.
- **Fecha:** Indica la fecha en el que se ha realizado la respectiva gestión del riesgo.
- **Probabilidad:** Representa el nivel de ocurrencia dentro de la realización del sistema.
- **Impacto:** Representa el nivel de afectación en el proceso de desarrollo del sistema.
- **Exposición:** Como su nombre lo indica es la exposición del riesgo sobre la realización del proyecto.
- **Prioridad:** Nivel de prioridad de gestión del riesgo.
- **Descripción:** Detalla el riesgo que se va a tratar, con el fin de que todos los interesados en el proyecto puedan entenderlo de manera simple y específica.
- **Refinamiento:** Dentro de este apartado tenemos dos subcampos importantes que definen tanto las causas como las consecuencias de un riesgo en específico.
- **Reducción:** Se definen las alternativas para solucionar o disminuir las probabilidades de que un riesgo afecte negativamente en el desarrollo de un proyecto.

- **Supervisión:** Son los argumentos para la realización de la supervisión de un riesgo, es decir, aquellas medidas útiles para tener en cuenta en la reducción de un riesgo.
- **Gestión:** Protocolos con los cuales se puede realizar la respectiva gestión de reducción del riesgo.
- **Responsable:** Indica el nombre de la persona encargada de realizar la gestión de un riesgo.

3.3.3.7. Historias de usuario

Las historias de usuario sirven para plasmar las necesidades del cliente, por lo tanto, cada una estas conforman el producto final y es necesario documentarlos con el fin de llevar el control. A continuación, en la **Tabla 22-3**, se define un ejemplo, el cual describe las especificaciones del requerimiento de autenticación de los usuarios.

Las historias de usuario que fueron necesarias para el desarrollo del sistema se encuentran detalladas en el **ANEXO A**.

Tabla 22-3: Autenticación de usuarios por número de cédula y contraseña

HISTORIA DE USUARIO	
Número: 01	Nombre de la Historia: Autenticación de los usuarios por número de cédula y contraseña
Modificación de historia de usuario:	
Usuario / Rol: Administrador, Directivo, Estudiante, Docente, Instructor, Practicante.	Sprint: 2
Prioridad en el Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 24
Riesgo en el Desarrollo: Alta (Alta / Media / Baja)	Puntos Reales: 24
Descripción: Como usuario del sistema necesito una página de inicio con el formulario de autenticación para el ingreso.	
Observaciones:	
Pruebas de aceptación:	
<ul style="list-style-type: none"> - Verificar que los campos de ingreso de datos estén de acuerdo con lo solicitado. - Verificar que el usuario esté registrado en el sistema. - Emitir un mensaje de alerta cuando se trate de enviar uno o más campos vacíos. - Emitir un mensaje de éxito al autenticarse correctamente. 	

Realizado por: Aldas W.; Pilataxi C., 2020

Cada historia de usuario cuenta con una o varias pruebas de aceptación como se muestra a continuación desde la **Tabla 23-3** hasta la **Tabla 26-3**, las cuales son especificaciones más detalladas de las tareas que se ha tenido que realizar para completar con la funcionalidad planteada por el usuario, lo cual determina si cumple con las necesidades previstas.

Tabla 23-3: Prueba de aceptación PA-01 de HU-01

PRUEBA DE ACEPTACIÓN	
Código: PA-01	Historia de Usuario: HU-01 Autenticación de los usuarios por número de cédula y contraseña
Nombre: Verificar que los campos de ingreso de datos estén de acuerdo con lo solicitado	
Responsable: Claudio Pilataxi	Fecha: 24/06/2020
Descripción: Verificar que los campos de ingreso de datos necesarios se encuentren en la vista de autenticación de usuario.	
Condiciones de Ejecución: <ul style="list-style-type: none"> - Tener conexión a internet - Debe haber una conexión a la base de datos - Tener cargado la página del sistema en el navegador. - El usuario debe estar registrado en el sistema 	
Pasos de ejecución: <ul style="list-style-type: none"> - Completar los campos requeridos (cédula y contraseña) - Clic en el botón Acceder 	
Resultado esperado: La codificación realizada se ejecuta correctamente y las vistas tienen los campos necesarios para el ingreso de los datos.	
Evaluación de la prueba: Exitosa.	
Realizado por: Aldas W.; Pilataxi C., 2020	

Tabla 24-3: Prueba de aceptación PA-02 de HU-01

PRUEBA DE ACEPTACIÓN	
Código: PA-02	Historia de Usuario: HU-01 Autenticación de los usuarios por número de cédula y contraseña
Nombre: Verificar que el usuario esté registrado en el sistema.	
Responsable: Claudio Pilataxi	Fecha: 24/06/2020
Descripción: Verificar que los datos del usuario se encuentren registrados.	
Condiciones de Ejecución: <ul style="list-style-type: none"> - Tener conexión a internet - Debe haber una conexión a la base de datos - Tener cargado la página del sistema en el navegador 	
Pasos de ejecución: <ul style="list-style-type: none"> - Completar los campos requeridos (cédula y contraseña) - Clic en el botón Acceder 	
Resultado esperado: Mostrar mensaje que indique que el usuario es incorrecto.	
Evaluación de la prueba: Exitosa.	
Realizado por: Aldas W.; Pilataxi C., 2020	

Tabla 25-3: Prueba de aceptación PA-03 de HU-01

PRUEBA DE ACEPTACIÓN	
Código: PA-03	Historia de Usuario: HU-01 Autenticación de los usuarios por número de cédula y contraseña.
Nombre: Emitir un mensaje de alerta cuando se trate de enviar uno o más campos vacíos.	
Responsable: Claudio Pilataxi	Fecha: 24/06/2020
Descripción: Verificar que los datos del usuario se encuentren registrados.	
Condiciones de Ejecución: <ul style="list-style-type: none"> - Tener conexión a internet - Tener cargado la página del sistema en el navegador 	
Pasos de ejecución: <ul style="list-style-type: none"> - Clic en el botón Acceder 	
Resultado esperado: Mostrar mensaje que indique que los datos no han sido completados.	
Evaluación de la prueba: Exitosa.	
Realizado por: Aldas W.; Pilataxi C., 2020	

Tabla 26-3: Prueba de aceptación PA-04 de HU-01

PRUEBA DE ACEPTACIÓN	
Código: PA-04	Historia de Usuario: HU-01 Autenticación de los usuarios por número de cédula y contraseña
Nombre: Emitir un mensaje de éxito al autenticarse correctamente.	
Responsable: Claudio Pilataxi	Fecha: 24/06/2020
Descripción: El usuario al proporcionar los datos al sistema debe esperar a que se notifique que sus datos han sido correctos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> - Tener conexión a internet - Debe haber una conexión a la base de datos - Tener cargado la página del sistema en el navegador - El usuario debe estar registrado en el sistema 	
Pasos de ejecución:	
<ul style="list-style-type: none"> - Completar los campos requeridos (cédula y contraseña) - Clic en el botón Acceder 	
Resultado esperado: Mostrar mensaje de éxito, donde se indique que los datos del usuario han sido validados.	
Evaluación de la prueba: Exitosa.	

Realizado por: Aldas W.; Pilataxi C., 2020

3.3.3.8. Tareas de ingeniería

Representan el desarrollo técnico de algunas de las dependencias que tiene el sistema pero que no son consideradas por el cliente, estas pueden ser scripts de consultas de bases de datos, apis de servicios externos, etc. Por lo general este tipo de actividades están ligadas a una historia de usuario, lo cual indica que sirven para completar su funcionamiento.

Cada tarea de ingeniería posee una o más pruebas de aceptación, los cuales sirven para demostrar que se ha cumplido con lo establecido.

Todas las tareas de ingeniería que fueron necesarias para el desarrollo de la aplicación se encuentran detalladas en el **ANEXO A**.

Tabla 27-3: Crear el controlador de autenticación de usuarios

TAREA DE INGENIERIA	
Historia de Usuario: HU-01	
Número de Tarea: TI-01	Nombre de Tarea: Crear el controlador de autenticación de usuarios
Tipo de Tarea: Desarrollo	Puntos Estimados: 8
Fecha Inicio: 22/06/2020	Fecha Fin: 22/06/2020
Programador Responsable: Claudio Pilataxi	
Descripción: Se debe crear el controlador de login de usuarios con su respectivo método de autenticar.	
Pruebas de aceptación:	
<ul style="list-style-type: none"> - Verificar que el controlador esté en la estructura correcta del proyecto. 	

Realizado por: Aldas W.; Pilataxi C., 2020

Tabla 28-3: Prueba de aceptación PA-05 de HU-01

PRUEBA DE ACEPTACIÓN	
Código: PA-05	Historia de Usuario: HU-01 Autenticación de los usuarios por número de cédula y contraseña
Nombre: Verificar que el controlador esté en la estructura correcta del proyecto.	
Responsable: Claudio Pilataxi	Fecha: 22/06/2020
Descripción: Comprobación de que exista el controlador solicitado dentro del directorio del proyecto.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> - Debe existir el controlador de Login de usuarios dentro del proyecto con su respectivo método de autenticar. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> - Ingresar al directorio de la aplicación - Ingresar a la carpeta app - Ingresar a la carpeta Http - Ingresar a la carpeta Contollers - Ingresar a la carpeta General - Clic sobre el archivo LoginController 	
Resultado esperado: Debe estar el controlador de Login con su respectivo método de autenticar.	
Evaluación de la prueba: Exitosa.	

Realizado por: Aldas W.; Pilataxi C., 2020

Tabla 29-3: Crear el modelo para la autenticación de usuarios

TAREA DE INGENIERIA	
Historia de Usuario: HU-01	
Número de Tarea: TI-02	Nombre de Tarea: Crear el modelo para la autenticación de usuarios
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 23/06/2020	Fecha Fin: 23/06/2020
Programador Responsable: Claudio Pilataxi	
Descripción: Definir el modelo de autenticación de usuarios en el respectivo directorio	
Pruebas de aceptación:	
<ul style="list-style-type: none"> - Verificar que el modelo esté en la estructura correcta de la aplicación. - Comprobar que exista el enlace del modelo con la tabla tusuario de la base de datos. 	

Realizado por: Aldas W.; Pilataxi C., 2020

Tabla 30-3: Prueba de aceptación PA-06 de HU-01

PRUEBA DE ACEPTACIÓN	
Código: PA-06	Historia de Usuario: HU-01 Autenticación de los usuarios por número de cédula y contraseña
Nombre: Verificar que el modelo esté en la estructura correcta del proyecto.	
Responsable: Claudio Pilataxi	Fecha: 23/06/2020
Descripción: Verificar que el modelo esté en la estructura correcta de la aplicación.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> - Debe existir el modelo Usuario dentro del proyecto 	
Pasos de ejecución:	
<ul style="list-style-type: none"> - Ingresar al directorio de la aplicación - Ingresar a la carpeta app - Ingresar a la carpeta Models - Clic sobre el archivo Usuario 	
Resultado esperado: Debe estar el modelo Usuario con los respectivos campos.	
Evaluación de la prueba: Exitosa.	

