



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS,
ELECTRÓNICA E INDUSTRIAL

CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES
E INFORMÁTICOS

TEMA:

IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PROGRESIVA PARA LA GESTIÓN DE PRUEBAS DE SIMULACIÓN PARA EL INGRESO A INSTITUCIONES MILITARES Y UNIVERSIDADES EN EL CENTRO DE CAPACITACIÓN Y NIVELACIÓN ACADÉMICA SMARTEL.

Trabajo de Titulación Modalidad: Proyecto de Investigación presentado previo la obtención del Título de Ingeniero en Sistemas Informáticos y Computacionales.

LÍNEA DE INVESTIGACIÓN:

Desarrollo de Software

AUTOR: Oñate Calderón, Wilmer Rolando

TUTOR: Ing. Clay Fernando Aldás Flores, Mg.

Ambato – Ecuador

Octubre, 2020

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PROGRESIVA PARA LA GESTIÓN DE PRUEBAS DE SIMULACIÓN PARA EL INGRESO A INSTITUCIONES MILITARES Y UNIVERSIDADES EN EL CENTRO DE CAPACITACIÓN Y NIVELACIÓN ACADÉMICA SMARTEL, desarrollado bajo la modalidad Proyecto de Investigación por el señor Wilmer Rolando Oñate Calderón, estudiante de la Carrera de Ingeniería Sistemas Informáticos y Computacionales, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, octubre 2020.



Firmado electrónicamente por:
**CLAY FERNANDO
ALDAS FLORES**

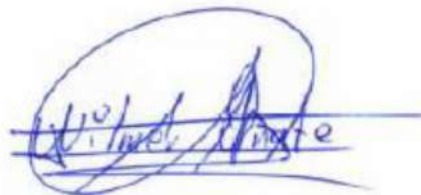
Ing. Clay Aldas

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado: IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PROGRESIVA PARA LA GESTIÓN DE PRUEBAS DE SIMULACIÓN PARA EL INGRESO A INSTITUCIONES MILITARES Y UNIVERSIDADES EN EL CENTRO DE CAPACITACIÓN Y NIVELACIÓN ACADÉMICA SMARTEL es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, octubre 2020.



Wilmer Rolando Oñate Calderón

C.C. 0605179969

AUTOR

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Wilmer Rolando Oñate Calderón estudiante de la Carrera de Ingeniería Sistemas Informáticos y Computacionales de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PROGRESIVA PARA LA GESTIÓN DE PRUEBAS DE SIMULACIÓN PARA EL INGRESO A INSTITUCIONES MILITARES Y UNIVERSIDADES EN EL CENTRO DE CAPACITACIÓN Y NIVELACIÓN ACADÉMICA SMARTEL, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, octubre 2020.



Firmado electrónicamente por:
**ELSA PILAR
URRUTIA**

Ing. Pilar Urrutia, Mg.
PRESIDENTA DEL TRIBUNAL



Firmado electrónicamente por:
**HERNAN FABRICIO
NARANJO AVALOS**

Ing. Hernán Naranjo
PROFESOR CALIFICADOR



Firmado electrónicamente por:
**OSCAR FERNANDO
IBARRA TORRES**

Ing. Fernando Ibarra
PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, octubre 2020.



Wilmer Rolando Oñate Calderon

C.C. 0605179969

AUTOR

DEDICATORIA

A mi madre Azucena Calderón por el apoyo incondicional brindado a lo largo de mi vida. Por sus cuidados, cariños, educación y valores impartidos que han formado y marcado mi vida, ya que sin ella no hubiera logrado esta meta en mi vida profesional. A mi padrastro Mesías Paredes, a mis hermanos y hermanas en especial a Carmen Oñate por su apoyo incondicional, consejos y compañía en todos los momentos importantes de mi vida.

A mi padre Ernesto Oñate (q.e.p.d.) por el tiempo que estuviste conmigo compartiendo tu amor y consejos.

Gracias.

Wilmer Rolando Oñate Calderón

AGRADECIMIENTO

Quiero expresar mi gratitud a Dios, quién con su bendición llena siempre mi vida y a toda mi familia, por ser el apoyo en aquellos momentos de dificultad y debilidad.

Gracias a mi madre y a mi padrastro: Azucena Calderón y Mesías Paredes por ser los motores de mi vida, por sus consejos, valores y principios que se me ha inculcado.

Agradezco a los docentes de la Carrera en Ingeniería en Sistemas Computacionales e Informáticos y en especial a mi tutor el Ing. Clay Aldas, por haber compartido sus conocimientos y haberme guiado en este largo proceso.

Muchas Gracias.

Wilmer Rolando Oñate Calderón.

ÍNDICE

APROBACIÓN DEL TUTOR	ii
AUTORÍA.....	iii
APROBACIÓN TRIBUNAL DE GRADO	iv
DERECHOS DE AUTOR	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
CAPÍTULO I	1
EL PROBLEMA.....	1
1.1. Tema de Investigación	1
1.2. Planteamiento del Problema	1
1.3. Delimitación	2
2.1.1. De Contenidos.....	2
2.1.2. Espacial.....	2
2.1.3. Temporal	2
1.4. Justificación.....	2
1.5. Objetivos	3
2.1.4. Generales	3
2.1.5. Específicos.....	3
CAPÍTULO II	4
MARCO TEÓRICO	4
2.1. Antecedentes Investigativos	4
2.2. Fundamentación Teórica	5
2.2.1. Aplicaciones Web.....	5
2.2.1.1. JavaScript	6
2.2.1.2. TypeScript	6
2.1.2.3. Single Page Application (SPA)	7
2.1.2.4. PWA (Progressive Web Apps - Aplicaciones Web Progresivas)	7
2.2.2. Patrones	10
2.2.3. Modelo Vista Controlador (MVC)	11
2.2.4. Modelo Vista Modelo de Vista (MVVM)	13
2.2.5. Tecnologías de Frond-End.....	13
2.2.5.1. Angular	14
2.2.5.2. React	14
2.2.5.3. Vue	15

2.2.6.	Tecnologías de Back-End.....	16
2.2.6.1.	NODE JS.....	16
2.2.6.2.	.NET CORE	16
2.2.7.	Metodologías de Desarrollo de Software	17
CAPÍTULO III	20
METODOLOGÍA	20
3.1.	Modalidad de la Investigación	20
3.2.	Recolección de Información.....	20
3.3.	Procesamiento y Análisis de Datos	20
3.4.	Desarrollo del Proyecto.....	20
CAPÍTULO IV	22
DESARROLLO DE LA PROPUESTA	22
4.1.	Metodologías Ágiles en el Desarrollo de Aplicaciones Web Progresivas	22
4.1.1.	Metodología XP (Extreme Programming – Programación extrema)	22
4.1.1.1.	Ventajas de la Metodología XP	25
4.1.1.2.	Desventajas de la Metodología XP.....	25
4.1.2.	Metodología Scrum.....	25
4.1.2.1.	Ventajas de Scrum	27
4.1.2.2.	Desventajas de Scrum	27
4.1.3.	Crystal Methodologies	27
4.1.3.1.	Ventajas de Crystal Methodologies	28
4.1.3.2.	Desventajas de Crystal Methodologies.....	28
4.1.4.	Características de Metodologías Ágiles	29
4.1.5.	Comparativa de Metodologías Ágiles	29
4.1.6.	Metodología Elegida	31
4.2.	Comparativa entre Tecnologías de Front-End	31
4.2.1.	Tecnología Front-End Elegida	36
4.3.	Comparación entre Tecnología de Back-End	37
4.3.1.	Comparación entre ASP.NET CORE y NODE.JS.....	37
4.3.2.	Tecnología de Back-End Elegida.....	41
4.4.	Arquitectura de la Aplicación Web Progresiva	41
4.5.	Desarrollo de la Metodología	42
4.5.1.	Roles del Proyecto.....	42
4.5.2.	Fase: Planificación del Proyecto.....	43
4.5.3.	Diagrama de Procesos.....	43
4.5.3.1.	Historias de Usuario	43

4.5.3.2.	Tareas	50
4.5.3.3.	Plan de Entrega del Proyecto	63
4.5.4.	Fase: Diseño	63
4.5.4.1.	Tarjetas CRC (Clase-Responsabilidad-Colaboración)	63
4.5.4.2.	Diseño de la Base de Datos	70
4.5.5.	Fase: Codificación.....	70
4.5.4.1.	Desarrollo del Servidor.....	70
4.5.4.2.	Desarrollo del Cliente.....	80
4.5.4.3.	Desarrollo de Iteraciones	90
4.5.5.	Fase: Prueba.....	103
4.5.5.1.	Pruebas Unitarias	103
4.5.5.2.	Pruebas de Aceptación	104
CAPÍTULO V		111
CONCLUSIONES Y RECOMENDACIONES.....		111
5.1.	Conclusiones	111
5.2.	Recomendaciones	112
MATERIAL DE REFERENCIA		113
Bibliografía		113
Glosario de Términos y Acrónimos		116

ÍNDICE DE FIGURAS

Figura 1.1: Diferencias entre metodologías ágiles y no ágiles.	19
Figura 4.2: Tendencias de búsqueda de Angular, Vue y React.....	32
Figura 4.3: Estadísticas de descargas NPM(Angular ,Vue, React).	33
Figura 4.4: Estadísticas de preguntas de Stack Overflow.	33
Figura 4.5: Rendimiento de .NET y NODE JS.....	38
Figura 4.6: Framework más usados en el año 2020	38
Figura 4.7: Lenguajes más usados en el año 2020.....	39
Figura 4.8: Tendencias de búsqueda entre .NET CORE y NODE JS.	39
Figura 4.9: Arquitectura de la Aplicación Web Progresiva.	42
Figura 4.10: Flujo de procesos de negocios del Centro de Capacitación y Nivelación Académica SMARTEL.....	45
Figura 4.11: Diseño de la Base de Datos parte 1.	71
Figura 4.12: Diseño de la Base de Datos parte 2.	72
Figura 4.13: Diseño de la Base de Datos parte 3.	73
Figura 4.14: Creación del modelo mediante Entity Framework Core	70
Figura 4.15: Clases creadas por Entity Framework Core.	74
Figura 4.16: Parte del código de la clase SmartelContext.	75
Figura 4.17: Creación de un proyecto angular mediante Angular CLI.....	80
Figura 4.18: Creación de una PWA en base a un proyecto existente.....	80
Figura 4.19: Configuración del archivo package.json.	81
Figura 4.20: Configuración del archivo package.json.	83
Figura 4.21: Referencia al archivo manifest en el archivo index.html.	83
Figura 4.22: Importación del módulo ServiceWorkerModule.....	84
Figura 4.23: Componente de presentación del módulo representante.....	86
Figura 4.24: Clase RepresentanteComponent.....	86
Figura 4.25: Uso de CanActive en appmodule.ts.....	88
Figura 4.26: Gestión de asignaturas.	91
Figura 4.27: FromGrop de la clase asignatura	92
Figura 4.28: Validación de código único correspondiente al módulo asignatura.	92
Figura 4.29: Pantalla de edición de un curso.....	93
Figura 4.30 : Validación de código único correspondiente la clase Curso Detalle.	94
Figura 4.31: Ingreso de un estudiante.....	94
Figura 4.32: Validación de estudiante, profesor y representante único al seleccionar una Persona	95
Figura 4.33: Ingreso de preguntas en el repositorio.	96
Figura 4.34: Etiqueta ngKatex-paragrah.....	96
Figura 4.35: Ingreso de fórmulas matemáticas	96
Figura 4.36: Confirmación para crear un usuario	97
Figura 4.37: Clase EmailSender.....	97
Figura 4.38: Pantalla de visualización de cursos.....	98
Figura 4.39: Asignación de preguntas a una evaluación.	99
Figura 4.40: Clase Opciones de una evaluación.	99
Figura 4.41: Revisión de evaluaciones por estudiante.	99
Figura 4.42: Casos de pruebas para el mantenimiento de asignatura.	105

ÍNDICE DE TABLAS

Tabla 4.1: Características de las metodologías ágiles XP, SCRUM Y CRYSTAL METHODOLOGIES.	30
Tabla 4.2: Comparativa entre Angular, React y Vue.....	35
Tabla 4.3: Ventajas y desventajas de Angular	35
Tabla 4: Ventajas y desventajas de React.....	36
Tabla 4.5 : Ventajas y desventajas de Vue	36
Tabla 6: Ventajas y desventajas de NODE JS	40
Tabla 7: Ventajas y desventajas de ASP.NET CORE	40
Tabla 4.8: Comparativa entre ASP.NET CORE y NODE JS.....	41
Tabla 4.9: Roles del proyecto a desarrollar	43
Tabla 4.10: Historia de Usuario- Inicio de sesión.....	44
Tabla 4.11: Historia de Usuario- Gestión de Asignatura.	44
Tabla 4.12: Historia de Usuario- Gestión de Módulos.....	44
Tabla 4.13: Historia de Usuario- Gestión de Cursos.	44
Tabla 4.14: Historia de Usuario- Gestión de Títulos Académicos.....	46
Tabla 4.15: Historia de Usuario- Gestión de Instituciones Educativas.	46
Tabla 4.16: Historia de Usuario - Gestión de Personas.	46
Tabla 4.17: Historia de Usuario - Gestión de Representantes.....	46
Tabla 4.18: Historia de Usuario - Gestión de Profesores.....	47
Tabla 4.19: Historia de Usuario - Gestión de estudiantes.	47
Tabla 4.20: Historia de Usuario - Gestión de tema de una pregunta.	47
Tabla 4.21: Historia de Usuario - Gestión de preguntas.....	48
Tabla 4.22: Historia de Usuario - Creación de usuarios.....	48
Tabla 4.23: Historia de Usuario - Gestión de Evaluaciones.	48
Tabla 4.24: Historia de Usuario - Revisión de evaluaciones.	48
Tabla 4.25: Historia de Usuario - Iniciar Evaluación.	49
Tabla 4.26: Historia de Usuario - Reporte de horario de clases.	49
Tabla 4.27: Historia de Usuario - Reporte de evaluaciones.	49
Tabla 4.28: Historia de Usuario - Ficha de Inscripción del estudiante.	49
Tabla 4.29: Diseño de las interfaces para la gestión de asignaturas.....	50
Tabla 4.30: Implementación de la gestión de asignaturas.	50
Tabla 4.31: Diseño de las interfaces para la gestión de módulos.	50
Tabla 4.32: Implementación de la gestión de módulos.	51
Tabla 4.33: Diseño de las interfaces para la gestión de cursos.	51
Tabla 4.34: Implementación de la gestión de módulos.....	51
Tabla 4.35: Asignación de asignaturas y profesores por curso.	51
Tabla 4.36: Asignación de horario de clases.....	52
Tabla 4.37: Diseño de las interfaces para la gestión de títulos académicos.	52
Tabla 4.38: Implementación de la gestión de títulos académicos.	52
Tabla 4.39: Diseño de las interfaces para la gestión de instituciones educativas.....	53
Tabla 4.40: Implementación de la gestión de instituciones educativas.....	53
Tabla 4.41: Diseño de las interfaces para la gestión de personas.....	53
Tabla 4.42: Implementación de la gestión de personas.....	53

Tabla 4.43: Asignación de teléfonos por persona.	54
Tabla 4.44: Diseño de las interfaces para la gestión de representantes.....	54
Tabla 4.45: Implementación de la gestión de representantes.....	54
Tabla 4.46: Diseño de las interfaces para la gestión de profesores.....	54
Tabla 4.47: Implementación de la gestión de profesor.....	55
Tabla 4.48: Implementación de la gestión de asignaturas por profesor.....	55
Tabla 4.49: Implementación de la asignación de títulos académicos.....	55
Tabla 4.50: Diseño de las interfaces para la gestión de estudiantes.....	55
Tabla 4.51: Implementación de la gestión de estudiantes.....	56
Tabla 4.52: Asignación de representantes por estudiante.....	56
Tabla 4.53: Implementación de la asignación de instituciones educativas.....	56
Tabla 4.54: Asignación de asignaturas por estudiante.....	56
Tabla 4.55: Diseño de las interfaces para la gestión de tema de una pregunta.....	57
Tabla 4.56: Implementación de la gestión de tema de una pregunta.....	57
Tabla 4.57: Diseño de las interfaces para la gestión de preguntas.....	57
Tabla 4.58: Implementación de la gestión de preguntas.....	57
Tabla 4.59: Diseño de las interfaces para el ingreso de fórmulas matemáticas.....	57
Tabla 4.60: Diseño de las interfaces para la gestión de usuarios.....	58
Tabla 4.61: Implementación de la creación de usuarios.....	58
Tabla 4.62: Implementación del reinicio de clave.....	58
Tabla 4.63: Implementación de recuperación de clave.....	58
Tabla 4.64: Implementación de cambio de clave.....	59
Tabla 4.65: Diseño de la interfaz para inicio de sección.....	59
Tabla 4.66: Implementación de cambio de clave.....	59
Tabla 4.67: Diseño de las interfaces para la gestión de Evaluaciones.....	59
Tabla 4.68: Diseño de las interfaces para la gestión de Evaluaciones.....	60
Tabla 4.69: Diseño de las interfaces para la gestión de evaluaciones.....	60
Tabla 4.70: Diseño de las interfaces para la gestión de evaluaciones.....	60
Tabla 4.71: Implementación de la acción de agregar una pregunta.....	60
Tabla 4.72: Activar una pregunta.....	60
Tabla 4.73: Diseño de las interfaces para la revisión de evaluaciones.....	61
Tabla 4.74: Implementación de revisión de evaluaciones.....	61
Tabla 4.75: Diseño de las interfaces para iniciar una evaluación.....	61
Tabla 4.76: Implementación de evaluaciones.....	61
Tabla 4.77: Implementación de reporte de horario de clases.....	62
Tabla 4.78: Implementación del reporte de evaluaciones.....	62
Tabla 4.79: Implementación del reporte de evaluaciones consolidadas.....	62
Tabla 4.80: Implementación de la ficha de inscripción.....	62
Tabla 4.81: Tarjeta CRC - Asignatura.....	63
Tabla 4.82: Tarjeta CRC - Módulo.....	63
Tabla 4.83: Tarjeta CRC - Representante.....	65
Tabla 4.84: Tarjeta CRC - Curso.....	65
Tabla 4.85: Plan de entrega del proyecto.....	64
Tabla 4.86: Tarjeta CRC – Título Académico.....	65
Tabla 4.87: Tarjeta CRC – Institución Educativa.....	65
Tabla 4.89: Tarjeta CRC - Pregunta.....	65
Tabla 4.90: Tarjeta CRC - Persona.....	65
Tabla 4.91: Tarjeta CRC – Teléfono persona.....	66

Tabla 4.92: Tarjeta CRC - Estudiante.	66
Tabla 4.93: Tarjeta CRC - Profesor.....	66
Tabla 4.94: Tarjeta CRC – Profesor - Asignatura.....	66
Tabla 4.95: Tarjeta CRC – Profesor – Título Académico.	66
Tabla 4.96: Tarjeta CRC – Curso Detalle.	66
Tabla 4.97: Tarjeta CRC - Horario.	67
Tabla 4.98: Tarjeta CRC – Estudiante - Institución.	67
Tabla 4.99: Tarjeta CRC – Estudiante - Representante.....	67
Tabla 4.100: Tarjeta CRC – Estudiante – Curso Detalle.	67
Tabla 4.101: Tarjeta CRC - Evaluación.	67
Tabla 4.102: Tarjeta CRC - Pregunta.....	67
Tabla 4.103: Tarjeta CRC - Evaluación.	68
Tabla 4.104: Tarjeta CRC – Evaluación - Pregunta.....	68
Tabla 4.105: Tarjeta CRC – Curso Vista.....	68
Tabla 4.106: Tarjeta CRC – Estudiante - Evaluación.	68
Tabla 4.107: Tarjeta CRC – Pregunta - Estudiante.....	68
Tabla 4.108: Tarjeta CRC – Evaluación - Consolidado.	68
Tabla 4.109: Tarjeta CRC – Evaluación Consolidado - Detalle.	68
Tabla 4.110: Tarjeta CRC – Evaluación Consolidado - Estudiante.	69
Tabla 4.111: Tarjeta CRC – Pregunta Consolidado.	69
Tabla 4.112: Tarjeta CRC - Account.	69
Tabla 4.113: Tarjeta CRC – Estado Civil.	69
Tabla 4.114: Tarjeta CRC – Instrucción Académica.	69
Tabla 4.115: Tarjeta CRC – Operadora Telefónica.....	69
Tabla 4.116: Tarjeta CRC – Tipo Identificación.....	69
Tabla 4.117: Tarjeta CRC – Tipo Pregunta.	69
Tabla 4.118: Tarjeta CRC – Tipo Teléfono.....	70
Tabla 4.119: Librerías utilizadas en el proyecto de Angular.....	90
Tabla 4.120: Iteración1.....	91
Tabla 4.121: Iteración 2.	93
Tabla 4.122: Iteración 3.	95
Tabla 4.123: Iteración 4.	100
Tabla 4.124: Casos de pruebas para el mantenimiento de asignatura.	104
Tabla 4.125: Prueba listar pregunta.	105
Tabla 4.126: Prueba agregar pregunta.....	105
Tabla 4.127: Prueba editar pregunta.....	106
Tabla 4.128: Prueba eliminar pregunta.....	106
Tabla 4.129: Prueba activar pregunta.	106
Tabla 4.130: Prueba cambiar clave.....	107
Tabla 4.131: Prueba reiniciar clave.....	107
Tabla 4.132: Prueba recuperar clave.....	107
Tabla 4.133: Prueba listar evaluaciones.	108
Tabla 4.134: Prueba agregar evaluación.	108
Tabla 4.135: Prueba editar evaluación.	108
Tabla 4.136: Prueba eliminar evaluación.	109
Tabla 4.137: Prueba activar pregunta.	109
Tabla 4.138: Prueba agregar pregunta a una evaluación.....	109
Tabla 4.139: Prueba iniciar evaluación.....	110

RESUMEN EJECUTIVO

Existe un gran número de tecnologías que facilita la creación de aplicaciones web, pero en los últimos años han surgido las PWA (Progressive Web Apps - Aplicaciones Web Progresivas), que es una tecnología que combina lo mejor de las aplicaciones web y las aplicaciones nativas, consiguiendo que una aplicación web se comporte como una aplicación nativa.

El presente proyecto describe la creación de una PWA encargada de realizar los procesos de gestión de pruebas de simulación para el ingreso a instituciones mliares y universidades en el Centro de Capacitación y Nivelación Académica SMARTEL, con la finalidad de dejar a un lado las hojas de cálculos y cuadernillos que utilizan hasta la fecha.

El proyecto está desarrollado mediante la metodología XP y el patrón de arquitectura MVC haciendo uso de los framework ASP.NET CORE 3.1.1 en el lado del servidor y Angular 8 en el lado del cliente.

ASP .NET CORE facilitó la creación del entorno de desarrollo del lado del servidor mediante sus herramientas como son Entity Framework y ASP .NET CORE Identity, por otro lado, Angular 8 facilitó la creación de la Aplicación Web Progresiva a través de su librería @angular/pwa.

Palabras clave: Aplicaciones Web Progresivas, Metodología XP, NET CORE, Angular

ABSTRACT

There are a large number of technologies that facilitate the creation of web applications, but in recent years PWA (Progressive Web Apps) has emerged, which is a technology that combines the best of web applications and native applications, getting a web application to behave like a native application.

This project describes the creation of a PWA to carry out the simulation test management processes for admission to institutions and universities in the SMARTEL Academic Training and Leveling Center, in order to streamline their processes and set aside the spreadsheets and booklets they use to date.

The project is developed using the XP methodology and MVC architecture pattern using the frameworks ASP.NET CORE 3.1.1 on the server side and Angular 8 on the side of the client.

ASP .NET CORE facilitates the creation of the server-side development environment using its tools such as Entity Framework and ASP .NET CORE Identity, on the other hand, Angular 8 facilitates the creation of the Progressive Web Application through its @angular/pwa library.

Keywords: Progressive Web Applications, XP Methodology, NET CORE, Angular

CAPÍTULO I

EL PROBLEMA

1.1. Tema de Investigación

“Implementación de una Aplicación Web Progresiva para la gestión de pruebas de simulación para el ingreso a Instituciones Militares y Universidades en el Centro de Capacitación y Nivelación Académica SMARTEL.”

1.2. Planteamiento del Problema

La competitividad originada entre las diferentes empresas ha llevado a buscar maneras de brindar mejores servicios y ser más competitivas en el mercado, es así, que en los sistemas informáticos se ha visto una gran oportunidad de mejorar la calidad de los servicios prestados, reducir tiempos, conseguir mercados y generar mayores ganancias [1].

En los últimos años los diferentes centros de capacitación a nivel nacional han tenido un gran incremento en su mercado, principalmente por la incorporación de pruebas para el ingreso a las diferentes universidades del país por parte del SNNA (Sistema Nacional de Nivelación y Admisión).

En la actualidad en el cantón Quero de la provincia de Tungurahua los diferentes centros de capacitaciones para el ingreso a universidades o institutos militares sus procesos son llevados de una forma manual, utilizando hojas de cálculos para llevar la información de los estudiantes que capacitan. De igual manera para la realización de simulaciones se utilizan cuadernillos o aplicaciones gratuitas, en el caso de estas últimas, no se ajustan a los requerimientos o nivel de enseñanza de los centros de capacitación, ocasionando un desbalance entre lo enseñado y lo evaluado. En este contexto, el empleo de cuadernillos se presenta como la opción más viable, pero esto con lleva una inversión en tiempo, recursos y personal. Sin embargo, este método de gestión imposibilita la capacidad de tabular resultados y la toma de decisiones.

Al tomar en cuenta estos antecedentes, el Centro de Capacitación y Nivelación Académica SMARTEL tienen como objetivo mejorar la calidad de sus servicios, reducir tiempos y recursos en el proceso de ejecución de pruebas de simulación implementando una aplicación web que soporte al proceso y a su vez permita la

gestión de pruebas de simulación, con el fin de obtener información de las simulaciones realizadas a los estudiantes, analizar dicha información y tomar decisiones en los procesos de enseñanza que el centro de capacitación brinda.

1.3. Delimitación

2.1.1. De Contenidos

Línea de Investigación: Desarrollo de Software.

Sub-línea de Investigación: Aplicación Web.

2.1.2. Espacial

La presente investigación se desarrollará en el Centro de Capacitación y Nivelación Académica SMARTEL ubicado en el cantón Quero.

2.1.3. Temporal

La presente investigación tendrá un período de duración de 4 meses la misma que se realizará en el período académico mayo 2020 – septiembre 2020.

1.4. Justificación

En el ciclo académico 2019-2020 el Ministerio de Educación informó la incorporación de 149.050 bachilleres únicamente en el régimen Sierra – Amazonía [2]. De este porcentaje de bachilleres un gran número optan por el ingreso a una de las diferentes universidades en el Ecuador o a una institución militar. La alta demanda de acceso a los centros de estudios de tercer nivel e instituciones militares obliga a los estudiantes pasar por un período de preparación previo para rendir y aprobar sus exámenes de ingreso. Los centros de capacitación y nivelación académica brindan estos servicios, en los cuales los estudiantes son evaluados constantemente mediante pruebas y test de simulación para evaluar sus avances y progresos.

Al momento existen diferentes páginas web y aplicaciones móviles que son de uso gratuito o fueron desarrolladas por los centros de capacitación y nivelación académica, en las cuales se pueden evaluar el conocimiento actual de los estudiantes en temáticas específicas. Las aplicaciones gratuitas que se pueden encontrar en la

web y apoyan el proceso de evaluación, estas son limitadas pues al ser versiones libres no cuentan con todas las funcionalidades deseadas y estos tipos de evaluación no pueden ser acoplados a la cadena de valor de información de las instituciones, en la mayoría de los casos.

Considerando estos antecedentes, el Centro de Capacitación y Nivelación Académica SMARTEL ha optado por el desarrollo de una PWA que facilite la gestión de pruebas de simulación, permitiendo actualizar constantemente las pruebas según sus necesidades y planes de enseñanza, obtener información sobre las simulaciones realizadas, analizar la información y tomar decisiones respecto a sus procesos de enseñanza. Además, teniendo en cuenta la necesidad del cliente de acceder a su información desde diferentes dispositivos, se presenta la necesidad de incorporar una PWA que evite la construcción de una o más aplicaciones nativas.

Con este recurso informático el Centro de Capacitación y Nivelación Académica SMARTEL podrá mejorar la entrega de servicios, ganar margen competitivo, disminuir tiempos de trabajo y sobre todo dejar a un lado los procesos manuales que se utilizan actualmente.

1.5. Objetivos

2.1.4. Generales

Implementar una Aplicación Web Progresiva para la gestión de pruebas de simulación para el Centro de Capacitación y Nivelación Académica SMARTEL.

2.1.5. Específicos

- Analizar la información y procesos que el Centro de Capacitación y Nivelación Académica SMARTEL utiliza para la realización de pruebas de simulación.
- Determinar una metodología ágil que se adapte al desarrollo de Aplicaciones Web Progresivas.
- Seleccionar la tecnología más adecuada de Front-End y Back-End para el desarrollo de la Aplicación Web Progresiva.
- Implementar la Aplicación Web Progresiva para la gestión de pruebas de simulación en el Centro de Capacitación y Nivelación Académica SMARTEL.

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes Investigativos

Entre los diferentes proyectos e investigaciones realizadas en los últimos años a nivel nacional e internacional podemos mencionar las siguientes:

El estudio realizado por Edward Washington Rubio Guerrero y Marcela Georgina Gómez Zermeño en el año 2016 titulado “Propuesta de diseño de un modelo educativo integral para capacitaciones corporativas” en el cual se expone que la tecnología y el uso de aplicaciones en los diferentes procesos de capacitación son necesarios y fundamentales para el aprendizaje, donde los estudiantes son los protagonistas. En una encuesta realizada, como parte de la propuesta, el 90% de profesores manifiestan que deben utilizarse diferentes tecnologías en los procesos de capacitación [1].

En la investigación “Construcción y Validación de un Sistema Web/Móvil como Herramienta de Intervención Tecnológica para la Evaluación Dinámica en Asignatura de Grado en Ingeniería en Sistemas de Información” describe la creación de un sistema cuestionario On-Line para el repaso, entrenamiento y puesta a prueba de los diferentes temas y conceptos desarrollados previamente en clases. Además, el sistema permite realizar un seguimiento y control continuo, en el cual, los docentes podrán obtener alertas tempranas de temas o conceptos de difícil aprendizaje. El sistema se llevó a cabo mediante un aplicativo web y móvil, este último dirigido únicamente para la resolución de cuestionarios. Esta investigación concluye que el sistema desarrollado ayuda a los estudiantes a prepararse y realizarse una autoevaluación previo a una evaluación y ayuda al docente a generar y analizar la información acerca del desempeño y aprendizaje de los estudiantes [3].

En la investigación “La aplicación SOCRATIVE como herramienta de evaluación y precursor de la participación en el aula”, desarrollada en el año 2017 en la cual se realiza la incorporación de una herramienta gratuita SOCRATIVE como un sistema de respuestas para el estudiante. Los docentes realizan preguntas con diferentes configuraciones donde se determina el grado de conocimientos adquiridos en las aulas por los estudiantes, permitiendo a los docentes adquirir información en tiempo

real de los temas o conceptos necesarios a estudiar o realizar una retroalimentación. Los resultados obtenidos en la investigación fueron positivos ya que los estudiantes manifestaron una experiencia agradable, el deseo de alargar las secciones y de tratar con preguntas más concretas de los temas tratados [4].

En la investigación titulada “Diseño de un software de evaluaciones en línea para los estudiantes y docentes de grado de la Universidad Tecnológica Empresarial de Guayaquil” concluye que el desarrollo de un sistema de evaluación en línea tiene un impacto positivo en los diferentes procesos que se llevan a cabo en la institución, ya que mejora el proceso de evaluación y disminuye el tiempo de trabajo al docente, permitiéndole dedicar este tiempo a actividades académicas o de investigación [5].

Se han realizado diferentes estudios y proyectos en los cuales se ha aplicado las PWA (Progressive Web Apps - Aplicaciones Web Progresivas) por las ventajas que brindan, entre los diferentes trabajos podemos mencionar el “Desarrollo de un sistema web progresivo para el control y toma de lecturas de agua potable” desarrollado por Marcillo Matute y William Alex en el cual se usa las PWA debido a que se puede incluir sistemas web y sistemas para dispositivos móviles en una sola aplicación web [6].

2.2. Fundamentación Teórica

2.2.1. Aplicaciones Web

Las aplicaciones web son herramientas informáticas que poseen funcionalidades específicas y se adaptan a las necesidades del usuario, son accedidas desde un navegador a través de internet o una red local [7].

Las aplicaciones web se distinguen de las aplicaciones de escritorio tradicional en que en vez de implementar la interfaz de usuario utilizando un lenguaje particular, se utilizan páginas web como punto de acceso a las aplicaciones [8].

Ventajas de las Aplicaciones Web:

- No necesitan ningún tipo de instalación, ya que se puede acceder a ellas a través de un navegador [9].

- Es multiplataforma y multidispositivos, Esto significa que nos podemos olvidar de que software tiene cada dispositivo que accede, y que puede acceder igualmente un ordenador, una tablet, un smartphone [10].
- El rendimiento de la aplicación no esta limitada por el dispositivo que se accede, ya que el rendimiento esta determinado por el servidor en el cual se encuentra alojada la aplicación [10].
- La aplicación puede estar en la nube, con lo que sería accesible para cualquier ordenador con acceso a internet [10].
- Adaptable, visualmente intuitiva y facil de realizar mantenimiento de ser necesario [10].

2.2.1.1. JavaScript

JavaScript fue desarrollado por Netscape, es un lenguaje de programación dinámico que soporta construcción de objetos basados en prototipos. Las capacidades dinámicas de JavaScript incluyen construcción de objetos en tiempo de ejecución, listas variables de parámetros, variables que pueden contener funciones, creación de scripts dinámicos, introspección de objetos y recuperación de código fuente [11].