Realizado por: Aldas W.; Pilataxi C., 2020

Tabla 31-3: Prueba de aceptación PA-07 de HU-01

PRUEBA DE ACEPTACIÓN	
Código: PA-07	Historia de Usuario: HU-01 Autenticación de los usuarios por número de cédula y contraseña
Nombre: Comprobar que exista el enlace del modelo con la tabla tusuario de la base de datos.	
Responsable: Claudio Pilataxi	Fecha: 23/06/2020
Descripción: Verificar que esté enlazado con la tabla tusuario de la base de datos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> - Debe existir el modelo Usuario dentro del proyecto - Debe estar creado la tabla tusuario en la base de datos 	
Pasos de ejecución:	
<ul style="list-style-type: none"> - Ingresar al directorio de la aplicación - Ingresar a la carpeta app - Clic sobre el archivo Usuario.php - Verificar que el atributo \$table tenga el nombre de la tabla tusuario. - Verificar que los atributos de la tabla tusuario estén en el array \$fillable 	
Resultado esperado: Debe estar el modelo Usuario con los respectivos campos y a su vez enlazado con la tabla de la base de datos.	
Evaluación de la prueba: Exitosa.	
Realizado por: Aldas W.; Pilataxi C., 2020	

Tabla 32-3: Crear el API REST de la autenticación de usuarios

TAREA DE INGENIERIA	
Historia de Usuario: HU-01	
Número de Tarea: TI-03	Nombre de Tarea: Crear el API REST de la autenticación de usuarios
Tipo de Tarea: Desarrollo	Puntos Estimados: 6
Fecha Inicio: 23/06/2020	Fecha Fin: 23/06/2020
Programador Responsable: Claudio Pilataxi	
Descripción: Definir el API REST de la autenticación de usuarios y que esté enlazado con su respectivo método y controlador.	
Pruebas de aceptación:	
<ul style="list-style-type: none"> - Verificar que exista la ruta Generales\LoginController en el archivo de los servicios API - Comprobar que la ruta esté enlazada con el controlador correspondiente. - Realizar el testing de la API REST creada. 	
Realizado por: Aldas W.; Pilataxi C., 2020	

Tabla 33-3: Prueba de aceptación PA-08 de HU-01

PRUEBA DE ACEPTACIÓN	
Código: PA-08	Historia de Usuario: HU-01 Autenticación de los usuarios por número de cédula y contraseña
Nombre: Verificar que exista la ruta Generales\LoginController en el archivo de los servicios API	
Responsable: Claudio Pilataxi	Fecha: 23/06/2020
Descripción: Definir el API REST correspondiente para que el sistema pueda obtener los datos necesarios.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> - Debe existir el servicio REST en el archivo api.php 	
Pasos de ejecución:	
<ul style="list-style-type: none"> - Ingresar al directorio de la aplicación - Ingresar a la carpeta routes - Clic sobre el archivo api.php 	
Resultado esperado: Debe estar el método solicitado dentro de la ruta Generales\LoginController	
Evaluación de la prueba: Exitosa.	
Realizado por: Aldas W.; Pilataxi C., 2020	

Tabla 34-3: Prueba de aceptación PA-09 de HU-01

PRUEBA DE ACEPTACIÓN	
Código: PA-09	Historia de Usuario: HU-01 Autenticación de los usuarios por número de cédula y contraseña
Nombre: Comprobar que la ruta esté enlazada con el controlador correspondiente.	
Responsable: Claudio Pilataxi	Fecha: 23/06/2020
Descripción: Verificar que esté enlazado el controlador de login de usuarios con el API REST.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> - Debe existir el controlador Login con el método de autenticación de usuarios y además estar enlazado a la ruta correspondiente. 	
Pasos de ejecución:	
<ul style="list-style-type: none"> - Ingresar al directorio de la aplicación - Ingresar a la carpeta routes - Clic sobre el archivo api.php - Verificar que la API REST Generales\LoginController esté enlazada con el controlador Login y su respectivo verbo apiResource, el cual contiene los métodos básicos como el post. 	
Resultado esperado: La API REST debe poder invocar al método post del controlador de autenticación.	
Evaluación de la prueba: Exitosa.	

Realizado por: Aldas W.; Pilataxi C., 2020

Tabla 35-3: Prueba de aceptación PA-10 de HU-01

PRUEBA DE ACEPTACIÓN	
Código: PA-10	Historia de Usuario: HU-01 Autenticación de los usuarios por número de cédula y contraseña
Nombre: Realizar el testing de la API REST creada.	
Responsable: Claudio Pilataxi	Fecha: 23/06/2020
Descripción: Verificar que el api devuelva datos correspondientes a lo solicitado	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> - Debe existir el servicio API REST - Tener una herramienta de solicitudes de APIS como postman - Tener conexión con la base de datos - Ingresar las credenciales del usuario - Tener la ruta del api 	
Pasos de ejecución:	
<ul style="list-style-type: none"> - Ejecutar postman - Ingresar la dirección <code>http://127.0.0.1:8000/api/login</code> - Seleccionar el método post - Clic en body - Clic en send 	
Resultado esperado: Se devuelven los datos en formato JSON con el status 200	
<pre>{ "severity": "success", "sms": "Usuario Autenticado", "status": "200", "type": "loginuser", "datos": { "codusuario": 3, "cedula": "0602155913", "usuario": "CARLOS ERNESTO", "apellido": "GUFFANTTE NARANJO", "password": "XkjgUIBKJhkKJghKHJkjhjGJHjbHBkjakjkjYRFgjhRYyUGyuUFGuuygYU", "foto": "CARLOS ERNESTO GUFFANTTE NARANJO_20210308142239.jpg", "codpersona": 133, "correo": "guffanttecarlos@hotmail.com", "celular": "0996383423", "direccion": "Conjunto San Antonio", "rol": "ADMINISTRADOR", "codrol": 15, "departamento": "DTIC" } }</pre>	
Evaluación de la prueba: Exitosa.	

Realizado por: Aldas W.; Pilataxi C., 2020

Tabla 36-3: Desarrollo del Frontend para la autenticación de usuarios

TAREA DE INGENIERIA	
Historia de Usuario: HU-01	
Número de Tarea: TI-04	Nombre de Tarea: Desarrollo del Frontend para la autenticación de usuarios
Tipo de Tarea: Desarrollo	Puntos Estimados: 8
Fecha Inicio: 24/06/2020	Fecha Fin: 24/06/2020
Programador Responsable: Claudio Pilataxi	
Descripción: Desarrollar la interfaz de usuario para la autenticación de los usuarios en el sistema.	
Pruebas de aceptación:	
<ul style="list-style-type: none"> - Verificar que esté creado la interfaz de usuario para la autenticación de usuarios al sistema. - Verificar la conexión con el Backend 	

Realizado por: Aldas W.; Pilataxi C., 2020

Tabla 37-3: Prueba de aceptación PA-11 de HU-01

PRUEBA DE ACEPTACIÓN	
Código: PA-11	Historia de Usuario: HU-01 Autenticación de los usuarios por número de cédula y contraseña
Nombre: Verificar que esté creado la interfaz de usuario para la autenticación de usuarios al sistema.	
Responsable: Claudio Pilataxi	Fecha: 24/06/2020
Descripción: Verificar que exista la interfaz de autenticación de usuarios para el acceso al sistema.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> - Debe existir el formulario de autenticación de usuarios 	
Pasos de ejecución:	
<ul style="list-style-type: none"> - Estar en la página principal del sistema - Ingresar los datos del usuario (usuario y contraseña) - Clic en el botón de Ingresar 	
Resultado esperado: Debe ingresar al sistema generando el token correspondiente.	
Evaluación de la prueba: Exitosa.	

Realizado por: Aldas W.; Pilataxi C., 2020

Tabla 38-3: Prueba de aceptación PA-12 de HU-01

PRUEBA DE ACEPTACIÓN	
Código: PA-12	Historia de Usuario: HU-01 Autenticación de los usuarios por número de cédula y contraseña
Nombre: Verificar la conexión con el Backend.	
Responsable: Claudio Pilataxi	Fecha: 24/06/2020
Descripción: Verificar que exista una conexión a la API REST api/login	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> - Debe existir el formulario de autenticación en la interfaz de usuario. - Debe existir conexión con la ruta api/login 	
Pasos de ejecución:	
<ul style="list-style-type: none"> - Ingresar al directorio de la aplicación - Ingresar a la carpeta resources/js/resources/views - Clic en el archivo Login.js - Ingresar a la carpeta resources/js/actions/ProcesoGestionUsuarios - Clic en el archivo Login.js 	
Resultado esperado: Debe estar el método de envío de datos al backend	
Evaluación de la prueba: Exitosa.	

Realizado por: Aldas W.; Pilataxi C., 2020

3.3.3.9. Historias técnicas

Las historias técnicas son aquellas especificaciones que no se han establecido abiertamente por

el cliente pero que sin embargo el equipo de desarrollo lo considera necesario para el funcionamiento del producto final. Este tipo de tareas son incluidas en la planificación de entrega del producto final, debido al esfuerzo que requieren. En la mayoría de los casos este tipo de actividades se los lleva a cabo al inicio de un proyecto, ya que permite establecer pautas a seguir, tal es el caso del diseño de una base de datos, diseño de la arquitectura del sistema, etc.

Las tareas de ingeniería planteadas para este trabajo son desarrolladas al inicio del proyecto, debido a que es ahí donde más se requieren de las especificaciones técnicas, sin embargo, debido a las necesidades planteadas, estas también se desarrollan a lo largo de todos los Sprints planteados, específicamente al final de cada uno de ellos, ya que se desarrollan lo que en un principio se ha definido en el plan de pruebas para cada una de las funcionalidades.

El ejemplo de la **Tabla 39-3**, muestra a detalle el desarrollo de una historia técnica perteneciente a de uno de los casos de prueba planificados. Las historias técnicas que fueron consideradas para el desarrollo del sistema se encuentran detalladas en el **ANEXO E**.

Tabla 39-3: Historia técnica

HISTORIA TÉCNICA	
ID: HT-06	Descripción: Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-01, HU-02, HU-03, HU-04, HU-06, HU-06, HU-07
Modificación de historia de usuario:	
Usuario: Desarrollador	Sprint Asignado: 2
Prioridad en el negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 8
Riesgo en el Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 8
Descripción: Como desarrollador necesito realizar las pruebas unitarias a las funcionalidades correspondientes a este Sprint	
Observaciones: Cada una de las pruebas unitarias deben ser realizadas con el framework Jest y ejecutada para probar su funcionalidad.	

Realizado por: Aldas W.; Pilataxi C., 2020

3.3.3.10. *Diseño y construcción de casos de prueba*

El desarrollo de los casos de prueba se llevará a cabo en cada uno de los Sprints definidos al comienzo de la planificación, ya que estos son tomados como historias técnicas.

Para la documentación de los casos de prueba desarrollados se establece un formato en el que se especifica los detalles técnicos necesarios para ser ejecutados, mismos que se describen a continuación:

- **Id:** Permite identificar el caso de prueba, para lo cual se ha establecido denominarlo con una letra “P” mayúscula seguido de un número

- **Caso de prueba:** Es la nomenclatura que define al caso de prueba, así que para nombrar un caso de prueba se ha decidido detallar el tipo de prueba a implementar y sobre cuál de las historias de usuario.
- **Descripción:** En esta sección se da a conocer sobre las funcionalidades que serán puesto a prueba.
- **Fecha:** Representa la fecha de inicio de la construcción del caso de prueba.
- **Funcionalidad/Característica:** Es una descripción más detallada de lo que se va a realizar en el caso de prueba.
- **Resultado esperado:** Especifica el resultado que se va a obtener al finalizar la ejecución de un determinado caso de prueba.
- **Código:** Muestra las líneas de código empleadas para llevar a cabo los casos de prueba, en cada uno de estos se establece los datos de entrada, los datos esperados y las condiciones a probar.
- **Dependencias con otros casos de Prueba:** Los casos de prueba planteados para este trabajo al ser de tipo unitario, no requieren de la dependencia de otras pruebas, sin embargo, es necesario establecerlo.
- **Pasos de Ejecución:** Detalla aquellos pasos necesarios para llevar a cabo la funcionalidad, y sobre los cuales la prueba debe superar.
- **Estado:** Representa el estado de ejecución de una prueba unitaria.
- **Última Fecha de Estado:** Es la fecha en la que se culmina con la construcción de los casos de prueba.
- **Observaciones:** Aquí se detallan todos los inconvenientes encontrados o alguna especificación que se quiera dar con el fin de documentar de mejor manera el caso de prueba.

En la **Tabla 40-3** se ejemplifica las partes del formato planteado para cada caso de prueba llevado a cabo en cada Sprint.

Los casos de pruebas que fueron consideradas durante el desarrollo del proyecto se encuentran detalladas en el **ANEXO E**.

Tabla 40-3: Caso de prueba

Id	P1-P5
Caso de Prueba	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-01
Descripción	Los casos de prueba presentados pertenecen a las siguientes funcionalidades: - Autenticación de usuarios
Fecha	10/07/2020
Funcionalidad/Característica	Las pruebas unitarias planteadas permiten probar las unidades de código que permiten interactuar al usuario con el sistema, mientras realiza las acciones de autenticación.
Datos/Acciones	- Deben ser importadas las librerías necesarias de testing

de Entrada	<ul style="list-style-type: none"> - Debe importarse el script a ser testeado - Las variables de testing deben estar inicializadas
Resultado Esperado	La prueba unitaria debe ser exitosa
Código	<pre> /* * PRUEBAS UNITARIAS DE LOS ACTIONS */ describe("ACTION Login", () => { let store; let expectedState; let user; beforeEach(async () => { moxios.install(); store = mockStore({}); expectedState = { severity: "success", sms: "Usuario autenticado", status: "200", type: "showLogin", datos: [{ codusuario: 1 }] }; moxios.stubRequest(`\${url}/api/login`, { status: 200, response: expectedState }); }); afterEach(() => { moxios.uninstall(); }); it("PU1.- Envio de datos de login vacios", function(done) { moxios.wait(function() { store.dispatch(postLogin(user)).then(() => { const actions = store.getActions(); expect(actions[0].user).not.toBeNull(); }); done(); }); }); user = { usuario: "0606060123", constrasenia: "0606060123" }; it("PU2.- Mensaje de estado de autenticación", function(done) { moxios.wait(function() { store.dispatch(postLogin(user)).then(() => { const actions = store.getActions(); expect(actions[0].type).toBe("POST_MESSAGE"); }); done(); }); }); it("PU3.- Ejecución de evento submit del formulario", function(done) { moxios.wait(function() { store.dispatch(postLogin(user)).then(() => { const actions = store.getActions(); expect(actions.length).toBe(1); }); done(); }); }); </pre>

	<pre> }); }); /* * PRUEBAS UNITARIAS DE LOS REDUCERS */ describe("REDUCER UserLocalStorage", () => { const resourceOne = { login: true, token: { apellido: "SECOND NAME", cedula: "0606060601", celular: "0989898988", codperiodoseccion: 1, codpersona: 1, codrol: 1, codusuario: 1, correo: "email@email.com", departamento: "DTIC", direccion: "donde sea, no importa", foto: "default.png", password: "SFFDG8888", rol: "DIRECTIVO", usuario: "FIRST NAME" } }; const dataArray = [resourceOne]; it("PU4.- REDUCER DEFAULT Retorna el estado por defecto", () => { const newState = usuariolocal(undefined, {}); expect(newState).toEqual([]); }); it("PU5.- SET_USERDATA: Recibe un array de usuariolocal con datos", () => { expect(usuariolocal([], { type: "SET_USERDATA", usuariolocal: dataArray })).toEqual(dataArray); }); }); }); </pre>
Dependencias con otros casos de Prueba	No.
Información para el seguimiento	
Pasos de Ejecución	<ol style="list-style-type: none"> 1. Ingresar al sistema 2. Completar los campos de autenticación (cédula y contraseña) 3. Clic en Ingresar
Estado	Éxito
Última Fecha de Estado	10/07/2020
Observaciones	Ninguna

Realizado por: Aldas W.; Pilataxi C., 2020

3.3.4 Fase de cierre

En esta fase se lleva a cabo las pruebas del sistema para garantizar el cumplimiento de los objetivos planteados al comienzo del proyecto. Además, se realizan capacitaciones a las personas que vayan a utilizar el producto final por lo que también se hace un análisis de la documentación

respectiva. (Trigas, 2019)

Las actividades desarrolladas se detallan en la **Tabla 41-3**, en el que además se establecen a los responsables de cada una de ellas.

Tabla 41-3: Actividades para la conclusión del proyecto

Actividad	Descripción	Responsable
Despliegue de la aplicación web y móvil	Despliegue de la aplicación web y móvil en el servidor de la empresa.	Wilmer Aldas, Claudio Pilataxi
Documentación de la aplicación	Elaboración del manual técnico y los manuales de usuario	Wilmer Aldas, Claudio Pilataxi
Capacitación a los usuarios	Capacitación a los usuarios para el uso correcto del sistema.	Wilmer Aldas, Claudio Pilataxi

Realizado por: Aldas W.; Pilataxi C., 2021

Tanto el despliegue de la aplicación web como el de la aplicación móvil se lo ha realizado en el hosting de la institución, al igual que la base de datos. La aplicación web permite el acceso a través de internet con el correspondiente dominio determinado para este, del mismo modo sucede con la aplicación móvil.

Para definir la guía de uso de los distintos procesos desarrollados en el proyecto, se han elaborado múltiples manuales de usuario, uno por cada rol que existe, mismos que se establecen en el **Anexo C**.

3.3.4.1. *BurnDown Chart*

Para una mejor apreciación del seguimiento del proyecto se muestra el BurnDown Chart, la **Gráfico 1-3** muestra la velocidad con la que se realizó cada Sprint.

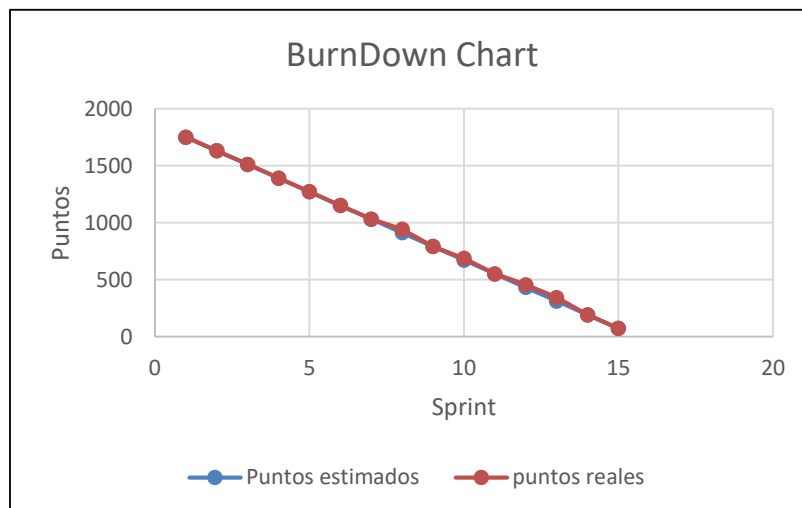


Gráfico 1-3: Avance del proyecto (BurnDownChart)

Realizado por: Aldas W.; Pilataxi C., 2020

Según el **Gráfico 1-3** se puede observar que se tiene 15 Sprints, y los puntos estimados dieron un total de 1752 puntos, sin embargo, se produjeron cambios en el sprint 8, 9, 12 y 13, ya que se ocupó un esfuerzo mayor al que se había previsto. pero estos no fueron significantes para el desarrollo de la aplicación.

CAPÍTULO IV

4. RESULTADOS

La investigación que se plantea es el de tipo cuasi experimental debido a que se pretende implementar un sistema web y móvil para realizar los distintos procesos administrativos de la escuela de conducción “CONDUESPOCH E.P” y a razón de esto, se procederá a analizar el comportamiento en cuanto a la eficiencia de tiempo que se emplea en realizar una tarea. El análisis consiste en la comparación de la ejecución de la tarea tanto de manera física como de manera virtual, por lo que la técnica que se utiliza es la observación y se da uso de un cronometro para la toma de tiempos.

4.1. Ejecución y resultados de casos de prueba

Al finalizar el proceso de pruebas planificadas, lo que queda es detallar el informe de resultados de los datos obtenidos, en el que se comprueba si se cumple o no con lo establecido al comienzo del proyecto.

Las partes del informe que se han considerado importantes a destacar son, el alcance de las pruebas, un resumen de lo obtenido, la cobertura de código, el listado de defectos encontrados y el análisis final, mismos que se describen a continuación:

4.1.1. Alcance de las pruebas

En este informe se recoge los resultados obtenidos del sistema de gestión de información administrativa de la escuela de conducción profesional CONDUESPOCH, en su primera versión. Los datos obtenidos corresponden a las pruebas unitarias aplicadas a las funcionalidades del sistema definidas por el cliente, para lo cual se mide la cobertura de código. La cobertura de código no es otra cosa más que medir aquellas líneas de código correspondientes a una funcionalidad que han sido puestas a prueba para obtener un resultado esperado. La manera más detallada de representar esta actividad es como se detalla a continuación:

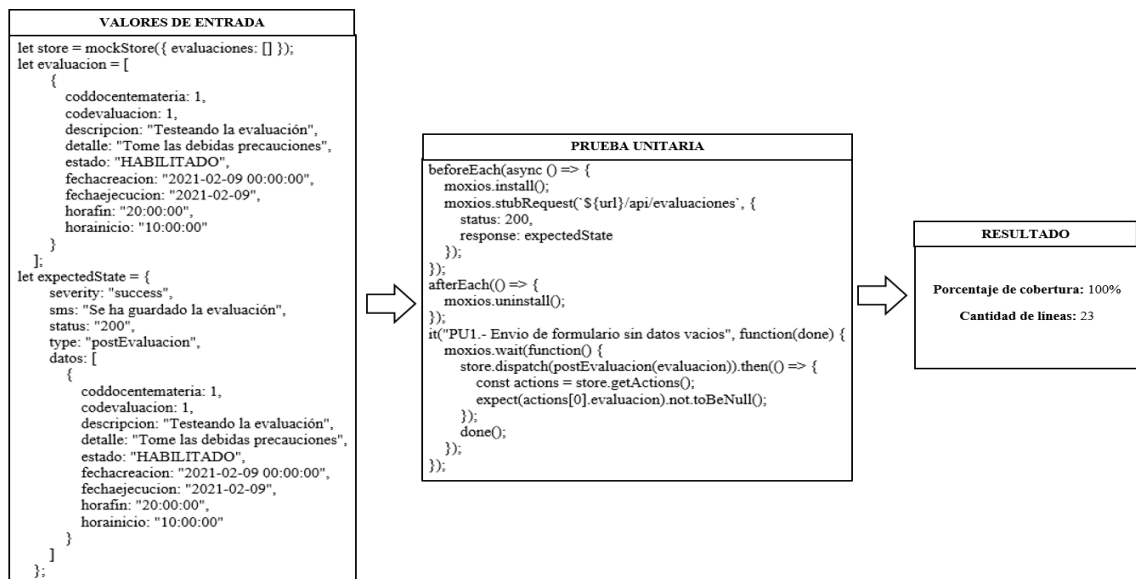


Figura 1-4: Estructura de una prueba unitaria

Realizado por: Aldas W.; Pilataxi C., 2020

En la parte izquierda de la **Figura 1-4** se puede observar los datos necesarios para un caso de prueba, mismos que son, “store”, “evaluación” y “expectedState”, en el que los dos primeros representan los datos enviados hacia la funcionalidad a probar, mientras que el ultimo es el arreglo de datos esperados.

Lo siguiente es el script de prueba en el que se tiene tres métodos que son “beforeEach”, “afterEach” y la prueba unitaria, las dos primeras estructuras representan aquellos ajustes técnicos que se deben hacer antes y después de ejecutarse el caso de prueba, por consiguiente, lo último es la prueba unitaria. Al final de todo esto en la parte derecha, se muestra el resultado obtenido, en el que se da a conocer la cantidad de líneas de código de la funcionalidad que se ha probado y el porcentaje de cobertura que esto representa. Tomando en cuenta esto, se puede ver el proceso que siguen cada una de las pruebas desarrolladas para este trabajo.

4.1.2. Cobertura de código

La **Tabla 1-4** muestra los resultados de cobertura de código obtenidos al ejecutar todos los casos de prueba del sistema de gestión de procesos administrativos de la escuela de conducción profesional CONDUESPOCH, en su primera versión.

Los datos que a continuación se detallan son, los sprints con las respectivas funcionalidades para las pruebas planteadas, la cantidad de líneas de código de cada funcionalidad con la cobertura deseada y finalmente la cantidad de líneas de código que han superado las pruebas con el porcentaje asignado.