2.2.1.2. TypeScript

TypeScript es un lenguaje de programación de código abierto desarrollado por Microsoft, el cual cuenta con herramientas de programación orientada a objetos. TypeScript convierte su código en JavaScript común. Es llamado también Superset de JavaScript, lo que significa que, si el navegador está basado en JavaScript, este nunca llegará a saber que el código original fue realizado con TypeScript y ejecutará el JavaScript como lenguaje original [7],[12].

La principal característica de TypeScript es el tipado estático [12]. Decimos que un lenguaje es de tipado estático cuando cumple con estas características principales:

- Las variables tienen un tipo de dato.
- Los valores sólo se pueden asignar a variables del tipo correspondiente.

Las aplicaciones web modernas no solo tienen mayores expectativas de los usuarios y mayores demandas, sino que también se espera que estén disponibles 24/7 desde cualquier parte del mundo y accesible desde cualquier dispositivo o tamaño de pantalla. Las aplicaciones web deben ser seguras, flexible y escalable para satisfacer

el impulso de la demanda, y capaz de manejar escenarios complejos y experiencias de usuario enriquecidas [10].

Existen dos enfoques generales para construir aplicaciones web hoy en día: aplicaciones web tradicionales que realizan la mayor parte de la lógica de la aplicación en el servidor, y aplicaciones de una sola página que realizan la mayor parte de la lógica de la interfaz de usuario en un navegador web, comunicándose con el servidor web principalmente por una web api [11].

2.1.2.3. Single Page Application (SPA)

La mayoría de las aplicaciones web en la actualidad utilizan SPA, que es una aplicación web que tienen un único punto de entrada, generalmente el archivo index.html. Todas las pantallas se muestran en la misma página, sin recargar el navegador como cualquier otra aplicación de escritorio [8].

En SPA todos los componentes como CSS (Cascade Style Sheets – Hojas de Estilo en cascada), imágenes, scripts y otros recursos requeridos se cargan al mismo tiempo en la carga inicial de la página y posteriormente el contenido, los componentes se cargan dinámicamente dependiendo de la interacción con el usuario. Cuando el usuario posee cargado una versión inicial, cada solicitud tomará menos tiempo en actualizar esa parte o región en lugar de recargar una página o aplicación completa. Las SPA permiten tratar los datos de una forma más flexible, mientras mejora la experiencia del usuario con interfaces interactivas [11].

2.1.2.4. PWA (Progressive Web Apps - Aplicaciones Web Progresivas)

Las PWA impulsadas por Google, surgen como una opción más simplificada para los desarrolladores y los medios de comunicación a la hora de implementar una aplicación [13].

Las PWA son el nuevo estándar en la era moderna del desarrollo web, son creadas con las mismas tecnologías web pero se comportan como una aplicación híbrida [14]. Aunque ya existían, se dan a conocer en mayor medida en el año 2015, de la mano de la diseñadora Frances Berriman y del ingeniero de Google Chrome, Alex Russell, quienes mencionan a las PWA con el objetivo de describir a las aplicaciones que utilizan service workers en los navegadores modernos.

Las PWA une lo mejor de las tecnologías web y aplicaciones nativas, no requieren instalación y están disponibles en un navegador. Presenta varias ventajas como es: carga rápida, envío de notificaciones, acceso directo en la pantalla principal, modo offline [8]. Usan las características de los navegadores para proporcionar una experiencia cercana a las aplicaciones nativas. Asimismo, facilita al usuario tener a la mano las PWA sin pasar por el proceso de visitar una tienda de aplicaciones.[10]

Service Workers

El service worker son eventos que se ejecutan en segundo plano en una aplicación, actúan como un proxy entre la red y la aplicación, por lo cual puede interceptar solicitudes de red creadas por la página web y retornar la respuesta de la red o de un almacenamiento local ,es por ello, que el service worker provee la capacidad a las PWA de imitar la carga instantánea y actualizaciones regulares de una aplicación nativa. También puede otorgarle funcionamiento en un escenario sin conexión al almacenar localmente el app shell y otros recursos después de la primera carga de la página [8],[11].

El service worker está registrado en la visita de la primera página del usuario, consiste en un archivo JavaScript que incorpora ganchos de ciclo de vida para la lógica empresarial y control de caché. El service workerses responsable de almacenar en caché todos los archivos, servir notificaciones push, actualizar contenido, manipulación de datos [14].

Manifest File

Manifest file o archivo manifiesto es un archivo de tipo JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) en el cual se especifica las características de inicio que poseerá la Aplicación Web Progresiva desde la presentación hasta la interacción con el usuario [8].

El propósito del archivo de manifiesto es exponer ciertas configuraciones modificables a los desarrolladores de aplicaciones. Al proporcionar un extracto de datos en formato JSON, es posible almacenar en caché esta información con la ayuda del service worker y luego usar app shell para cargar reglas CSS. El service worker almacena en caché el extracto JSON y físicamente en la shell de la aplicación. Este shell de aplicación es un contenedor de autocontenido que tiene todas las hojas de estilo, scripts, imágenes, fuentes y salidas HTML (HyperText

Markup Language) necesarias para que un usuario presente una página completa [14].

Beneficios del manifest file para las Aplicaciones Web Progresivas:

- Ejecuta la aplicación en modo de pantalla completa, sin la barra URL (Uniform Resource Locator - Localizador de Recursos Uniforme).
- Mejora la experiencia de usuario.
- Ejecución de la aplicación desde la URL y desde la pantalla de inicio.

Responsive Web Desing

Es un conjunto de técnicas que permiten que una interfaz se adapte a diferentes dispositivos de forma automática [10].

App Shell

El grupo de Fundamentos web de Google define el app shell como el mínimo HTML, CSS y JavaScript que alimenta una interfaz de usuario, almacenado en caché y mostrando contenido dinámico, a medida que los datos se extraen de las API (Application Programming Interface - Interfaz de Programación de Aplicaciones) externas [14].

La app Shell se encarga de darle un enfoque similar que las apps nativas a las Aplicaciones Web Progresivas permitiendo una carga confiable y al instante de las pantallas [8].

Características

Las PWA enrola un conjunto de características que se describen a continuación:

- **Progresiva:** Funcionan para cualquier usuario en cualquier navegador.
- **Adaptable:** Se adaptan a cualquier pantalla móviles, tablet o PC.
- **Independiente de conectividad:** Puede trabajar con redes que posean una mala calidad o sin conexión mediante el uso de Service Workers.
- **Segura:** Trabaja bajo el protocolo HTTPS(HyperText Transfer Protocol Secure,-Protocolo de transferencia de hipertexto seguro) con lo cual garantiza que el contenido no sea manipulado por terceras personas.
- **Vinculable:** Permite compartir fácilmente mediante la URL.

- **Posibilidad de volver interactuar:** Facilidad de Interacción mediante la implementación de notificaciones push.

2.2.2. Patrones

Los patrones son una disciplina de resolución de problemas que han surgido con mayor énfasis en la comunidad de orientación de objetos, aunque puede ser utilizada en cualquier ámbito de la informática o la ciencia en general [15]. Los patrones de diseños tuvieron relevancia a partir del año de 1990 tras la publicación del libro “Design Patterns” escrito por el grupo “Gang of Four (GoF)”, compuesto por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, en el cual se recopila 23 patrones de diseños comunes [13].

Los patrones este compuesto por una regla de 3 partes que expresan una relación entre un contexto, un problema y una solución.

- **Contexto:** Describe la situación actual del problema dando inicio al diseño del problema [15].
- **Problema:** Representa el problema recurrente que nace a partir del contexto dado, comienza con su especificación general, determinando el problema en concreto que se debe resolver y tratando de balancear sus fuerzas [15].
- **Solución:** Muestra la manera de resolver un problema recurrente o como balancear las fuerzas asociadas a él [15].

Una buena definición de un patrón permite entenderlo inmediatamente, y además provee todos los detalles necesarios para implementarlo y considerar las consecuencias de su aplicación [15].

Categorías de Patrones

Los patrones se agrupan en función de un rango similar de abstracción en tres categorías: patrones arquitectónicos, patrones de diseño e idioms [15].

- **Patrones Arquitectónicos**

Proveen un conjunto de subsistemas predefinidos, especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos [15].

- **Patrones de Diseño**

Un patrón de diseño provee un esquema para refinar componentes de un sistema de software y la forma en que se relacionan entre sí. Describe una estructura generalmente recurrente de comunicación de componentes que resuelve un problema de diseño general dentro de un contexto particular [15]. Los patrones de diseño solucionan los problemas basándose en sus 4 objetivos principales: Proporcionar catálogos de componentes reutilizables para el diseño de sistemas de software, estandarizar una terminología común entre diseñadores, estandarizar la manera en que se realiza el diseño y facilitar el aprendizaje de futuros diseñadores integrando un conocimiento ya existente [13].

- **Idioms**

Los Idioms están relacionados con la implementación de diseño de problemas particulares. Un Idiom es un patrón de bajo nivel específico para un lenguaje de programación que describen cómo resolver problemas de implementación específicos en un lenguaje de programación determinado [15].

2.2.3. Modelo Vista Controlador (MVC)

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de presentación de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Fue diseñado en base a las ideas de reutilización de código y la separación de conceptos para facilitar el desarrollo de aplicaciones y su posterior mantenimiento. [13]

El modelo MVC propone la construcción de 3 componentes: el modelo, la vista y el controlador.

- **El modelo:** Es el conjunto de clases encargado de representar la información con que trabaja el usuario [13].
- **La Vista:** Las vistas son las delegadas de representar gráficamente el modelo y de brindar las operaciones de los controladores para que el usuario pueda interactuar con el sistema diseñado [13].
- **El controlador:** Es un módulo o una sección intermedia de código, que hace de intermediario de la comunicación entre el modelo y la vista, y unifica la

validación. Organiza la interacción entre las vistas y el modelo recibiendo peticiones del usuario, interactúa con el modelo realizando consultas y modificaciones a este, decide qué vista se muestra como respuesta y le proporciona los datos requeridos para su renderizado, o delega la respuesta a otra acción de otro controlador [13].

Ventajas:

- **Múltiples vistas del mismo modelo:** Múltiples vistas pueden ser implementadas y usadas como un simple modelo [15].
- **Vistas Sincronizadas:** El mecanismo de propagación de cambios asegura que todos los observadores registrados son notificados de los cambios en los datos de la aplicación. Esto sincroniza todas las vistas y controladores dependientes [15].
- **Cambiabilidad:** Al ser el modelo independiente de todo el código de la interfaz de usuario, exportar una aplicación MVC a una nueva plataforma no afectaría la funcionalidad central de la aplicación, solo es necesario la implementación conveniente para esa plataforma, de los componentes vistas y controladores [15].

Desventajas:

- **Incremento de complejidad:** Al seguir estrictamente la estructura MVC, el uso excesivo de su estructura puede ocasionar un aumento de la complejidad sin ganar mucha flexibilidad [15].
- **Excesivos números de actualización:** Si una sola acción del usuario implica varias actualizaciones, el modelo debe pasar por alto las notificaciones intermedias innecesarias para evitar un consumo de recursos innecesarios [15].
- **Acoplamiento de vistas y controladores con un modelo:** Los componentes vista y controlador hacen llamadas directas al modelo. Esto implica que cambios en la interfaz del modelo probablemente rompan el código de vista y controlador [15].

- **Conexión entre vistas y controladores:** Controladores y vistas son componentes separados, pero estrechamente relacionados que impiden su rehusó individual [15].

2.2.4. Modelo Vista Modelo de Vista (MVVM)

En la década de los 80 el Modelo Vista Modelo de vista fue introducido para superar las limitaciones del Modelo MVC (Modelo Vista Controlador) y aprovechar sus fortalezas [16]. El patrón MVVM es una variación del patrón de arquitectura MVC, está diseñado para plataformas de desarrollo de interfaz de usuario modernas donde la vista es responsabilidad de un diseñador en lugar de un desarrollador [9]. El modelo MVVM igual que el modelo MVC se basa en la separación de conceptos aumentando la calidad del código [16].

El modelo MVVM introduce un nuevo objeto denominado “Modelo de vista” (ViewModel) el cual tiene la lógica para preparar los datos para la Vista [16].

- **Modelo:** Se encarga del encapsulamiento de la lógica de negocios y datos. Para maximizar las oportunidades de reutilización, los modelos no deben contener ningún caso de uso o conducta específica del usuario o lógica de la aplicación [9].
- **Vista:** Encapsula la interfaz de usuario y la lógica de la interfaz del usuario. Las vistas definen la interfaz de usuario específica para una parte de la aplicación [9].
- **Modelo de Vista:** Encapsula la lógica de presentación y estado. Lógica de presentación se define como la lógica de la aplicación que se ocupa de casos de uso de la aplicación y define el comportamiento lógico y la estructura de la aplicación. Para maximizar las oportunidades de reutilización, el ViewModel no debe tener ninguna referencia a las clases específicas de interfaz de usuario, elementos, controles o comportamiento [9].

2.2.5. Tecnologías de Frond-End

Los Framework son estructuras de código genérico y reusable que ayuda a los desarrolladores en las tareas de programación con el fin de aumentar la productividad.

2.2.5.1. Angular

Angular es un framework de JavaScript basado en TypeScript, desarrollado y mantenido por Google, lanzado originalmente como AngularJS, fue creado en el año de 2008 por los empleados de Google Misko Hevery y Adam Abrons, en aquel entonces se conocía como AngularJs [17],[14].

Angular permite la creación de aplicaciones web de una sola página del lado del cliente con herramientas que simplifican y aceleran la creación de contenido web dinámico y de aplicaciones de una sola página, enlace de datos, enrutamiento de vistas, la inyección de dependencias y el enrutamiento [18].

En el año 2016 Angular fue completamente reescrito por el equipo de desarrolladores original y nombrado Angular 2, lo que elimina la mayor parte de la compatibilidad con versiones anteriores de AngularJs. Muchos de los conceptos centrales del framework cambiaron mientras que AngularJS se centró en ámbitos y controladores como patrón de arquitectura, Angular 2 se basa completamente en una jerarquía de componentes [17], [18].

Angular, como muchos otros frameworks está basado en componentes, que son los principales bloques de construcción. La mejor práctica sugiere que los componentes consisten en tres archivos separados: un archivo HTML para plantilla, un archivo CSS para diseño y un archivo TS(TypeScript) para el controlador [17]. Los proyectos en angular consisten en NgModules, los NgModules consisten principalmente en componentes y archivos de servicio. Los componentes manejan la lógica de negocios de la aplicación y los servicios son paquetes de código que contienen código reutilizable y común [18].

Angular usa TypeScript que es un superconjunto de JavaScript fuertemente tipado. El tipado reduce el riesgo de conversión accidental de tipo de datos en el código y facilita la capacidad de prueba ya que cualquier intento de asignar un valor de tipo de datos incorrecto a una variable de tipo diferente causa un error [18].

2.2.5.2. React

React es una biblioteca JavaScript declarativa y basada en componentes que puede ser usada para construir interfaces de usuario, desarrollada por Facebook y

publicado como software de código abierto en el año 2013 para la creación de interfaces de usuario para la web [17].

El núcleo de React está hecho de componentes los cuales pueden escribirse bajo dos enfoques diferentes: componentes como funciones y componentes como clases ES6.

Componentes como funciones: Son funciones puras que devuelven exactamente un `ReactElement`, el nombre del componente es también el nombre de la función. Sin embargo, este enfoque tiene sus limitaciones: ni el estado puede modificarse ni los métodos del ciclo de vida [17].

Componentes como clases ES6: El nombre del componente está representado por el nombre de la clase. La clase siempre tiene que extender la superclase `React.Component` [17]. Los desarrolladores de React recomiendan usar los componentes de la función tanto como sea posible. Eso debería permitir una reutilización eficiente de componentes simples.

2.2.5.3. Vue

Vue a menudo se describe como un marco progresivo que se puede utilizar para construir interfaces de usuario para la web. Si bien no está estrictamente asociado con el Modelo-Vista-Patrón de modelo de vista (MVVM), los principios de diseño de Vue se inspiraron en parte en eso. Fue lanzado por primera vez en febrero 2014 por el ex empleado de Google llamado Evan You como un marco progresivo [17].

Vue es un framework basado en componentes, la biblioteca principal se centra solo en la capa de vista y es fácil de recoger e integrar con otras bibliotecas o proyectos existentes.

Según Evan You, considera que Vue es más similar con React, ya que su idea central es el enlace de datos y los componentes. En comparación con React, Vue pone más énfasis en la experiencia del usuario, lo que facilita su aprendizaje, si el usuario conoce los conceptos básicos: HTML, JavaScript y CSS.

Desde su lanzamiento, Vue ha sido todo un éxito, especialmente dado el hecho de que está desarrollado y mantenido por un equipo de docenas de desarrolladores. Vue es utilizado por Alibaba, Baidu, Expedia, Nintendo, GitLab y otros [14].

2.2.6. Tecnologías de Back-End

2.2.6.1. NODE JS

Node.js es un entorno de tiempo de ejecución de JavaScript que utiliza el motor de Google Chrome V8 para proporcionar un buen rendimiento, usa programación asíncrona controlada por eventos y especialmente diseñada para servicios en la red [19].

Node.js utiliza un modelo de entrada / salida asíncrona sin bloqueo lo que le permite administrar de mejor manera los recursos y el rendimiento de las aplicaciones, usa un solo hilo para las diferentes solicitudes e implementa un bucle de eventos que evita un bloqueo de entrada/salida, además, Node.js es un enfoque de eventos basado entrada/salida evitando realizar bloqueos permitiéndole manejar todas las solicitudes del servidor, además brinda una instalación fácil por medio del administrador de paquetes NPM (Node Package Manager – Administrador de Paquetes Node) [20].

Express

Express es el framework más popular de Node.js que proporciona un marco flexible para la creación de aplicaciones web. Express brinda funciones robustas y limpias por lo que la creación de aplicaciones web es más fácil en Express que en módulos nativos de Node.js [20].

2.2.6.2. .NET CORE

.NET Core es un framework gratuito y de código abierto. Es multiplataforma de código abierto construido desde cero. El proyecto es desarrollado principalmente por Microsoft y lanzado bajo la licencia MIT. Por encima de .NET CORE se construyen diferentes bibliotecas especializadas en tareas concretas.

ASP.NET CORE

Es un marco de desarrollo de código abierto y multiplataforma de alto rendimiento para la creación de aplicaciones basadas en la nube y conectadas a internet, entre las cuales destaca servicios web, aplicaciones IoT [21].

Los principales beneficios que se obtiene con la utilización de ASP.NET Core son muchos, pero a continuación menciono los más relevantes [21]:

- Solicitudes HTTP (HyperText Transfer Protocol - Protocolo de transferencia de hipertexto) ligeras, modulares y de alto rendimiento.
- Incorporación de inyección de dependencias.
- Integración de marcos de trabajo modernos al lado del cliente.
- De código abierto.
- Capacidad para desarrollar y ejecutar en Windows, macOS y Linux.
- Simplifica el desarrollo web.

Las aplicaciones creadas pueden hospedarse en apache, IIS (Internet Information Service - Servicio de Información de Internet), Nginx (Engine x), etc.

2.2.7. Metodologías de Desarrollo de Software

El uso de las metodologías de software es de gran importancia para cumplir los objetivos en la creación o modificación de software de calidad. Es importante conocer la definición de metodología y desarrollo antes de definir que es una metodología de desarrollo.

Metodología se deriva de tres vocablos griegos: metà(“más allá”), odós(“camino”), y logos(“estudio”). Metodología es el conjunto de métodos coherentes y relacionados por principios comunes”. El desarrollo es incrementar, agrandar, ampliar o aumentar características de algo físico. Por lo tanto la metodología de desarrollo es el estudio y determinación de métodos adecuados para dar incremento a algo [22].

Las metodologías de desarrollo de software deben poseer los siguientes requisitos [23]:

- Visión del producto.
- Vinculación con el cliente.
- Establecer un modelo de ciclo de vida.
- Gestión de los requisitos.
- Plan de desarrollo.
- Integración del proyecto.
- Medidas de progreso del proyecto.
- Métricas para evaluar la calidad.
- Maneras de medir el riesgo.

- Como gestionar los cambios.
- Establecer una línea de meta.

Clasificación de Metodologías de Desarrollo

Las metodologías de desarrollo de software se pueden dividir en dos grupos las metodologías tradicionales que posee una disciplina de trabajo sobre el proceso de desarrollo de software y las metodologías ágiles que son más adaptivas que predictivas.

Metodologías Tradicionales:

Imponen una disciplina de trabajo sobre el proceso de desarrollo de software con el fin de generar un software eficiente y predecible. Hace hincapié en la planificación total del trabajo a realizar, una vez realizado este proceso comienza con el ciclo de desarrollo del software, además son consideradas por ser rígidas y dirigidas por la documentación que se genera en cada una de las actividades desarrolladas [24].

Entre las diferentes metodologías tradicionales podemos mencionar:

- RUP (Rational Unified Process)
- MSF (Microsoft Solution Framework)
- Win-Win Spiral Model
- Iconix

Metodologías Ágiles:

Febrero del año 2001 tras una reunión en Utah-EEUU nace el término ágil como una alternativa a las metodologías tradicionales como reacción a la pesadez, lentitud y exceso de documentación de las metodologías existentes y a la expansión de la red, las aplicaciones web y móviles como al crecimiento del open source. Tras esta reunión se creó la organización “The Agile Alliance” promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos [25].

Las metodologías ágiles se centran en capturar los requisitos cambiantes y la gestión de riesgos donde su principal objetivo es la reducción de tiempo, generando un producto completo, entregable e incremental, donde el producto se construye bloque a bloque durante todo el ciclo de vida del desarrollo de software.

Entre las diferentes metodologías ágiles podemos mencionar [22]:

- Agile Modeling
- Crystal Methods
- Dynamic Systems development methods
- Pragmatic programming
- Scrum
- Extreme Programming
- Evolutionary Project Management
- Story cards driven development
- Agile Unified Process

Diferencias entre Metodologías Ágiles y Tradicionales

La diferencia que poseen las dos metodologías se presenta por los diferentes enfoques y objetivos diferentes. Las metodologías ágiles son adaptativas más que predictivas. Las metodologías tradicionales potencian la planificación detallada de prácticamente todo el desarrollo software a largo plazo. Las metodologías ágiles proponen procesos que se adaptan y progresan con el cambio [24].

Las metodologías ágiles están orientadas al personal más que orientadas al proceso [24].

A continuación, en la tabla 1 se describe las diferencias entre las metodologías ágiles y las metodologías tradicionales:

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Figura 1.1: Diferencias entre metodologías ágiles y no ágiles.

Fuente: [23]

CAPÍTULO III

METODOLOGÍA

3.1. Modalidad de la Investigación

La investigación se desarrollará basándose en dos modalidades de investigación:

- Investigación Bibliográfica -Documental:

Se realizará una investigación bibliográfica - documental ya que se basará en artículos relacionados publicados al rededor del mundo, de los cuales se tomará información relevante, metodologías, procesos con los cuales se está desarrollando la Aplicación Web Progresiva, además de utilizarse para la construcción del marco teórico del presente proyecto.

- Investigación Aplicada:

Se realizará una investigación aplicada ya que se emplearán los conocimientos obtenidos a lo largo de los años para así lograr alcanzar los objetivos propuestos para el desarrollo del presente proyecto.

3.2. Recolección de Información

Para la recolección de información se utilizará las siguientes técnicas:

Técnica documental: Se considera una técnica documental porque se utilizará diferentes fuentes bibliográficas como son artículo, proyectos de investigación de los distintos repositorios, con el objetivo de conocer y profundizar el tema establecido.

3.3. Procesamiento y Análisis de Datos

El primer paso que se realizar será la recopilación de información sobre la metodología y la tecnología de Back-End y Front-End a utilizarse. De igual manera se analizará los diferentes módulos a desarrollar que conformaran el aplicativo web progresivo juntamente con sus procesos para así poder cumplir con los objetivos planteados anteriormente.

3.4. Desarrollo del Proyecto

Para el desarrollo del presente proyecto se considerará lo siguiente:

Las actividades por realizarse para el cumplimiento del objetivo “Analizar la información y procesos que el centro de capacitación y nivelación académica SMARTEL utiliza para la realización de pruebas de simulación” son:

- Recolectar información y datos de los diferentes procesos para la realización de las pruebas de simulación.
- Analizar la información obtenida para la realización de las pruebas de simulación.

El cumplimiento del objetivo “Determinar una metodología ágil que se adapte al desarrollo de Aplicaciones Web Progresivas” se realizara mediante las siguientes actividades:

- Investigar las diferentes metodologías ágiles para el desarrollo de Aplicaciones Web Progresivas.
- Escoger una metodología ágil que permita el desarrollo de Aplicaciones Web Progresivas o se adapte a su desarrollo.

El objetivo “Seleccionar la tecnología más adecuada de Front-End y Back-End para el desarrollo de la Aplicación Web Progresiva” se llevará a cabo mediante las siguientes tareas:

- Investigar las diferentes tecnologías para el desarrollo de Aplicaciones Web Progresivas.
- Seleccionar la tecnología más adecuada de Front-End y Back-End para el desarrollo de Aplicaciones Web Progresiva.

Para la realización del objetivo “Implementar la Aplicación Web Progresiva para la gestión de pruebas de simulación en el centro de capacitación y nivelación académica SMARTEL” se deberá cumplir las siguientes actividades:

- Elaboración de la Aplicación Web Progresiva.
- Realización de pruebas de funcionamiento.

CAPÍTULO IV

DESARROLLO DE LA PROPUESTA

4.1. Metodologías Ágiles en el Desarrollo de Aplicaciones Web Progresivas

El uso de metodologías ágiles en el desarrollo de la aplicación web esta denotado por los factores de riesgo que presenta en el uso de metodologías tradicionales tales como la rigidez en la gestión de cambios, tiempo de desarrollo extenso, gran cantidad de documentación, poca comunicación con el usuario [26]. Además las metodologías ágiles son aplicables y beneficiosas, debido a que cumplen con la mayor parte de características establecidas en el desarrollo de aplicaciones web como es obtener el producto final en el menor tiempo posible, obtener versiones del preliminares previo a la obtención del producto final, se lleve a cabo mediante un proceso de desarrollo adaptable a los cambios, documentación no extensa [24].

El desarrollo de Aplicaciones Web Progresivas no está limitada al desarrollo de una metodología ágil en específico, por lo que se deberá determinar una metodología ágil que se adapte a las necesidades y características del desarrollo de la Aplicación Web Progresiva. Antes de determinar la metodología ágil que se adapte al desarrollo de la Aplicación Web Progresiva se realizará una descripción de las metodologías ágiles seleccionadas para posteriormente compararlas y realizar la selección de la metodología ágil adecuada.

Las metodologías ágiles seleccionadas para realizar la comparación son: Programación Extrema, Scrum y Crystal Methodologies, las cuales fueron seleccionadas teniendo en cuenta los beneficios, popularidad y requerimientos del desarrollo.

4.1.1. Metodología XP (Extreme Programming – Programación extrema)

Programación extrema o XP es una metodología ágil que fue desarrollada por Kent Beck, se centra en potenciar las relaciones interpersonales dentro del equipo de trabajo, promueve el trabajo en equipo, el aprendizaje de los desarrolladores y proporciona un buen clima de trabajo. Se basa en la realimentación continua entre cliente y el equipo de desarrollo, una comunicación fluida entre todos los

participantes. Se define como una metodología adecuada para proyectos donde los requisitos son imprecisos y cambiantes y poseen un alto riesgo técnico [27], [28].

La metodología XP mejora un proyecto de software de cuatro maneras esenciales como es:

- **Comunicación:** El equipo de desarrollo se comunica constantemente con sus clientes y entre ellos. El cliente es considerado parte del equipo de desarrollo [27].
- **Simplicidad:** El equipo de trabajo se encarga de la autoorganización con el fin de mantener un diseño simple y limpio. No se utilizan recursos para la realización de actividades complejas, solo se realiza lo que el cliente demanda de la forma más sencilla posible [29],[27].
- **Retroalimentación:** La retroalimentación se realiza en dos sentidos, por parte del equipo de trabajo a los clientes, con el fin de brindar información sobre la evolución de sistema y desde el cliente hacia el equipo de trabajo en los aportes de construcción del proyecto [27].
- **Valentía:** El equipo de trabajo debe estar preparado para enfrentarse a los diferentes cambios que pueden originarse en el transcurso del proyecto, mediante la retroalimentación pueden actuar valientemente a los diferentes cambios tanto de requisitos y de tecnologías, y poder exponer los problemas o dudas que se posea [27].

Fases de la Metodología XP

La metodología XP está definida por cuatro fases que se detallan a continuación:

Planeación

En esta fase se definen los requisitos que permite generar una idea del contexto del negocio y recoger las principales características y funcionalidades, con esto se genera las historias de usuarios que describen lo que se va a realizar en el software [29].

Las historias de usuario son escritas conjuntamente por el cliente, teniendo en cuenta todos los requisitos necesarios y generando una prioridad para las mismas. Después, el personal encargado analizará cada historia de usuario asignándole un

costo. Este costo está medido en semanas, si es necesario se puede mandar a descomponer en historias de usuarios más pequeñas de ser necesario y cada historia de usuario no puede durar más de tres semanas [29].

Diseño

En XP se realiza la fase del diseño bajo el pensamiento de siempre realizarlo de forma sencilla, el diseño se encarga de guiar la implementación de las historias de usuario tal y como se describen. XP usa las tarjetas CRC (clase-responsabilidad-colaborador) como un mecanismo eficaz para pensar en el software en un contexto orientado a objetos. Las tarjetas CRC son el único producto del diseño que se genera [29].

El diseño se debe realizar tanto antes y después que comience la codificación, es decir el diseño se realiza de manera continua mediante se construye el sistema.

Codificación

Después que se lleva acabo las historias de usuario y el diseño preliminar se desarrolla una serie de pruebas unitarias de las historias de usuario. Después de realizar las pruebas unitarias el desarrollador está concentrado en lo que se debe implementar. Al finalizar la implementación de las historias de usuario en código se debe aplicar pruebas unitarias [29].

XP recomienda que dos personas trabajen juntas con el objetivo de crear código para una historia de usuario con el fin de solucionar problemas de forma rápida y asegurar la calidad [29].

Pruebas

La creación de pruebas unitarias es un elemento clave en el enfoque que posee XP, se deben realizar de manera que sea posible automatizarlas para poder realizarlas varias veces y con facilidad.

Las pruebas de aceptación XP o también conocidas como pruebas del cliente se centran en las características y funcionalidades generales del sistema que son visibles y revisables por el cliente [29].

4.1.1.1. Ventajas de la Metodología XP

La metodología XP ofrece una serie de ventajas, entre las cuales podemos recalcar las siguientes [30]:

- Permite disminuir la tasa de error, permite corregir los diferentes errores antes de generar nuevas funcionalidades al sistema.
- El proceso de integración es continuo, logrando integrar todo el trabajo con mayor facilidad.
- Permite satisfacer las necesidades del usuario con más exactitud.
- Desarrolla un código simple y fácil de entender.
- Permite realizar modificaciones de una manera fácil.
- Genera un equipo de trabajo motivado y existe una mayor integración entre el equipo de trabajo.

4.1.1.2. Desventajas de la Metodología XP

Entre las desventajas que posee la metodología XP podemos mencionar las siguientes [30]:

- Es recomendable la implementación solo en proyectos a corto plazo.
- No es aconsejable para empresas que producen software masivo.
- Los usuarios pueden no desear frecuentes versiones del software.
- No siempre puede resultar más fácil que el desarrollo tradicional.
- Imposible de prever futuros errores antes de programas.

4.1.2. Metodología Scrum

Scrum se define como un proceso empírico, iterativo e incremental de desarrollo, diseñado para la colaboración eficaz del equipo de trabajo empleando un conjunto de reglas y roles para la generación de una estructura de correcto funcionamiento [28], [31].

La metodología scrum define tres roles principales, los cuales son: El Scrum master, product owner y el equipo de desarrollo o team. El Scrum master es el líder del equipo y se encarga de que todo el equipo cumpla las reglas y procesos de la metodología. El Product owner es el cliente o la persona que representa a los clientes. El team es el grupo de desarrolladores encargado de convertir la lista de requerimientos en funcionalidades del software [31].

Scrum hace uso de sprint que corresponde a una etapa de trabajo generando una versión utilizable del producto, es decir, cada sprint es considerado como un proyecto individual o mini proyecto. Un sprint está formado por diferentes elementos, que se describen a continuación [29]:

Reunión de planeación del Sprint

Es un plan de tiempo variable en el que colabora todo el equipo Scrum para determinar el trabajo a realizarse en el sprint. La reunión de planificación del Sprint consta de dos partes, donde se define: que se va a entregar en el próximo Sprint y como se va a realizar el trabajo [29].

En conclusión, la principal función de la reunión de planificación del sprint es definir el objetivo del Sprint.

Scrum diario

El Scrum diario permite identificar y eliminar obstáculos mediante una reunión de 15 minutos, donde el equipo de desarrollo sincroniza actividades y establece un plan para las próximas 24 horas. El Scrum diario aporta comunicación entre interesados y promueve la toma de decisiones rápidas [29].

Trabajo de desarrollo durante el Sprint

Mientras el Sprint se encuentra desarrollo se debe asegurar que no se realicen cambios, que se cumplan los objetivos establecidos y no se disminuya la calidad. En el proceso de desarrollo es importante mantener una comunicación con el cliente para tener claro el proceso que se realiza o realizara modificaciones al alcance del proyecto de ser necesario [29].

Revisión del Sprint

Es una reunión informal donde asiste el equipo Scrum y el cliente o interesados al finalizar el Sprint. Se revisa el Product Backlog del Sprint y se definen los cambios a realizarse de ser necesario, estos cambios serán planificados para el siguiente Sprint [29].

Retrospectiva del Sprint

La retrospectiva del Sprint representa una oportunidad para el equipo de desarrollo de autoevaluarse y crear un plan para mejorar su proceso de trabajo y aplicar estos

cambios a los siguientes Sprint. Esta etapa tiene como objetivos los siguientes puntos:[29]

- Revisar las relaciones, procesos y herramientas que se utilizó el último Sprint.
- Identificar y listar los temas que salieron bien y mejoras a realizar.
- Crear un plan para la implementación de mejoras.