Tabla 1-4: Resultado de cobertura de código por requerimientos

Sprint	Requerimientos	Líneas de código	Cobertura deseado	Cantidad Real (Líneas de código)	Porcentaje Real
Sprint 2	HU-01	22	100%	22	100%
	HU-02	18	100%	18	100%
	HU-03	44	100%	44	100%
	HU-04	63	100%	57	90%
	HU-05	44	100%	44	100%
	HU-06	22	100%	22	100%
	HU-07	22	100%	22	100%
Sprint 3	HU-08	13	100%	13	100%
	HU-09	16	100%	16	100%
	HU-10	4	100%	4	100%
	HU-11	20	100%	18	90%
	HU-12	21	100%	19	90%
Sprint 4	HU-13	10	100%	10	100%
	HU-14	27	100%	27	100%
	HU-15	6	100%	6	100%
	HU-16	6	100%	4	67%
	HU-17	16	100%	16	100%
	HU-18	21	100%	19	90%
	HU-19	20	100%	18	90%
	HU-20	4	100%	4	100%
Sprint 5	HU-21	7	100%	7	100%
	HU-22	6	100%	6	100%
	HU-23	10	100%	8	80%
	HU-24	13	100%	10	77%
	HU-25	6	100%	6	100%
	HU-26	6	100%	6	100%
	HU-27	6	100%	6	100%
	HU-28	18	100%	16	89%
	HU-29	4	100%	4	89%
	HU-30	7	100%	4	57%
	HU-31	7	100%	7	100%
Sprint 6	HU-32	6	100%	6	100%
	HU-33	4	100%	4	100%
	HU-34	7	100%	7	100%
	HU-35	16	100%	16	100%
	HU-36	7	100%	7	100%
	HU-37	6	100%	6	100%
	HU-38	39	100%	34	87%
	HU-39	3	100%	3	100%
	HU-40	3	100%	3	100%
Sprint 7	HU-41	19	100%	19	100%
	HU-42	6	100%	6	100%
	HU-43	60	100%	56	93%
	HU-44	7	100%	7	100%
	HU-45	7	100%	4	57%
	HU-46	47	100%	43	92%
	HU-47	6	100%	6	100%
	HU-48	6	100%	6	100%
Sprint 8	HU-49	6	100%	6	100%
	HU-50	6	100%	6	100%
	HU-51	4	100%	4	100%
	HU-52	6	100%	6	100%
	HU-53	6	100%	6	100%
	HU-54	25	100%	20	80%
	HU-55	6	100%	6	100%
	HU-56	6	100%	3	50%
HU-57	6	100%	6	100%	

Sprint	Requerimientos	Líneas de código	Cobertura deseado	Cantidad Real (Líneas de código)	Porcentaje Real
Sprint 9	HU-58	6	100%	6	100%
	HU-59	6	100%	6	100%
	HU-60	6	100%	6	100%
	HU-61	6	100%	6	100%
	HU-62	19	100%	18	95%
	HU-63	57	100%	54	95%
Sprint 10	HU-64	58	100%	55	95%
	HU-65	10	100%	10	100%
	HU-66	6	100%	3	50%
	HU-67	6	100%	6	100%
Sprint 11	HU-68	24	100%	23	96%
	HU-69	6	100%	6	100%
	HU-70	6	100%	6	100%
	HU-71	6	100%	6	100%
	HU-72	19	100%	18	95%
Sprint 12	HU-73	6	100%	6	100%
	HU-74	39	100%	37	95%
	HU-75	39	100%	37	95%
	HU-76	117	100%	112	96%
	HU-77	9	100%	9	100%
	HU-78	36	100%	28	78%
Sprint 13	HU-79	6	100%	6	100%
	HU-80	6	100%	6	100%
	HU-81	6	100%	6	100%
	HU-82	36	100%	31	96%
	HU-83	6	100%	6	100%
	HU-84	6	100%	6	100%
	HU-85	6	100%	6	100%
Sprint 14	HU-86	5	100%	5	100%
	HU-87	21	100%	16	76%
	HU-88	20	100%	16	80%
	HU-89	8	100%	8	100%
	HU-90	9	100%	9	100%
Sprint 15	HU-91	36	100%	28	78%
	HU-92	6	100%	6	100%
	HU-94	20	100%	20	100%
	HU-95	34	100%	34	100%
Sprint 15	HU-96	51	100%	51	100%
	HU-97	48	100%	48	100%
	HU-98	99	100%	45	50%
Promedio de porcentaje de cobertura					94%

Realizado por: Aldas W.; Pilataxi C., 2020

Para la realización de este trabajo se utilizó el patrón de arquitectura de datos redux, el cual nos permite enviar y recibir datos desde un solo almacén sin tener que interactuar con un componente padre o en caso contrario con datos del componente hijo. La ventaja de implementar este patrón es que permite ubicar el código de manera ordenada, es decir que podemos separar los recursos para generar interfaces gráficas de los que sirven para conectar servicios externos como conexiones y consultas a bases de datos, etc.

Por otro lado, el Framework Jest planteada para el desarrollo de pruebas, nos permite conocer de manera sencilla la cobertura de código de una prueba unitaria, para lo cual los resultados obtenidos son de manera gráfica, en la **Figura 2-4**, podemos ver la cobertura de código de los

actions creators, que no es otra cosa más que un nivel de abstracción que ofrece redux para crear métodos de manipulación de datos desde un almacén compartido denominado store.

Cada vez que el usuario interactúa en la interfaz del sistema con eventos como el clic de un formulario, se dispara un action creator, el cual evalúa los datos solicitados y los devuelve dependiendo del estado del almacén de datos ya que, en caso de ser necesario, un action creator puede conectar con un servicio externo como una API.

Tomando en cuenta lo anterior, lo más viable para la implementación de las pruebas unitarias ha sido en este nivel, ya que la finalidad de este trabajo es únicamente la de probar las funcionalidades, excluyendo otro tipo de pruebas como las de interfaces de usuario.

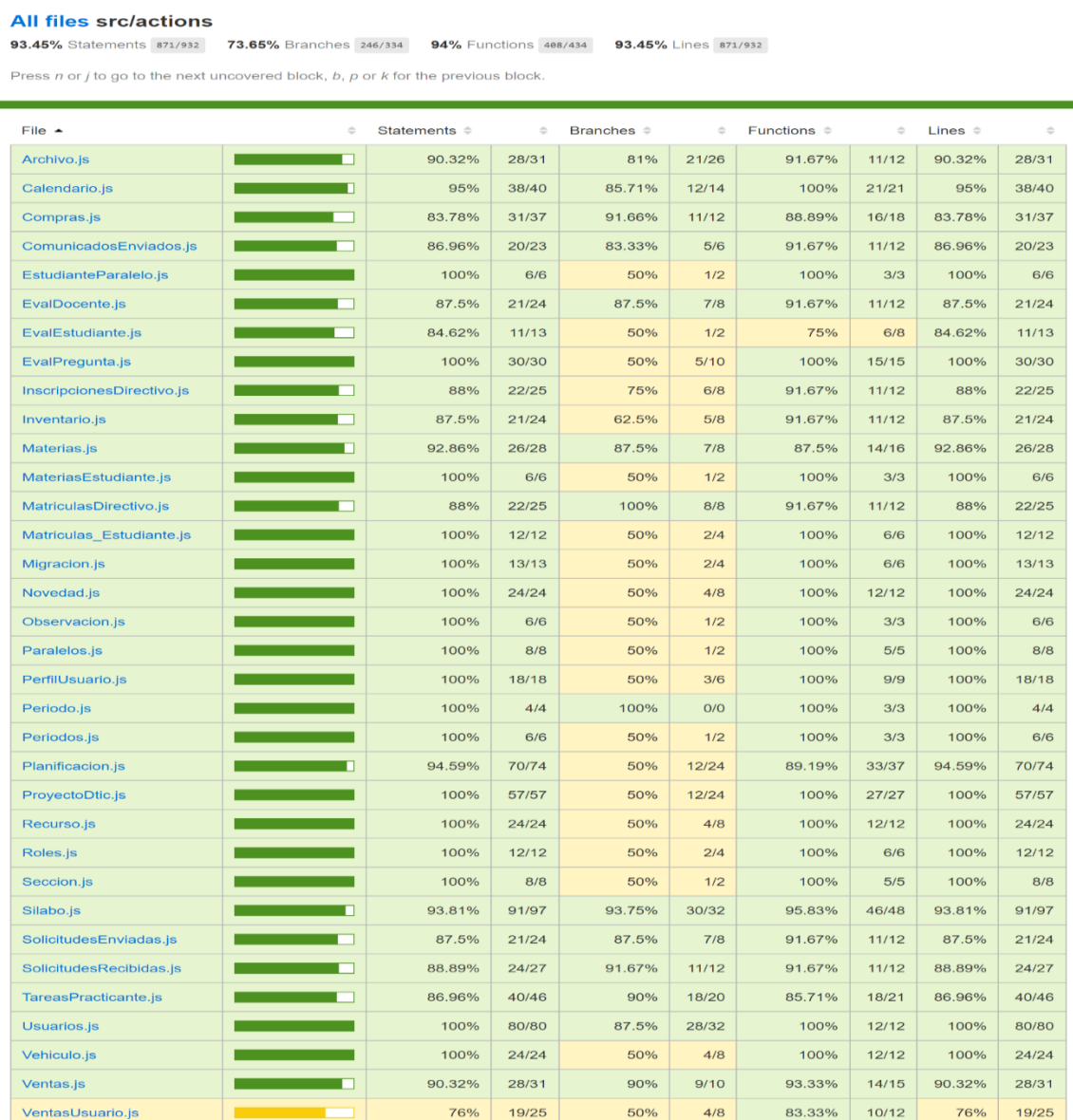


Figura 2-4: Cobertura de código de los actions creators del sistema

Realizado por: Aldas W.; Pilataxi C., 2020

Otro de los elementos de redux son las funciones reductoras, los cuales permiten a grandes rasgos modificar los datos del store o almacén compartido del sistema, los cuales han sido manipulados por los actions creators y deben mostrar las actualizaciones en la interfaz de usuario. Es así que en esta parte se controla lo que obtiene el usuario al haber interactuado con el sistema.

En la **Figura 3-4**, se muestra los resultados obtenidos tras la ejecución de pruebas unitarias de las funciones reductoras.

All files src/reducers
91.73% Statements **388/423** **85.83%** Branches **212/247** **98.17%** Functions **107/109** **91.73%** Lines **388/423**
 Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

File	Statements	Branches	Functions	Lines
Administradores.js	92.31%	12/13	87.5%	7/8
Archivo.js	82.76%	24/29	68.42%	13/19
Calendario.js	100%	16/16	100%	12/12
Compras.js	81.82%	9/11	42.86%	3/7
ComunicadosEnviados.js	88.89%	16/18	85.71%	6/7
Directivos.js	89.47%	17/19	81.82%	9/11
Docentes.js	92.31%	12/13	87.5%	7/8
EncabezadoSilabo.js	100%	14/14	100%	6/6
Estudiantes.js	92.86%	13/14	87.5%	7/8
EstudianteParalelo.js	100%	4/4	100%	3/3
EvalDocente.js	92.31%	12/13	87.5%	7/8
EvalEstudiante.js	100%	5/5	100%	4/4
EvalPregunta.js	92.31%	12/13	87.5%	7/8
Inscripcion_Estudiante.js	100%	4/4	100%	3/3
InscripcionesDirectivo.js	93.75%	15/16	85.71%	6/7
Instructores.js	92.31%	12/13	87.5%	7/8
MateriasDocente.js	100%	6/6	100%	4/4
MateriasEstudiante.js	100%	4/4	100%	3/3
Matricula_Estudiante.js	100%	4/4	100%	3/3
MatriculasDirectivo.js	93.75%	15/16	85.71%	6/7
Paralelos.js	100%	4/4	100%	3/3
Periodo.js	100%	4/4	100%	3/3
Periodos.js	100%	4/4	100%	3/3
Planificacion.js	100%	4/4	100%	3/3
Practicantes.js	92.31%	12/13	87.5%	7/8
ProyectoSolicitudDtic.js	100%	4/4	100%	3/3
ProyectosDtic.js	88.57%	31/35	76.47%	13/17
Recurso.js	92.31%	12/13	87.5%	7/8
Roles.js	100%	4/4	100%	3/3
Seccion.js	100%	4/4	100%	3/3
Show.js	100%	4/4	100%	3/3
Silabo.js	100%	4/4	100%	3/3
SolicitudesEnviadas.js	87.5%	14/16	85.71%	6/7
SolicitudesRecibidas.js	92.86%	13/14	85.71%	6/7
TareasPracticanteDtic.js	83.87%	26/31	80%	12/15
Vehiculo.js	83.33%	10/12	87.5%	7/8
Ventas.js	100%	4/4	66.67%	2/3
VentasUsuario.js	100%	4/4	66.67%	2/3

Figura 3-4: Cobertura de código de las funciones reductoras del sistema

Realizado por: Aldas W.; Pilataxi C., 2020

La ejecución de las pruebas unitarias en el Framework Jest, muestra una lista resumida en el que se detalla la cantidad de suites con las pruebas ejecutadas y el tiempo transcurrido en superar todas las condiciones planteadas, es así que en la **Figura 4-4**, se puede verificar estos datos referentes a los casos de prueba de la aplicación web, mientras que en la **Figura 4-5**, está el resultado obtenido de las pruebas aplicadas a la aplicación móvil.

```
Test Suites: 114 passed, 114 total
Tests:      639 passed, 639 total
Snapshots:  0 total
Time:       45.947 s
Ran all test suites.
```

Figura 4-4: Casos de prueba ejecutados en la aplicación web

Realizado por: Aldas W.; Pilataxi C., 2020

```
Test Suites: 1 passed, 1 total
Tests:      19 passed, 19 total
Snapshots:  0 total
Time:       4.244 s
Ran all test suites.
```

Figura 5-4: Casos de prueba ejecutados en la aplicación móvil

Realizado por: Aldas W.; Pilataxi C., 2020

4.13. Resumen de pruebas

El número de casos de prueba de prueba planificados en su respectivo artefacto de pruebas, es de 658, de los cuales 639 se han llevado a cabo en la aplicación web, mientras que 19 se han implementado en la aplicación móvil. Estas pruebas han sido realizadas y ejecutadas de forma exitosa por lo que no se ha tenido inconvenientes al momento de documentarlos, de este modo se muestra en la **Tabla 2-4**.

Tabla 2-4: Resultados de casos de prueba

Categoría	Valor
Casos de prueba planificados	658
Casos de prueba ejecutados	658
Casos de prueba exitosos	658
Casos de prueba fallidos	0
Casos de prueba bloqueados	0

Realizado por: Aldas W.; Pilataxi C., 2020

En el **Gráfico 1-4**, se muestra la cobertura de código por sprint, en el que cada uno de los 15 planteados es representado por dos barras verticales, el primero de color naranja indica la cantidad de líneas de código a probar con el porcentaje esperado, mientras que la barra azul detalla el número de líneas reales que se han cubierto con su respectivo porcentaje.

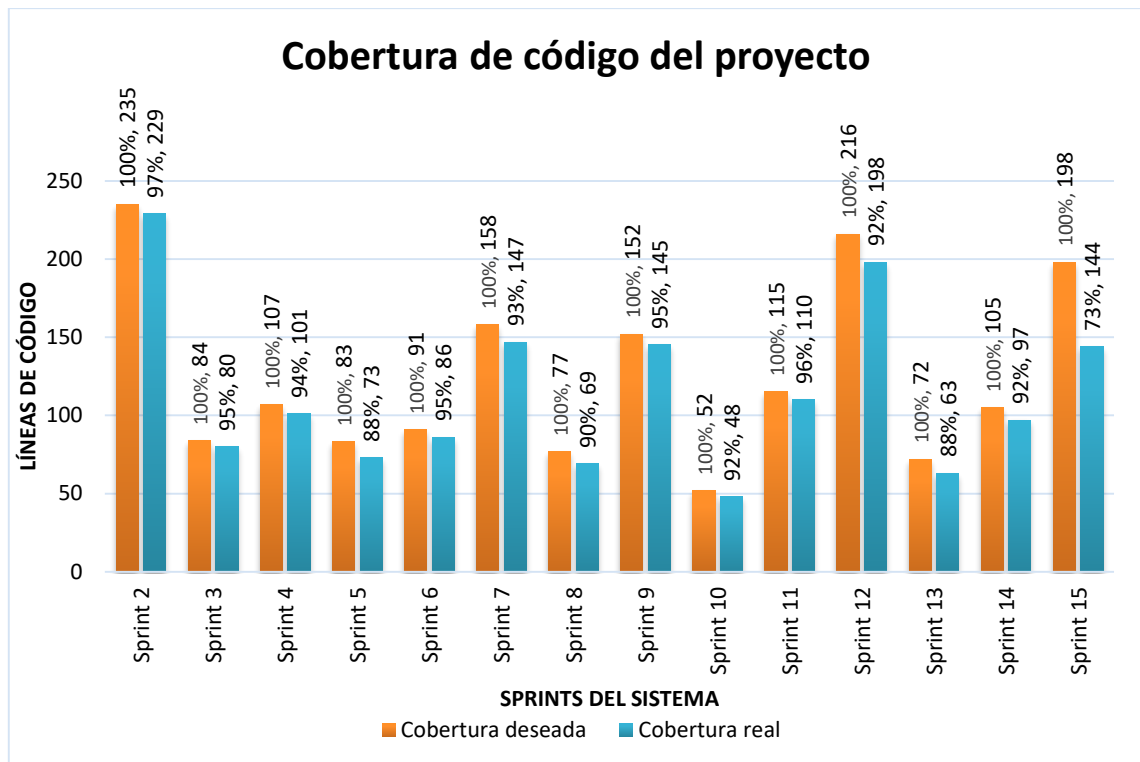


Gráfico 1-4: Cobertura de código por Sprints

Realizado por: Aldas W.; Pilataxi C., 2020

El gráfico aquí expuesto, es una forma reducida de mostrar los datos establecidos en la **Tabla 1-4**, con esto se indica que los casos de pruebas planificados han sido llevados a cabo de manera exitosa por lo que los resultados reflejan que la gran mayoría de líneas de código de las funcionalidades han sido probadas, y como detalle final se ha obtenido que el promedio de cobertura de pruebas en todo el sistema tanto web como móvil es del 94%.

4.2. Tiempos de respuesta

Cada uno de los requerimientos que se tomaron en cuenta para medir la eficiencia de tiempo del sistema son los siguientes:

- **Registro de datos de inscripción del estudiante:** El tiempo que se toma en consideración va desde que el alumno solicita información y entrega los datos a secretaría, partiendo de esto, se ha tomado los tiempos que se demoran en realizar dicha actividad 3 estudiantes en diferentes momentos.
- **Registrar un usuario:** Para la toma de tiempo de esta tarea se ha considerado desde el momento en que se adquiere los datos del usuario a registrar, el completado de los campos necesarios y su posterior almacenamiento, actividad que ha sido realizada mediante el

registro de 3 usuarios del personal administrativo en diferentes instancias.

- **Ingresar una solicitud al departamento Gerencial:** La medida de tiempo adquirida al ejecutar esta tarea, se ha llevado a cabo desde que se empieza a ingresar los datos de una nueva solicitud para el departamento gerencial hasta su correspondiente envío, lo cual ha sido ejecutado por 3 usuarios del personal administrativo.
- **Ingreso de solicitud al departamento de DTIC:** El tiempo a considerar para el ingreso de una solicitud es desde el instante en que se ingresa los datos correspondientes y finaliza cuando se obtiene la respuesta de que se ha enviado de manera correcta, para lo cual también se ha tomado las medidas de 3 usuarios del personal administrativo.
- **Ingresar comunicado:** Los comunicados enviados a los diferentes usuarios consisten en el ingreso de la información a transmitir, enviar y luego obtener el estado de envío satisfactorio, proceso sobre el cual se ha medido el tiempo de ejecución mediante 3 usuarios del personal administrativo.
- **Ingreso de solicitud de tipo general o justificación:** Para la toma de tiempo de este tipo de solicitud se ha recurrido al usuario del personal administrativo de contabilidad, quien al ingresar 3 solicitudes en determinados momentos se ha obtenido los tiempos respectivos para el análisis.
- **Ingreso pago de mensualidad de los estudiantes:** Para la medición de tiempo de ejecución de esta tarea, se ha tomado en consideración desde el instante en que se ingresa los datos solicitados, el envío y la obtención de respuesta de que se ha realizado correctamente, lo cual se ha logrado con el ingreso de 3 pagos de mensualidad de 3 estudiantes.
- **Buscar datos de inscripción por cédula de identidad o nombres del usuario:** Esta tarea empieza desde la selección del parámetro (cédula de identidad o nombres de usuario), el ingreso del dato de búsqueda y finaliza al obtener los correspondientes resultados, todo esto es llevado a cabo por medio de 3 usuarios del personal administrativo, los cuales arrojaron sus respectivos tiempos de demora.
- **Ingresar un nuevo proyecto de DTIC:** El tiempo medido para esta tarea va desde el momento en que se ingresan los datos necesarios para poner en marcha un nuevo proyecto hasta su respectivo registro, para lo cual se han tomado 3 medidas de análisis.
- **Registrar solicitudes desde la aplicación móvil:** Tiempo necesario para ingresar 3 nuevas solicitudes de examen psicosensométrico por parte de un usuario desde la aplicación móvil.

42.1. *Procesos actuales*

En la actualidad los procesos internos que se ejecutan en la Escuela de Conducción Profesional

CONDUESPOCH son realizados de manera manual, para la toma de los tiempos se escogieron las tareas que son más usadas dentro de la institución, dichos tiempos fueron registrados cada 48 horas para tener un tiempo prudente entre las mediciones como se muestran en la **Tabla 3-4**.

Tabla 3-4: Tiempo requerido para realizar una actividad de forma manual

Tarea	Tiempo1 (Min)	Tiempo2 (Min)	Tiempo3 (Min)	Promedio (Min)
Registro de datos de inscripción del estudiante	15.52	12.87	16.1	14.83
Registrar un usuario	20	16.25	18.14	18.13
Ingresar una solicitud al departamento Gerencial	15.38	19.16	15.66	16.73
Ingreso de solicitud al departamento de DTIC	15.43	14.95	16.37	15.58
Ingresar comunicado	15	12.67	17	14.89
Ingreso de solicitud de tipo general o justificación	10.31	12.16	10.54	11
Ingreso pago de mensualidad de los estudiantes	30.33	26.14	29.85	28.77
Buscar datos de inscripción por cédula de identidad o nombres del usuario	15.25	11.68	13.61	13.51
Ingresar un nuevo proyecto de DTIC	5.12	8.42	7.21	6.91
Permitir que los usuarios(clientes) puedan registrar un horario para el examen psicosenométrico.	4.21	5.7	4.5	4.8

Realizado por: Aldas W.; Pilataxi C., 2020

4.2.2. *Procesos automatizados*

Para la toma de tiempos con el sistema se los realizó de la misma manera que en el método manual, es decir con los mismos procesos, pero utilizando la aplicación. Estos tiempos fueron registrados cada 48 horas para tener un tiempo prudente entre las mediciones como se muestran en la **Tabla 4-4**.

Tabla 4-4: Tiempo requerido para realizar una actividad de forma automatizada

Tarea	Tiempo1 (Min)	Tiempo2 (Min)	Tiempo3 (Min)	Promedio (Min)
Registro de datos de inscripción del estudiante	2.89	2.35	3.18	2.80
Registrar un usuario	1.3	1.7	1.1	1.36
Ingresar una solicitud al departamento Gerencial	1.28	1.6	1.4	1.42
Ingreso de solicitud al departamento de DTIC	1.25	1.5	1.32	1.35
Ingresar comunicado	0.6	1	0.9	0.83
Ingreso de solicitud de tipo general o justificación	0.43	0.68	0.5	0.53
Ingreso pago de mensualidad de los estudiantes	0.51	0.74	0.67	0.64
Buscar datos de inscripción por cédula de identidad o nombres del usuario	0.16	0.35	0.2	0.23
Ingresar un nuevo proyecto de DTIC	1.46	1.2	1.33	1.33
Permitir que los usuarios(clientes) puedan registrar un horario para el examen psicosenométrico.	1.08	0.88	1.2	1.05

Realizado por: Aldas W.; Pilataxi C., 2020

4.2.3. Comparación de tiempos

Después de que los tiempos fueron recopilados tanto por el método manual y el automatizado, en la **Tabla 5-4** se puede observar que, al utilizar el método automatizado, el tiempo disminuye notablemente.

Al utilizar el método manual para la realización de los procesos, se obtuvo un promedio de 14.51 minutos (100%), por otra parte, en el método automatizado se obtuvo un tiempo promedio de 1.15 minutos (7.93%), dando como diferencia un tiempo de 13.36 minutos, lo que, en términos de porcentaje significa que se ha reducido un 92.07% de tiempo.

Tabla 5-4: Diferencia de tiempos entre la forma manual y automatizada

Tarea	Tiempo proceso automatizado (Min)	Tiempo proceso manual (Min)	Tiempo de diferencia (Proceso manual y automatizado) (Min)
Registro de datos de inscripción del estudiante	2.80	14.83	12.03
Registrar un usuario	1.36	18.13	16.77
Ingresar una solicitud al departamento Gerencial	1.42	16.73	15.31
Ingreso de solicitud al departamento de DTIC	1.35	15.58	14.23
Ingresar comunicado	0.83	14.89	14.06
Ingreso de solicitud de tipo general o justificación	0.53	11	10.47
Ingreso pago de mensualidad de los estudiantes	0.64	28.77	28.13
Buscar datos de inscripción por cédula de identidad o nombres del usuario	0.23	13.51	13.28
Ingresar un nuevo proyecto de DTIC	1.33	6.91	5.58
Permitir que los usuarios(clientes) puedan registrar un horario para el examen psicosenométrico.	1.05	4.8	3.75
Promedio	1.15	14.51	13.36

Realizado por: Aldas W.; Pilataxi C., 2020

4.3. Pruebas estadísticas

4.3.1. Análisis de normalidad

Una de las técnicas brindadas por la estadística descriptiva es el análisis de la normalidad, que sirve para determinar la forma de la distribución de los datos, a través del planteamiento de hipótesis que son denominadas hipótesis nula (H_0) e hipótesis alternativa (H_1), lo cual posteriormente una de ellas será rechazada y la otra aprobada, dependiendo del caso de estudio. Por consiguiente, para llevar a cabo el análisis de normalidad de los datos del presente proyecto se utilizó el software estadístico R, con el test de Shapiro-Wilk, y las correspondientes hipótesis se definen a continuación:

Planteamiento de la hipótesis para determinar una distribución normal

Hipótesis Nula (Ho) = Los datos no provienen de una distribución normal.