4.1.2.1. Ventajas de Scrum

La metodología Scrum en el desarrollo de un proyecto, ofrece ventajas como [32]:

- La adaptabilidad lo cual permite que se incorpore cambios al proyecto de una manera fácil.
- Ejecuta una planificación detallada por adelantado del proyecto, dar énfasis en la gestión y la solución en base a costos, horarios y gestión de parámetros.
- La transparencia que ofrece a través de las herramientas que muestran los avances del proyecto como el Scrumboard y Sprint Burndown Chart.
- Ofrece una retro alimentación continua.
- Se centra en el cliente haciendo énfasis en el valor del negocio y da énfasis en un trabajo colaborativo.

4.1.2.2. Desventajas de Scrum

En la metodología Scrum podemos identificar las siguientes desventajas [32]:

- Funciona correctamente con equipos de desarrollos reducidos.
- Exige una alta formación académica por parte de los desarrolladores.
- Requiere de una auto organización por parte de los integrantes del grupo de trabajo.
- Requiere una adecuada y exhaustiva definición de las tareas y su tiempo de entrega.

4.1.3. Crystal Methodologies

Crystal Methodologies consiste en un conjunto estrategias que se caracteriza por centrar a las personas que forman parte del equipo de desarrollo y clientes, teniendo en cuenta la reducción máxima del número de entregables producidos. El equipo de desarrollo es un factor clave, por lo que se invierte esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo bien definidos contemplando el tamaño del equipo [27].

Establece siete procesos y propiedades para cada proyecto, de los cuales tres son obligatorios y cuatro son de prevención y seguridad, los cuales son [27]:

- Entrega frecuente
- Mejora continua
- Seguridad personal
- Enfoque
- Fácil acceso a usuarios expertos
- Medio Ambiente, técnica de prueba automatizada, gestión e Integración frecuente

Crystal Methodologies no se rige a una serie de pasos específicos , sino que, tiene la libertad de usar técnicas de otras metodologías. Sin embargo, hay que tener en cuentas estos 3 pasos para el desarrollo de un proyecto [27].

- **Planificación del siguiente incremento:** Se debe planificar lo que se realizara en base a los requerimientos funcionales a implementar en el incremento.
- **Revisiones:** En cada incremento se realizar las revisiones correspondientes para determinar el correcto funcionamiento de cada incremento.
- **Monitoreo:** Cada proceso de una iteración es monitoreado a partir de las diferentes entregas.

La Crystal Methodologies posee una serie de ventajas y desventajas las cuales se enumeran a continuación.

4.1.3.1. Ventajas de Crystal Methodologies

- Se concentra especialmente en el tamaño del equipo de trabajo, lo que facilita tener control del desarrollo [28].
- Permite aplicar y adaptar diferentes técnicas y estrategias dependiendo del tipo de desarrollo que se va a llevar acabo [28].
- Permite una planificación transparente [28].

4.1.3.2.Desventajas de Crystal Methodologies

- Es una metodología fácil de aprender e implementar, pero no es una metodología disciplinada [28].

- Permite la eliminación y agregación de fases en la metodología lo que puede ocasionar una pérdida del control del desarrollo [28].

4.1.4. Características de Metodologías Ágiles

A continuación, se muestra un resumen de las principales características de las tres metodologías ágiles expuestas anteriormente XP, Scrum y Crystal Methodologies. Ver tabla 4.1

4.1.5. Comparativa de Metodologías Ágiles

En la tabla número 4.2, se presenta una comparación entre las metodologías ágiles para el desarrollo de software: XP, Scrum y Crystal Methodologies.

Metodología	Características
XP	<ul style="list-style-type: none"> • Metodología basada en prueba y error. • Cliente bien definido. • Los requisitos pueden y van a cambiar. • Grupo pequeño y muy integrado (máximo 12 personas). • Fundamentada en Valores. • Implementa compatibilidad y usabilidad con otras metodologías. • Está orientada hacia quién produce y uso del software. • Reduce el costo del cambio en todas las etapas del ciclo de vida del sistema. • Combina las que han demostrado ser las mejores prácticas para desarrollar software y las lleva al extremo. • Programación organizada. • Reduce la tasa de errores.
Scrum	<ul style="list-style-type: none"> • Permite la gestión de diferentes equipos industriales. • Aplicación de las metodologías ágiles en los procesos de calidad, entregando resultados óptimos. • Su adopción en prácticas orientadas a planes como CMMI ó PSP/TSP. • La productividad y competitividad de las empresas de software al implementar este tipo de metodología. • El aporte a la gestión de proyectos a través del proceso. • Aporte efectivo dentro de la gerencia de proyectos de software.

Crystal Methodologies	<ul style="list-style-type: none"> • Entregas frecuentes, en base a un ciclo de vida iterativo e incremental. • Cuantas más personas estén implicadas, más grande debe ser la metodología. • Entorno técnico con pruebas automatizadas, gestión de la configuración e integración continua. • Si el proyecto tiene mucha densidad, un error no detectado puede ser crítico. • El aumento de tamaño o densidad añade un coste considerable al proyecto. • Tamaño de un equipo (número de componentes) • Comunicación entre los componentes. • Distintas políticas para seguir. • Mejora reflexiva
------------------------------	---

Tabla 4.1: Características de las metodologías ágiles XP, SCRUM Y CRYSTAL METHODOLOGIES.
Fuente: [25]

En la presente comparación se la realiza con una valoración de uno a cinco (uno - mínimo, cinco - máximo) desde tres parámetros, los cuales son [33]:

- Sistema como algo cambiante
- Colaboración
- Características metodológicas

PARAMETROS	XP	SCRUM	Crystal
Sistema como algo cambiante	5	5	4
Colaboración	5	5	5
Características Metodológicas (CM)			
1. Resultados	5	5	5
2. Simplicidad	5	5	4
3. Adaptabilidad	3	4	5
4. Excelencia técnica	4	3	3
5. Prácticas de colaboración	5	4	5
Media CM	4.4	4.2	4.4
Media total	4.8	4.7	4.5

Tabla 4.2: Comparativa entre metodologías ágiles XP, SCRUM y CRYSTAL METHODOLOGIES
Fuente: [29]

Según la valoración de la Tabla 4.2, la metodología XP presenta la valoración más alta con un puntaje de 4.8 seguida de cerca por la metodología Scrum y por último tenemos a la metodología Crystal Methodologies.

4.1.6. Metodología Elegida

Después de analizar las diferentes metodologías ágiles considerando sus características y valoraciones respecto a la comparativa entre metodologías ágiles, además, teniendo en cuenta las características y necesidades del desarrollo de la Aplicación Web Progresiva se ha seleccionado para el desarrollo del presente proyecto la metodología ágil XP teniendo en cuenta lo siguiente:

- La metodología XP en la comparativa de metodologías presenta una calificación de 4.8 en su media total, siendo esta metodología la que presentan la valoración más alta, lo que dada como resultado que, la metodología XP aporta mayores beneficios al desarrollo del proyecto de la Aplicación Web Progresiva.
- Permite un desarrollo rápido manteniendo una comunicación permanente con el cliente, lo cual es fundamental para un desarrollo óptimo, además, proporciona un desarrollo incremental facilitando una retroalimentación lo que genera una menor tasa de error en el proyecto.
- La metodología XP está dirigida especialmente a pequeños y medianos grupos de trabajos, y al poseer un grupo reducido para el desarrollo del presente proyecto esta, metodología se adapta perfectamente.

4.2. Comparativa entre Tecnologías de Front-End

Existen diferentes tecnologías para el desarrollo del Front-End entre los cuales podemos mencionar Angular, React, Vue, Ember, Backbone, Polymer. Angular, React y Vue son las tecnologías más populares y usadas en la actualidad para el desarrollo del Front-End , además, cada una de las tecnologías mencionadas brindan herramientas que facilitan el desarrollo de Aplicación Web Progresivas.

En este apartado se realizará una comparativa entre las 3 tecnologías mencionadas anteriormente teniendo en cuenta la popularidad y características de cada una de ellas, con el fin de seleccionar la tecnología que se adapte a lo requerido para el desarrollo del presente proyecto.

- **Popularidad**

La popularidad de las diferentes tecnologías en el mercado nos dará indicios de porcentajes de aceptación y utilización, para el análisis se tomó en cuenta las estadísticas de los principales sitios de desarrollo y búsquedas relacionadas a las tecnologías realizadas en el último año.

Analizando la figura 4.2, se observa que la tendencia de búsquedas en el último año (2019-2020) ha predominado la tecnología React seguida de cerca por la tecnología de Angular y por último la tecnología de Vue. Cabe recalcar que en los últimos dos años React ha presentado un incremento con lo cual ha superado a Angular el mismo que ha presentado un decremento en el mismo tiempo.

Un punto importante que da a conocer el interés y posible uso de las tecnologías de Front-End a nivel mundial es el número de descargas que poseen a través de NPM. Analizando la Figura 4.3, podemos notar que React posee un gran número de descargas en NPM consiguiendo una gran diferencia tanto de Angular y Vue, para la presente estadística se considera las dos versiones existentes de Angular.

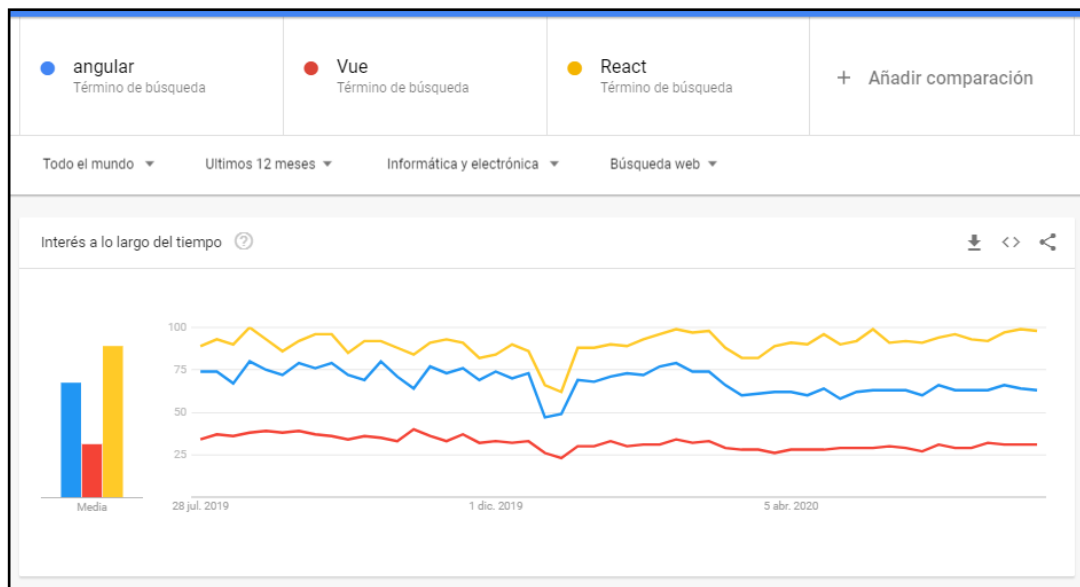


Figura 4.2: Tendencias de búsqueda de Angular, Vue y React.

Fuente: Google Trends (www.google.com/trends).

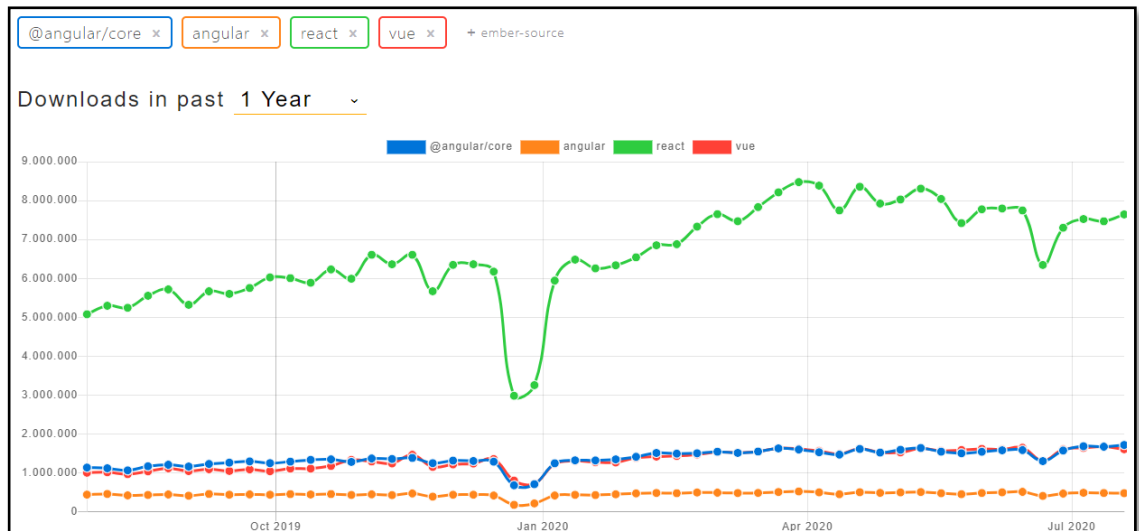


Figura 4.3: Estadísticas de descargas NPM(Angular ,Vue, React).
Fuente: NPM Trends (www.npmtrends.com)

Otra estadística para tener en cuenta es el número de preguntas realizadas en la web para poder tener una idea de la comunidad que posee la respectiva tecnología en el mundo. Ver Figura 4.4.

La Figura 4.4 muestra que hasta el 2020 Angular predominaba las estadísticas de Stack Overflow en cuanto a preguntas realizadas, a partir de este año React presenta un incremento y domina las estadísticas seguido de cerca por Angular. Tanto React y Angular presenta un gran porcentaje de diferencia en relación con Vue.

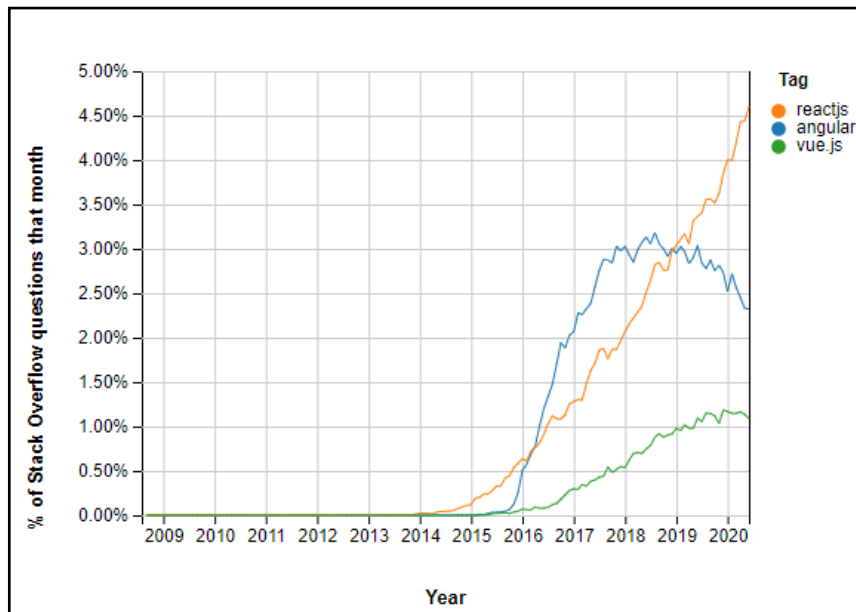


Figura 4.4: Estadísticas de preguntas de Stack Overflow.
Fuente: Stack Overflow Trends (<https://insights.stackoverflow.com/trends>).

- **Características Técnicas y Diferencias**

A continuación, se revisará las principales características que poseen tanto Angular, Vue y React, denotando las diferencias entre ellos.

Angular y Vue son frameworks el primero orientado a la creación de aplicaciones y el segundo orientado a la creación de interfaces de usuario, mientras que React es una librería.

Angular y Vue están obligados a trabajar de una manera ya establecida, mientras React presenta una mayor libertad en la forma de trabajo, ocasionando una mayor flexibilidad [12].

Un punto que está siempre a discusión es el rendimiento de estas 3 tecnologías, siendo un punto difícil de determinar. Según las diferentes pruebas de rendimiento Vue es la tecnología con un rendimiento más rápido, Vue en relación Angular que es cinco veces más rápido y dos veces más rápido que React. La gran diferencia de rendimiento entre Angular, Vue y React está definida en el manejo del DOM (Document Object Model- Modelo de Objetos del Documento), Angular utiliza el DOM nativo del navegador, en cambio Vue y React utilizan un DOM virtual, creando una copia de memoria, consiguiendo generar un número menor de operaciones sobre el DOM nativo [7], [12].

Angular y Vue comparten un enfoque parecido al manejo de componente tanto en plantilla, estilo y lógica, ofreciendo manejarlo ya sea en un archivo o dividirlo en 3 archivos por separado. Angular distingue entre componentes, servicios y vistas, teniendo en cuenta a la inyección de dependencias como eje central de la interoperabilidad y reusabilidad, utiliza HTML con interpolación y directivas propias para la creación de templates. React de igual manera está basado en componentes y no dispone de un contenedor de servicios y como sistema de templates utiliza JSX (JavaScript XML) un lenguaje mezclado de XML y JavaScript. Vue utiliza un sistema de templates HTML con interpolación y directivas propias, pero a diferencia de Angular permite el uso JSX. [12]

En el enlace de datos Angular y Vue utilizan un modelo de doble enlace que permite la creación de un enlace entre el valor del modelo y vista, es decir si cualquier modificación en el origen del modelo o vista la actualización se realizara en ambos. En cambio, React utiliza un modelo simple en el cual el modelo es el responsable

de actualizar el valor teniendo en cuenta las diferentes notificaciones o eventos emitidos por la vista.[17]

Un punto clave en la comparativa entre los tres frameworks es la robustez, Vue y React son considerados como un enfoque para la creación de interfaces de usuario y Angular está orientado a la creación de aplicaciones, por lo cual Angular es más robusto que Vue y React, sin embargo, estas dos últimas tecnologías pueden crecer a medida que el proyecto lo necesite.

	Angular	React	Vue
Creador	Google	Facebook	Evan You
Tipo	Framework	Librería	Framework
Curva de aprendizaje	Media	Media/Fácil	Fácil
Flexibilidad	Baja	Alta	Alta
Lenguaje	TypeScript	JavaScript	JavaScript/JSX
Modelo DOM	Nativo	Virtual	Virtual
Robustez	Grande	Mediana	Mediana

Tabla 4.2: Comparativa entre Angular, React y Vue.

Fuente: [15]

En base a las características y funcionalidad de las diferentes tecnologías a continuación, se expone las ventajas y desventajas de las 3 tecnologías seleccionadas:

Ventajas y Desventajas de Angular

Ventajas	Desventajas
<ul style="list-style-type: none"> • Angular posee un enlace de datos en ambos sentidos, permitiendo minimizar errores. • Es altamente reusable al trabajar con componentes. • Permite un mantenimiento y actualización de código de una manera eficiente. • A través del compilado su código en TypeScript es convertido a JavaScript. • Es un framework muy robusto o que permite un desarrollo completo y escalable. • Posee una documentación detallada y una gran comunidad que facilita la resolución de problemas. 	<ul style="list-style-type: none"> • Angular posee una curva de aprendizaje muy alta debido a la robustez que posee. • Puede presentarse problemas al migrar de una versión a otra.

Tabla 4.3: Ventajas y desventajas de Angular

Fuente: Elaborado por el investigador

Ventajas y Desventajas de React

Ventajas	Desventajas
<ul style="list-style-type: none">• Posee una curva de aprendizaje muy baja debido a sus simplicidades en términos de sintaxis.• Maneja un DOM Virtual con el fin de generar menos operaciones sobre el DOM nativo.• Posee un framework muy ligero.• Facilidad al realizar migraciones de una versión a otra.	<ul style="list-style-type: none">• No posee un estándar de desarrollo definido.• Necesita mucho tiempo y profundo conocimiento para dominarlo.

Tabla 4: Ventajas y desventajas de React

Fuente: Elaborado por el investigador

Ventajas Y Desventajas de Vue

Ventajas	Desventajas
<ul style="list-style-type: none">• Posee una excelente integración, ya que permite la construcción de aplicaciones de una sola página o aplicaciones con complejas interfaces.• Pose un tamaño reducido conservando su flexibilidad y rendimiento.• Posee un gran rendimiento al trabajar con un DOM Virtual al igual que React.	<ul style="list-style-type: none">• No posee una documentación extensa o comunidad que ayude con la resolución de problemas.• Puede ser muy flexible y ocasionar problemas al integrar a grandes proyectos.

Tabla 4.5 : Ventajas y desventajas de Vue

Fuente: Elaborado por el investigador

4.2.1. Tecnología Front-End Elegida

Después de analizar las diferentes características y tendencias de las 3 tecnologías mencionadas en el apartado anterior, se ha decidido utilizar el framework Angular al ser uno de los framework más robusto, permitiendo un desarrollo completo y escalable. Y considerando que, la valoración de la curva de aprendizaje que posee es debido a su robustez y la valoración de la flexibilidad es debido a que al ser un framework posee una forma de trabajo definida, lo que disminuye su flexibilidad en comparación a otros framework.

Además de esto, Google es la compañía que en la actualidad más apuesta por el desarrollo de las Aplicaciones Web Progresivas y siendo el creador y la compañía que mantiene a Angular posee cierta ventaja de las demás tecnologías.

4.3. Comparación entre Tecnología de Back-End

Existen diferentes tecnologías de Back-End en el mercado dentro de las cuales podemos mencionar ASP.NET, ASP.NET CORE , PHP, JAVA , NODE JS, etc. Se seleccionó dos tecnologías de Back-End : ASP.NET CORE y NODE.JS , considerando que son unas de las tecnologías más usadas y sobre todo se integran perfectamente con la tecnología de Front-End seleccionada.

Se realizó una comparativa de las tecnologías seleccionadas, con el fin de determinar la tecnología con mejores características para el desarrollo de nuestra Aplicación Web Progresiva.

4.3.1. Comparación entre ASP.NET CORE y NODE.JS

- **Rendimiento**

Comparar el rendimiento de ASP.NET CORE con NODE.JS en el lado del servidor es un tema difícil de catalogar y que genera diferentes puntos de opinión como los que se menciona a continuación:

Según el trabajo de investigación titulado “Análisis de frameworks de desarrollo de API REST y su impacto en el rendimiento de aplicaciones web con arquitectura SPA” concluye que el framework NODE.JS posee un mejor rendimiento en comparación a ASP.NET CORE en el consumo de una API REST (Representational State Transfer - Transferencia de Estado Representacional), llegando a una diferencia del 8%. Hay que tener en cuenta que este porcentaje va en aumento al considerar un mayor número de usuarios concurrentes. Es decir, a un menor número de usuarios concurrentes el porcentaje disminuirá considerablemente.[20]

Por otra parte, ASP.NET en su página oficial, asegura poseer un mayor rendimiento en cuanto a solicitudes por segundo, basándose en las pruebas obtenidas por TechEmpower Round 19. Sin duda, ASP.NET CORE Y NODE.JS poseen un excelente rendimiento, sin embargo, cada uno de ellos presenta una disminución de su rendimiento al presentarse situaciones específicas por ejemplo NODE.JS es superado por ASP.NET CORE al realizar operaciones de cálculo y su viceversa ASP.NET CORE es superado por NODE.JS en aplicaciones que necesiten estar conectadas continuamente.

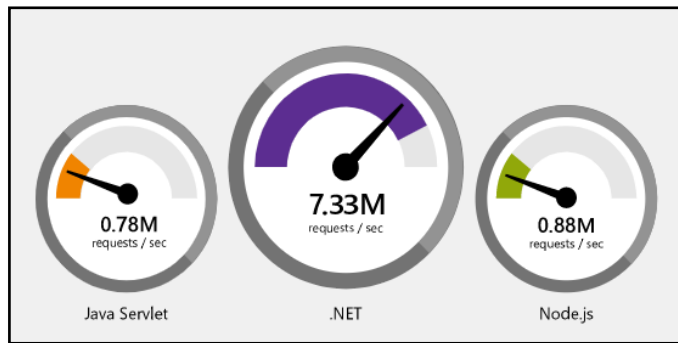


Figura 4.5: Rendimiento de .NET y NODE JS
Fuente : Asp.Net (<https://dotnet.microsoft.com/apps/aspnet>)

- **Popularidad**

Otro punto importante es la popularidad de los framework que existe entre los desarrolladores, para lo cual se ha tomado las estadísticas expuestas por StackOverFlow del año 2020. La popularidad de los diferentes framework nos da a valoración de la comunidad y un potencial uso por parte de los desarrolladores.

La Figura 4.6 resalta que el framework más usado en la actualiza es NODE.JS seguido de ASP.NET Y ASP.NET CORE en las estadísticas de framework más usados.

La Figura 4.7 muestra que JavaScript es el lenguaje más usado, mientras que C# ocupa la séptima posición de la encuesta. JavaScript y C# son los lenguajes de programación de NODE JS Y ASP.NET CORE respectivamente.

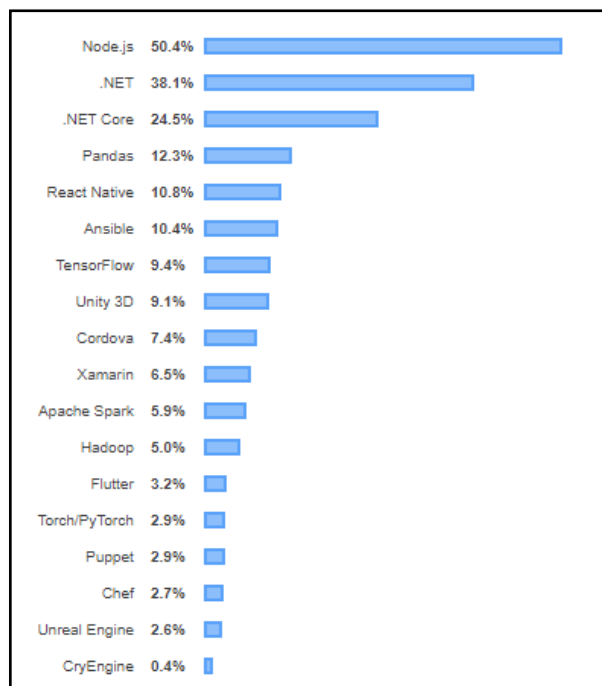


Figura 4.6: Framework más usados en el año 2020
Fuente: Stack Overflow Trends (<https://insights.stackoverflow.com/trends>).

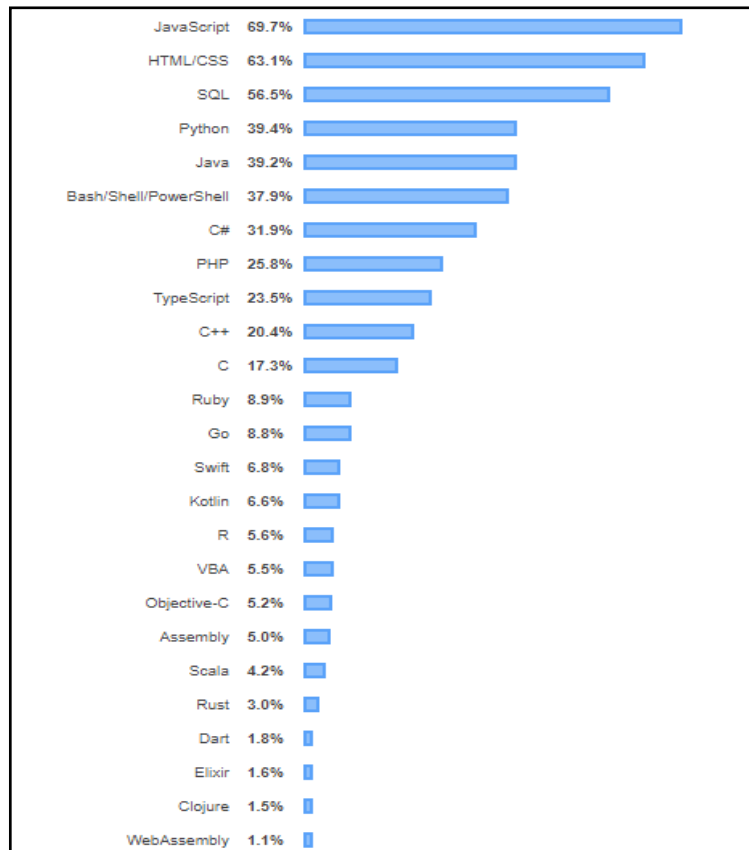


Figura 4.7: Lenguajes más usados en el año 2020.

Fuente: Stack Overflow Trends (<https://insights.stackoverflow.com/trends>).



Figura 4.8: Tendencias de búsqueda entre .NET CORE y NODE JS.

Fuente: Google Trends (www.google.com/trends)

La Figura 4.8 muestra las estadísticas de las búsquedas en los últimos 4 años, la Figura 4.8 revela que ASP.NET CORE ha tenido un auge en cuanto a búsquedas por tema superando en el año 2018 a NODE.JS.

Las diferentes estadísticas presentadas anteriormente nos indica que NODE JS predomina el mercado en la actualidad, pero ASP.NET CORE está teniendo un gran auge y aceptación por los desarrolladores.

Ventajas y Desventajas de NODEJS

Ventajas	Desventajas
<ul style="list-style-type: none"> • El desarrollo en NODE.JS se realiza a través del lenguaje JavaScript. • Basado en el motor de Google Chrome V8 lo que proporciona un rendimiento eficaz. • Operaciones de Entrada/Salida sin bloqueos. • NODE.JS es orientada a eventos. • Permite realizar un desarrollo liviano y eficiente. 	<ul style="list-style-type: none"> • Requiere un cambio de mentalidad para el desarrollo debido a su estructura asíncrona. • Dificultad de aprendizaje debido a su estructura asíncrona. • Posee una menor potencia de cálculo en comparación a otras tecnologías.

Tabla 6: Ventajas y desventajas de NODE JS

Fuente: Elaborado por el investigador

Ventajas y Desventajas de ASP.NET CORE

Ventajas	Desventajas
<ul style="list-style-type: none"> • NET CORE es compatible con los tres principales sistemas operativos en la actualidad, como son: Windows, Linux y MacOS. • Está disponible en GitHub lo que favorece al desarrollo y a la existencia de una comunidad activa y comprometida. • Facilita el desarrollo mediante dos modos: basada en marco e implementación de autocontenida. • Es modular debido a que se publica a través de Nuget en paquete ensamblados reducidos, permitiendo un desarrollo ágil y la optimización en el desarrollo. 	<ul style="list-style-type: none"> • Elevado consumo de recursos de la máquina tanto en memoria RAM y procesamiento de CPU para un desarrollo correcto en cuanto a funcionalidad y compilación.

Tabla 7: Ventajas y desventajas de ASP.NET CORE

Fuente: Elaborado por el investigador.

- **Características de ASP.NET CORE Y NODE JS**

La siguiente tabla muestra las principales características de ASP.NET CORE Y NODE.JS:

CARACTERISTICAS	ASP.NET CORE	NODE.JS
Creador	Microsoft	Ryan Dahl
Lenguaje	C#	JavaScript
Comunidad	Alta	Alta
Herramientas	Alta	Media
Plataformas	Linux, MacOS, Microsoft	Linux, MacOS, Microsoft
Procesamiento	Multiproceso	Un solo hilo

Tabla 4.8: Comparativa entre ASP.NET CORE y NODE JS

Fuente: Elaborado por el investigador.

4.3.2. Tecnología de Back-End Elegida

La tecnología que se ha seleccionado para el desarrollo de la Aplicación Web Progresiva es ASP.NET CORE, ya que posee las siguientes características que ayudan al desarrollo del presente proyecto:

- Posee herramientas que ayuda a agilizar el proceso de desarrollo de Back-End, como es el caso Entity Framework Core y Framework Identity ayudando a la construcción del modelado de datos e identificación de la aplicación.
- Posee marco trabajo más ágil y moderno, permite la integración de librerías como Angular, React, Blazor, entre otras.
- Posee un rendimiento superior en comparación a NODE JS.
- ASP.NET CORE está enfocado en la construcción de aplicaciones web, lo que permite un mayor rendimiento en operaciones de cálculos.

4.4. Arquitectura de la Aplicación Web Progresiva

La presente Aplicación Web Progresiva se desarrollará mediante el modelo arquitectura cliente-servidor según las tecnologías seleccionadas. Para la construcción de la Aplicación Web Progresiva se aplicó el modelo ASP.NET MVC (Modelo-Vista-Controlador).

El modelo constará de los datos y la lógica de negocio, el controlador se encargará de gestionar la comunicación entre el modelo y la vista. En cambio, la vista procesara la información procedente del modelo y visualizar al usuario la información mediante las diferentes interfaces de usuario. La vista se desarrollará

bajo la tecnología de Angular 8 la misma que estará agrupada por módulos y desarrollada bajo el modelo MVC, los servicios representaran a los modelos, los componentes de presentación representaran las vistas y los componentes de página representaran los controladores.

Además, en el servidor se aumentó los DTO (Objetos de transferencia de datos) que define como se enviará la información desde el controlados hacia la vista, con lo cual nos permitirá eliminar las referencias circulares, ocultar o agregar datos extras a la información enviada por el modelo.

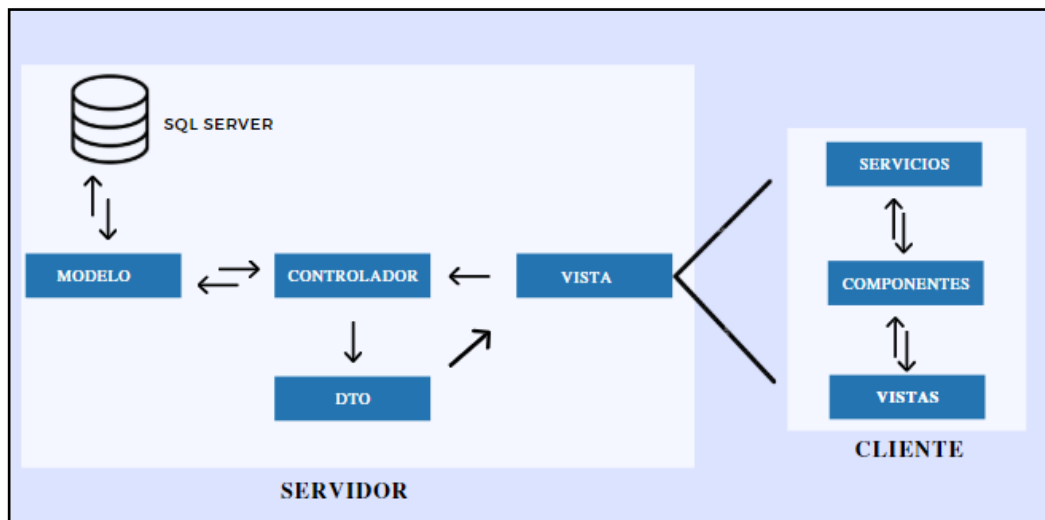


Figura 4.9: Arquitectura de la Aplicación Web Progresiva.
Fuente: Elaborado por el investigador.

4.5. Desarrollo de la Metodología

El presente proyecto se llevará a cabo mediante la metodología XP, la cual permite de una manera rápida e interactiva la realización del diseño, codificación y pruebas mediante sus diferentes fases que se mencionan a continuación:

1. Fase: Planificación del Proyecto
2. Fase: Diseño
3. Fase: Codificación
4. Fase: Pruebas

4.5.1. Roles del Proyecto

Para el desarrollo del proyecto se definieron los siguientes roles:

Rol	Responsable	Función
Usuario o Cliente	Centro de capacitación y nivelación académica SMARTEL	Determinar los procesos a implementar en el desarrollo de la Aplicación Web Progresiva.
Programador	Wilmer Rolando Oñate Calderon	Desarrollar la Aplicación Web Progresiva en base a las necesidades del cliente.
Tester	Ing. Miller Patricio Bayas Remache	Realizar las diferentes pruebas de aceptación.

Tabla 4.9: Roles del proyecto a desarrollar
Fuente: Elaborado por el investigador.

4.5.2. Fase: Planificación del Proyecto

En la planificación del proyecto se define el alcance y la prioridad mediante el uso de historias usuario, las mismas que se realizaron juntamente con el usuario. Las historias de usuario es una representación de los requisitos expresados en frases cortas y utilizando un lenguaje natural. Para el desarrollo de las historias de usuario, se realizó la recolección de la información mediante dos reuniones con el cliente donde se observó los procesos que posee y realiza el Centro de Capacitación y Nivelación Académica SMARTEL.

4.5.3. Diagrama de Procesos

El diagrama de procesos es un conjunto de actividades y acciones que son representadas en forma gráfica. A continuación, se muestra el diagrama de procesos para la toma de pruebas de simulación en el Centro de Capacitación y Nivelación Académica SMARTEL, desde la creación de módulos hasta la generación de reportes por parte de los profesores. Ver Figura 4.10

A continuación, se muestran las historias de usuarios elaboradas para el desarrollo de la PWA, para la estimación de las historias de usuario se utilizó la técnica “Planning Poker”.

4.5.3.1. Historias de Usuario

HISTORIA DE USUARIO	
Código: H1	Usuarios: Todos
Nombre Historia: Inicio de sección.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 3	Interacción Asignada: 3
Descripción: Antes de iniciar la aplicación se debe solicitar un nombre de usuario y una contraseña, para que tengan acceso a las diferentes funcionalidades del sistema.	

Existen tres tipos de usuarios, Administrador, Profesor y Estudiante, con distintas funcionalidades.
Observación: El nombre de los usuarios estará formado por el primer apellido del usuario seguido de la primera letra del rol en mayúscula seguido de los últimos 4 dígitos de su identificación

Tabla 4.10: Historia de Usuario- Inicio de sesión.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H2	Usuarios: Administrador
Nombre Historia: Gestión de Asignaturas.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 3	Interacción Asignada: 1
Descripción: El administrador de la aplicación podrá realizar las acciones de listar, buscar, crear, editar y eliminar asignaturas.	
Observación: Las asignaturas se listarán en activos e inactivos, la acción de eliminar se realizará mediante una eliminación lógica.	

Tabla 4.11: Historia de Usuario- Gestión de Asignatura.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H3	Usuarios: Administrador
Nombre Historia: Gestión de Módulos.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 2	Interacción Asignada: 1
Descripción: El administrador de la aplicación podrá realizar las acciones de listar, buscar, crear, editar y eliminar los diferentes módulos existentes.	
Observación: Los Módulos se agruparán en activos e inactivos, la acción de eliminar se realizará mediante una eliminación lógica.	

Tabla 4.12: Historia de Usuario- Gestión de Módulos.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H4	Usuarios: Administrador
Nombre Historia: Gestión de Cursos.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 5	Interacción Asignada: 1
Descripción: El administrador de la aplicación podrá realizar las acciones de listar, buscar, crear, editar y eliminar los diferentes cursos existentes, además de seleccionar y asociar las asignaturas, profesores y horario de clases que posee un curso.	
Observación: Los Cursos se agruparán en activos e inactivos, la acción de eliminar se realizará mediante una eliminación lógica de ser necesario.	

Tabla 4.13: Historia de Usuario- Gestión de Cursos.

Fuente: Elaborado por el investigador.

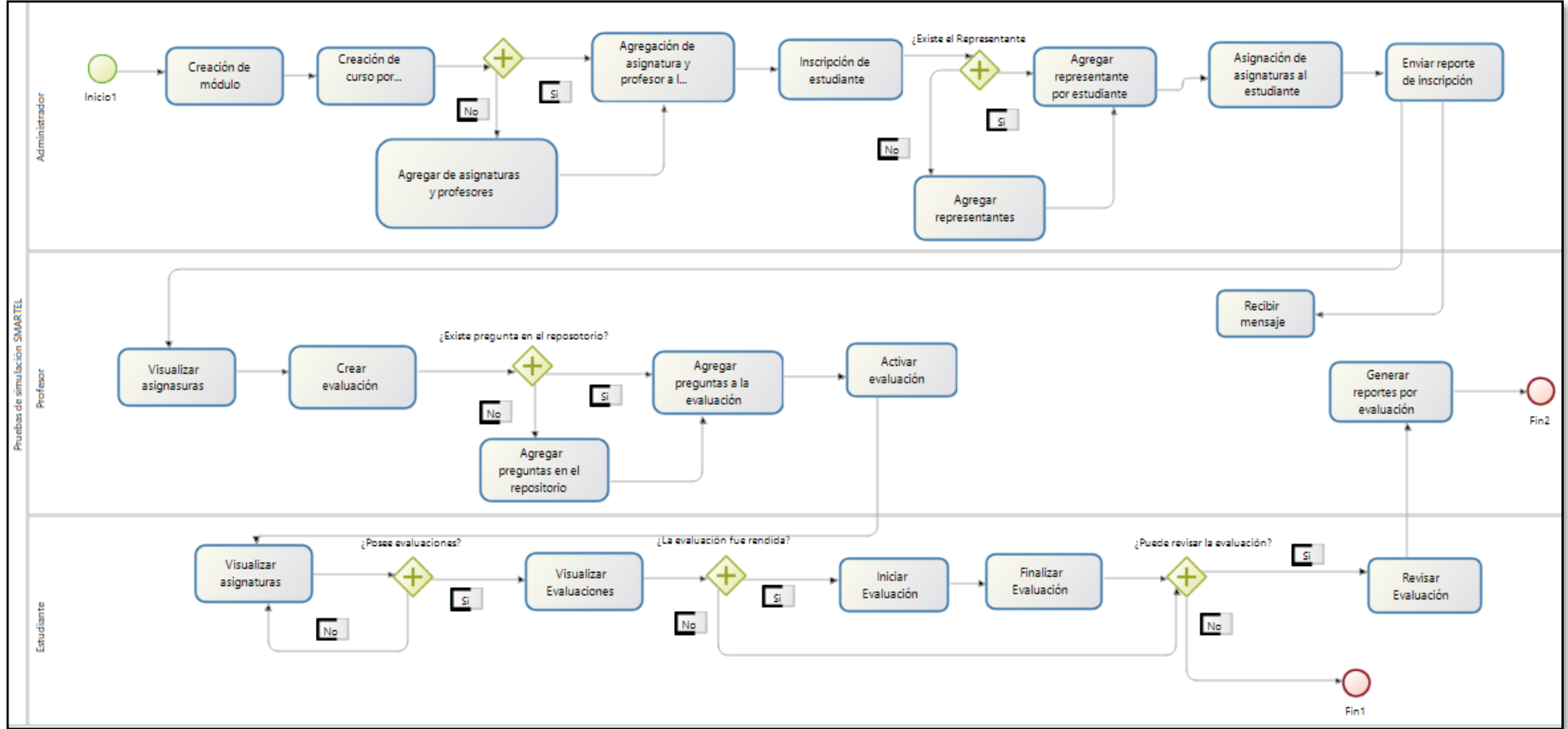


Figura 4.10: Flujo de procesos de negocios del Centro de Capacitación y Nivelación Académica SMART EL.
Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H5	Usuarios: Administrador
Nombre Historia: Gestión de Títulos Académicos.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 2	Interacción Asignada: 1
Descripción: El administrador de la aplicación podrá realizar las acciones de listar, buscar, crear, editar y eliminar los diferentes títulos académicos.	
Observación: Los Títulos Académicos se agruparán en activos e inactivos, la acción de eliminar se realizará mediante una eliminación lógica.	

Tabla 4.14: Historia de Usuario- Gestión de Títulos Académicos.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H6	Usuarios: Administrador
Nombre Historia: Gestión de Instituciones Educativas.	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 2	Interacción Asignada: 1
Descripción: El administrador de la aplicación podrá realizar las acciones de listar, buscar, crear, editar y eliminar las diferentes Instituciones Educativas.	
Observación: Las Instituciones Educativas se agruparán en activas e inactivas, la acción de eliminar se realizará mediante una eliminación lógica.	

Tabla 4.15: Historia de Usuario- Gestión de Instituciones Educativas.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H7	Usuarios: Administrador
Nombre Historia: Gestión de Personas.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 3	Interacción Asignada: 2
Descripción: El administrador de la aplicación podrá realizar las acciones de listar, buscar, crear, editar y eliminar personas, además de seleccionar y asociar los teléfonos que posee una persona.	
Observación: Las personas se agruparán en activos e inactivos, la información de email, contraseña email e imagen de la identificación serán opcionales.	

Tabla 4.16: Historia de Usuario - Gestión de Personas.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H8	Usuarios: Administrador
Nombre Historia: Gestión de Representantes.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 2	Interacción Asignada: 2
Descripción: Se podrá realizar las acciones de listar, buscar, crear y eliminar representantes. Un representante se creará a partir de una persona previamente registrada.	
Observación: Los representantes se agruparán en activos e inactivos, la acción de eliminar se realizará mediante una eliminación lógica.	

Tabla 4.17: Historia de Usuario - Gestión de Representantes.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H9	Usuarios: Administrador
Nombre Historia: Gestión de Profesores.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 5	Interacción Asignada: 2
Descripción: Se podrá realizar las acciones de listar, buscar, crear, y eliminar profesores. Un profesor se creará a partir de una persona previamente registrada, además de seleccionar y asociar las asignaturas, títulos académicos que posee un profesor.	
Observación: Los profesores se agruparán en activos e inactivos, la acción de eliminar se realizará mediante una eliminación lógica.	

Tabla 4.18: Historia de Usuario - Gestión de Profesores.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H10	Usuarios: Administrador
Nombre Historia: Gestión de Estudiantes.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 5	Interacción Asignada: 2
Descripción: Se podrá realizar las acciones de listar, buscar, crear y eliminar estudiantes. Un estudiante se creará a partir de una persona previamente registrada. Además de seleccionar y asociar los representantes, instituciones educativas y asignaturas que posee un estudiante.	
Observación: Los estudiantes se agruparán en activos e inactivos, la acción de eliminar se realizará mediante una eliminación lógica.	

Tabla 4.19: Historia de Usuario - Gestión de estudiantes.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H11	Usuarios: Administrador, Profesor
Nombre Historia: Gestión de tema de una pregunta.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 2	Interacción Asignada: 1
Descripción: Se realizará las acciones de listar, buscar, crear, editar y eliminar para temas de preguntas.	
Observación: Los temas de pregunta se agruparán en activos e inactivos, la acción de eliminar se realizará mediante una eliminación lógica.	

Tabla 4.20: Historia de Usuario - Gestión de tema de una pregunta.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H12	Usuarios: Administrador, Profesor
Nombre Historia: Gestión de preguntas.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 8	Interacción Asignada: 3
Descripción: Se generará un repositorio de preguntas el mismo que está organizado por tema de pregunta. Cada pregunta constará de un texto y como mínimo un ítem de selección, tanto en el texto de la pregunta y los ítems se podrá ingresar una o varias imagen y fórmula matemática, además de elegir cual ítem es la respuesta correcta a la pregunta por ingresar. Se añadirá las acciones de listar, editar y eliminar.	

Observación: El tipo de pregunta será únicamente de opción múltiple, pero se considerará la opción de incrementar los tipos de preguntas a ingresar posteriormente.

Tabla 4.21: Historia de Usuario - Gestión de preguntas.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H13	Usuarios: Administrador
Nombre Historia: Gestión de usuarios.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 8	Interacción Asignada: 3
Descripción: Los usuarios se crearán automáticamente cuando se ingrese un profesor y un estudiante respectivamente, se asignará una contraseña por defecto a todos los usuarios que se crearan automáticamente, además de permitir cambiar, reiniciar y recuperas la clave de usuario.	
Observación: Los roles que se crearan automáticamente son estudiante y profesor.	

Tabla 4.22: Historia de Usuario - Creación de usuarios.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H14	Usuarios: Profesor
Nombre Historia: Gestión de Evaluaciones.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 13	Interacción Asignada: 3
Descripción: Los profesores visualizaran las asignaturas que dictan clases y podrán crear, editar y eliminar evaluaciones, evaluaciones consolidadas, además de agregar preguntas procedentes del repositorio y posteriormente activar dichas evaluaciones.	
Observación: La evaluación se creará inactiva, una evaluación consolidada se formará de la evaluación de las diferentes asignaturas que posee el estudiante.	

Tabla 4.23: Historia de Usuario - Gestión de Evaluaciones.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H15	Usuarios: Profesor, estudiante
Nombre Historia: Revisión de evaluaciones.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 3	Interacción Asignada: 4
Descripción: Un profesor y un estudiante podrá revisar las evaluaciones realizadas por el estudiante donde se visualizará la calificación obtenido por pregunta y por evaluación en caso del profesor.	
Observación: Se mostrará las respuestas correctas e incorrectas de la evaluación.	

Tabla 4.24: Historia de Usuario - Revisión de evaluaciones.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H16	Usuarios: Estudiante
Nombre Historia: Iniciar Evaluación.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 5	Interacción Asignada: 4

Descripción: El estudiante podrá visualizar las diferentes evaluaciones agrupadas por asignatura e iniciar la evaluación. Al iniciar la evaluación el estudiante podrá contestar las preguntas que presentarán al estudiante de forma progresiva una por una.
Observación: La duración de la evaluación empezará a contar desde que el estudiante confirme el inicio de la evaluación.

Tabla 4.25: Historia de Usuario - Iniciar Evaluación.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H17	Usuarios: Estudiante, Profesor
Nombre Historia: Reporte de horario de clases.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 3	Interacción Asignada: 4
Descripción: Los usuarios con los roles estudiantes y profesor podrán visualizar y descargar el horario de clases.	
Observación: El horario de clases se descargará en formato PDF.	

Tabla 4.26: Historia de Usuario - Reporte de horario de clases.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H18	Usuarios: Profesor
Nombre Historia: Reporte de evaluaciones.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 3	Interacción Asignada: 4
Descripción: Se generará un reporte de las evaluaciones de todos los estudiantes que realizaron la evaluación seleccionada, el mismo que costara de un encabezado donde se especificara el curso, la asignatura, el profesor, el nombre de la evaluación y la valoración de la evaluación. El cuerpo del reporte constara de identificación del estudiante, nombre del estudiante, calificación, numero de respuestas correctas e incorrectas y no contestadas.	
Observación: El reporte se generará en formato PDF.	

Tabla 4.27: Historia de Usuario - Reporte de evaluaciones.

Fuente: Elaborado por el investigador.

HISTORIA DE USUARIO	
Código: H19	Usuarios: Administrador
Nombre Historia: Ficha de Inscripción del estudiante.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 3	Interacción Asignada: 3
Descripción: Se generará una ficha de inscripción donde se constatará la información proporcionada por el estudiante, además del horario de clases del estudiante.	
Observación: El reporte se generará en formato PDF de acuerdo con el formato proporcionado por cliente.	

Tabla 4.28: Historia de Usuario - Ficha de Inscripción del estudiante.

Fuente: Elaborado por el investigador.

4.5.3.2. Tareas

Una tarea es una actividad o conjunto de actividades que poseen un inicio y un final, son asignadas a un responsable o encarado, el cual debe completar dicha tarea en un tiempo determina [29].

En base a las historias de usuarios y las iteraciones definidas se procedió a la generación de las diferentes tareas a desarrollar. En las siguientes tablas se muestran las tareas a realizar en base a cada historia de usuario.

Historia de usuario: Gestión de Asignaturas.

TAREA	
Número de Tarea: T1	Código de Historia: H2
Nombre de Tarea: Diseño de las interfaces para la gestión de asignaturas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,5
Fecha Inicio: 01/04/2020	Fecha Fin: 01/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará el diseño de las interfaces, en el cual los usuarios podrán realizar las acciones de listar, buscar, agregar, editar y eliminar una asignatura.	

Tabla 4.29: Diseño de las interfaces para la gestión de asignaturas.
Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T2	Código de Historia: H2
Nombre de Tarea: Implementación de la gestión de asignaturas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1,5
Fecha Inicio: 02/04/2020	Fecha Fin: 03/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee una asignatura.	

Tabla 4.30: Implementación de la gestión de asignaturas.
Fuente: Elaborado por el investigador.

Historia de usuario: Gestión de Módulos.

TAREA	
Número de Tarea: T3	Código de Historia: H3
Nombre de Tarea: Diseño de las interfaces para la gestión de módulos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,5
Fecha Inicio: 06/04/2020	Fecha Fin: 06/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará el diseño de las interfaces, en el cual los usuarios podrán realizar las acciones de listar, buscar, agregar, editar y eliminar un módulo.	

Tabla 4.31: Diseño de las interfaces para la gestión de módulos.
Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T4	Código de Historia: H3
Nombre de Tarea: Implementación de la gestión de módulos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1,5
Fecha Inicio: 06/04/2020	Fecha Fin: 07/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee un módulo.	

Tabla 4.32: Implementación de la gestión de módulos.

Fuente: Elaborado por el investigador.

Historia de usuario: Gestión de Cursos.

TAREA	
Número de Tarea: T5	Código de Historia: H4
Nombre de Tarea: Diseño de las interfaces para la gestión de cursos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 09/04/2020	Fecha Fin: 09/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará el diseño de las interfaces para la gestión de un curso, en el cual los usuarios podrán realizar las acciones de listar, buscar, agregar, editar y eliminar un curso como también la asignación de asignaturas, profesores y horario de clases.	

Tabla 4.33: Diseño de las interfaces para la gestión de cursos.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T6	Código de Historia: H4
Nombre de Tarea: Implementación de la gestión de cursos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 10/04/2020	Fecha Fin: 10/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee un curso.	

Tabla 4.34: Implementación de la gestión de módulos.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T8	Código de Historia: H4
Nombre de Tarea: Asignación de asignaturas y profesores por curso.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 10/04/2020	Fecha Fin: 10/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, buscar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee las asignaturas y profesores correspondientes a un curso.	

Tabla 4.35: Asignación de asignaturas y profesores por curso.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T9	Código de Historia: H4
Nombre de Tarea: Asignación de horario de clases.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 10/04/2020	Fecha Fin: 10/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, agregar, editar y eliminar, teniendo en cuenta el diseño y los requerimientos de horario de clases por asignatura y profesor correspondientes a un curso.	

Tabla 4.36: Asignación de horario de clases.

Fuente: Elaborado por el investigador.

Historia de usuario: Gestión de Títulos Académicos.

TAREA	
Número de Tarea: T10	Código de Historia: H5
Nombre de Tarea: Diseño de las interfaces para la gestión de títulos académicos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,5
Fecha Inicio: 13/04/2020	Fecha Fin: 13/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará el diseño de las interfaces para la gestión de títulos académicos, en el cual los usuarios podrán realizar las acciones de listar, buscar, agregar, editar y eliminar un título académico.	

Tabla 4.37: Diseño de las interfaces para la gestión de títulos académicos.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T11	Código de Historia: H5
Nombre de Tarea: Implementación de la gestión de títulos académicos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1,5
Fecha Inicio: 14/04/2020	Fecha Fin: 14/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee un módulo.	

Tabla 4.38: Implementación de la gestión de títulos académicos.

Fuente: Elaborado por el investigador

Historia de usuario: Gestión de Instituciones Educativas.

TAREA	
Número de Tarea: T12	Código de Historia: H6
Nombre de Tarea: Diseño de las interfaces para la gestión de instituciones educativas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,5
Fecha Inicio: 15/04/2020	Fecha Fin: 15/04/2020
Programador Responsable: Wilmer Oñate	

Descripción: Se realizará el diseño de las interfaces para la gestión de instituciones educativas, en el cual los usuarios podrán realizar las acciones de listar, buscar, agregar, editar y eliminar una institución educativa.

Tabla 4.39: Diseño de las interfaces para la gestión de instituciones educativas.

Fuente: Elaborado por el investigador

TAREA	
Número de Tarea: T13	Código de Historia: H6
Nombre de Tarea: Implementación de la gestión de instituciones educativas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1,5
Fecha Inicio: 16/04/2020	Fecha Fin: 16/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee una institución educativa.	

Tabla 4.40: Implementación de la gestión de instituciones educativas.

Fuente: Elaborado por el investigador.

Historia de usuario: Gestión de Personas.

TAREA	
Número de Tarea: T14	Código de Historia: H7
Nombre de Tarea: Diseño de las interfaces para la gestión de personas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 20/04/2020	Fecha Fin: 20/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará el diseño de las interfaces para la gestión de personas, en el cual los usuarios podrán realizar las acciones de listar, buscar, agregar, editar y eliminar una persona, además se realizará el diseño de la asignación de teléfonos por persona.	

Tabla 4.41: Diseño de las interfaces para la gestión de personas.

Fuente: Elaborado por el investigador

TAREA	
Número de Tarea: T15	Código de Historia: H7
Nombre de Tarea: Implementación de la gestión de personas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 21/04/2020	Fecha Fin: 21/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee una persona.	

Tabla 4.42: Implementación de la gestión de personas.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T16	Código de Historia: H7
Nombre de Tarea: Asignación de teléfonos por persona.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22/04/2020	Fecha Fin: 22/04/2020
Programador Responsable: Wilmer Oñate	

Descripción: Se llevará a cabo la implementación de las acciones de listar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee un teléfono por persona.

Tabla 4.43: Asignación de teléfonos por persona.

Fuente: Elaborado por el investigador.

Historia de usuario: Gestión de Representantes.

TAREA	
Número de Tarea: T17	Código de Historia: H8
Nombre de Tarea: Diseño de las interfaces para la gestión de representantes.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23/04/2020	Fecha Fin: 23/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará el diseño de las interfaces para la gestión de representantes, en el cual los usuarios podrán realizar las acciones de listar, buscar, agregar, editar y eliminar una representante.	

Tabla 4.44: Diseño de las interfaces para la gestión de representantes.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T18	Código de Historia: H8
Nombre de Tarea: Implementación de la gestión de representantes.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 24/04/2020	Fecha Fin: 24/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, buscar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee un representante.	

Tabla 4.45: Implementación de la gestión de representantes.

Fuente: Elaborado por el investigador.

Historia de usuario: Gestión de Profesores.

TAREA	
Número de Tarea: T19	Código de Historia: H9
Nombre de Tarea: Diseño de las interfaces para la gestión de profesores.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1,5
Fecha Inicio: 27/04/2020	Fecha Fin: 27/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará el diseño de las interfaces para la gestión de profesores, en el cual los usuarios podrán realizar las acciones de listar, buscar, agregar, editar y eliminar un profesor. Además de las interfaces para gestionar asignaturas y títulos académicos correspondientes a un profesor.	

Tabla 4.46: Diseño de las interfaces para la gestión de profesores.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T20	Código de Historia: H9
Nombre de Tarea: Implementación de la gestión de profesores.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 28/04/2020	Fecha Fin: 29/04/2020

Programador Responsable: Wilmer Oñate
Descripción: Se llevará a cabo la implementación de las acciones de listar, buscar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee un profesor.

Tabla 4. 47: Implementación de la gestión de profesor.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T21	Código de Historia: H9
Nombre de Tarea: Asignación de asignaturas por profesor.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1,5
Fecha Inicio: 30/04/2020	Fecha Fin: 30/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, buscar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee una asignatura correspondiente a un profesor.	

Tabla 4.48: Implementación de la gestión de asignaturas por profesor.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T22	Código de Historia: H9
Nombre de Tarea: Asignación de títulos académicos por profesor.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 01/05/2020	Fecha Fin: 01/05/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, buscar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee un título académico correspondiente a un profesor.	

Tabla 4.49: Implementación de la asignación de títulos académicos.

Fuente: Elaborado por el investigador.

Historia de usuario: Gestión de Estudiantes.

TAREA	
Número de Tarea: T23	Código de Historia: H10
Nombre de Tarea: Diseño de las interfaces para la gestión de estudiantes.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 04/05/2020	Fecha Fin: 04/05/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará el diseño de las interfaces para la gestión de estudiantes, en el cual los usuarios podrán realizar las acciones de listar, buscar, agregar, editar y eliminar un estudiante. Además del diseño de la gestión de representantes, instituciones educativas y asignaturas asociadas a un estudiante.	

Tabla 4.50: Diseño de las interfaces para la gestión de estudiantes.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T24	Código de Historia: H10
Nombre de Tarea: Implementación de la gestión de estudiantes.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 05/05/2020	Fecha Fin: 05/05/2020
Programador Responsable: Wilmer Oñate	

Descripción: Se llevará a cabo la implementación de las acciones de listar, buscar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee un estudiante.

Tabla 4.51: Implementación de la gestión de estudiantes.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T25	Código de Historia: H10
Nombre de Tarea: Asignación de representantes por estudiante.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 06/05/2020	Fecha Fin: 06/05/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, buscar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee un representante correspondiente a un estudiante.	

Tabla 4.52: Asignación de representantes por estudiante.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T26	Código de Historia: H10
Nombre de Tarea: Asignación de instituciones educativas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 07/05/2020	Fecha Fin: 07/05/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, buscar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee una institución educativa correspondiente a un estudiante.	

Tabla 4.53: Implementación de la asignación de instituciones educativas.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T27	Código de Historia: H10
Nombre de Tarea: Asignación de asignaturas por estudiante.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 08/05/2020	Fecha Fin: 08/05/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, buscar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee una asignatura correspondiente a un estudiante.	

Tabla 4-54: Asignación de asignaturas por estudiante.

Fuente: Elaborado por el investigador.

Historia de usuario: Gestión de tema de una pregunta.

TAREA	
Número de Tarea: T28	Código de Historia: H11
Nombre de Tarea: Diseño de las interfaces para la gestión de tema de una pregunta.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,5
Fecha Inicio: 17/04/2020	Fecha Fin: 17/04/2020
Programador Responsable: Wilmer Oñate	

Descripción: Se realizará el diseño de las interfaces, en el cual los usuarios podrán realizar las acciones de listar, buscar, agregar, editar y eliminar un tema de pregunta.

Tabla 4.55: Diseño de las interfaces para la gestión de tema de una pregunta.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T29	Código de Historia: H11
Nombre de Tarea: Implementación de la gestión de tema de una pregunta.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,5
Fecha Inicio: 17/04/2020	Fecha Fin: 17/04/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, buscar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información que posee un tema de pregunta.	

Tabla 4.56: Implementación de la gestión de tema de una pregunta.

Fuente: Elaborado por el investigador.

Historia de usuario: Gestión de preguntas.

TAREA	
Número de Tarea: T30	Código de Historia: H12
Nombre de Tarea: Diseño de las interfaces para la gestión preguntas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 11/05/2020	Fecha Fin: 14/05/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará el diseño de las interfaces, en el cual los usuarios podrán realizar las acciones de listar, agregar, editar y eliminar una pregunta.	

Tabla 4.57: Diseño de las interfaces para la gestión de preguntas.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T31	Código de Historia: H12
Nombre de Tarea: Implementación de la gestión de preguntas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 15/05/2020	Fecha Fin: 20/05/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación listar, agregar, editar y eliminar una pregunta del repositorio, teniendo en cuenta el diseño y la información que posee una pregunta.	

Tabla 4.58: Implementación de la gestión de preguntas.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T32	Código de Historia: H12
Nombre de Tarea: Ingreso de fórmulas matemáticas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 21/05/2020	Fecha Fin: 22/05/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se implementará el ingreso y visualización de fórmulas matemáticas en el ingreso y edición de las preguntas del repositorio.	

Tabla 4.59: Diseño de las interfaces para el ingreso de fórmulas matemáticas.

Fuente: Elaborado por el investigador.

Historia de usuario: Gestión de usuarios.

TAREA	
Número de Tarea: T33	Código de Historia: H13
Nombre de Tarea: Diseño de las interfaces para la gestión usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 25/05/2020	Fecha Fin: 26/05/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará el diseño de las interfaces para el cambio, recuperación y reinicio de claves.	

Tabla 4.60: Diseño de las interfaces para la gestión de usuarios.
Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T34	Código de Historia: H13
Nombre de Tarea: Implementación de la creación de usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 27/05/2020	Fecha Fin: 28/05/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará la creación de usuarios al momento del registro de profesores y estudiantes.	

Tabla 4.61: Implementación de la creación de usuarios.
Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T35	Código de Historia: H13
Nombre de Tarea: Implementación del reinicio de clave.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 29/05/2020	Fecha Fin: 01/06/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se implementará el reinicio de clave para todos los usuarios registrados.	

Tabla 4.62: Implementación del reinicio de clave.
Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T36	Código de Historia: H13
Nombre de Tarea: Implementación de recuperación de clave.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 02/05/2020	Fecha Fin: 03/05/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de la recuperación de clave de usuario.	

Tabla 4.63: Implementación de recuperación de clave.
Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T37	Código de Historia: H13
Nombre de Tarea: Implementación de cambio de clave.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 04/05/2020	Fecha Fin: 05/05/2020
Programador Responsable: Wilmer Oñate	

Descripción: Se llevará a cabo la implementación del cambio de clave de un usuario.

Tabla 4.64: Implementación de cambio de clave.

Fuente: Elaborado por el investigador.

Historia de usuario: Inicio de sección.

TAREA	
Número de Tarea: T38	Código de Historia: H1
Nombre de Tarea: Diseño de la interfaz para inicio de sección.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22/06/2020	Fecha Fin: 22/06/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo el diseño de la pantalla de inicio de sección y la pantalla principal de la aplicación.	

Tabla 4.65: Diseño de la interfaz para inicio de sección.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T39	Código de Historia: H1
Nombre de Tarea: Implementación inicio de sección.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 23/06/2020	Fecha Fin: 24/06/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará la implementación del inicio de sección para los tres usuarios identificados (administrador, profesor y estudiante).	

Tabla 4.66: Implementación de cambio de clave.

Fuente: Elaborado por el investigador.

Historia de usuario: Gestión de Evaluaciones.

TAREA	
Número de Tarea: T40	Código de Historia: H14
Nombre de Tarea: Diseño de la interfaz para la gestión de evaluaciones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 08/06/2020	Fecha Fin: 09/06/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará el diseño de las interfaces la gestión de evaluaciones, en el cual los usuarios podrán realizar las acciones de listar, agregar, editar y eliminar una evaluación.	

Tabla 4.67: Diseño de las interfaces para la gestión de Evaluaciones.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T40	Código de Historia: H14
Nombre de Tarea: Diseño de la interfaz para la gestión de evaluaciones consolidadas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 10/06/2020	Fecha Fin: 10/06/2020
Programador Responsable: Wilmer Oñate	

Descripción: Se realizará el diseño de las interfaces la gestión de evaluaciones consolidadas, en el cual los usuarios podrán realizar las acciones de listar, agregar, editar y eliminar una evaluación.

Tabla 4.68: Diseño de las interfaces para la gestión de Evaluaciones.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T41	Código de Historia: H14
Nombre de Tarea: Implementación de la gestión de evaluaciones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 11/06/2020	Fecha Fin: 15/06/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información de una evaluación.	

Tabla 4.69: Diseño de las interfaces para la gestión de evaluaciones.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T41	Código de Historia: H14
Nombre de Tarea: Implementación de la gestión de evaluaciones consolidadas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 16/06/2020	Fecha Fin: 17/06/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, agregar, editar y eliminar, teniendo en cuenta el diseño y la información de una evaluación consolidada.	

Tabla 4.70: Diseño de las interfaces para la gestión de evaluaciones.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T42	Código de Historia: H14
Nombre de Tarea: Implementación de la acción de agregar una pregunta.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 18/06/2020	Fecha Fin: 18/06/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar, agregar, y eliminar una pregunta a una evaluación.	

Tabla 4.71: Implementación de la acción de agregar una pregunta.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T43	Código de Historia: H14
Nombre de Tarea: Activar una evaluación.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 19/06/2020	Fecha Fin: 19/06/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará la activación de una evaluación cambiando el estado de esta.	

Tabla 4.72: Activar una pregunta.

Fuente: Elaborado por el investigador.

Historia de usuario: Revisión de evaluaciones.