Hipótesis Alternativa (H1) = Los datos provienen de una distribución normal.

Test de Shapiro Wilk del promedio del tiempo sin utilizar el sistema

```
Shapiro-Wilk normality test
data: Tiempo.proceso.manual.Min.
W = 0.92303, p-value = 0.3829
```

Figura 6-4: Test Shapiro-Wilk del tiempo sin utilizar el sistema

Realizado por: Aldas W.; Pilataxi C., 2020

Test de Shapiro Wilk del promedio del tiempo al utilizar el sistema

```
Shapiro-Wilk normality test
data: Tiempo.proceso.automatizado.Min.
W = 0.88295, p-value = 0.1411
```

Figura 7-4: Test Shapiro Wilk del tiempo al utilizar el sistema

Realizado por: Aldas W.; Pilataxi C., 2020

Con los datos obtenidos en la **Figura 6-4** referente al promedio de tiempo en ejecutar tareas sin el sistema, y los resultados obtenidos en la **Figura 7-4** sobre el promedio al ejecutar las mismas tareas, pero haciendo uso del sistema, se ha conseguido demostrar que p-value es mayor a 0.05 en cada uno de los casos, lo cual implica que se rechaza la hipótesis nula(Ho) y se acepta la hipótesis alternativa(H1), lo cual indica que existe una distribución normal y por ende se puede continuar con la realización de la prueba T-Student.

4.3.2. Test T-Student para datos relacionados

Planteamiento de la Hipótesis

Se determina la hipótesis con la finalidad de poder evaluar la eficiencia de los tiempos de respuesta, el cual ayudará a aceptar o rechazar una afirmación con respecto a los indicadores que se va a evaluar.

Hipótesis nula(Ho): No existe diferencia significativa en los tiempos de respuesta en el método manual frente al método automatizado.

Hipótesis alterna(H1): Existe diferencia significativa en los tiempos de respuesta en el método manual frente al método automatizado.

Debido a la distribución normal que presentan los datos referentes a los tiempos de ejecución de las tareas observadas en distintas condiciones tanto de forma manual como automatizada, se ha establecido el uso de la prueba T-Student para datos relacionados, ya que también la cantidad de datos obtenidos que son 30, corroboran al uso de este tipo de pruebas.

Nivel de Significancia

El nivel de significancia que se utilizó fue de 0.05 ($\alpha = 5\%$) ya que se considera un margen de error aceptable.

Medición por objetivo específico

Determinar el nivel de eficiencia del sistema en relación al tiempo empleado en la gestión de procesos administrativos.

Para dicho test se ha utilizado la distribución T-Student, con un porcentaje de significancia del 95%.

```

Paired t-test

data: Tiempo.proceso.manual.Min. and Tiempo.proceso.automatizado.Min.
t = 6.3549, df = 9, p-value = 0.0001321
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 8.604858 18.117142
sample estimates:|
mean of the differences
                13.361
    
```

Figura 8-4: Prueba T-Student pareado

Realizado por: Aldas W.; Pilataxi C., 2020

En la **Figura 8-4** se muestra los datos obtenidos de la prueba T-Student mediante el software estadístico R, que para un uso más práctico se ha detallado en la **Tabla 6-4** y que a continuación de esta se detalla el significado de cada uno de los valores conseguidos.

Tabla 6-4: Valores para la prueba T-Student

Grados de libertad	9
Nivel de significancia	0.05
Valor crítico para dos colas	± 2.26
Prueba T	6.35
P-Value	0.0001321

Realizado por: Aldas W.; Pilataxi C., 2020

En la **Tabla 6-4** se obtuvieron los datos relacionados con el test T-Student para datos relacionados, en dichos datos se puede notar que p-value tiene como resultado 0.0001321 y es menor a 0.05 por lo tanto se procede a rechazar la hipótesis nula (H_0) y se acepta la hipótesis alternativa (H_1), es decir que existe una diferencia significativa en los tiempos de respuesta en

el método manual frente al método automatizado.

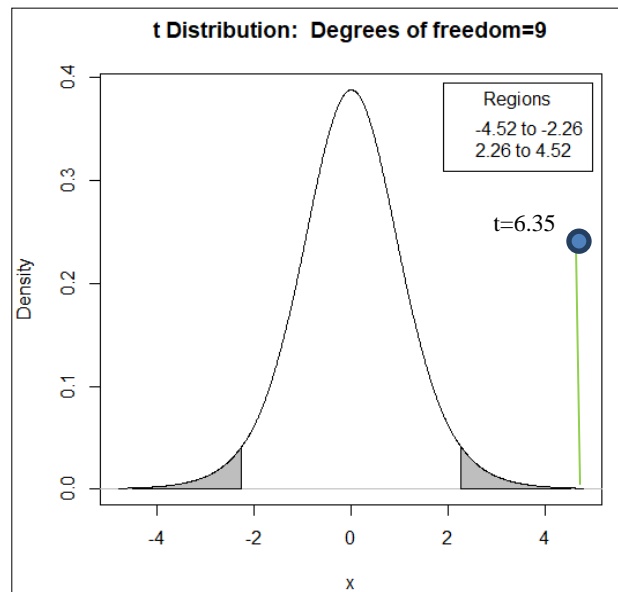


Gráfico 2-4: Prueba T-Student pareado gráfico

Realizado por: Aldas W.; Pilataxi C., 2020

El **Gráfico 2-4** se muestra el resultado obtenido de la prueba T-student mediante la campana de gauss, en el que t es igual a 6.35, lo cual indica claramente que es mayor al valor crítico de las dos colas que poseen un valor de 2.26 respectivamente, por lo tanto, no está dentro de la zona de aceptación de la hipótesis nula, y de esta manera se acepta la hipótesis alternativa.

433. *Toma de decisiones*

Al observar los resultados de los test se puede notar que existe una diferencia significativa en los tiempos de respuesta en el método manual frente al método automatizado.

En el **Gráfico 3-4** se puede notar de manera más simple que existe tal diferencia en los tiempos de respuesta. En el promedio de tiempos de la ejecución de las tareas planteadas para el análisis, se obtuvo un valor de 1.15 minutos (7.93%) en el método automatizado, frente al método manual que tomó un valor de 14.51 minutos (100%). El tiempo del método manual se considera como el total de tiempo que se demora un usuario en realizar una tarea, es decir el 100%, al comparar con el tiempo automatizado se ha obtenido que se utiliza el 7.93% del tiempo total, como conclusión se puede decir que se ha comprobado la reducción del tiempo de un 92.07% en realizar dichas tareas.

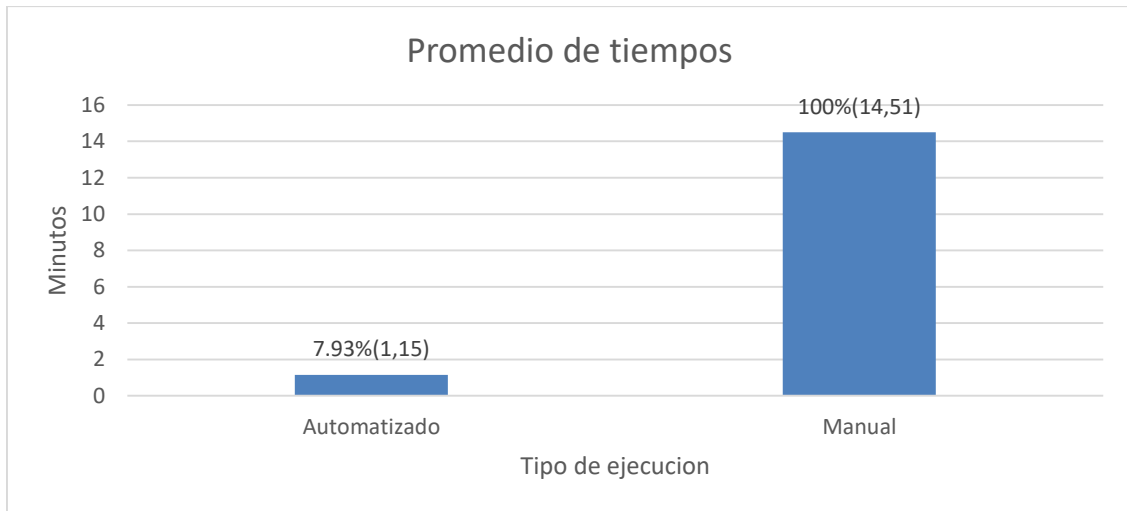


Gráfico 3-4: Tiempos de ejecución de procesos con aplicación y sin aplicación

Realizado por: Aldas W.; Pilataxi C., 2020

Los resultados obtenidos permiten ver la reducción considerable de tiempo empleado en ejecutar procesos de forma manual comparada con la automatizada, por lo que se argumenta que, con la implementación del sistema llevado a cabo en este proyecto, se ha ayudado a mejorar la eficiencia de consumo de tiempo en la realización de las tareas planteadas para este.

CONCLUSIONES

- Se ha establecido y documentado las actividades administrativas llevadas a cabo en la escuela de conducción profesional “CONDUESPOCH” mediante el análisis de las gráficas de procesos proporcionados por el personal laboral de la institución, con lo cual se ha conseguido identificar 22 procesos, mismos que están distribuidos por departamentos de la siguiente manera, 2 Procesos Académicos, 8 Procesos de Gestión de vehículos e instructores, 2 Procesos de Desarrollo y mantenimiento de aplicaciones, 2 Procesos de Gestión de Informes y notificaciones, 2 Procesos de Gestión de sílabos y docentes , 6 Procesos de Gestión de ventas y pagos, todo esto ha servido además para definir el desarrollo de los módulos planteados.
- Mediante el análisis de distintas herramientas de desarrollo web y móvil se ha podido determinar que el framework React JS y React Native respectivamente, ofrecen características que permiten desarrollar interfaces de usuario con mucha sencillez, mediante su concepto de estructuración por componentes reutilizables, mientras que para el manejo y almacenamiento de datos se ha usado Laravel con PostgreSQL, las cuales han sido seleccionadas tomando en cuenta la disponibilidad de las herramientas de soporte para entornos virtuales en la institución al momento del despliegue, lo cual se llevó a cabo sin ninguna complicación.
- Al finalizar con la recopilación y análisis de los datos referentes a los procesos llevados a cabo en CONDUESPOCH, se han determinado los requerimientos de usuario que posteriormente fueron implementados en los correspondientes módulos del sistema, mismos que para desarrollarlos se dividieron en 98 historias de usuario, 92 tareas de ingeniería, 19 historias técnicas y 597 pruebas de aceptación, lo cual se ha manejado con la metodología ágil SCRUM en 15 Sprints, dando como resultado exitoso la culminación de todo lo planificado.
- Para demostrar el uso de la metodología de pruebas que consiste en identificar, diseñar, construir, ejecutar y comparar resultados obtenidos, se han implementado las pruebas unitarias a las funcionalidades de las aplicaciones web y móvil, mediante los denominados artefactos de prueba que consisten en, el plan de pruebas con el cual se ha identificado 658 casos de prueba, de los cuales 639 pertenecen a la aplicación web y 19 a la aplicación móvil,

el siguiente artefacto que son los scripts de prueba los cuales contienen las suites de prueba y estas a su vez las condiciones a superar, fueron diseñadas y construidas con el Framework Jest, y el último artefacto que es el informe de resultados ha permitido detallar los datos obtenidos de la cobertura de código, que en promedio entre las dos aplicaciones se ha obtenido un 94% de asertividad, lo cual es muy aceptable y demuestra que se ha tenido éxito en la implementación del proceso de prueba.

- Se ha determinado el nivel de eficiencia del sistema en relación al tiempo empleado en la gestión de procesos administrativos haciendo uso del estándar 25010, dando como resultado que la aplicación implementada mejora los resultados de tiempos, y es una forma más eficiente comparado con el método manual. El tiempo que es requerido para realizar las tareas se redujo en un 92.07%, es decir, por medio del método manual se obtuvo un tiempo de 14.51 minutos (100%) y en el método automatizado se obtuvo un tiempo de 1.15 minutos (7.93%), resultado que es favorable para los usuarios del sistema.