TAREA	
Número de Tarea: T44	Código de Historia: H15
Nombre de Tarea: Diseño de la interfaz para la revisión de evaluaciones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 24/06/2020	Fecha Fin: 24/06/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará el diseño de las interfaces para la revisión de evaluaciones tanto para los roles de profesor y estudiante.	

Tabla 4.73: Diseño de las interfaces para la revisión de evaluaciones.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T45	Código de Historia: H15
Nombre de Tarea: Implementación de revisión de evaluaciones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 25/06/2020	Fecha Fin: 26/06/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la implementación de las acciones de listar evaluación, teniendo en cuenta el diseño y la información de una evaluación realizadas por los estudiantes.	

Tabla 4.74: Implementación de revisión de evaluaciones.

Fuente: Elaborado por el investigador.

Historia de usuario: Iniciar Evaluación.

TAREA	
Número de Tarea: T46	Código de Historia: H16
Nombre de Tarea: Diseño de la interfaz para iniciar una evaluación.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 29/05/2020	Fecha Fin: 30/05/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará el diseño de las interfaces la gestión de evaluaciones, en el cual los usuarios podrán realizar las acciones de listar, agregar, editar y eliminar una evaluación.	

Tabla 4.75: Diseño de las interfaces para iniciar una evaluación.

Fuente: Elaborado por el investigador.

TAREA	
Número de Tarea: T47	Código de Historia: H16
Nombre de Tarea: Implementación de evaluaciones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 01/07/2020	Fecha Fin: 03/07/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo la activación e implementación de la evaluación tanto para profesores y estudiantes respectivamente.	

Tabla 4.76: Implementación de evaluaciones.

Fuente: Elaborado por el investigador.

Historia de usuario: Reporte horario de clases.

TAREA	
Número de Tarea: T48	Código de Historia: H17
Nombre de Tarea: Implementación de reporte de horario de clases.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 06/07/2020	Fecha Fin: 08/07/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se realizará el diseño e implementación de reporte de horario de clases de profesores y estudiantes.	

Tabla 4.77: Implementación de reporte de horario de clases.

Fuente: Elaborado por el investigador.

Historia de usuario: Reporte de evaluaciones.

TAREA	
Número de Tarea: T49	Código de Historia: H17
Nombre de Tarea: Implementación del reporte de evaluaciones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 08/07/2020	Fecha Fin: 10/07/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo el diseño e implementación del reporte de evaluación de asignaturas.	

Tabla 4.78: Implementación del reporte de evaluaciones.

Fuente: Elaborado por el investigador.

Historia de usuario: Reporte de evaluaciones por consolidado.

TAREA	
Número de Tarea: T50	Código de Historia: H18
Nombre de Tarea: Implementación del reporte de evaluaciones consolidadas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 13/07/2020	Fecha Fin: 15/07/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo el diseño e implementación del reporte de evaluación consolidadas.	

Tabla 4.79: Implementación del reporte de evaluaciones consolidadas.

Fuente: Elaborado por el investigador.

Historia de usuario: Ficha de Inscripción.

TAREA	
Número de Tarea: T51	Código de Historia: H19
Nombre de Tarea: Implementación de la ficha de inscripción.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 16/05/2020	Fecha Fin: 17/05/2020
Programador Responsable: Wilmer Oñate	
Descripción: Se llevará a cabo el diseño e implementación del reporte de la ficha de inscripción del estudiante.	

Tabla 4.80: Implementación de la ficha de inscripción.

Fuente: Elaborado por el investigador.

4.5.3.3. Plan de Entrega del Proyecto

Al definir las historias de usuario y las tareas para el desarrollo de la Aplicación Web Progresiva, se ha elaborado un plan de entrega que un listado de las diferentes actividades a desarrollar en un tiempo establecido. En el plan de entrega se muestra las historias de usuario que se llevarán a cabo en cada iteración.

Para la elaboración del plan de entrega se ha tomado en cuenta el esfuerzo de cada historia de usuario y se ha agrupado en 4 módulos de desarrollo (administración, profesores, estudiantes, reportes). Ver Tabla 4.85

4.5.4. Fase: Diseño

La metodología XP sugiere el desarrollo del diseño sea simple, sencillo, entendible e implementable lo que generara un costo menos en tiempo y esfuerzo de desarrollo.

4.5.4.1. Tarjetas CRC (Clase-Responsabilidad-Colaboración)

La metodología XP recomienda el uso de las tarjetas CRC (Clase-Responsabilidad-Colaboración) que permiten realizar un diseño orientado a objetos. En la parte superior de la tarjeta CRC se escribe el nombre de la clase a la que pertenece el objeto, en una columna a la izquierda se escriben las responsabilidades a cumplir y en la columna a la derecha las clases que colaboran para cumplir las responsabilidades.

En las siguientes tablas se muestran las diferentes clases expuestas en las tarjetas CRC:

Clase: Asignatura	
Responsabilidades	Colaboradores
Agregar asignatura Editar asignatura Eliminar asignatura Listar asignatura	

Tabla 4.81: Tarjeta CRC - Asignatura.

Fuente: Elaborado por el investigador

Clase: Módulo	
Responsabilidades	Colaboradores
Agregar módulo. Editar módulo. Eliminar módulo. Listar módulos.	

Tabla 4.82: Tarjeta CRC - Módulo.

Fuente: Elaborado por el investigador

Modulo	Historias	Nombre Historia	Iteraciones	Prioridad	Esfuerzo	Fecha Inicio	Fecha Fin
Administración	Historia 2	Gestión de Asignaturas.	1	Media	2	01/04/2020	03/04/2020
	Historia 3	Gestión de Módulos.	1	Media	2	06/04/2020	08/04/2020
	Historia 4	Gestión de Cursos.	1	Media	5	09/04/2020	10/04/2020
	Historia 5	Gestión de Títulos Académicos.	1	Media	2	13/04/2020	14/04/2020
	Historia 6	Gestión de Instituciones Educativas.	1	Baja	2	15/04/2020	16/04/2020
	Historia 7	Gestión de Personas.	2	Alta	3	20/04/2020	22/04/2020
	Historia 8	Gestión de Representantes.	2	Media	2	23/04/2020	24/04/2020
	Historia 9	Gestión de Profesores.	2	Media	5	27/04/2020	01/05/2020
	Historia 10	Gestión de Estudiantes.	2	Media	5	04/05/2020	08/05/2020
	Profesores Estudiantes	Historia 11	Gestión de tema de una pregunta.	1	Media	2	17/04/2020
Historia 12		Gestión de preguntas.	3	Alta	8	11/05/2020	22/05/2020
Historia 13		Gestión de usuarios.	3	Alta	8	25/05/2020	05/06/2020
Historia 14		Gestión de Evaluaciones.	3	Alta	13	08/05/2020	19/05/2020
Historia 1		Inicio de sección.	3	Alta	3	22/05/2020	24/05/2020
Reportes	Historia 15	Revisión de evaluaciones por estudiante.	4	Media	3	24/05/2020	26/05/2020
	Historia 16	Iniciar Evaluación.	4	Alta	5	29/05/2020	03/06/2020
	Historia 17	Reporte de horario de clases.	4	Alta	3	06/06/2020	10/06/2020
	Historia 18	Reporte de evaluaciones por asignatura y consolidado.	4	Alta	3	13/06/2020	15/06/2020
	Historia 19	Ficha de Inscripción del estudiante.	4	Media	3	16/06/2020	17/06/2020

Tabla 4.83: Plan de entrega del proyecto.

Fuente: Elaborado por el investigador.

Clase: Título Académica	
Responsabilidades	Colaboradores
Agregar título académico. Editar título académico. Eliminar título académico. Listar título académico.	

Tabla 4.84: Tarjeta CRC – Título Académico.

Fuente: Elaborado por el investigador.

Clase: Representante	
Responsabilidades	Colaboradores
Agregar representante. Eliminar representante. Listar representante.	

Tabla 4.85: Tarjeta CRC - Representante.

Fuente: Elaborado por el investigador.

Clase: Curso	
Responsabilidades	Colaboradores
Agregar Curso Listar Curso Editar Curso Eliminar Curso	

Tabla 4.86: Tarjeta CRC - Curso.

Fuente: Elaborado por el investigador

Clase: Institución Educativa	
Responsabilidades	Colaboradores
Agregar institución educativa. Editar institución educativa. Eliminar institución educativa. Listar institución educativa.	

Tabla 4.87: Tarjeta CRC – Institución Educativa.

Fuente: Elaborado por el investigador.

Clase: Tema Pregunta.	
Responsabilidades	Colaboradores
Agregar tema pregunta. Editar tema pregunta. Eliminar tema pregunta. Listar tema pregunta.	

Tabla 4.88: Tarjeta CRC - Pregunta.

Fuente: Elaborado por el investigador.

Clase: Persona	
Responsabilidades	Colaboradores
Agregar persona. Editar persona. Eliminar persona. Listar personas.	

Tabla 4.89: Tarjeta CRC - Persona.

Fuente: Elaborado por el investigador.

Clase: Teléfono Persona	
Responsabilidades	Colaboradores
Agregar teléfono persona. Editar teléfono persona. Eliminar teléfono persona. Listar teléfono persona.	Persona.

Tabla 4.90: Tarjeta CRC – Teléfono persona.

Fuente: Elaborado por el investigador.

Clase: Estudiante	
Responsabilidades	Colaboradores
Agregar estudiante. Eliminar estudiante. Listar estudiante.	Account.

Tabla 4.91: Tarjeta CRC - Estudiante.

Fuente: Elaborado por el investigador.

Clase: Profesor	
Responsabilidades	Colaboradores
Agregar profesor. Eliminar profesor. Listar profesor.	Account.

Tabla 4.92: Tarjeta CRC - Profesor.

Fuente: Elaborado por el investigador.

Clase: Profesor - Asignatura	
Responsabilidades	Colaboradores
Agregar profesor asignatura. Editar profesor asignatura. Eliminar profesor asignatura. Listar profesor asignatura.	Profesor. Asignatura.

Tabla 4.93: Tarjeta CRC – Profesor - Asignatura.

Fuente: Elaborado por el investigador.

Clase: Profesor - Título Académico	
Responsabilidades	Colaboradores
Agregar profesor - título académico. Editar profesor - título académico. Eliminar profesor - título académico. Listar profesor - título académico.	Profesor. Título académico.

Tabla 4.94: Tarjeta CRC – Profesor – Título Académico.

Fuente: Elaborado por el investigador.

Clase: Curso Detalle	
Responsabilidades	Colaboradores
Agregar curso – detalle. Editar curso – detalle. Eliminar curso – detalle. Listar curso – detalle.	Curso. Asignatura. Profesor.

Tabla 4.95: Tarjeta CRC – Curso Detalle.

Fuente: Elaborado por el investigador.

Clase: Horario	
Responsabilidades	Colaboradores
Agregar horario. Editar horario. Eliminar horario. Listar horario.	Curso Detalle.

Tabla 4.96: Tarjeta CRC - Horario.
Fuente: Elaborado por el investigador.

Clase: Estudiante - Institución.	
Responsabilidades	Colaboradores
Agregar estudiante - institución. Editar estudiante - institución. Eliminar estudiante - institución. Listar estudiante - institución.	Estudiante. Institución.

Tabla 4.97: Tarjeta CRC – Estudiante - Institución.
Fuente: Elaborado por el investigador.

Clase: Estudiante - Representante.	
Responsabilidades	Colaboradores
Agregar estudiante - representante. Editar estudiante - representante. Eliminar estudiante - representante. Listar estudiante - representante.	Estudiante. Representante.

Tabla 4.98: Tarjeta CRC – Estudiante - Representante.
Fuente: Elaborado por el investigador.

Clase: Estudiante - Curso Detalle	
Responsabilidades	Colaboradores
Agregar estudiante - curso detalle. Editar estudiante - curso detalle. Eliminar estudiante - curso detalle. Listar estudiante - curso detalle.	Estudiante. Curso Detalle.

Tabla 4.99: Tarjeta CRC – Estudiante – Curso Detalle.
Fuente: Elaborado por el investigador.

Clase: Evaluación	
Responsabilidades	Colaboradores
Agregar evaluación. Editar evaluación. Eliminar evaluación. Listar evaluación.	Curso Detalle.

Tabla 4.100: Tarjeta CRC - Evaluación.
Fuente: Elaborado por el investigador.

Clase: Pregunta	
Responsabilidades	Colaboradores
Agregar pregunta. Editar pregunta. Eliminar pregunta. Listar pregunta.	Tema pregunta, Tipo pregunta, Ítem, Imagen – Ítem, Imagen – Pregunta.

Tabla 4.101: Tarjeta CRC - Pregunta.
Fuente: Elaborado por el investigador.

Clase: Evaluación	
Responsabilidades	Colaboradores
Agregar y listar tema pregunta. Editar tema pregunta. Eliminar tema pregunta.	

Tabla 4.102: Tarjeta CRC - Evaluación.
Fuente: Elaborado por el investigador.

Clase: Evaluación - Pregunta	
Responsabilidades	Colaboradores
Agregar y listar evaluación - pregunta. Editar evaluación - pregunta. Eliminar evaluación - pregunta.	Evaluación. Pregunta.

Tabla 4.103: Tarjeta CRC – Evaluación - Pregunta.
Fuente: Elaborado por el investigador.

Clase: Curso Vista	
Responsabilidades	Colaboradores
Listar curso por estudiante. Listar curso por profesor.	Curso Detalle, Estudiante, Profesor.

Tabla 4.104: Tarjeta CRC – Curso Vista.
Fuente: Elaborado por el investigador.

Clase: Estudiante - Evaluación	
Responsabilidades	Colaboradores
Agregar estudiante - evaluación. Listar estudiante - evaluación.	Estudiante. Evaluación.

Tabla 4.105: Tarjeta CRC – Estudiante - Evaluación.
Fuente: Elaborado por el investigador.

Clase: Pregunta - Estudiante	
Responsabilidades	Colaboradores
Agregar estudiante - evaluación. Listar estudiante - evaluación.	Estudiante - Evaluación. Pregunta.

Tabla 4.106: Tarjeta CRC – Pregunta - Estudiante.
Fuente: Elaborado por el investigador.

Clase: Evaluación - Consolidado	
Responsabilidades	Colaboradores
Agregar y listar evaluación consolidado. Eliminar evaluación consolidado.	Curso detalle.

Tabla 4.107: Tarjeta CRC – Evaluación - Consolidado.
Fuente: Elaborado por el investigador.

Clase: Evaluación Consolidado - Detalle	
Responsabilidades	Colaboradores
Listar evaluación consolidado - detalle.	Evaluación - consolidado. Evaluación.

Tabla 4.108: Tarjeta CRC – Evaluación Consolidado - Detalle.
Fuente: Elaborado por el investigador.

Clase: Evaluación Consolidado - Estudiante	
Responsabilidades	Colaboradores
Agregar estudiante - evaluación. Listar estudiante - evaluación.	Estudiante. Evaluación - consolidado.

Tabla 4.109: Tarjeta CRC – Evaluación Consolidado - Estudiante.
Fuente: Elaborado por el investigador.

Clase: Pregunta Consolidado	
Responsabilidades	Colaboradores
Agregar y listar pregunta - consolidado. Listar pregunta - consolidado.	Evaluación Consolidado – Estudiante. Pregunta.

Tabla 4.110: Tarjeta CRC – Pregunta Consolidado.
Fuente: Elaborado por el investigador.

Clase: Account	
Responsabilidades	Colaboradores
Ingreso y salida. Agregar usuarios. Editar usuarios. Listar usuarios.	Persona.

Tabla 4.111: Tarjeta CRC - Account.
Fuente: Elaborado por el investigador.

Clase: Estado Civil	
Responsabilidades	Colaboradores
Listar estado civil.	

Tabla 4.112: Tarjeta CRC – Estado Civil.
Fuente: Elaborado por el investigador

Clase: Instrucción Académica	
Responsabilidades	Colaboradores
Listar instrucción académica.	

Tabla 4.113: Tarjeta CRC – Instrucción Académica.
Fuente: Elaborado por el investigador.

Clase: Operadora Telefónica	
Responsabilidades	Colaboradores
Listar operadora telefónica.	

Tabla 4.114: Tarjeta CRC – Operadora Telefónica.
Fuente: Elaborado por el investigador.

Clase: Tipo Identificación	
Responsabilidades	Colaboradores
Listar tipo identificación.	

Tabla 4.115: Tarjeta CRC – Tipo Identificación.
Fuente: Elaborado por el investigador.

Clase: Tipo Pregunta	
Responsabilidades	Colaboradores
Listar tipo pregunta.	

Tabla 4.116: Tarjeta CRC – Tipo Pregunta.
Fuente: Elaborado por el investigador.

Clase: Tipo Teléfono	
Responsabilidades	Colaboradores
Listar tipo teléfono.	

Tabla 4.117: Tarjeta CRC – Tipo Teléfono.

Fuente: Elaborado por el investigador.

4.5.4.2. Diseño de la Base de Datos

La Base de datos para la Aplicación Web Progresiva se desarrolló en SQL Server 2017, donde se tomó toda la información y procesos brindada por el cliente y se llevó acabo el modelado de la base de datos. En la siguiente imagen se expone el modelo entidad relación de la base de datos desarrollada. Ver Figura 4.11- Figura 4.13

4.5.5. Fase: Codificación

4.5.5.1. Desarrollo de la Aplicación Web Progresiva

La Aplicación Web Progresiva será desarrollada bajo las tecnologías ASP.NET CORE y Angular 8, para el desarrollo del servidor y cliente respectivamente, mediante un el patrón de arquitectura de software MVC.

Para la construcción del servidor se utilizó ASP.NET CORE 3.1.1, se creó un proyecto ASP.NET CORE MVC plantilla API, donde se eliminó la vista del modelo, la cual será remplazada por el proyecto desarrollado en Angular 8.

4.5.4.1. Desarrollo del Servidor

La solución de servidor contiene las siguientes carpetas:

Modelo: El modelo posee la representación de los datos y la lógica de negociación, la representación de los datos fue creado mediante Entity Framework Core 3.1.1 que es el ORM (Mapeo Objeto - Relación) por defecto de Asp.Net Core.

•**Entity Framework Core:** Con Entity Framework Core se generó automáticamente el acceso a los datos a partir de una base de datos existente. Este método de creación de datos es conocido como Database First.

Para la creación del modelo de acceso de datos se utilizó Scaffold-DbContext mediante la siguiente sentencia que se muestra a continuación:

```
PM> Scaffold-DbContext "Server=localhost;Database=Smartel1;User ID=sa;Password= "
Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models
```

Figura 4.11: Creación del modelo mediante Entity Framework Core .

Fuente: Elaborado por el investigador.

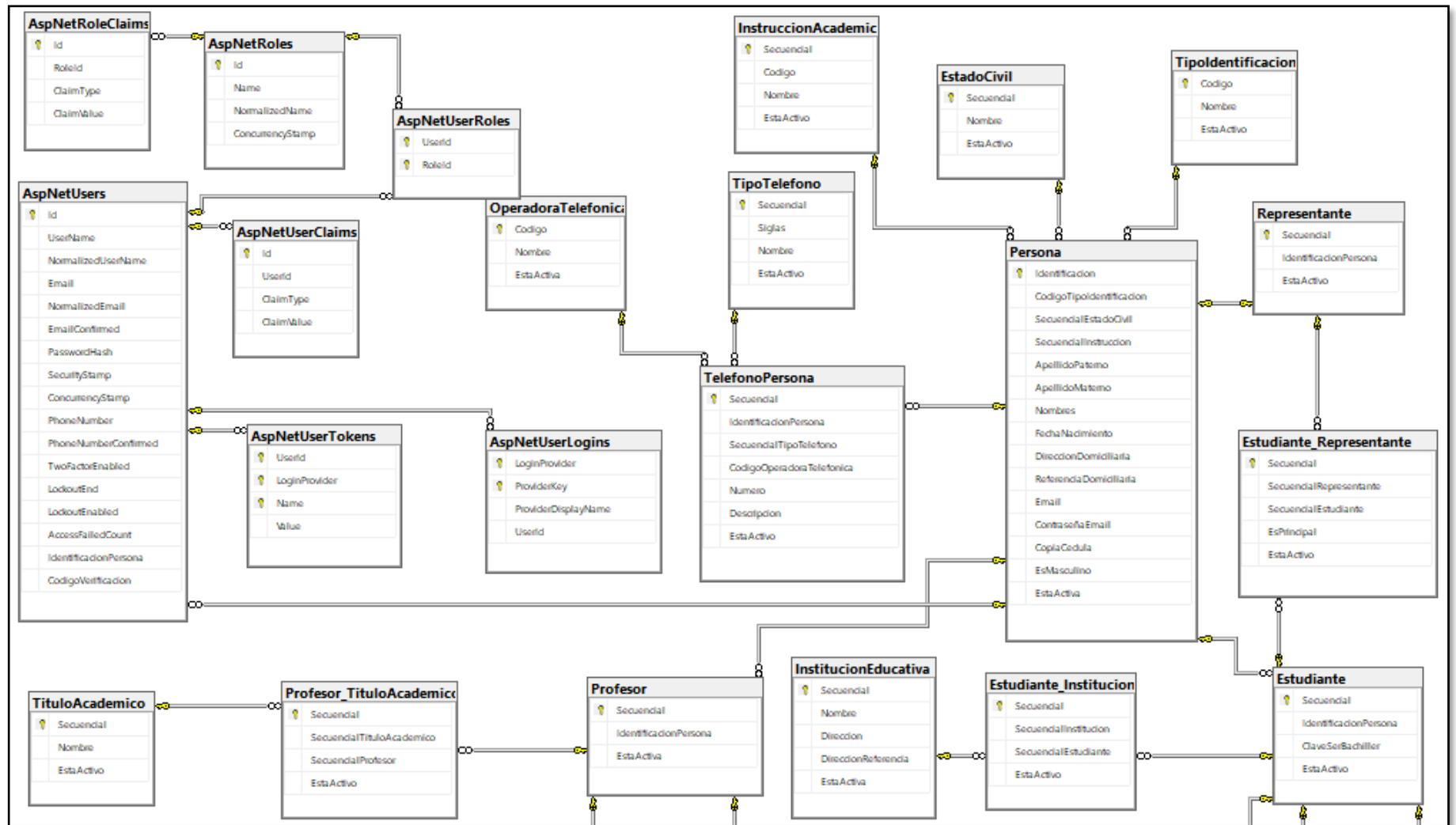


Figura 4.12: Diseño de la Base de Datos parte 1.

Fuente: Elaborado por el investigador.

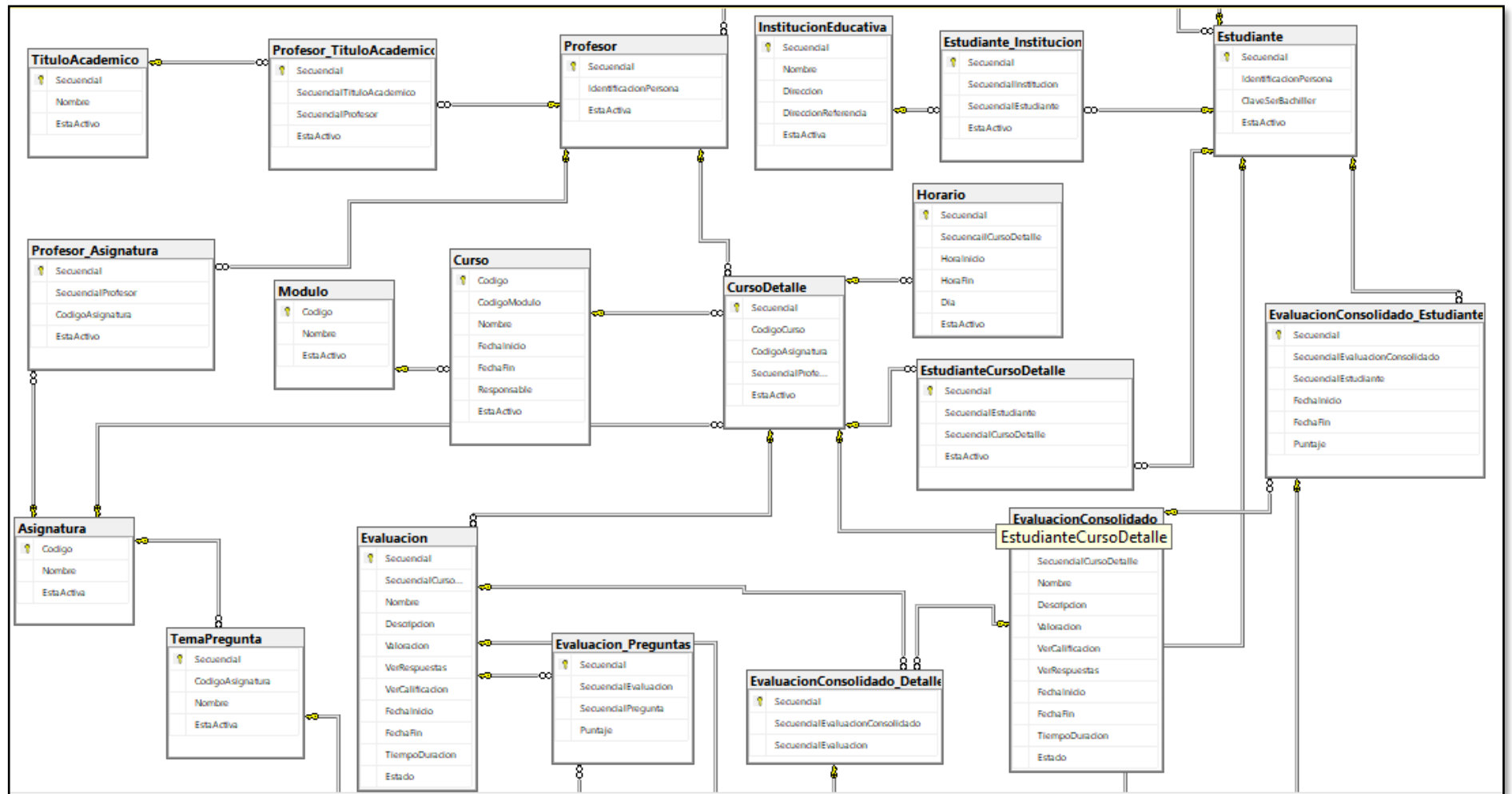


Figura 4.13: Diseño de la Base de Datos parte 2.
Fuente: Elaborado por el investigador.

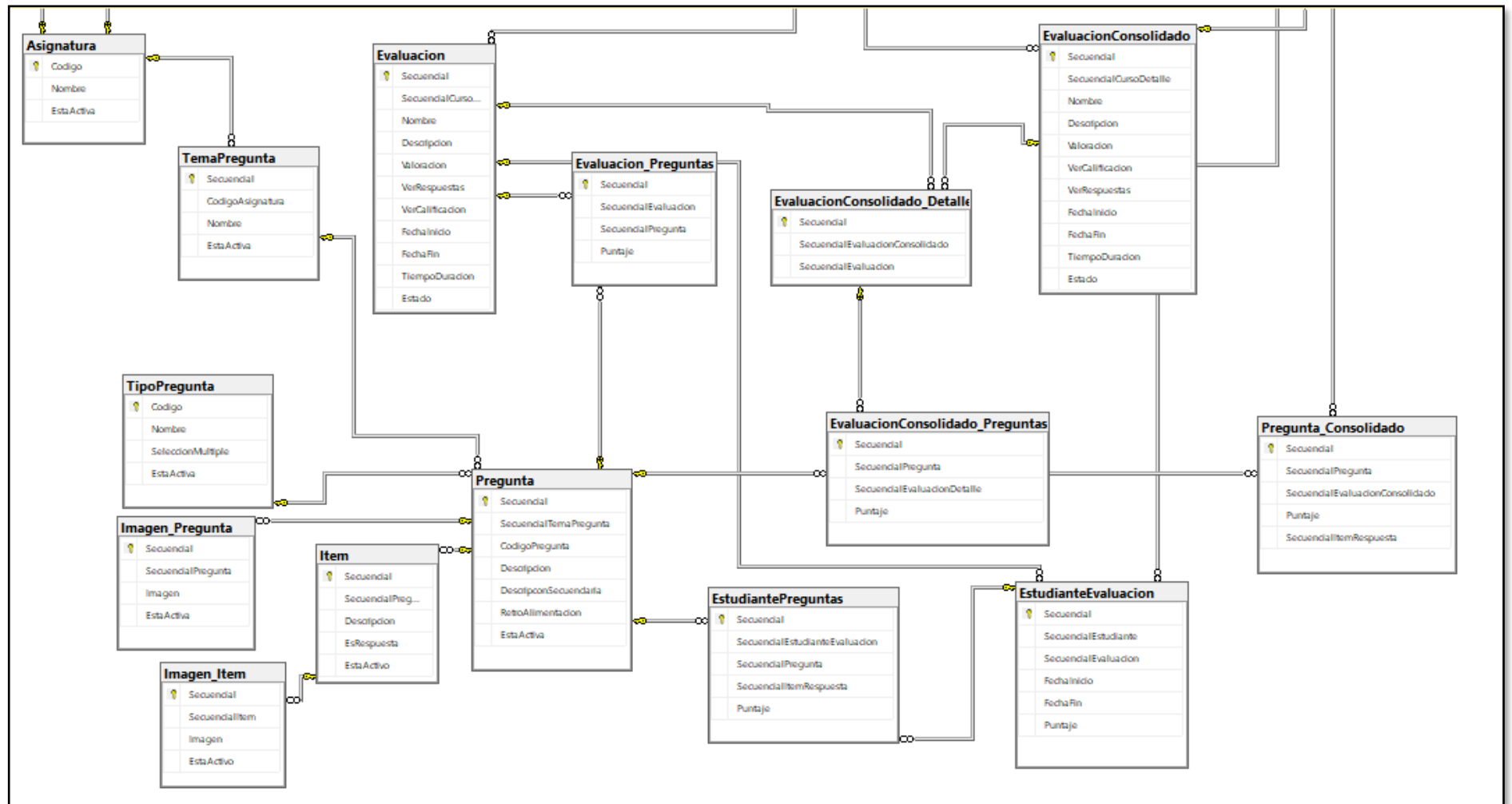


Figura 4.14: Diseño de la Base de Datos parte 3.
Fuente: Elaborado por el investigador.

La sentencia Scaffold-DbContext hace referencia a los siguientes parámetros:

- Cadena de conexión a la base de datos.
- Proveedor por usar.
- Directorio en el cual se guardará los diferentes archivos.

Entity Framework Core genera los diferentes archivos de los cuales, la mayoría son clases que representan a las entidades de la base de datos y la clase llamada SmartelContext que posee los diferentes DbSet de las entidades mapeadas.

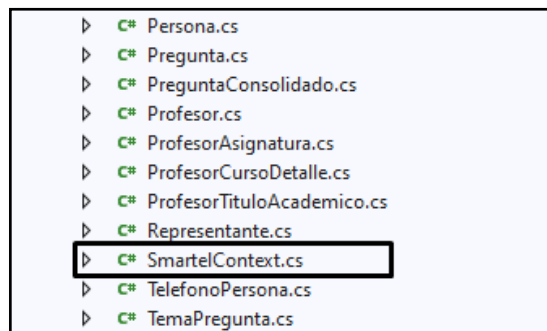


Figura 4.15: Clases creadas por Entity Framework Core.

Fuente: Elaborado por el investigador.

A continuación, se muestra el código generado por Entity Framework Core para la clase Representantes.cs.

```
1. using System;
2. using System.Collections.Generic;
3. namespace ServerApp.Models
4. {
5.     public partial class Representante
6.     {
7.         public Representante()
8.         {
9.             EstudianteRepresentante = new
10.             HashSet<EstudianteRepresentante>();
11.         }
12.         public int Secuencial { get; set; }
13.         public string IdentificacionPersona { get; set; }
14.         public bool EstaActivo { get; set; }
15.         public virtual Persona
16.         IdentificacionPersonaNavigation { get;
17.         set; }
18.         public virtual ICollection<EstudianteRepresentante>
19.         EstudianteRepresentante { get; set; }
20.     }
```

A continuación, se muestra una parte del código generado por Entity Framework Core para la clase SmartelContext.cs. La clase SmartelContext posee las diferentes DbSet de las entidades mapeadas.

```

80 referencias
public partial class SmartelContext : IdentityDbContext<ApplicationUser>
{
    0 referencias
    public SmartelContext()
    {
    }
    0 referencias
    public SmartelContext(DbContextOptions<SmartelContext> options) {...}
    23 referencias
    public virtual DbSet<Asignatura> Asignatura { get; set; }
    30 referencias
    public virtual DbSet<Curso> Curso { get; set; }
    31 referencias
    public virtual DbSet<CursoDetalle> CursoDetalle { get; set; }
    3 referencias
    public virtual DbSet<EstadoCivil> EstadoCivil { get; set; }
    23 referencias
    public virtual DbSet<Estudiante> Estudiante { get; set; }
    17 referencias
    public virtual DbSet<EstudianteCursoDetalle> EstudianteCursoDetalle { get; set; }
    13 referencias
    public virtual DbSet<EstudianteEvaluacion> EstudianteEvaluacion { get; set; }
    14 referencias
    public virtual DbSet<EstudianteInstitucion> EstudianteInstitucion { get; set; }
    7 referencias
    public virtual DbSet<EstudiantePreguntas> EstudiantePreguntas { get; set; }
    11 referencias
    public virtual DbSet<EstudianteRepresentante> EstudianteRepresentante { get; set; }
    20 referencias
    public virtual DbSet<Evaluacion> Evaluacion { get; set; }
    13 referencias
    public virtual DbSet<EvaluacionConsolidado> EvaluacionConsolidado { get; set; }
    8 referencias
    public virtual DbSet<EvaluacionConsolidadoDetalle> EvaluacionConsolidadoDetalle { get; set; }
    13 referencias
    public virtual DbSet<EvaluacionConsolidadoEstudiante> EvaluacionConsolidadoEstudiante { get; set; }
}

```

Figura 4.16: Parte del código de la clase SmartelContext.

Fuente: Elaborado por el investigador.

Controlador: La carpeta controladora posee los diferentes servicios que serán expuestos al cliente mediante los métodos estándar de HTTP como son Get, Post, Put y Delete [34].

A continuación, se muestra el método HttpGet del controlador Representante el mismo que filtra las personas activas y retorna una entidad RepresentanteIS de la clase RepresentanteDTO.

```

1. [HttpGet]
2. [QueryStringConstrainAttribute("estaActivo", true)]
3. public IEnumerable<RepresentanteIS> GetRepresentante(bool estaActivo)
4. {
5.     var representante = _context.Representante;
6.     var representanteDTO = from aux in representante
7.     where aux.EstaActivo == estaActivo
8.     orderby aux.Secuencial ascending
9.     select new RepresentanteIS()
10.    {
11.        Secuencial = aux.Secuencial,
12.        IdentificacionPersona = aux.IdentificacionPersona,
13.        EstaActivo = aux.EstaActivo,
14.        NombresPersona = aux.IdentificacionPersonaNavigation.Nombres,
15.        ApellidoPaternoPersona =
16.        aux.IdentificacionPersonaNavigation.ApellidoPaterno,
17.        ApellidoMaternoPersona =
18.        aux.IdentificacionPersonaNavigation.ApellidoMaterno

```

```

19.     };
20.     return representanteDTO;
21. }

```

El método `HttpDelete` del Controlador `Representante` recibe del cliente un secuencial, procede a buscar el respectivo representante en la base de datos para posteriormente realizar un borrado lógico. En caso de no encontrar el representante o de ocurrir algún error en el proceso se genera una excepción a través de la clase `NotFoundCustomException` que recibe como parámetros un mensaje, una descripción y un título. El funcionamiento de la clase `NotFoundCustomException` se estará tratando más adelante.

```

1.  [[HttpDelete()]]
2.  [QueryStringConstrainAttribute("secuencial", true)]
3.  public async Task<IActionResult> DeleteRepresentante(int secuencial)
4.  {
5.      if (!ModelState.IsValid)
6.          throw new NotFoundCustomException("Error en los datos enviados",
7.          $"Por favor revise el parámetro secuencial enviado",
8.          "Representante");
9.      try
10.     {
11.         var representante = await _context.Representante.
12.         FindAsync(secuencial);
13.         if (representante == null)
14.             throw new NotFoundCustomException("Error al cargar datos",
15.             "No se encontraron registros para el secuencial: "+
16.             secuencial.ToString(), "Representante");
17.         representante.EstaActivo = false;
18.         _context.Representante.Update(representante);
19.         await _context.SaveChangesAsync();
20.         return Ok(representante);
21.     }
22.     catch (Exception ex)
23.     {
24.         throw new NotFoundCustomException("Error
25.         al Eliminar", ex.Message,
26.         "Representante");
27.     }

```

DTO: Los objetos de transferencia de datos permitirá definir la información de entrada, salida, entrada y salida desde el controlador y hacia el controlado con el objetivo de evitar referencias circulares, ocultar o agregar datos extras a la información enviada por el modelo.

El archivo `RepresentantesDTO.cs` expone dos clases `RepresentanteIE` y `RepresentanteIS` que representa la información de entrada y salida para el Controlador

Representante. En el siguiente fragmento de código se expone el archivo RepresentantesDTO.cs.

```
1. public class RepresentanteIE
2. {
3.     public int Secuencial { get; set; }
4.     public string IdentificacionPersona { get; set; }
5.     public bool EstaActivo { get; set; }
6. }
7. public class RepresentanteIS
8. {
9.     public int Secuencial { get; set; }
10.    public string IdentificacionPersona { get; set; }
11.    public bool EstaActivo { get; set; }
12.    public string NombresPersona { get; set; }
13.    public string ApellidoPaternoPersona { get; set; }
14.    public string ApellidoMaternoPersona { get; set; }
15. }
16. }
```

- **Manejo de Excepciones**

El manejo de excepciones en el servidor se realizó mediante la creación de la clase NotFoundCustomException que a su vez hereda de la clase BaseCustomException. BaseCustomException hereda de la clase Exception y posee 3 atributos que son título, descripción y código de error. En los diferentes controladores generaremos una nueva excepción a través de la clase NotFoundCustomException, esta clase envía los mensajes que visualizara el usuario al generarse un error.

```
1. public class NotFoundCustomException : BaseCustomException
2.     {
3.         public NotFoundCustomException(string message, string des
4.             cription,
5.             string titulo): base(message, description, titulo,
6.             (int)HttpStatusCode.NotFound)
7.     }
8. }
```

```
1. public class BaseCustomException : Exception
2.     {
3.         private int _code;
4.         private string _description;
5.         private string _titulo;
6.         public string Titulo
7.             {
8.                 get => _titulo;
9.             }
10.        public int Code
11.            {
12.                get => _code;
13.            }
14.        public string Description
15.            {
```

```

16.         get => _description;
17.     }
18.
19.     public BaseCustomException(string message, string description
20.     '
21.         string titulo, int code) : base(me
22.     ssage)
23.     {
24.         _code = code;
25.         _description = description;
26.         _titulo = titulo;
27.     }
28. }

```

- **Manejo de Usuarios**

Para la gestión de usuarios se utilizó ASP.NET CORE IDENTITY, la cual facilitó la creación de las tablas y clases necesarias para la autenticación de la PWA. ASP.NET CORE IDENTITY crea un controlador llamado AccountController, en la cual, llevará a cabo las diferentes acciones sobre los usuarios. AccountController posee referencia a las clases UserManager y SignInManager, estas dos clases se utiliza para implementar la lógica de autenticación y la lógica para trabajar con los usuarios respectivamente.

Para realizar la incorporación de datos adicionales al modelo se creó la clase ApplicationUser en el modelo, ApplicationUser hereda de la clase IdentityUser. En esta clase se incorpora los datos y claves foráneas adicionales como se muestra a continuación:

```

1. public class ApplicationUser : IdentityUser
2. {
3.     [Column(TypeName = "varchar(10)")]
4.     public string IdentificacionPersona { get; set; }
5.     [Column(TypeName = "varchar(10)")]
6.     public string CodigoVerificacion { get; set; }
7.
8.     [ForeignKey("IdentificacionPersona")]
9.     public virtual Persona { get; set; }
10. }

```

Posteriormente se realizará una actualización a la base de datos, en este proceso se creará los diferentes atributos agregados para la tabla AspNetUsers en la base de datos correspondiente.

Autenticación

La autenticación de los usuarios en la Aplicación Web Progresiva se la realiza mediante JWT (JSON Web Token) a través del método llamado BuildToken

ubicado en AccountCroller, JWT es generado en el servidor y válido en el cliente.

A continuación, se muestra el método de creación del JWT:

```
1. private IActionResult BuildToken(ApplicationUser userInfo,
2.   IList<string> role)
3. {
4.   IdentityOptions _options = new IdentityOptions();
5.   var claims = new[]
6.   {
7.     new Claim(JwtRegisteredClaimNames.UniqueName,
8.       userInfo.UserName),
9.     new Claim("role" , role.FirstOrDefault()),
10.    new Claim("IdentificcionPersona",
11.      userInfo.IdentificacionPersona),
12.    new Claim(JwtRegisteredClaimNames.Jti,
13.      Guid.NewGuid().ToString())
14.  };
15. var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes("****
16.   ****"));
17. var creds = new SigningCredentials(key,
18.   SecurityAlgorithms.HmacSha256);
19. var expiration = DateTime.UtcNow.AddDays(1);
20. JwtSecurityToken token = new JwtSecurityToken(
21.   issuer: "yourdomain.com",
22.   audience: "yourdomain.com",
23.   claims: claims,
24.   expires: expiration,
25.   signingCredentials: creds);
26. return Ok(new
27. {
28.   token = new JwtSecurityTokenHandler().WriteToken(token),
29.   expiration = expiration
30. });
31. }
```

En el método BuildToken se crea el JwtSecurityToken el mismo que estará conformado por las diferentes Claim con la información del usuario, una fecha de expiración, una credencial.

- **Configuración de Cors (Access-Control-Allow-Origin)**

Es necesario habilitar el intercambio de recursos de origen cruzado ya que tanto el cliente y el servidor estarán alojados en direcciones diferentes. Para habilitar Cors se agregar la siguiente porción de código en la clase Startup en el método “ConfigureServices”, este método se encarga de configurar los servicios de la aplicación.

```
1. services.AddCors(options =>
2.   {
3.     options.AddPolicy(
4.       "CorsPolicy",
```

```

5.         builder => builder.WithOrigins("https://localhost
      :8080")
6.         .AllowAnyHeader()
7.         .WithMethods("GET", "HEAD", "POST", "PUT", "DELETE",
8.             "OPTIONS", "TRACE")
9.         .AllowCredentials();
10.});

```

El anterior fragmento de código permite realizar solicitudes “Get”, ”Head”, ”Post”, ”Put”, ”Delete”, “Options”, ”Trace” procedentes del origen “https://localhost:8080” que es el origen del cliente en un ambiente de pruebas.

Posteriormente se deberá añadir el uso de Cors en el método “Configure” de la clase Startup.

```

1. app.UseCors("CorsPolicy");

```

4.5.4.2. Desarrollo del Cliente

El cliente se llevó a cabo mediante Angular 8 el proyecto está formado por componentes, servicios y modelos y a su vez estos elementos están agrupados en módulos, los que son enlazados mediante un sistema de enrutamiento.

- **Desarrollo de la PWA:**

La librería @angular/pwa permite crear una Aplicación Web Progresiva a partir de una aplicación angular, @angular/pwa instala y configura los archivos necesarios para convertir una aplicación angular en una PWA. Esta librería podemos usarla mediante Angular CLI (Command-Line Interface) a través del siguiente comando:

```

PS E:\AplicacionPWA\ClientApp> ng new SmartelClient
--directory /ClientApp --routing true --style scss_

```

Figura 4.17: Creación de un proyecto angular mediante Angular CLI.
Fuente: Elaborado por el investigador.

```

PS E:\AplicacionPWA\ClientApp> ng add @angular/pwa

```

Figura 4.18: Creación de una PWA en base a un proyecto existente
Fuente: Elaborado por el investigador.

@angular/pwa crea iconos y configura los archivos ngsw-config.json, manifest.json y ngsw-worker.js, mientras que los archivos angulares.json, app.module.ts, index.html, y package.json son modificados. A continuación, se describe los diferentes archivos creados y modificados por la librería @angular/pwa:

Archivo package.json: El archivo es modificado por la librería @angular/pwa para agregar la dependencia "@ angular / serviceworker" como se muestra en la Figura 4.19.

```
package.json - x
Esquema: https://json.schemastore.org/package
10   "e2e": "ng e2e"
11 },
12   "private": true,
13   "dependencies": {
14     "@angular/animations": "~8.1.2",
15     "@angular/cdk": "~8.2.3",
16     "@angular/common": "~8.1.2",
17     "@angular/compiler": "~8.1.2",
18     "@angular/core": "~8.1.2",
19     "@angular/forms": "~8.1.2",
20     "@angular/material": "^8.2.3",
21     "@angular/platform-browser": "~8.1.2",
22     "@angular/platform-browser-dynamic": "~8.1.2",
23     "@angular/pwa": "^0.803.25",
24     "@angular/router": "~8.1.2",
25     "@angular/service-worker": "~8.1.2",
26     "angular-bootstrap-md": "^8.9.0",
27     "angular-popover": "^2.0.1",
28     "bootstrap": "^4.4.1",
29     "hammerjs": "^2.0.8",
30     "jquery": "^3.4.1",
31     "material-design-icons": "^3.0.1",
```

Figura 4.19: Configuración del archivo package.json.
Fuente: Elaborado por el investigador.

Archivo manifest.webmanifest: Es un archivo JSON que contiene metadatos que le indica al navegador y a el sistema operativo de qué manera tratar a la aplicación, posee información como nombre , iconos , descripción, etc. Además, ofrece al usuario la opción de instalar la Aplicación Web Progresiva.[35]

A continuación, se muestra la configuración del archivo manifest.webmanifest.

```
1. "short_name": "Smartel",
2.   "theme_color": "#40749c",
3.   "background_color": "#fafafa",
4.   "display": "standalone",
5.   "scope": "/",
6.   "start_url": "./",
7.   "icons": [
8.     {
9.       "src": "assets/icons/icon-72x72.png",
10.      "sizes": "72x72",
11.      "type": "image/png"
12.    },
13.    {
14.      "src": "assets/icons/icon-96x96.png",
15.      "sizes": "96x96",
16.      "type": "image/png"
17.    },
```

```

18.   {
19.     "src": "assets/icons/icon-128x128.png",
20.     "sizes": "128x128",
21.     "type": "image/png"
22.   },
23.   {
24.     "src": "assets/icons/icon-144x144.png",
25.     "sizes": "144x144",
26.     "type": "image/png"
27.   },
28.   {
29.     "src": "assets/icons/icon-152x152.png",
30.     "sizes": "152x152",
31.     "type": "image/png"
32.   },
33.   {
34.     "src": "assets/icons/icon-192x192.png",
35.     "sizes": "192x192",
36.     "type": "image/png"
37.   },
38.   {
39.     "src": "assets/icons/icon-384x384.png",
40.     "sizes": "384x384",
41.     "type": "image/png"
42.   },
43.   {
44.     "src": "assets/icons/icon-512x512.png",
45.     "sizes": "512x512",
46.     "type": "image/png"
47.   }
48. ]
49. }

```

Archivo ngsw-config.json: El archivo ngsw-config posee la configuración de Service Worker. El archivo del service worker se agregara al momento de compilar el proyecto para un ambiente de producción [34].

A continuación, se muestra la configuración del archivo ngsw-config.json.

```

1.  {
2.    "$schema": "./node_modules/@angular/service-
  worker/config/schema.json",
3.    "index": "/index.html",
4.    "assetGroups": [
5.      { "name": "app",
6.        "installMode": "prefetch",
7.        "resources": {
8.          "files": [ "/favicon.ico", "/index.html", "/*.css",
9.                    "/*.js" ]
10.       }
11.    }, {
12.      "name": "assets",
13.      "installMode": "lazy",
14.      "updateMode": "prefetch",
15.      "resources": {
16.        "files": [ "/assets/**",
17.                  "/*. (eot|svg|cur|jpg|png|webp|gif|otf|ttf|woff|woff
2|ani)"

```

```
18.     ]
19.     }
20.   }
21. ]
22. }
```

Iconos: Añade los diferentes iconos por defecto en la dirección `src/assets/icons/` los mismo que serán utilizados por el archivo `manifest.webmanifest`.

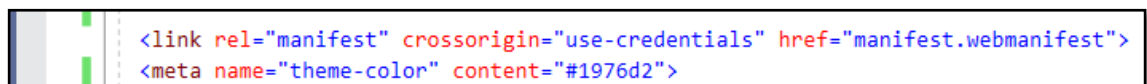
Archivo Angular.json: Se agrega la propiedad `serviceWorker` a la configuración de producción para informar a angular CLI que esta característica está habilitada [34].



```
Esquema: node_modules\@angular\cli\lib\config\schema.json
16     "architect": {
17     "build": {
18       "builder": "@angular-devkit/build-angular:browser",
19       "options": {...},
20     "configurations": {
21       "production": {
22         "fileReplacements": [...],
23         "optimization": true,
24         "outputHashing": "all",
25         "sourceMap": false,
26         "extractCss": true,
27         "namedChunks": false,
28         "aot": true,
29         "extractLicenses": true,
30         "vendorChunk": false,
31         "buildOptimizer": true,
32         "budgets": [...],
33         "serviceWorker": true,
34         "ngswConfigPath": "ngsw-config.json"
35       }
36     }
37   }
38 }
```

Figura 4.20: Configuración del archivo package.json.
Fuente: Elaborado por el investigador.

Archivo Index.html: El archivo es modificado para exponer el archivo manifest e indicar al navegar cual es el archivo de configuración para el manifiesto del proyecto.



```
<link rel="manifest" crossorigin="use-credentials" href="manifest.webmanifest">
<meta name="theme-color" content="#1976d2">
```

Figura 4.21: Referencia al archivo manifest en el archivo index.html.
Fuente: Elaborado por el investigador.

Archivo app.module: Se importa el `ServiceWorkerModule` para agregar las capacidades y características del service worker al proyecto.

```
293 imports: [  
294   ReactiveFormsModule.withConfig({ warnOnNgModelWithFormControl: 'never' })),  
295   BrowserModule,  
296   AppRoutingModule,  
297   BrowserAnimationsModule,  
298   ServiceWorkerModule.register('ngsw-worker.js', { enabled: environment.production })),
```

Figura 4.22: Importación del módulo ServiceWorkerModule

Fuente: Elaborado por el investigador.

Los diferentes módulos desarrollados en Angular están compuestos por los siguientes archivos:

Modelos: Definen la información que se recibirá o se enviará a los diferentes servicios expuestos por el cliente. A continuación, se muestra el modelo de datos del módulo representantes.

```
1. export interface IRepresentanteIE {  
2.   secuencial: number;  
3.   identificacionPersona: string;  
4.   estaActivo: boolean;  
5. }  
6. export interface IRepresentanteIS {  
7.   secuencial: number;  
8.   identificacionPersona: string;  
9.   estaActivo: boolean;  
10.  nombresPersona: string;  
11.  apellidoPaternoPersona: string;  
12.  apellidoMaternoPersona: string;  
13. }
```

Componentes: Existen dos tipos de componentes de página y presentación:

- **Componentes de página:** Es un archivo TypeScript que permite la interacción entre los servicios y componente de presentación. Ver Figura 4.24
- **Componentes de presentación:** Es un archivo HTML en el cual se define un conjunto de características y comportamientos que se mostrará en el navegador. Ver figura 4.23

Servicios: Los servicios es una clase TypeScript que permite consumir los servicios expuesto por el servidor. La siguiente porción de código corresponde a la clase ServicioPersona, la misma que implementa diferentes métodos que se encargan de realizar la llamada a los métodos HTTP definidos en el servidor.

```
1. import { HttpClient, HttpParams } from '@angular/common/http';  
2. import { IPersona } from './personas';  
3. import { Observable } from 'rxjs';  
4. import { Injectable } from '@angular/core';  
5. import { AppSettings } from '../appSettings';
```

```

6. @Injectable()
7. export class PersonaService {
8.   private apiUrl = AppSettings.defaultUrl + "/api/persona";
9.   constructor(private http: HttpClient) { }
10.  getPersona(parametro: string): Observable<IPersona[]> {
11.    let params = new HttpParams().set("estaActiva", parametro);
12.    return this.http.get<IPersona[]>(this.apiUrl, { params: params
    });
13.  }
14.  getPersonaPorIdentificacion(personaIdentificacion: string):
15.  Observable<IPersona> {
16.    let params = new HttpParams().set("personaIdentificacion",
    personaIdentificacion);
17.    return this.http.get<IPersona>(this.apiUrl, { params: params
    });
18.  }
19.  getPersonaPorIdentificacionLista(personaIdentificacionSelec: st
    ring): Observable<IPersona[]> {
20.    let params = new HttpParams().set("personaIdentificacionSelec",
    personaIdentificacionSelec);
21.
22.    return this.http.get<IPersona[]>(this.apiUrl, { params: params
    });
23.  }
24.
25.  create_ActualizarPersona(persona: IPersona): Observable<IPerson
    a> {
26.    return this.http.post<IPersona>(this.apiUrl, persona)
27.  }
28.  updatePersona(personaId: string): Observable<IPersona> {
29.    return this.http.put<IPersona>(this.apiUrl , personaId);
30.  }
31.  deletePersona(personaId: string): Observable<IPersona> {
32.    return this.http.delete<IPersona>(this.apiUrl + "/" + personaId
    );
33.  }
34.  }

```

Validaciones: Además de estos archivos mencionados anteriormente, un módulo puede poseer un archivo TypeScript para la realización de validación de datos, las validaciones de datos son realizadas mediante los fromGroup definidos en los componentes de presentación.

A continuación, se presenta una parte del código de la clase ValidacionPersona del módulo Persona. El método validarIdentificacionUnica consulta al servidor si éxito la identificación ingresada en la base de datos:

```

1. export class ValidacionPersona {
2.   public static validarIdentificacionUnica(servicio: PersonaServi
    ce, esEdicion: boolean): ValidatorFn {
3.     return (control: AbstractControl): { [key: string]: any } => {
4.
5.       var v = control.value;
6.       return new Promise( (resolve, reject) => {
7.         servicio.getPersonaPorIdentificacionLista(v).subscribe(
8.           data => {
9.             if (esEdicion) {

```

```

<div class="container-fluid">
  <div class="form-group">
    <p class="font-weight-normal">Home > Representante </p>
    <hr class="divicionNavegacion" />
  </div>
  <div class="alert alert-secondary">...</div>
  &nbsp;
  <div class="col-md-12 form-group">...</div>
  <div class="row">...</div>
  <div class="table-responsive">
    <p *ngIf="!(representantes && representantes.length)">
      No existen registros para mostrar
    </p>
    <table class="table table-hover" *ngIf="representantes && representantes.length">
      <thead>
        <tr>
          <th>Secuencial</th>
          <th>Identificación</th>
          <th>Nombre</th>
          <th>></th>
        </tr>
      </thead>
      <tbody>
        <tr *ngFor="let representante of representantes | representanteFilter:buscar |
          paginate: {id:'representantes', itemsPerPage:numeroRegistrosPaginacion, currentPage: pageActualRepresentante}">
          <td>{{ representante.secuencial }}</td>
          <td>{{ representante.identificacionPersona }}</td>
          <td>{{ representante.apellidoPaternoPersona }} {{ representante.apellidoMaternoPersona }} {{ representante.nombresPersona }}</td>
          <td>...</td>
          <td>...</td>
          <td>...</td>
        </tr>
      </tbody>
    </table>
    <pagination-controls id="representantes" (pageChange)="pageActualRepresentante = $event"></pagination-controls>
  </div>
</div>

```

Figura 4.23: Componente de presentación del módulo representante.
Fuente: Elaborado por el investigador.

```

@Component({
  selector: 'app-representante',
  templateUrl: './representante.component.html'
})
export class RepresentanteComponent implements OnInit {
  representantes: IRepresentanteIS[];
  buscar: string = "";
  pageActualRepresentante: number = 1;
  ignorarExistenCambiosPendientes: boolean = false;

  activosInactivos: string;
  filtroValor: string;
  filtros: string[] = ['Activo', 'Inactivo'];
  numeroRegistrosPaginacion = AppSettings.numeroRegistrosPaginacion;

  constructor(... )
  ngOnInit(... )
  onChange(... )
  cargarData(parametro: string)...
  delete(representante: IRepresentanteIE)...
  activarRepresentante(representante: IRepresentanteIE)...
  abrirBusquedaPersona(... )
  save(representante: IRepresentanteIE)...
  buscarRepresentantePorIdentificacion(identificacionRepresentante: string): boolean...
  verInformacionRepresentante(representante: IRepresentanteIE)...
}

```

Figura 4.24: Clase RepresentanteComponent
Fuente: Elaborado por el investigador.

```

10.     resolve({ validarIdentificacionUnica: true });
11.     } else {
12.     resolve(null);    }
13.     } else {
14.     if (data.length > 0) {
15.     resolve({ validarIdentificacionUnica: true });
16.     } else {
17.     resolve(null);
18.     }
19.     }
20.     },
21.     err => { resolve({ validarIdentificacionUnica: true }),
22.     }
23.     )
24.     })
25.     }
26.}
27.static isPresent(obj: any): boolean {
28.    return obj !== undefined && obj !== null;
29.}
30.}

```

Autenticación en el Cliente:

Con Angular Auth Guards se agregó una capa de autenticación basada en permisos, se recuperará el JWT enviado por el Servidor al momento de autenticarse y se mostraran los diferentes módulos en base a los permisos asignados a cada rol. Se utiliza el Guard CanActive para determinar si se activa o no la ruta solicitada en base al rol o roles asignados. CanActive realiza la respectiva verificación antes de cargar los diferentes compones. Ver Figura 4.25

A continuación, se expone la clase AuthGuardService encargada de activar o no la ruta solicitada en base a al rol del usuario.

```

1. @Injectable()
2. export class AuthGuardService implements CanActivate {
3.     constructor(private accountService: AccountService,
4.         private router: Router,
5.         private toastr: ToastrService, ) { }
6.
7.     canActivate(route: ActivatedRouteSnapshot,
8.         state: RouterStateSnapshot): boolean {
9.
10.        if (this.accountService.estaLogueado()) {
11.            let
12.            roles = route.data['permittedRoles'] as Array<string>;
13.            if (roles) {
14.                if (this.accountService.roleMatch(roles)) return true;
15.                else {
16.                    this.toastr.info("No posee los permisos necesarios,
17.                    se redireccionará a la página de inicio."+
18.                    "Si el problema persiste comuníquese con servicio
19.                    técnico", 'Advertencia');
20.                    this.router.navigate(['']);
21.                    return false;

```

```

21.     }
22.     }
23.     return true;
24.
25.   } else {
26.     this.toastr.info("No posee los permisos necesarios, se
27.       redireccionará a la página de inicio." +
28.       "Si el problema persiste comuníquese con servicio
29.       técnico",
30.       'Advertencia');
31.     this.router.navigate(['']);
32.
33.     return false;
34.   }
35. }
36.}

```

```

RouterModule.forRoot([
  { path: '', component: HomeComponent, pathMatch: 'full' },
  {
    path: 'asignatura', component: AsignaturaComponent, canActivate: [AuthGuardService],
    data: { permittedRoles: ['Administrador'] }
  },
  {
    path: 'modulo', component: ModuloComponent, canActivate: [AuthGuardService],
    data: { permittedRoles: ['Administrador'] }
  },
],

```

Figura 4.25: Uso de CanActive en appmodule.ts
Fuente: Elaborado por el investigador.

La clase AuthInterceptor se encarga de interceptar las solicitudes HTTP de la Aplicación Web Progresiva y verificar que el usuario se encuentre autenticado para poder realizar las solicitudes http.

```

1. @Injectable()
2. export class AuthInterceptor implements HttpInterceptor {
3.
4.   constructor(private router: Router) { }
5.
6.   intercept(req: HttpRequest<any>, next: HttpHandler):
7.     Observable<HttpEvent<any>> {
8.     if (localStorage.getItem('token') != null) {
9.       const clonedReq = req.clone({
10.        headers: req.headers.set('Authorization', 'Bearer ' +
11.        localStorage.getItem('token'))
12.      });
13.       return next.handle(clonedReq).pipe(
14.        tap(
15.          succ =>
16.          { },
17.          err => {
18.            if (err.status == 401){
19.              localStorage.removeItem('token');
20.              this.router.navigateByUrl('/register-login');
21.            }
22.            else if(err.status == 403)
23.              this.router.navigateByUrl('');
24.          }
25.        )

```



```

26.     )
27.     }
28.     else
29.     return next.handle(req.clone());
30.     }
31. }

```

Clase de Configuración “AppSettings”

Se crea una clase denominada AppSettings la cual contendrá las diferentes configuraciones que poseerá el proyecto cliente de la Aplicación Web Progresiva. A continuación, se muestra la clase AppSetting.

```

1.  export class AppSettings {
2.
3.  static defaultUrl: string = "https://localhost:5001";
4.
5.  static appUrl: string = "http://localhost:4200";
6.
7.  static numeroRegistrosPaginacion = 5;
8.
9.  static numeroRegistrosPaginacionAnidada = 5;
10.
11. static numeroRegistrosPaginacionEvaluacion = 1;
12.
13. static passwordDefecto = "*****!";
14. static codigoAsignaturaConsolidado = "CS";
15.
16. static meses = ["Enero", "Febrero", "Marzo", "Abril", "Mayo",
17. "Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviemb
re",
18. "Diciembre"];
19. static dias = ["Lunes", "Martes", "Miercoles", "Jueves", "Vie
rnes",
20. "Sabado", "Domingo"];
21.
22. static horasClase = ["08:00 - 09:00", "09:00 - 10:00", "10:00
-
23. 11:00", "15:00 - 16:00", "16:00 - 17:00", "17:00 - 18:00"];
24. static EstadosEvaluacion = {
25.   Inactiva: 0,
26.   Activa : 1,
27.   Finalizada : 2,
28. };
29. }

```

Las configuraciones que posee el archivo AppSettings se describen a continuación:

- **defaultUrl:** Representa la dirección Url del servidor.
- **appUrl:** Representa la dirección Url del cliente.
- **numeroRegistrosPaginacion:** Representa el número de registros que se mostraran en una paginación.
- **numeroRegistrosPaginacionAnidada:** Representa el número de registros que se mostraran en una paginación doble.

- **numeroRegistrosPaginacionEvaluacion:** Representa el número de registros (preguntas) que se mostraran al realizar una evaluación.
- **passwordDefecto:** Contraseña por defecto al crear y resetear la clave de los usuarios.
- **codigoAsignaturaConsolidado:** Código de asignatura Consolidado .
- **estadosEvaluacion:** Los estados que posee una evaluación.
- **días:** Representa los días de la semana que se dicta clases.
- **horas:** Representa el horario de clases que se imparte a los estudiantes

Librerías Adicionales:

Se utilizó una serie de librerías externas que realizarán funciones específicas y ayudan a mejorar la apariencia de la Aplicación Web Progresiva, en la siguiente tabla se detalla las diferentes librerías externas utilizadas:

Librería	Descripción	Comando de instalación
Angular material	Es una librería de estilos realizada por el equipo de desarrollo de angular, se integra perfectamente a los proyectos desarrollados con angular.	add @angular/material
Material design icons	Esta librería ofrece una serie de símbolos diseñados para acciones y artículos específicos.	add @angular/material
ngKatex	ng-Katex permite visualizar fórmulas matemáticas escritas a través de sintaxis LaTeX.	npm install ng-katex --save
PdfMake Wrapper	Es una librería que nos permite la creación de documentos en formato PDF del lado del cliente.	npm install pdfmake --save npm install pdfmake-wrapper --save
ngx-toastr	Esta librería nos permite visualizar notificaciones a los usuarios.	npm install ngx-toastr --save
ngx-pagination	Brinda una solución simple y eficaz para realizar paginación en angular.	npm install ngx-pagination --save

Tabla 4.118: Librerías utilizadas en el proyecto de Angular.

Fuente: Elaborado por el investigador.

4.5.4.3. Desarrollo de Iteraciones

El desarrollo de la Aplicación Web Progresiva esta dividida en 4 iteraciones en las cuales se implementa las diferentes historias de usuarios que comprende cada iteración. A continuación, se describe el desarrollo de cada iteración y sus diferentes

historias de usuarios, se describirá el desarrollo de las historias de usuarios más relevantes:

Iteración 1

Las diferentes historias de usuario se desarrollaron bajo los mismos conceptos de diseño y arquitectura que a continuación se describe, se describirá los puntos más relevantes que se desarrolló en esta iteración.

Iteración	Numero	Historia de usuario	Fecha Inicio	Fecha Fin
1	2	Gestión de Asignaturas.	01/04/2020	03/04/2020
1	3	Gestión de Módulos.	06/04/2020	08/04/2020
1	4	Gestión de Cursos.	09/04/2020	10/04/2020
1	5	Gestión de Títulos Académicos.	13/04/2020	14/04/2020
1	6	Gestión de Instituciones Educativas.	15/04/2020	16/04/2020
1	11	Gestión de tema de una pregunta.	17/04/2020	17/04/2020

Tabla 4.119: Iteración1.

Fuente: Elaborado por el investigador.

Gestión de Asignaturas: La gestión de asignaturas comprende el ingreso edición y eliminación de asignaturas como se muestra en la siguiente imagen.

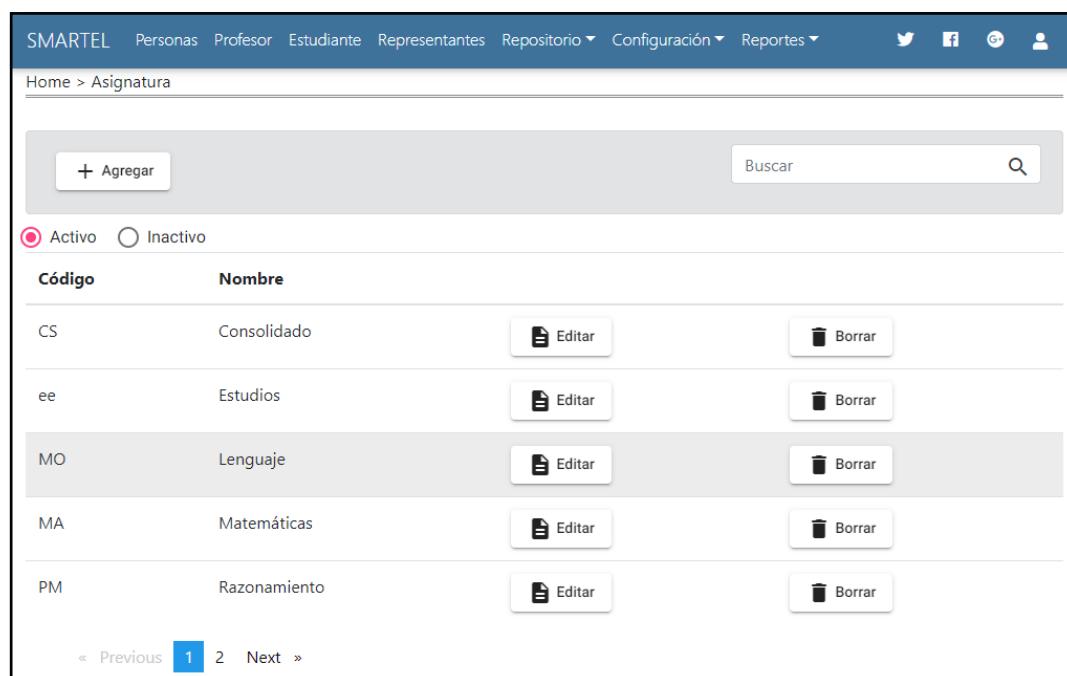


Figura 4.26: Gestión de asignaturas.

Fuente: Elaborado por el investigador.