RECOMENDACIONES

- Laravel es un Framework optimizado para la creación de APIs del lado del backend, que pueden ser usados por una gran variedad de clientes tal como se ha hecho en este trabajo, sin embargo, se recomienda NodeJs, el cual presenta algunas características adicionales como su simplicidad o el uso de Javascript para interactuar con la base de datos, lo cual permite a los desarrolladores conocer un solo lenguaje, puesto que React es una derivación de Node, no habría complicaciones en combinarlos, pero tomando en cuenta que para ponerlo en producción se debería contar con un servicio de Host que soporte este Framework.
- El desarrollo basado en componentes que ofrece tanto React JS como React Native ha permitido que los resultados obtenidos sean escalables debido a la reutilización de código, lo cual hace posible implementar más funcionalidades de forma rápida, como por ejemplo un módulo de evaluaciones a los estudiantes o el de ingreso de tareas.
- React al ser un framework para la creación de interfaces de usuario, no debería contener demasiada lógica de negocio en su desarrollo, salvo que sean de mucha importancia como es el caso de validaciones de formularios o cálculos rápidos, ya que de implementar todo en esta capa de usuario hará que el cambio de tecnología, de ser el caso, sea muy difícil, así que lo recomendable es desarrollar la lógica de negocio en el backend.
- La metodología de pruebas utilizada en este trabajo, si bien no es una manera definitiva de llevar a cabo el ciclo de pruebas, se considera como la base típica para otras metodologías, debido a que se puede agregar etapas dependiendo el tipo de pruebas a realizar, con lo cual se obtendría resultados de mejor calidad, sin embargo, se recomienda investigar sobre metodologías de pruebas de software orientadas a entornos de desarrollo de software ágil ya que estas presentan condiciones específicas para este tipo de ambientes.

GLOSARIO

Arquitectura: Estructuración del sistema que, idealmente, se crea en etapas tempranas del desarrollo. (Cervantes,2019, párr.1)

Backend: Es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione. (Arjonilla,2019, párr.3)

Burn-Down Chart: Un gráfico de trabajo pendiente a lo largo del tiempo muestra la velocidad a la que se está completando los objetivos/requisitos. (proyectosagiles,2016, párr.1)

Debugger: Es un programa que permite detectar y diagnosticar fallos en programas informáticos (ionos,2020, párr.11)

Framework: Es un patrón o esquema que ayuda a la programación a estructurar el código y a ahorrar tiempo y esfuerzos a los programadores. (fp.uoc.fje.edu,2019, párr.1)

Frontend: Es la parte del desarrollo web que se dedica a la parte frontal de un sitio web, en pocas palabras del diseño de un sitio web. (Garza, 2020, párr.1)

Funcionalidad: Es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas. (Riascos,2018, párr.3)

Hook: Es la posibilidad que le da el código de cambiar el comportamiento original de algo sin cambiar su código de la clase correspondiente. (Souvick, 2017, párr.1)

Mitigar: Minimizar el impacto del riesgo, o reducir las posibilidades de que ocurra, es también una acción válida dentro de un proceso de Gestión de Riesgos eficaz. (isotools, 2017, párr.2)

Prototipo: Es una implementación parcial pero concreta de un sistema, que se crea para explorar aspectos diversos durante el desarrollo del mismo. (Lara, 2020, párr.7)

Requerimiento: Son las necesidades de los interesados que requiere que el Sistema deba de cumplir de manera Satisfactoria. (Rodríguez, 2012, párr.2)

SCRUM: Es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. (proyectosagiles,2019, párr.1)

Sprint: Se trata de un miniproyecto de no más de un mes (ciclos de ejecución muy cortos -entre una y cuatro semanas), cuyo objetivo es conseguir un incremento de valor en el producto que estamos construyendo. (BBVA,2019, párr.1)

WebMaster: Es una persona que maneja y gestiona un sitio Web. (Arimetrics, 2018, , párr.9)

BIBLIOGRAFÍA

ACCOMAZZO, Anthony(ed.). *Fullstack React, the complete guide to ReactJS and Friends*. USA : FullStack.io,2017. 0991344626, pp. 294.

ANDRADER. *Parte 1: Introducción a Pruebas de Software | Tips de Desarrollo.* [en línea], 2021, [Consulta: 17 septiembre 2021]. Disponible en: <https://marceloandrader.github.io/blog/2021/03/23/parte-1-introducci%C3%B3n-a-pruebas-de-software/>.

ANDRADER. *Parte 2: Tipos De Pruebas de Software | Tips de Desarrollo.* [en línea], 2021, [Consulta: 17 septiembre 2021]. Disponible en: <https://marceloandrader.github.io/blog/2021/04/01/parte-2-tipos-de-pruebas-de-software/>.

ARIMETRICS. *Qué es webmaster - Definición, significado y ejemplos.* [en línea], 2018, [Consulta: 17 septiembre 2021]. Disponible en: <https://www.arimetrics.com/glosario-digital/webmaster>.

ARJONILLA. *Qué es el Backend de una web y por qué es tan importante.* Rafa Arjonilla [en línea], 2019,[Consulta: 17 septiembre 2021]. Disponible en: <https://rafarjonilla.com/que-es/backend/>.

BACK4APP. *Los diez mejores lenguajes de backend | Low-code backend to build modern apps.* *Back4App Blog* [en línea], 2021. [Consulta: 17 septiembre 2021]. Disponible en: <https://blog.back4app.com/es/los-diez-mejores-lenguajes-de-backend/>.

BBVA. *Metodología «scrum»: ¿Qué es un «sprint»? BBVA NOTICIAS* [en línea], 2019, [Consulta: 17 septiembre 2021]. Disponible en: <https://www.bbva.com/es/metodologia-scrum-que-es-un-sprint/>.

BLANCO, D. *Software para la escuela de conducción General Santander ESCUGESAN.* [En Línea],2014, [Consulta: 8 de junio de 2020]. Disponible en: <https://repository.unad.edu.co/handle/10596/20194>.

CASPERS, M. *React and Redux.* [En Línea],2016, [Consulta: 8 de junio de 2020]. Disponible en: <https://uol.de/f/2/dept/informatik/ag/svs/download/reader/reader-seminar-ws2016.pdf#page=14>.

CEROIDEAS. *LA IMPORTANCIA DE LAS PRUEBAS UNITARIAS PARA COMPROBAR FRAGMENTOS DE CÓDIGO.* *Cero Ideas* [en línea], 2020. [Consulta: 17 septiembre 2021]. Disponible en: <https://ceroideas.es/la-importancia-de-las-pruebas-unitarias-para-comprobar-fragmentos-de-codigo/>.

CERVANTES. *Arquitectura de Software.* *SG Buzz* [en línea], 2019. [Consulta: 17 septiembre 2021]. Disponible en: <https://sg.com.mx/revista/27/arquitectura-software>.

CODIGOFACILITO. *MVC (Model, View, Controller) explicado.* *CódigoFacilito* [en línea], 2017, [Consulta: 17 septiembre 2021]. Disponible en: <https://www.codigofacilito.com/articulos/mvc-model-view-controller-explicado>.

CONTINUOSDEV. *Prueba de caja blanca - Tecnología - 2021.* *continuousdev* [en línea], 2021, [Consulta: 17 septiembre 2021]. Disponible en: <https://es.continuousdev.com/3891-white-box-testing-1970>.

CREHANA. *Principales tipos de aplicaciones móviles: Qué son y para qué sirven.* <https://www.crehana.com> [en línea], 2020, [Consulta: 17 septiembre 2021]. Disponible en: <https://www.crehana.com/ec/blog/desarrollo-web/tipos-aplicaciones-moviles/>.

DELOITTE. *¿Qué es React Native?* *Deloitte Spain* [en línea], 2020, [Consulta: 17 septiembre 2021]. Disponible en: <https://www2.deloitte.com/es/es/pages/technology/articles/que-es-react-native.html>.

DESARROLLOWEB. *Qué es Redux.* [en línea], 2019, [Consulta: 17 septiembre 2021]. Disponible en: <https://desarrolloweb.com/articulos/que-es-redux.html>.

EDGAR. *Automatizar la gestión de matrícula y notas en entorno web de la Escuela de Capacitación para Conductores Profesionales del Sindicato de Chóferes del cantón Mejía parroquia Machachi.*[En Línea], 2018, [Consulta: 08 de mayo de 2020]. Disponible en: <http://www.dspace.uce.edu.ec/handle/25000/16274>.

FEWSTER, Mark. *Software Test Automation Effective use of test execution tools.* New York: Addison-Wesley,1999. 0201331403, pp. 7-17

FP.UOC.FJE.EDU. *¿Qué tipos de aplicaciones web existen?* *FP Online* [en línea], 2017. [Consulta: 17 septiembre 2021]. Disponible en: <https://fp.uoc.fje.edu/blog/que-tipos-de-aplicaciones-web-existen/>.

FP.UOC.FJE.EDU. *JavaScript, ¿para qué sirve y cómo funciona?* *FP Online* [en línea], 2019. [Consulta: 17 septiembre 2021]. Disponible en: <https://fpuocx.azurewebsites.net/blog/javascript-para-que-sirve/>.

FP.UOC.FJE.EDU. *Qué es un framework en programación.* *FP Online* [en línea], 2019, [Consulta: 17 septiembre 2021]. Disponible en: <https://fp.uoc.fje.edu/blog/que-es-un-framework-en-programacion/>.

FREECODECAMP. *Angular vs React:Cuál Elegir Para tu Aplicación.* *freeCodeCamp.org* [en línea], 2020. [Consulta: 17 septiembre 2021]. Disponible en: <https://www.freecodecamp.org/espanol/news/angular-vs-react-cual-elegir-para-su-aplicacion/>.

GARCIA,J.M.B. *Analizamos las características de la librería Axios, un ligero cliente HTTP para JavaScript.* [En Línea], 2019, [Consulta: 16 de enero de 2021]. Disponible en: <https://www.arsys.es/blog/programacion/axios/>.

GARZA, B.G. *Backend y Frontend, ¿Qué es y cómo funcionan en la programación?* [en línea],

2020,[Consulta: 16 septiembre 2021]. Disponible en: <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programación-de-una-aplicación-web>.

GEWARREN. *Características del editor de código—Visual Studio*. [En Línea],2017, [Consulta: 8 de junio de 2020]. Disponible en: <https://docs.microsoft.com/es-es/visualstudio/ide/writing-code-in-the-code-and-text-editor>.

GRISALES. *¿Qué Es Un Framework De Desarrollo Backend? Julian Grisales Full Stack Developer* [en línea].2018. [Consulta: 16 septiembre 2021]. Disponible en: https://juliangrisales.dev/que-es-un-framework-de-desarrollo-backend/?fbclid=IwAR0h8mxynXphoMqxb8nPHfHOEkw5I4_SzBw3MDyLGiwNZwjta3_-WGoCA8.

HOSTGATOR. *¿Qué es Laravel? Conoce su historia y estructura. Blog HostGator México* [en línea], 2021, [Consulta: 16 septiembre 2021]. Disponible en: <https://www.hostgator.mx/blog/que-es-laravel/>.

IBM. *IBM Docs Test running test suite*. [en línea], 2021a. [Consulta: 16 septiembre 2021]. Disponible en: <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/es/elm/6.0?topic=tests-running-test-suite>.

IBM. *IBM Docs Testing test case suite overview*. [en línea], 2021b. [Consulta: 16 septiembre 2021]. Disponible en: <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/es/elm/6.0.3?topic=testing-test-case-test-suite-overview>.

IONOS. *Depuradores: importantes herramientas para detectar errores de software. IONOS Digitalguide* [en línea], 2020, [Consulta: 16 septiembre 2021]. Disponible en: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/depurador/>.

ISO25000. *NORMAS ISO 25000*. [en línea], 2019, [Consulta: 16 septiembre 2021]. Disponible en: <https://iso25000.com/index.php/normas-iso-25000>.

ISOTOOLS. *5 acciones para un proceso de Gestión de Riesgos eficaz - Software ISO*. [en línea], 2017. [Consulta: 16 septiembre 2021]. Disponible en: <https://www.isotools.org/2017/10/08/5-acciones-proceso-de-gestion-de-riesgos-eficaz/>.

ITPEDIA. *Prueba de caja blanca bajo el microscopio - Conocimiento de TI. IT Knowledge* [en línea], 2018, [Consulta: 16 septiembre 2021]. Disponible en: <https://es.itpedia.nl/2018/02/05/white-box-testing-onder-de-loep/>.