Para ingresar una asignatura verificamos que el código ingresado no se encuentre registrado previamente, para realizar el respectivo control utilizamos las validaciones proporcionadas por el formulario reactivo mediante los métodos expuestos en la clase de validación del módulo.

```
this.formGroup = this.fb.group({
  codigo: [{ value: '', disabled: this.modoEdicion }, [Validators.required, Validators.maxLength(2)],
  [ValidacionAsignatura.validarCodigoUnico(this.asignaturaService)]],
  nombre: ['', [Validators.required, Validators.maxLength(50), Validators.minLength(5)]],
  estaActiva: [{ value: true, disabled: !this.modoEdicion } ]
});
```

Figura 4.27: FromGrop de la clase asignatura

Fuente: Elaborado por el investigador.

En la figura 4.27 podemos apreciar el respectivo FromGroup y los FromControl definidos para código, nombre y esta activa.

```
import { AsignaturaService } from '../asignatura.service';
import { AbstractControl } from '@angular/forms';

export class ValidacionAsignatura {
  public static validarCodigoUnico(servicio: AsignaturaService) {
    return (control: AbstractControl): { [key: string]: any } => {
      var v = control.value;
      return new Promise((resolve, reject) => {
        servicio.getAsignaturaPorCodigo(v).subscribe(
          data => {
            if (this.isPresent(data.codigo)) {
              resolve({ validarCodigoUnico: true });
            } else {
              resolve(null);
            }
          },
          err => { resolve({ validarCodigoUnico: true }) }
        )
      })
    }
  }

  static isPresent(obj: any): boolean {
    return obj !== undefined && obj !== null;
  }
}
```

Figura 4.28: Validación de código único correspondiente al módulo asignatura.

Fuente: Elaborado por el investigador.

Gestión de Cursos: La gestión de cursos está formado por dos procesos. El primer proceso es el ingreso de cursos y el segundo es la asignación de profesores y asignaturas a un curso en específico. La asignación de asignatura y profesores se la realizará mediante la opción de editar curso, se mostrará un desplegable donde se podrá realizar los mantenimientos respectivos sobre la asignación de profesores y asignaturas. Ver Figura 4.27

De igual manera las diferentes validaciones para la asignación de una asignatura y profesor se la realizó mediante el respectivo formulario reactivo con ayuda de los métodos de la clase de validación. Estos métodos toman como parámetros los diferentes servicios de los módulos para realizar consultas al servidor y realizar el respectivo control. Ver Figura 4.28

Iteración 2

En la iteración 2 se realizó el ingreso de la información de los diferentes actores de la PWA como son los Estudiantes, Profesores y Representantes, para poder ingresar uno de estos actores se deberá ingresar previamente una Persona.

Iteración	Numero	Historia de usuario	Fecha Inicio	Fecha Fin
2	7	Gestión de Personas.	20/04/2020	22/04/2020
2	8	Gestión de Representantes.	23/04/2020	24/04/2020
2	9	Gestión de Profesores.	27/04/2020	01/05/2020
2	10	Gestión de Estudiantes.	04/05/2020	08/05/2020

Tabla 4.120: Iteración 2.

Fuente: Elaborado por el investigador.

Gestión de Personas: Una persona contendrá la información esencial que poseerá un profesor , estudiante o un representante. De igual manera se realizó los respectivos mantenimientos para Persona y la asignación de teléfonos a una Persona en base a la estructura definida para el cliente y servidor.

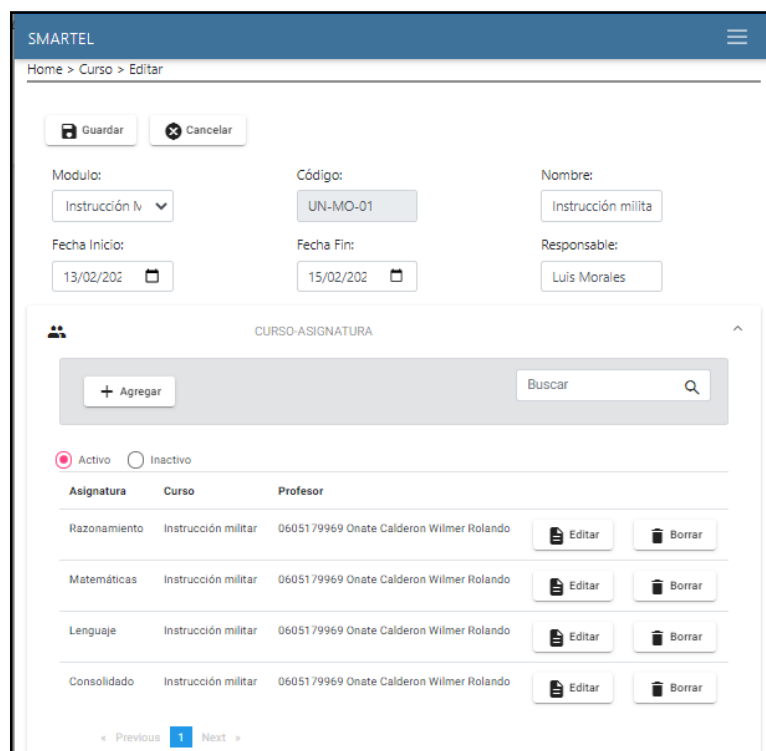


Figura 4.29: Pantalla de edición de un curso.

Fuente: Elaborado por el investigador.

```

export class ValidacionCursoDetalle {

    public static validarCursoDetalleUnico(servicio: CursoDetalleService, secuencialCursoDetalle:number): ValidatorFn {
        return (control: AbstractControl): { [key: string]: any } => {
            var v = control.value;
            return new Promise((resolve, reject) => {
                servicio.getCursoDetallePorCodigoAsignatura(secuencialCursoDetalle, servicio.codigoCursoSeleccionadoGet(),
                    servicio.codigoAsignaturaSeleccionadoGet(), v).subscribe(
                    data => {
                        if (data.length>0) {
                            resolve({ validarAsignaturaUnica: true });
                        } else {
                            resolve(null);
                        }
                    },
                    err => { resolve({ validarAsignaturaUnica: true }), console.log(err) }
                )
            })
        }
    }

    public static validarcursoDetalleProfesor(): ValidatorFn{...}

    static isPresent(obj: any): boolean {
        return obj !== undefined && obj !== null && obj !== 0;
    }
}

```

Figura 4.30 : Validación de código único correspondiente la clase Curso Detalle.

Fuente: Elaborado por el investigador.

Estudiantes, Profesores y Representantes: Para crear un estudiante, profesor o representante es necesario seleccionar una persona ya existente.

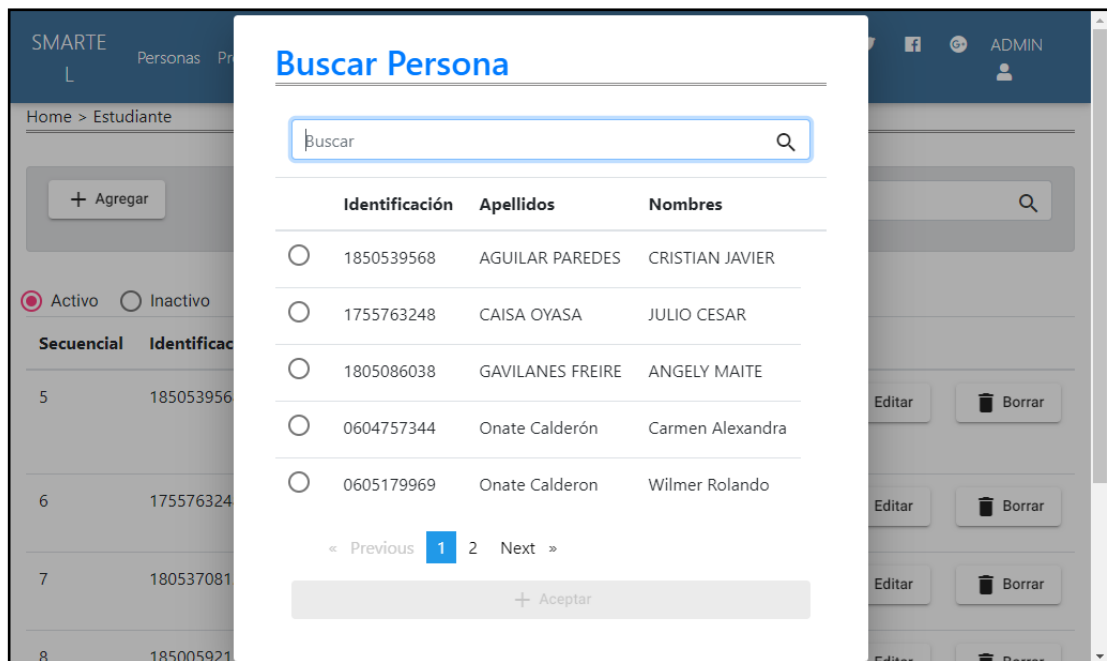


Figura 4.31: Ingreso de un estudiante.

Fuente: Elaborado por el investigador.

Una vez seleccionada la persona se realiza una validación para verificar que la persona seleccionada no se encuentre ingresada. A continuación, se detalla el método encargado de la validación el mismo que recibe un servicio encargado de la consulta

en el servidor y un modelo de datos que permite identificar si es un estudiante, profesor o representante.

```

public static validarBusquedaPersona(busqueda: IPersonaBusquedaExistencia, busquedaService: PersonaBusquedaService): ValidatorFn {
    return (control: AbstractControl): { [key: string]: any } => {
        var v = control.value;
        return new Promise((resolve, reject) => {
            if (busqueda.esRepresentante) else if (busqueda.esProfesor) {
                busquedaService.getProfesorPorIdentificacionPersona(v).subscribe(
                    data => {
                        if (data) {
                            resolve({ validarIdentificacionUnica: true });
                        } else {
                            resolve(null);
                        }
                    },
                    err => { resolve({ validarExistencia: true }); console.log(err) }
                )
            } else if (busqueda.esEstudiante) {
                busquedaService.getEstudiantePorIdentificacionPersona(v).subscribe(
                    data => {
                        if (this.isPresent(data.identificacionPersona)) {
                            resolve({ validarIdentificacionUnica: true });
                        } else {
                            resolve(null);
                        }
                    },
                    err => { resolve({ validarExistencia: true }); console.log(err) }
                )
            } else {
                resolve(null);
            }
        })
    }
}

```

Figura 4.32: Validación de estudiante, profesor y representante único al seleccionar una Persona
Fuente: Elaborado por el investigador.

De igual manera en esta iteración se creó los diferentes mantenimientos para el ingreso de información complementaria para profesores, estudiantes y representantes por ejemplo la asignación de representantes a estudiantes, la asignación de asignaturas a un estudiante y profesores etc.

Iteración 3

Iteración	Numero	Historia de usuario	Fecha Inicio	Fecha Fin
3	12	Gestión de Preguntas.	11/05/2020	22/05/2020
3	12	Gestión de Usuarios.	25/05/2020	05/06/2020
3	14	Gestión de Evaluaciones.	08/06/2020	19/06/2020
3	1	Inicio de sección.	22/06/2020	24/06/2020

Tabla 4.121: Iteración 3.
Fuente: Elaborado por el investigador.

Gestión de Preguntas. Los usuarios con el rol de administrador y profesor podrán acceder a la opción de repositorio donde podrán ingresar preguntas según la asignatura y el tema previamente seleccionada. Ver Figura 4.33.

Para el ingreso de formular matemáticas en la pregunta se utilizó la librería ngKatex la cual permite visualizar una fórmula matemática previamente ingresada en formato LaTeX. La etiqueta ng-katex-paragraph será la encargada de mostrar el texto junto a la fórmula matemática, la etiqueta ng-katex-paragraph diferencia un

texto de una formula por el símbolo \$ que indica el inicio y final de la fórmula matemática. Ver Figura 4.34- 4.35.

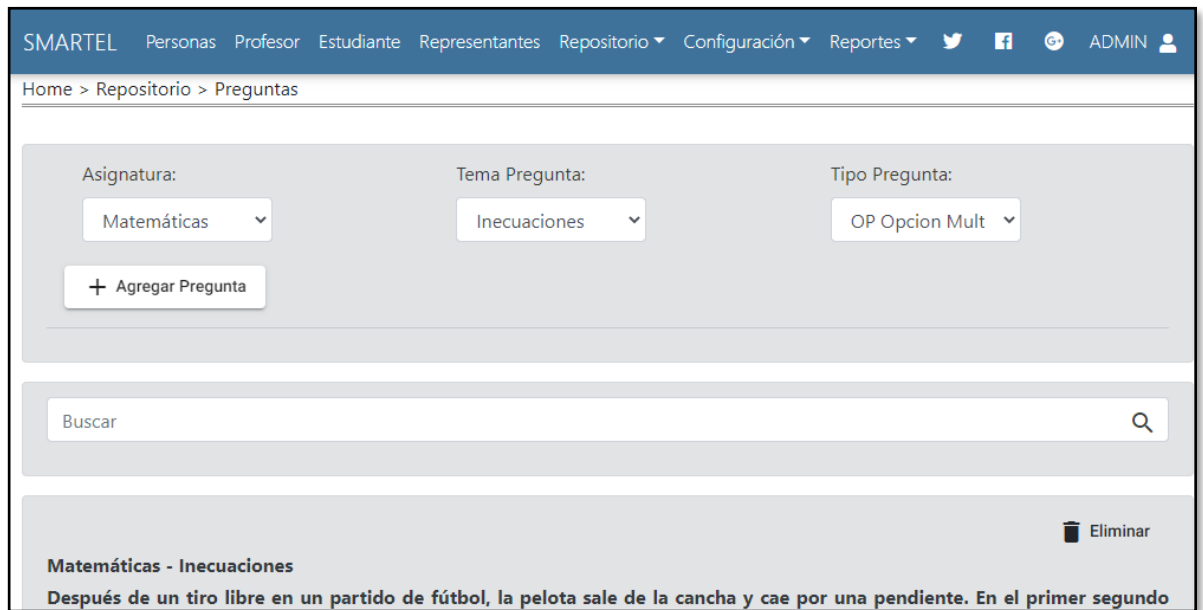


Figura 4.33: Ingreso de preguntas en el repositorio.
Fuente: Elaborado por el investigador.

```
<ng-katex-paragraph [paragraph]="formulaMatematica"></ng-katex-paragraph>
```

Figura 4.34: Etiqueta ngKatex-paragraph
Fuente: Elaborado por el investigador.

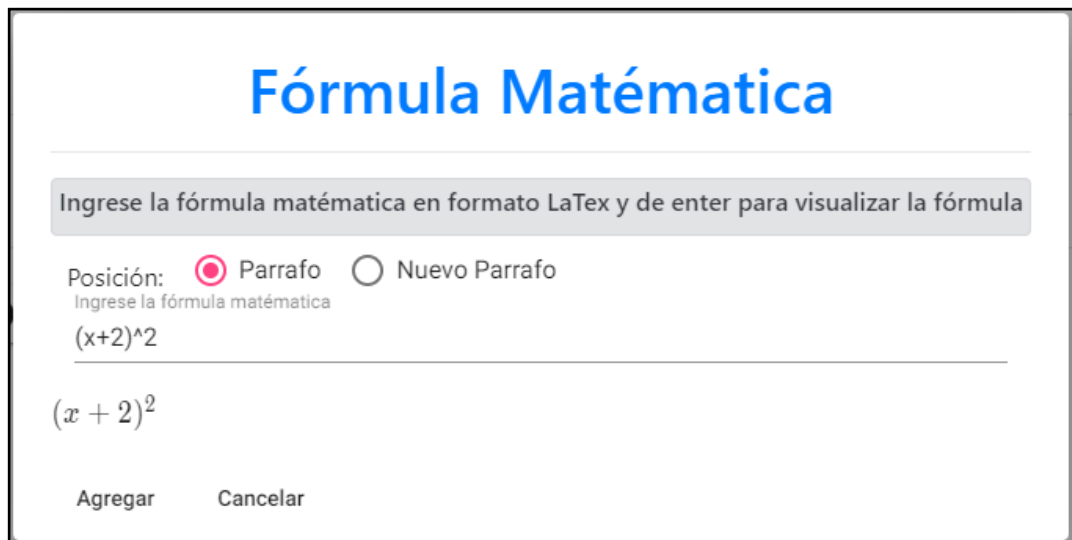


Figura 4.35: Ingreso de fórmulas matemáticas
Fuente: Elaborado por el investigador.

Gestión de Usuarios: Los usuarios con el rol estudiante y profesor serán creados cuando el administrador ingrese un estudiante o profesor, los usuarios se crearán en

base a la información ingresada en el formulario de personas y poseerá una clave predetermina confirmada por el cliente.

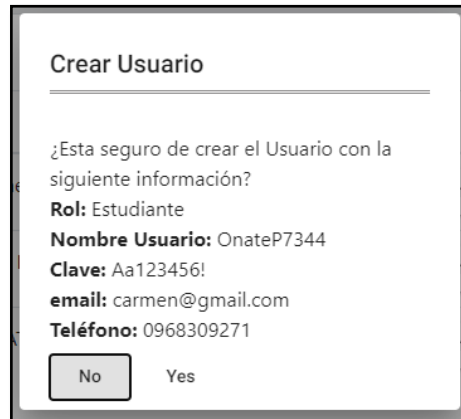


Figura 4.36: Confirmación para crear un usuario
Fuente: Elaborado por el investigador.

Al momento de un usuario ingresar al sistema el servidor devolverá un token el mismo que será interceptado por el servicio AuthInterceptor y validado por AuthGuardService.

Recuperación de Clave de Usuario

La recuperación de clave de un usuario se la realizo mediante la confirmación de un código único enviado al correo electrónico del usuario o mediante la restauración de la clave por medio del administrador del sistema. Para él envío de correo electrónico al usuario se utilizó Smtplib el mismo que se muestra a continuación:

```
public class EmailSender : IEmailSender
{
    private string host;
    private int port;
    private bool enableSSL;
    private string userName;
    private string password;
    public EmailSender()
    {
        this.host = "smtp.gmail.com";
        this.port = 587;
        this.enableSSL = true;
        this.userName = "*****";
        this.password = "*****"
    }
    public Task SendEmailAsync(string email, string subject,
        string htmlMessage)
    {
        var client = new SmtplibClient(host, port)
        {
            Credentials = new NetworkCredential(userName, password),
            EnableSsl = enableSSL
        };
        return client.SendMailAsync(
            new MailMessage(userName, email, subject, htmlMessage) {
                IsBodyHtml = true
            }
        );
    }
}
```

Figura 4.37: Clase EmailSender
Fuente: Elaborado por el investigador.

Gestión de Evaluaciones:

Las evaluaciones se presentarán a los estudiante y profesores agrupados por módulos y cursos con acciones y opciones diferentes:

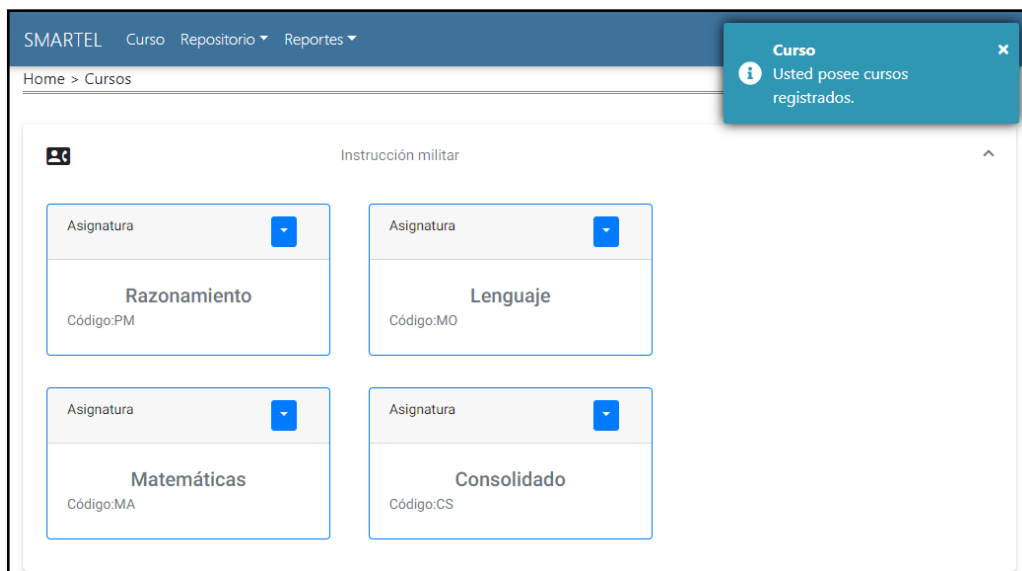


Figura 4.38: Pantalla de visualización de cursos.

Fuente: Elaborado por el investigador.

Evaluaciones con el Rol Profesor: Los profesores crear evaluación y evaluaciones consolidadas y posteriormente podrán agregar las respectivas preguntas a la evaluación, las preguntas a ingresar serán seleccionadas y se ingresara un puntaje a cada una . Ver Figura 4.39

La diferencia entre evaluaciones y evaluaciones consolidadas es que las evaluaciones pertenecen a una asignatura y el profesor seleccionaran las preguntas del repositorio y las evaluaciones consolidadas pertenecen al curso y se seleccionara aleatoriamente las preguntas de las evaluaciones creadas e iniciadas en el módulo.

Activar Evaluación

Una evaluación podrá ser activa posteriormente de agregar las preguntas correspondientes, una vez activa la evaluación no se podrá ingresar más preguntas. Ver figura 4.40

Una vez activada la evaluación y realizada por los estudiantes los profesores podrán visualizar las evaluaciones rendidas por los estudiantes que pertenecen al curso y asignatura correspondiente. Ver Figura 4.41

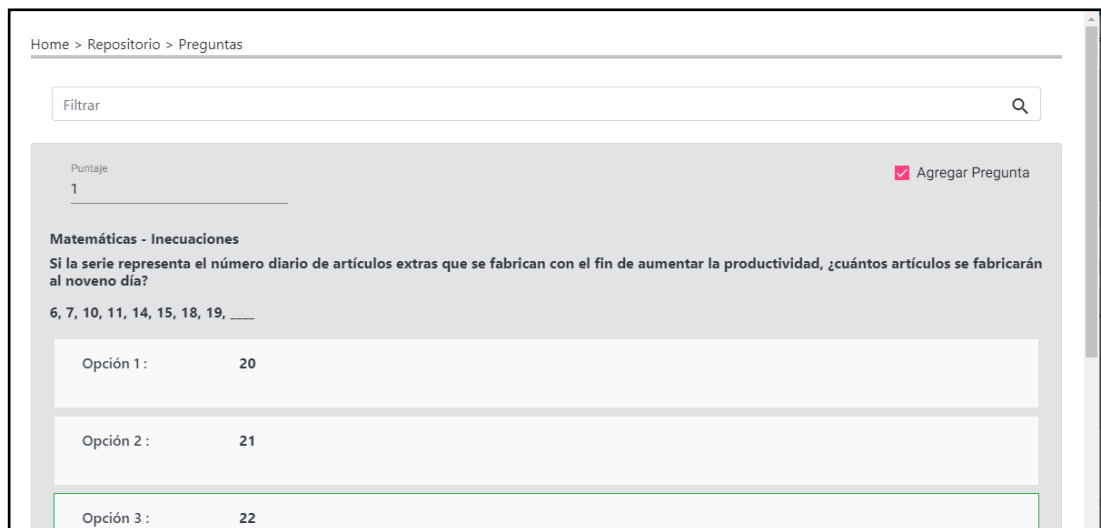


Figura 4.39: Asignación de preguntas a una evaluación.

Fuente: Elaborado por el investigador.

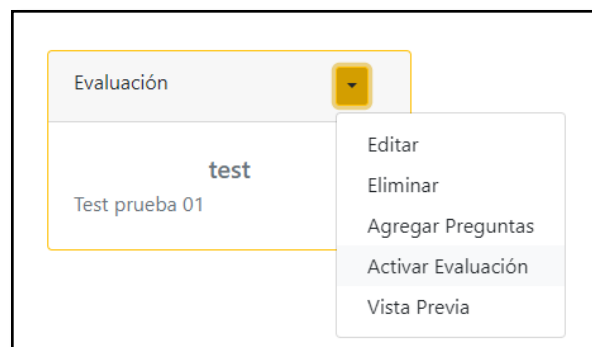


Figura 4.40: Clase Opciones de una evaluación.

Fuente: Elaborado por el investigador.

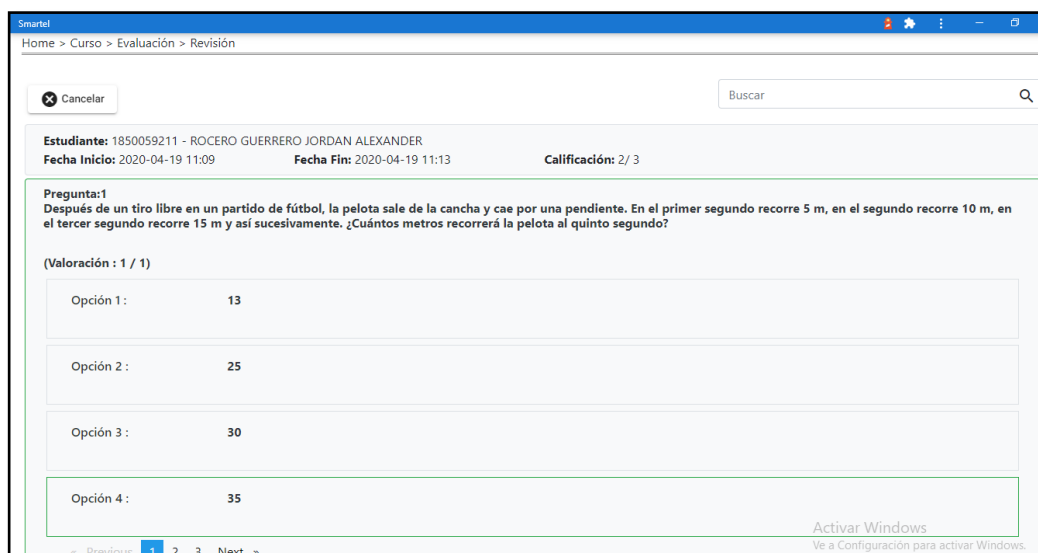


Figura 4.41: Revisión de evaluaciones por estudiante.

Fuente: Elaborado por el investigador.

Iteración 4

Iteración	Numero	Historia de usuario	Fecha Inicio	Fecha Fin
4	15	Revisión de evaluaciones por estudiante	24/06/2020	26/06/2020
4	16	Iniciar Evaluación.	29/06/2020	03/07/2020
4	17	Reporte de horario de clases.	06/07/2020	10/07/2020
4	18	Reporte de evaluaciones por asignatura y consolidado.	13/07/2020	15/07/2020
4	19	Ficha de Inscripción del estudiante.	16/07/2020	17/07/2020

Tabla 4.122: Iteración 4.

Fuente: Elaborado por el investigador.

Iniciar Evaluación: Los diferentes estudiantes visualizarán las evaluaciones activas e inactivas las mismas se diferenciarán por poseer un color diferente de borde. El estudiante iniciará la evaluación y se registra la hora de inicio de la evaluación y se iniciará un `timerInterval` con el tiempo establecido en la evaluación, al finalizar el tiempo la evaluación se guardará la evaluación con las respuestas ingresadas A continuación se muestra el código de `timerInterval` encargado de realizar la cuenta regresiva del tiempo de la evaluación, los atributos horas, minutos y segundos es inicializada al cargar los datos de la respectiva evaluación.

```
1. private timer(): void {
2.     if (--this.segundos < 0) {
3.         this.segundos = 59;
4.         if (--this.minutos < 0) {
5.             this.minutos = 59;
6.             if (--this.horas < 0) {
7.                 const dialogRef =
8.                 this.dialog.open(DialogConfirmationMessageComponent
9.                 , {
10.                    data: {
11.                        accion: "Guardar",
12.                        mensaje: "El tiempo de la Evaluación termino.
13.                        Se guardará la evaluación."
14.                    }, disableClose: true
15.                });
16.                this.save();
17.                this.minutos = 0;
18.                this.segundos = 0;
19.                this.horas = 0;
20.            }
21.        }
22.    }
```

Revisión de Evaluaciones por Estudiante: Los estudiantes podrán revisar las diferentes evaluaciones, claro si la configuración de la evaluación lo permite, se

mostrará las respuesta elegida y correcta además de una justificación si la respuesta fue contestada incorrectamente.

Las historias de usuarios comprendidas entre los números 17 y 19 se realizaron en base a la estructura de la creación de reportes que se describe a continuación:

Creación de Reportes

Para la creación de reporte se utilizó la librería Pdf Make Wrapper la cual utiliza internamente la librería Pdf Make. Pdf Make Wrapper permite generas documentos PDF de manera fácil a través de métodos descriptibles y de fácil comprensión. Pdf Make Wrapper permite la personalización de fuentes, iconos, estilos y la implementación de seguridad a través de contraseñas.

Para la implementación de Pdf Make Wrapper en el proyecto de Angular se creó tres diferentes métodos encargados de inicializar el PDF, crear el cuerpo del PDF y generar el PDF.

- **Inicializar PDF:** El método inicializarPdf se encarga de configurar el formato del pdf y definir el diseño y configuración del header , footer y background.

```
1. inicializarPdf() {
2.
3.     this.pdf = new PdfMakeWrapper();
4.     this.pdf.pageMargins([40, 80, 40, 60]);
5.
6.     this.pdf.pageSize('A4');
7.
8.     new Img(AppSettings.appURL +
9.         '/assets/imagenes/logoEncabezado.jpeg')
10.    .width(130).build().then(img => {
11.        this.pdf.header(img);
12.    });
13.
14.    new Img(AppSettings.appURL + '/assets/imagenes/footer.png')
15.    .
16.    alignment("center").width(400).opacity(0.5).build().then(im
17.    g => {
18.        this.pdf.footer(img);
19.    });
20.
21.    new Img(AppSettings.appURL + '/assets/imagenes/fondoReporte
22.    s.jpg')
23.    .width(700).height(600).alignment("center").opacity(0.1).bu
24.    ild()
25.    .then(imgW => {
26.        this.pdf.background(imgW);
27.    });
28. }
```

```

25.     this.pdf.pageOrientation('portrait');
26.     this.pdf.info({
27.         title: 'Smartel',
28.         author: 'smartel-app',
29.         subject: 'subject of document',
30.     });
31.
32.     this.pdf.footer(new Txt('PORQUE CREEMOS EN TUS SUEÑOS,
33.     TRABAJAMOS POR TU
34.     ÉXITO').alignment('center').opacity(0.5).end);
34. }

```

- **Cuerpo del PDF:** En este método se incorpora la información a mostrar.

```

1. generarCurpoPDF() {
2.     this.pdf.add(new Txt('HORARIO DE CLASES').fontSize(12)
3.     .alignment('center').bold().end);
4.     this.pdf.add(this.pdf.ln(1));
5.     this.pdf.add(new Txt("Identificación: " +
6.     this.persona.identificacion).fontSize(12).alignment('left')
7.     .bold().end);
8.     this.pdf.add(new Txt("Profesor: " +
9.     this.persona.apellidoPaterno + " " +
10.    this.persona.apellidoMaterno + " " +
11.    this.persona.nombres).fontSize(12).alignment('left')
12.    .bold().end);
13.    this.pdf.add(this.pdf.ln(1));
14.    let tablaEncabezado = new Table([
15.        [
16.            new Txt('').alignment('center').bold().end,
17.            new Txt('08:00 -
18.            09:00').alignment('center').bold().end,
19.            new Txt('09:00 -
20.            10:00').alignment('center').bold().end,
21.            new Txt('10:00 -
22.            11:00').alignment('center').bold().end,
23.            new Txt('15:00 -
24.            16:00').alignment('center').bold().end,
25.            new Txt('16:00 -
26.            17:00').alignment('center').bold().end,
27.            new Txt('17:00 -
28.            18:00').alignment('center').bold().end,
29.        ]
30.    ]).widths(['auto', 'auto', 'auto', 'auto', 'auto', 'auto',
31.    'auto']).end;
32.    for (var i = 0; i < this.dias.length; i++) {
33.        let horariosPorDia = this.horarioPorEstudiante
34.        .filter(aux => aux.dia == this.dias[i]);
35.        let filas: any[] = [];
36.        if (horariosPorDia.length > 0)
37.            filas= [this.dias[i], "", "", "", "", "", ""];
38.        for (var j = 0; j < horariosPorDia.length; j++) {
39.            for (var h = 0; h < this.horasClase.length; h++) {
40.                let horaInicio = moment(this.horasClase[h].split(" -
41.                ") [0],
42.                "HH:mm");
43.                let horaFin = moment(this.horasClase[h].split(" -
44.                ") [1],
45.                "HH:mm");

```

```

39.         let horaInicioAsignatura =
40.             moment(horariosPorDia[j].horaInicio, "HH:mm");
41.         let horaFinAsignatura =
42.             moment(horariosPorDia[j].horaFin, "HH:mm");
43.         if (
44.             ((horaInicio >= horaInicioAsignatura &&
45.              horaFin <= horaFinAsignatura) ||
46.              (horaInicio >= horaInicioAsignatura &&
47.               horaInicio < horaFinAsignatura) || (horaFin >
48.               horaInicioAsignatura && horaFin <=
49.               horaFinAsignatura))) {
50.             filas[h + 1] = (new Txt(horariosPorDia[j].nombreAsi
gnatura
51.                 + "\n\n" ).alignment('center').end)
52.         }
53.     }
54. }
55.     if (horariosPorDia.length>0)
56.         tablaEncabezado.table.body.push(filas);
57. }
58. this.pdf.add(tablaEncabezado);
59. }

```

- **Generar el PDF:** El método generar llama los métodos de inicializar y generar cuerpo, además de visualizar el archivo.

```

1. generarPdf() {
2.     this.generarCurpoPDF();
3.     this.pdf.create().open();
4.     this.inicializarPdf();
5.     this.dialog.closeAll();
6. }

```

4.5.5. Fase: Prueba

La fase de pruebas es fundamental en la metodología XP para comprobar el correcto funcionamiento de las aplicaciones. Para garantizar el correcto funcionamiento de la Aplicación Web Progresiva se realizaron las siguientes pruebas:

- Pruebas Unitarias
- Pruebas de Aceptación

4.5.5.1. Pruebas Unitarias

Las pruebas unitarias se realizan para comprobar el correcto funcionamiento de partes de código de la Aplicación Web Progresiva.