ITPEDIA. *Pruebas de caja negra: software en el rack - Conocimiento de TI. IT Knowledge* [en línea], 2019, [Consulta: 16 septiembre 2021]. Disponible en: <https://es.itpedia.nl/2019/01/23/black-box-testing-software-op-de-pijnbank/>.

JESTJS. *Utilizando Comparadores · Jest*. [en línea], 2019, [Consulta: 16 septiembre 2021]. Disponible en: <https://jestjs.io/es-ES/docs/using-matchers>.

KHORIKOV, Vladimir. *Unit Testing: Principles, Practices, and Patterns*. New York: Manning, 2020. 1617296279, pp.8-10.

KSHIRASAGAR, Naik. *SOFTWARE TESTING AND QUALITY ASSURANCE Theory and Practice*. New Jersey: Wiley, 2008. 0471789119, pp.8-17.

LA SALLE. *Escogiendo un framework front-end | Blogs La Salle | Campus Barcelona*. [En Línea], 2020, [Consulta: 16 de enero de 2021]. Disponible en: <https://blogs.salleurl.edu/es/node/10390>

LARA. *Prototipo de interfaz de usuario: qué es y qué herramientas hay para crearlo*. Garage [en línea]. [Consulta: 16 septiembre 2021]. Disponible en: <https://ar.godaddy.com/blog/prototipo-interfaz-usuario-definicion-herramientas/>.

LENGUAJECSS. *¿Qué es CSS? - CSS en español*. [en línea], 2019, [Consulta: 16 septiembre 2021]. Disponible en: <https://lenguajecss.com/css/introduccion/que-es-css/>.

LENJUAGEJS. *¿Qué es NPM? - Javascript en español*. [en línea], 2018, [Consulta: 16 septiembre 2021]. Disponible en: <https://lenguajejs.com/npm/introduccion/que-es/>.

MICROSOFT. *Características del editor de código - Visual Studio (Windows)*. [en línea], 2020, [Consulta: 16 septiembre 2021 a]. Disponible en: <https://docs.microsoft.com/es-es/visualstudio/ide/writing-code-in-the-code-and-text-editor>.

MONOCUBET. *10 Best Front end Frameworks for Web Development in 2021*. [En Línea], 2021, [Consulta: 02 de febrero de 2021]. Disponible en: <https://www.monocubed.com/best-front-end-frameworks/>.

MORAN, T. *Implementación de un sistema para automatizar los procesos académicos y administrativos de la escuela de conducción de choferes profesionales de los ríos Manuel Bhruniss Villacres de la ciudad de Babahoyo*. [En Línea], 2012, [Consulta: 8 de junio de 2020]. Disponible en: <http://dspace.utb.edu.ec/handle/49000/625>

MORELLI, V. *¿Qué es un framework de desarrollo backend? | Wheelhub. WheelHub - Blog* [en línea]. 2019. [Consulta: 16 septiembre 2021]. Disponible en: <https://wheelhub.es/blog/que-es-un-framework-de-desarrollo-backend/>.

NEXT UNIVERSITY. *¿Qué es HTML y Para Qué Sirve? | Next University. NextU LATAM* [en línea], 2016. [Consulta: 16 septiembre 2021]. Disponible en: <https://www.nextu.com/blog/que-es-html/>.

NEXTUNIVERSITY. *¿Qué es y cómo funciona React.js? NextU LATAM* [en línea], 2016, [Consulta: 16 septiembre 2021]. Disponible en: <https://www.nextu.com/blog/que-es-y-como-funciona-react-js/>.

NTASKMANAGER. *What's the Role of Burndown Chart in the World of Scrum? nTask* [en línea], 2020, [Consulta: 16 septiembre 2021]. Disponible en: <https://www.ntaskmanager.com/blog/burndown-chart-in-scrum/>.

NUÑEZ PÉREZ, F.S. APLICACIÓN WEB PARA LA GESTIÓN AUTOMATIZADA DE LOS SERVICIOS QUE PRESTA LA ESCUELA DE CAPACITACIÓN PARA CONDUCTORES DEL SINDICATO DE CHOFERES PROFESIONALES DEL CANTÓN MOCACHE. 2018. pp. 61.

OLIVEIRA, W. *Los dos principales sistemas de gestión de procesos organizativos.* HEFLO ES [en línea], 2017, [Consulta: 16 septiembre 2021]. Disponible en: <https://www.heflo.com/es/blog/bpm/gestion-procesos-organizativos/>.

PARADIGMADIGITAL. *Test unitarios en Javascript: una pequeña introducción.* [en línea], 2018, [Consulta: 16 septiembre 2021]. Disponible en: <https://www.paradigmadigital.com/dev/test-unitarios-javascript-introduccion/>.

PMOINFORMATICA. *La Oficina de Proyectos de Informática.* [en línea], 2018, [Consulta: 16 septiembre 2021]. Disponible en: <http://www.pmoinformatica.com//2017/02/pruebas-de-caja-negra-ejemplos.htm>.

PMOINFORMATICA. *Modelo de informe de ejecución de pruebas de software.* [en línea], 2015, [Consulta: 16 septiembre 2021 a]. Disponible en: <http://www.pmoinformatica.com/2015/06/modelo-informe-pruebas-software.html>.

PMOINFORMATICA. *Pruebas de software: 10 pasos para elaborar el plan de pruebas.* [en línea], 2016, [Consulta: 16 septiembre 2021 b]. Disponible en: <http://www.pmoinformatica.com/2016/01/elaborar-plan-pruebas-software.html>.

POSTGRESQL. *PostgreSQL: ¿Qué es? Características, Ventajas y Desventajas.* [en línea], 2017, [Consulta: 16 septiembre 2021]. Disponible en: <https://hostingpedia.net/postgresql.html>.

POSTGRESQL. *PostgreSQL: Ventajas y Desventajas.* PostgreSQL [en línea], 2017, [Consulta: 16 septiembre 2021]. Disponible en: <https://postgresql2013.blogspot.com/p/ventajas-y-desventajas.html>.

PROGRAMACION EN CASTELLANO, S.L. *Comparación de frameworks para desarrollar apps para móvil híbridas.* Programación en Castellano. [en línea], 2019, [Consulta: 16 septiembre 2021]. Disponible en: http://programacion.net/articulo/comparacion_de_frameworks_para_desarrollar_apps_para_movil_hibridas_1370.

PROJECTMANAGER. *Burndown Chart: What Is It & How Do I Use It?* ProjectManager.com [en línea], 2019. [Consulta: 16 septiembre 2021]. Disponible en: <https://www.projectmanager.com/blog/burndown-chart-what-is-it>.

PROYECTOSAGILES. *Gráficos de trabajo pendiente (Burndown charts).* Proyectos Ágiles [en línea], 2008. [Consulta: 16 septiembre 2021]. Disponible en: <https://proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts/>.

PROYECTOSAGILES. *Qué es SCRUM.* Proyectos Ágiles [en línea], 2008, [Consulta: 16

septiembre 2021]. Disponible en: <https://proyectosagiles.org/que-es-scrum/>.

RIASCOS. *Funcionalidad · Atributos de calidad y patrones de diseño.* [en línea], 2018, [Consulta: 16 septiembre 2021]. Disponible en: <https://fjimenezg.gitbooks.io/atributos-de-calidad-y-patrones-de-diseno/content/funcionalidad.html>.

RJCODEADVANCE. *Patrón MVC, Modelo-Vista-Controlador + Ejemplo (Cap 4) – RJ Code Advance.* [en línea], 2018. [Consulta: 16 septiembre 2021]. Disponible en: <https://rjcodeadvance.com/patrones-de-software-patron-mvc-ejemplo-parte-4/>.

RODRIGUEZ. *Técnicas efectivas para la toma de requerimientos.* Northware [en línea], 2012, [Consulta: 16 septiembre 2021]. Disponible en: <https://www.northware.mx/blog/tecnicas-efectivas-para-la-toma-de-requerimientos/>.

SAWAKINOME. *¿Cuál es la diferencia entre las pruebas manuales y las pruebas automatizadas? / Tecnología. La diferencia entre objetos y términos similares.* [en línea], 2019, [Consulta: 16 septiembre 2021]. Disponible en: <https://es.sawakinome.com/articles/technology/what-is-the-difference-between-manual-testing-and-automated-testing.html>.

SCRUM. *Scrum: qué es y cómo funciona esta metodología.* [en línea], 2020, [Consulta: 16 septiembre 2021]. Disponible en: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>.

SCRUMGUIDE. *Scrum-Guide-ES.pdf* [en línea], 2019, [Consulta: 16 septiembre 2021]. Disponible en: <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>.

SEOSTUDIOS. *¿Qué es un framework?: Descubre todas sus ventajas.* Seoestudios [en línea], 2020, [Consulta: 16 septiembre 2021]. Disponible en: <https://www.seoestudios.es/blog/que-es-un-framework/>.

SISTEMAWEB. *Que es un sistema web.* [En Línea], 2016, [Consulta: 17 de septiembre de 2020]. Disponible en: <https://sistemaweb.pe/que-es-un-sistema-web>

SOUVIK. *Software Development: What is a Hook? Let's Know About Few Types of Hook.* RS Web Solutions [en línea], 2017, [Consulta: 16 septiembre 2021]. Disponible en: <https://www.rswebsols.com/tutorials/programming/software-development-hook-hooking>.

SUMATD. *Los mejores lenguajes backend que debes probar en 2019.* [En Línea], 2019, [Consulta: 16 de agosto de 2020]. Disponible en: <https://sumatd.com/blog/mejores-lenguajes-backend-2019/>.

TAJAMAR. *Material UI / Caching en React.* TechClub Tajamar [en línea], 2020. [Consulta: 16 septiembre 2021]. Disponible en: <https://techclub.tajamar.es/material-ui-caching-en-react/>.

TESTINGCOLOMBIA. *Casos de prueba que son como se hacen y para qué sirven.* [En Línea], 2016, [Consulta: 08 de mayo de 2020]. Disponible en: <https://www.testingcolombia.com/casos-de-prueba-que-son-como-se-hacen-y-para-que-sirven/>.

TIJERO, P. *¿Qué es Redux y cómo funciona? Get on Board Dev* [en línea], 2019,[Consulta: 16 septiembre 2021]. Disponible en: <https://medium.com/get-on-board-dev/qu%C3%A9-es-redux-y-c%C3%B3mo-funciona-11466487a032>.

TRIGAS. *GESTION DE PROYECTOS INFORMÁTICOS METODOLOGÍA SCRUM.* [en línea], 2019, [Consulta: 16 septiembre 2021]. Disponible en: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>.

TUSPLUS. *8 cosas que hacen de Jest el mejor Framework de prueba de React. Code Envato Tuts+* [en línea], 2020. [Consulta: 16 septiembre 2021]. Disponible en: <https://code.tutsplus.com/es/tutorials/8-things-that-make-jest-the-best-react-testing-framework--cms-30534>.

UTPL. *PLAN DE Pruebas DE Software - Fundamentos de programación - UTPL. StuDocu* [en línea], 2018, [Consulta: 16 septiembre 2021]. Disponible en: <https://www.studocu.com/ec/document/universidad-tecnica-particular-de-loja/fundamentos-de-programacion/plan-de-pruebas-de-software/5638932>.

V&V. *Diferencia entre verificación y validación.* [en línea], 2020, [Consulta: 16 septiembre 2021 a]. Disponible en: <https://es.gadget-info.com/difference-between-verification>.

VARDY, Adam. *SCRUM! THE ULTIMATE BEGINNERS GUIDE TO MASTERING SCRUM TO BOOST PRODUCTIVITY & BEAT DEADLINES.* USA: CreateSpace Independent Publishing Plataform, 2016. 1530076552, pp. 20-23.

VISCARDI, Stacia. *The profesional ScrumMaster's Handbook.* USA: Packt Publishing,2013. 1849688028, pp. 23-24

WYSE, M. *REACT: A Framework for Rapid Exploration of Approximate Computing Techniques.* [En Línea],2015, [Consulta: 8 de junio de 2020]. Disponible en: <https://homes.cs.washington.edu/~wysem/website/publications/react-wax15.pdf>

YEEPLY. *¿Qué son las pruebas unitarias y cómo llevar una a cabo? Yeeply* [en línea], 2019. [Consulta: 16 septiembre 2021]. Disponible en: <https://www.yeeply.com/blog/que-son-pruebas-unitarias/>.