Para la realización de pruebas unitarias se utilizó NUnit, que es un framework que permite realizar pruebas unitarias en .NET y Mono. Se creó un proyecto adicional con el nombre “PWANUnitTest” donde se procedió a realizar las pruebas unitarias de la Aplicación Web Progresiva.

La realización de las pruebas unitarias se llevó a cabo mediante una base de datos en memoria a partir del contexto “SmartelContext” del proyecto principal.

```

1. public SmartelContext ObtenerContexto()
2. {
3.
4.     var options = new DbContextOptionsBuilder<SmartelContext>()
5.         .ConfigureWarnings
6.         (x => x.Ignore(InMemoryEventId.TransactionIgnoredWarning))
7.         .UseInMemoryDatabase(databaseName: "Test").Options;
8.     //Inicializamos el contexto SmartelContext
9.     var context = new SmartelContext(options);
10.    //Inicializamos los datos a utilizar en las pruebas
11.    unitarias
12.    InicializaDatos.Inicializar(context);
13.    return context;
14. }

```

A continuación, se detalla un ejemplo de los casos de pruebas realizados en la Aplicación Web Progresiva.

Casos de Pruebas de Asignaturas

Método	Caso de Prueba
Listar asignatura por activo o inactivo	<ul style="list-style-type: none"> • Validar que el número de registros devueltos sea igual a los existentes en la Base de Datos. • Verificar que los campos del objeto DTO devuelto contenga información recuperada de la Base de Datos.
Existe asignatura	<ul style="list-style-type: none"> • Comprobar que devuelva verdadero si existe la asignatura en la base de datos.
Eliminar asignatura	<ul style="list-style-type: none"> • Comprobar que realice un borrado lógico a la asignatura correspondiente.
Agregar asignatura	<ul style="list-style-type: none"> • Comprobar que añada la asignatura en la Base de Datos
Modificar asignatura	Comprobar que se realice una modificación correcta de la asignatura en la Base de Datos

Tabla 4.123: Casos de pruebas para el mantenimiento de asignatura.

Fuente: Elaborado por el Investigador.

Se implementaron las diferentes pruebas unitarias para el mantenimiento de asignaturas, se generaron pruebas con error y prueba sin error para validar el correcto funcionamiento de los diferentes métodos utilizados para realizar el mantenimiento de asignaturas. Ve Figura 4.42

4.5.5.2. Pruebas de Aceptación

Además, de comprobar si lo implementado en cada historia de usuario es lo deseado por parte del cliente.

Para el presente proyecto se realizó las pruebas de aceptación, las misma que se desarrolló juntamente con el cliente en un ambiente de pruebas instalado en los equipos de la entidad. A continuación, se expone los resultados de las principales pruebas de aceptación, teniendo en cuenta el riesgo y complejidad de las historias de usuario.

Prueba	Condición	Duración
✓ PWANUnit (30)	Correcta	34 s
✓ NUnitTestProject1 (30)	Correcta	34 s
✓ AsignaturaTest (5)	Correcta	8 s
✓ DeleteAsignatura	Correcta	1 s
✓ GuardartAsignatura	Correcta	980 ms
✓ listarAsignaturaActivos	Correcta	1 s
✓ listarAsignaturalnactivos	Correcta	3 s
✓ listarAsignaturaPorCod...	Correcta	948 ms
✓ CursoTest (5)	Correcta	5 s

Resumen de los detalles de la prueba	
✓	listarAsignaturaPorCodigo
📄	Origen: AsignaturaTest.cs línea 44
🕒	Duración: 948 ms

Figura 4.42: Casos de pruebas para el mantenimiento de asignatura.

Fuente: Elaborado por el Investigador.

PRUEBA DE ACEPTACIÓN	
Código: P86	Código historia: H12- Gestión de preguntas.
Nombre: Prueba listar preguntas.	
Condición de ejecución: Iniciar sección con el rol de administrador, profesor.	
Proceso de Ejecución: El usuario seleccionara la opción de Repositorio. Posteriormente seleccionar la opción de Preguntas.	
Resultado: Se mostrarán las preguntas del repositorio.	
Evaluación: La prueba se realizó de manera exitosa.	

Tabla 4.124: Prueba listar pregunta.

Fuente: Elaborado por el investigador.

PRUEBA DE ACEPTACIÓN	
Código: P87	Código historia: H12- Gestión de preguntas.
Nombre: Prueba agregar pregunta.	
Condición de ejecución: Iniciar sección con el rol de administrador, profesor.	
Proceso de Ejecución: El usuario seleccionara la opción Repositorio. Posteriormente seleccionar la opción de Preguntas. El usuario seleccionará la asignatura, el tema y tipo de pregunta y dará clic en el botón de Agregar. El usuario ingresara la información correspondiente. El usuario dará clic en el botón Guardar. El usuario aceptara el mensaje de confirmación para agregar una pregunta.	
Resultado: Se mostrará un mensaje de éxito al agregar una pregunta.	
Evaluación: La prueba se realizó de manera exitosa.	

Tabla 4.125: Prueba agregar pregunta.

Fuente: Elaborado por el investigador.

PRUEBA DE ACEPTACIÓN	
Código: P88	Código historia: H12- Gestión de preguntas.
Nombre: Prueba editar pregunta.	
Condición de ejecución: Iniciar sección con el rol de administrador, registros de preguntas existentes.	
Proceso de Ejecución: El usuario seleccionara la opción Repositorio. Posteriormente seleccionar la opción de Preguntas. El usuario dará clic en el botón Editar de la respectiva pregunta. El usuario ingresara la información a editar. El usuario dará clic en botón Guardar. El usuario aceptara el mensaje de confirmación para editar una pregunta.	
Resultado: Se mostrará un mensaje de éxito al editar una pregunta.	
Evaluación: La prueba se realizó de manera exitosa.	

Tabla 4.126: Prueba editar pregunta.

Fuente: Elaborado por el investigador.

PRUEBA DE ACEPTACIÓN	
Código: P89	Código historia: H12- Gestión de preguntas.
Nombre: Prueba eliminar tema de una pregunta.	
Condición de ejecución: Iniciar sección con el rol de administrador o profesor, registros de preguntas existentes.	
Proceso de Ejecución: El usuario seleccionara la opción Repositorio. Posteriormente seleccionar la opción de Preguntas. El usuario dará clic en botón Eliminar de la respectiva pregunta. El usuario aceptara el mensaje de confirmación para eliminar una pregunta.	
Resultado: Se mostrará un mensaje de éxito al eliminar una pregunta.	
Evaluación: La prueba se realizó de manera exitosa.	

Tabla 4.127: Prueba eliminar pregunta.

Fuente: Elaborado por el investigador.

PRUEBA DE ACEPTACIÓN	
Código: P90	Código historia: H12- Gestión de preguntas.
Nombre: Prueba activar una pregunta.	
Condición de ejecución: Iniciar sección con el rol de administrador o profesor, registros de preguntas inactivas existentes.	
Proceso de Ejecución: El usuario seleccionara la opción Repositorio. Posteriormente seleccionar la opción de Buscar. El usuario buscara los registros inactivos. El usuario dará clic en el botón Activar de la respectiva pregunta. El usuario aceptara el mensaje de confirmación para activar una pregunta.	
Resultado: Se mostrará un mensaje de éxito al activar un tema de una pregunta.	
Evaluación: La prueba se realizó de manera exitosa.	

Tabla 4.128: Prueba activar pregunta.

Fuente: Elaborado por el investigador.

PRUEBA DE ACEPTACIÓN	
Código: P91	Código historia: H13- Gestión de usuarios.
Nombre: Prueba cambiar clave.	
Condición de ejecución: Iniciar sección.	
Proceso de Ejecución: El usuario seleccionara la opción de Cambiar Contraseña. Posteriormente seleccionar la opción de Preguntas. El usuario ingresara la contraseña actual, la nueva contraseña y la confirmación de contraseña. El usuario dará clic en el botón Cambiar contraseña. El usuario confirmara el mensaje de confirmación cambiar la contraseña.	
Resultado: Se mostrará un mensaje de éxito al cambiar la contraseña.	
Evaluación: La prueba se realizó de manera exitosa.	

*Tabla 4.129: Prueba cambiar clave.
Fuente: Elaborado por el investigador.*

PRUEBA DE ACEPTACIÓN	
Código: P92	Código historia: H13- Gestión de usuarios.
Nombre: Prueba reiniciar clave.	
Condición de ejecución: Iniciar sección con el rol de administrador.	
Proceso de Ejecución: El usuario seleccionara la opción Usuarios. Posteriormente dará clic en la opción de Reset Clave del respectivo usuario.	
Resultado: Se mostrará un mensaje de éxito al reiniciar una clave.	
Evaluación: La prueba se realizó de manera exitosa.	

*Tabla 4.130: Prueba reiniciar clave.
Fuente: Elaborado por el investigador.*

PRUEBA DE ACEPTACIÓN	
Código: P93	Código historia: H13- Gestión de usuarios.
Nombre: Prueba recuperar clave.	
Condición de ejecución: Iniciar sección.	
Proceso de Ejecución: El usuario seleccionara la opción “Has olvidado tu clave”. Posteriormente el usuario ingresara el usuario y correo electrónico. El sistema enviara un correo electrónico con un código único. El usuario ingresara el código de comprobación	
Resultado: Se mostrará un mensaje de éxito al recuperar clave.	
Evaluación: La prueba se realizó de manera exitosa.	

*Tabla 4.131: Prueba recuperar clave.
Fuente: Elaborado por el investigador.*

PRUEBA DE ACEPTACIÓN	
Código: P94	Código historia: H14- Gestión de evaluaciones.
Nombre: Prueba listar evaluaciones.	
Condición de ejecución: Iniciar sección con el rol de profesor.	
Proceso de Ejecución: El usuario seleccionara la opción de Curso.	

Posteriormente seleccionar la opción de Ver Evaluaciones en la asignatura correspondiente.
Resultado: Se mostrarán las evaluaciones correspondientes a la asignatura.
Evaluación: La prueba se realizó de manera exitosa.

*Tabla 4.132: Prueba listar evaluaciones.
Fuente: Elaborado por el investigador.*

PRUEBA DE ACEPTACIÓN	
Código: P95	Código historia: H14- Gestión de evaluaciones.
Nombre: Prueba agregar evaluaciones.	
Condición de ejecución: Iniciar sección con el rol de profesor.	
Proceso de Ejecución: El usuario seleccionara la opción Curso. Posteriormente seleccionar la opción Crear Evaluación en la asignatura correspondiente. El usuario ingresara la información correspondiente. El usuario dará clic en el botón Guardar. El usuario aceptara el mensaje de confirmación para agregar una evaluación.	
Resultado: Se mostrará un mensaje de éxito al agregar una evaluación.	
Evaluación: La prueba se realizó de manera exitosa.	

*Tabla 4.133: Prueba agregar evaluación.
Fuente: Elaborado por el investigador.*

PRUEBA DE ACEPTACIÓN	
Código: P96	Código historia: H14- Gestión de evaluaciones.
Nombre: Prueba editar evaluación.	
Condición de ejecución: Iniciar sección con el rol de profesor, registros de evaluaciones existentes.	
Proceso de Ejecución: El usuario seleccionara la opción Curso. Posteriormente seleccionar la opción Editar Evaluación en la asignatura correspondiente. El usuario ingresara la información correspondiente. El usuario dará clic en el botón Guardar. El usuario aceptar el mensaje de confirmación para editar una evaluación.	
Resultado: Se mostrará un mensaje de éxito al editar una evaluación.	
Evaluación: La prueba se realizó de manera exitosa.	

*Tabla 4.134: Prueba editar evaluación.
Fuente: Elaborado por el investigador.*

PRUEBA DE ACEPTACIÓN	
Código: P97	Código historia: H14- Gestión de evaluaciones.
Nombre: Prueba eliminar evaluación.	
Condición de ejecución: Iniciar sección con el rol de profesor, registros de evaluaciones existentes.	
Proceso de Ejecución: El usuario seleccionara la opción Curso. Posteriormente dará clic en la opción Eliminar Evaluación en la asignatura correspondiente. El usuario aceptara el mensaje de confirmación para eliminar una evaluación.	

Resultado: Se mostrará un mensaje de éxito al eliminar una evaluación.
Evaluación: La prueba se realizó de manera exitosa.

Tabla 4.135: Prueba eliminar evaluación.

Fuente: Elaborado por el investigador.

PRUEBA DE ACEPTACIÓN	
Código: P98	Código historia: H14- Gestión de evaluaciones.
Nombre: Prueba activar evaluación.	
Condición de ejecución: Iniciar sección con el rol de profesor, registros de evaluaciones existentes.	
Proceso de Ejecución: El usuario seleccionara la opción Curso. Posteriormente dará clic en la opción Activar Evaluación en la asignatura correspondiente. El usuario confirmara el mensaje de confirmación para activar una evaluación.	
Resultado: Se mostrará un mensaje de éxito al activar una pregunta.	
Evaluación: La prueba se realizó de manera exitosa.	

Tabla 4.136: Prueba activar pregunta.

Fuente: Elaborado por el investigador.

PRUEBA DE ACEPTACIÓN	
Código: P99	Código historia: H14- Gestión de evaluaciones.
Nombre: Prueba agregar pregunta a una evaluación.	
Condición de ejecución: Iniciar sección con el rol de profesor, registros de evaluaciones y preguntas existentes.	
Proceso de Ejecución: El usuario seleccionara la opción Curso. Posteriormente dará clic en la opción Ver Evaluación en la asignatura correspondiente. Dar clic en la opción Agregar Preguntas en la asignatura correspondiente. El usuario ingresar la valoración de la pregunta y seleccionara la pregunta. El usuario dará clic en el botón Guardar.	
Resultado: Se mostrará un mensaje de éxito al agregar una pregunta.	
Evaluación: La prueba se realizó de manera exitosa.	

Tabla 4.137: Prueba agregar pregunta a una evaluación.

Fuente: Elaborado por el investigador.

PRUEBA DE ACEPTACIÓN	
Código: P102	Código historia: H16- Iniciar Evaluación.
Nombre: Prueba iniciar evaluación.	
Condición de ejecución: Iniciar sección con el rol de estudiante.	
Proceso de Ejecución: El usuario seleccionara la opción Curso. Posteriormente dará clic en la opción Ver Evaluación en la asignatura correspondiente. Dar clic en la opción Iniciar Evaluación en la asignatura correspondiente. El usuario confirmara el mensaje de confirmación para Iniciar una evaluación El usuario contestara las preguntas correspondientes. El usuario dará clic en guardar.	

El usuario confirmara el mensaje de confirmación para guardar una evaluación
Resultado: Se mostrará un mensaje de éxito al iniciar y finalizar una evaluación.
Evaluación: La prueba se realizó de manera exitosa.

Tabla 4.138: Prueba iniciar evaluación.

Fuente: Elaborado por el investigador.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- El desarrollo de la Aplicación Web Progresiva facilitó el proceso de realización, toma y seguimiento de pruebas de simulación, ahorrando tiempo y recursos para el centro de Capacitación y Nivelación Académica SMARTEL. De esta manera se consiguió dejar a un lado las hojas de cálculos y cuadernillos que ha sido de uso diario en la institución, lo que ocasionaba malestar entre los docentes por el esfuerzo y tiempo que requería su utilización.
- El análisis de la información y procesos que realiza el Centro de Capacitación y Nivelación Académica SMARTEL se realizó juntamente con el cliente, donde se detalló paso a paso los requerimientos expuesto por el mismo y junto a la metodología XP se llevó acabo el desarrollo de la Aplicación Web Progresiva permitiendo un desarrollo organizado y flexible, manteniendo una continua comunicación con el cliente.
- Angular 8, la tecnología de Front-End seleccionada fue la correcta, permitió realizar un desarrollo rápido y completo mediante las diferentes características que posee, entre las cuales podemos mencionar el “FromControl”, que ayudo a la realización y validación de formularios reactivos. Además, un punto a favor es la compatibilidad que posee con un gran número de librerías, que ayudaron a resolver problemas específicos como es el ingreso y visualización de fórmulas matemáticas, que fue resultado mediante la librería ngKatex.
- ASP.NET CORE permitió agilizar el proceso de desarrollo en el servidor a través de la creación del modelo de datos y la autenticación de usuarios mediante el uso de Entity Framework y ASP.NET CORE IDENTITY respetivamente, además de poseer una gran comunidad que facilitó la resolución de problemas presentados en el desarrollo y publicación del proyecto.

5.2. Recomendaciones

- Se recomienda a los nuevos desarrolladores que ingresan al mundo del desarrollo de las Aplicaciones Web Progresivas, realizar un análisis previo, para determinar si la aplicación web a construir cumple con las características que ofrece las Aplicaciones Web Progresiva, ya que hasta el momento no pueden realizar funciones avanzadas tales como acceder a la lista de contactos, características del sistema entre otras.
- Se sugiere al Centro de Capacitación y Nivelación Académica SMARTEL continuar con el desarrollo e incorporación de módulos restantes como cobranza, cursos en línea, con el fin de brindar nuevos servicios y llevar un control total y automatizado de los procesos que realizan.
- Se aconseja al Centro de Capacitación y Nivelación Académica SMARTEL implementar características adicionales soportadas en las Aplicación Web Progresiva como notificaciones, para brindar un mejor servicio y lograr una mayor interacción con el usuario.

MATERIAL DE REFERENCIA

Bibliografía

- [1] E. W. Rubio Guerrero and M. G. Gómez Zermeño, “Propuesta de Diseño de un Modelo Educativo Integral para Capacitaciones Corporativas,” *Acad. y Virtualidad*, vol. 9, no. 1, p. 52, 2016.
- [2] Ministerio de Educación, “Graduación virtual de Bachilleres régimen Sierra-Amazonía 2019-2020,” 2020. [Online]. Available: <https://educacion.gob.ec/graduacion-virtual-de-bachilleres-regimen-sierra-amazonia-2019-2020/>. [Accessed: 25-Aug-2020].
- [3] R. W. Garbarini, P. Cigliuti, and L. Recchini, “Construcción y Validación de un Sistema Web / móvil como Herramienta de Intervención Tecnológica para la Evaluación Dinámica en Asignatura de Grado en Ingeniería en Sistemas de Información,” vol. 65, pp. 65–79, 2018.
- [4] M. T. Parra-Santos, J. M. Molina Jordá, G. Luna Sandoval, I. Milanovic, G. Casanova Pastor, and F. Castro Ruiz, “La aplicación SOCRATIVE como herramienta de evaluación y precursor de la participación en el aula,” in *Investigación en docencia universitaria: Diseñando el futuro a partir de la innovación educativa*, 2017, pp. 677–683.
- [5] F. C. Troya, “Diseño de un software de evaluaciones en línea para los estudiantes y docentes de grado de la Universidad Tecnológica Empresarial De Guayaquil,” no. 9, 2017.
- [6] M. Matute and W. Alex, “Desarrollo de un sistema web progresivo para el control y toma de lecturas de agua potable,” Universidad de Guayaquil, 2019.
- [7] E. Saks, “JavaScript frameworks : Angular vs React vs Vue,” Haaga-Helia University of Applied Sciences, 2019.
- [8] J. J. Tanta Diaz, “Comparación del rendimiento entre angular4 y reactjs, basado en el modelo rail, en la progressive web app de Glup S.A.,” Universidad César Vallejo, 2017.
- [9] A. Perez et al., “Implementación de una aplicación web progresiva (PWA), para la identificación y notificación de mascotas extraviadas usando placas con código de respuesta rápida (QR) en collares,” 2018.
- [10] A. P. E. Figueroa, “El Ecosistema Móvil y las noticias en la era de las plataformas digitales,” Universidad Pompeu Fabra, 2018.
- [11] W. A. M. Matute, “Desarrollo de un sistema web progresivo para el control y toma de lecturas de agua potable,” Universidad de las Américas, 2019.
- [12] F. Robledo, “Desarrollo de aplicación SPA en REACT, apoyada en API RESTFull para la escucha de emisoras de radio online,” Universidad de

Alcalá Escuela Politécnica Superior, 2019.

- [13] G. Arcos-Medina, J. Menéndez, and J. Vallejo, “Comparative Study of Performance and Productivity of MVC and MVVM design patterns,” *KnE Eng.*, vol. 1, no. 2, p. 241, 2018.
- [14] S. Ivanova and G. Georgiev, “Using modern web frameworks when developing an education application: a practical approach,” *2019 42nd Int. Conv. Inf. Commun. Technol. Electron. Microelectron.*, pp. 1485–1491, 2019.
- [15] A. Sandra Almeida and V. Perez Cavenago, “Arquitectura de Software: Estilos y Patrones,” Universidad Nacional De La Patagonia San Juan Bosco Argentina, 2007.
- [16] M. Aljamea and M. Alkandari, “MMVMi: A validation model for MVC and MVVM design patterns in iOS applications,” *IAENG Int. J. Comput. Sci.*, vol. 45, no. 3, pp. 377–389, 2018.
- [17] E. Wohlgethan, “Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js,” Hamburg University of Applied Sciences, 2018.
- [18] T. Nevalainen, “MEAN Software Stack,” Degree Program. Bus. Inf. Technol. ICT, p. 23 p., 2018.
- [19] L. Liang, L. Zhu, W. Shang, D. Feng, and Z. Xiao, “Express supervision system based on NodeJS and MongoDB,” *Proc. - 16th IEEE/ACIS Int. Conf. Comput. Inf. Sci. ICIS 2017*, pp. 607–612, 2017.
- [20] D. G. Valencia Altamirano, “Análisis de frameworks de desarrollo de api rest y su impacto en el rendimiento de aplicaciones web con arquitectura Spa.,” UNIVERSIDAD TÉCNICA DEL NORTE, 2018.
- [21] S. L. Daniel Roth, Rick Anderson, “Introduction to ASP.NET Core,” 2019. [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-3.1>. [Accessed: 29-Jan-2019].
- [22] G. J. F. y H. L. RIVAS Carlos, CORONA Verónica Paola, “Metodologías actuales de desarrollo de software,” *Rev. Tecnol. e Innovación*, vol. 2, pp. 67–73, 2015.
- [23] R. Tinoco Gómez, López, P. Pablo, and U. N. M. de S. M. P. Bacalla, Salas, “Criterios de selección de metodologías de desarrollo de software,” *Ind. Data*, vol. 13, no. 2, pp. 70–74, 2010.
- [24] P. Cáceres, E. Marcos, and G. Kybele, “Procesos Ágiles para el Desarrollo de Aplicaciones Web,” *Taller Web Eng. las Jornadas Ing. del Softw. y Bases Datos*, vol. 2001, 2001.
- [25] M. Penadés and P. Letelier Torres, “Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP),” *Técnica Adm.*, vol. 5, no. 26, p. 1, 2006.
- [26] Y. A. Ramírez Jesús, “Identificación de los factores de riesgo en el desarrollo de aplicaciones web utilizando metodologías tradicionales,” Universidad

Técnica de Machala, 2019.

- [27] S. L. M. Montoya, C. J. M. Sepúlveda, and R. L. M. Jiménez, “Análisis comparativo de las metodologías ágiles en el desarrollo de software aplicadas en Colombia,” *Cimted*, no. October, pp. 450–464, 2017.
- [28] A. Calderón and S. Valverde, Rebaza, “Metodologías Ágiles,” *Esc. Informática. Trujillo Univ. Nac. Trujillo*, p. 37, 2007.
- [29] A. López Gil, “Estudio comparativo de metodologías tradicionales y ágiles para proyectos de Desarrollo de Software,” *Universidad de Valladolid*, 2018.
- [30] D. Armando, “Desarrollo del sistema académico aplicando la herramienta ASP.NET CORE 2 para la Escuela Eufrasia Pelletier,” *Universidad Técnica del Norte*, 2019.
- [31] B. M. Montero, H. V. Cevallos, and Jefferson Dávila Cuesta, “Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software,” *Espirales Rev. Multidiscip. Investig. ISSN 2550-6862*, vol. 2, no. 17, pp. 114–121, 2018.
- [32] M. Ramírez, M. Salgado, H. B. Ramírez, E. Manrique, N. Osuna, and R. Rosales, “Metodología SCRUM y desarrollo de Repositorio Digital,” *RISTI - Rev. Ibérica Sist. y Tecnol. Inf.*, vol. 17, pp. 1062–1073, 2019.
- [33] P. Letelier and C. Penadés, “Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP),” *Conf. Extrem. Program. Agil. Process. Softw. Eng.*, 2012.
- [34] S. Edition, *Essential Angular for ASP. NET Core MVC 3: A practical guide to successfully using both in your projects*, Second Edition. 2019.
- [35] M. Hajian, *Progressive Web Apps with Angular*. 2019.

Glosario de Términos y Acrónimos

PWA:	Progressive Web Apps - Aplicaciones Web Progresivas.
SPA:	Single Page Application- Aplicaciones de una sola página.
CSS:	Cascade Style Sheets – Hojas de Estilo en cascada.
JSON:	JavaScript Object Notation - Notación de Objetos de JavaScript.
JSX:	JavaScript XML.
HTML:	HyperText Markup Language.
URL:	Uniform Resource Locator - Localizador de Recursos Uniforme.
API:	Application Programming Interface - Interfaz de Programación de Aplicaciones.
REST:	Representational State Transfer - Transferencia de Estado Representacional.
HTTPS:	HyperText Transfer Protocol Secure - Protocolo de transferencia de hipertexto Seguro.
HTTP:	HyperText Transfer Protocol,-Protocolo de transferencia de hipertexto.
MVC:	Modelo Vista Controlador.
MVVM:	Modelo Vista Modelo de Vista.
NPM:	Node Package Manager – Administrador de Paquetes Node.
IIS:	Internet Information Service - Servicio de Información de Internet.
Nginx:	Engine x.
RUP:	Rational Unified Procces.
MSF:	Microsoft Solution Framework.
XP:	Extreme Programming – Programación extrema.
CRC:	Clase-responsabilidad-colaborador.
CMMI:	Capability Maturity Model Integration - Integración de sistemas modelos de madurez de capacidades.
DTO:	Objetos de transferencia de datos.
ORM:	Mapeo Objeto – Relación.
JWY:	JSON Web Token.
CLI:	Command-Line Interface.

Anexos y Apéndices

Anexo B: Manual de usuario

IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PROGRESIVA PARA LA GESTIÓN DE PRUEBAS DE SIMULACIÓN PARA EL INGRESO A INSTITUCIONES MILITARES Y UNIVERSIDADES EN EL CENTRO DE CAPACITACIÓN Y NIVELACIÓN ACADÉMICA SMARTEL.

Manual de Usuario

Versión: 001

Fecha: 02/08/2020

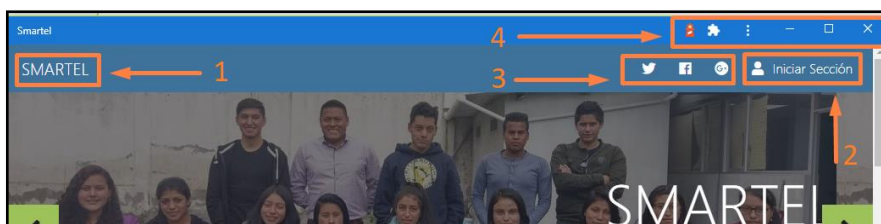
INTRODUCCIÓN

La presente aplicación web progresiva fue desarrollada como Proyecto de Titulación de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato

El presente manual usuario fue desarrollado con el fin de facilitar y guiar al usuario en los diferentes procesos que realiza la aplicación web progresiva para la gestión de pruebas de simulación en el Centro de Capacitación y Nivelación Académica SMARTEL.

1. PÁGINA DE INICIO

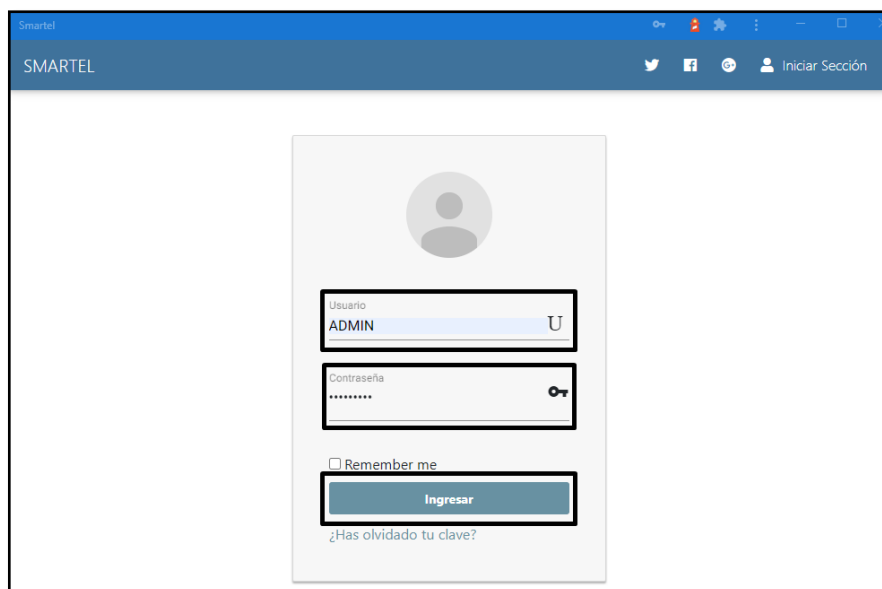
Muestra la ventana principal de la aplicación web progresiva.



1. **SAMRTEL:** Este enlace permite al usuario navegar directamente a la pantalla de inicio desde cualquier parte de la aplicación web progresiva.
2. **Inicio Sección:** Este enlace dirige al usuario a la venta de ingreso de sección de usuarios.
3. **Redes Sociales:** Estos enlaces redirecciona a las principales redes sociales del Centro de Capacitación y Nivelación Académica SMARREL.
4. **Opciones de Cerrar, minimizar y maximizar:** Permite cerrar maximizar y minimizar la aplicación web progresiva.

1.2. Inicio de sección

El usuario ingresará el nombre del usuario y contraseña y dará clic en Ingresar. El sistema validará el usuario y contraseña de ser incorrecta la aplicación mostrara una notificación con el respectivo error.



Al ingresar el usuario a la aplicación se visualizará y activará un menú horizontal correspondiente al rol asignado al usuario, además de un menú vertical con la opción de cambiar clave de usuario y resetear clave en caso del rol administrador.

2. ROL ADMINISTRADOR



2.1. Cambiar Clave:

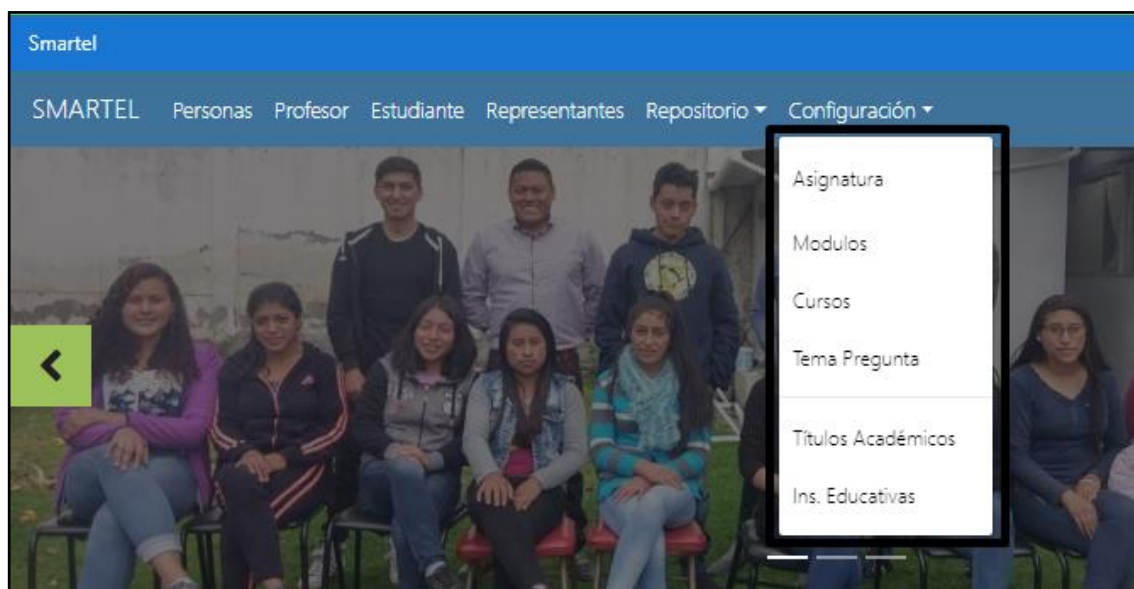
Al dar clic en la opción Cambiar Clave del menú vertical la aplicación nos redireccionara a un formulario en donde el usuario ingresara la contraseña actual, nueva contraseña y la confirmación de la contraseña. La nueva contraseña deberá contar con una longitud superior a 6 caracteres y estar conformada entre números, letras mayúsculas, letras minúsculas y caracteres alfanuméricos. Después de cumplir con las diferentes validaciones se activará el botón de “Cambiar Contraseña” y el usuario procederá a dar clic y confirmará que desea cambiar la contraseña. Al realizarse con éxito la aplicación el usuario tendrá que ingresar nuevamente a la aplicación.

Two screenshots of the password change form. The left screenshot shows the form with the 'Cambiar Contraseña' button disabled (greyed out). The right screenshot shows the same form but with the 'Cambiar Contraseña' button enabled (blue and active). Both forms have the following fields: 'Usuario' (ADMIN), 'Contraseña' (masked with dots), 'Nueva Contraseña' (masked with dots), and 'Confirmación Contraseña' (masked with dots). Each field has a key icon to toggle visibility. The background of the form is light grey with a user icon placeholder at the top.

El menú vertical posee una serie de opciones diferentes para cada rol de la aplicación, a continuación, se expone cada una de las opciones que posee este menú.

2.2. Configuración

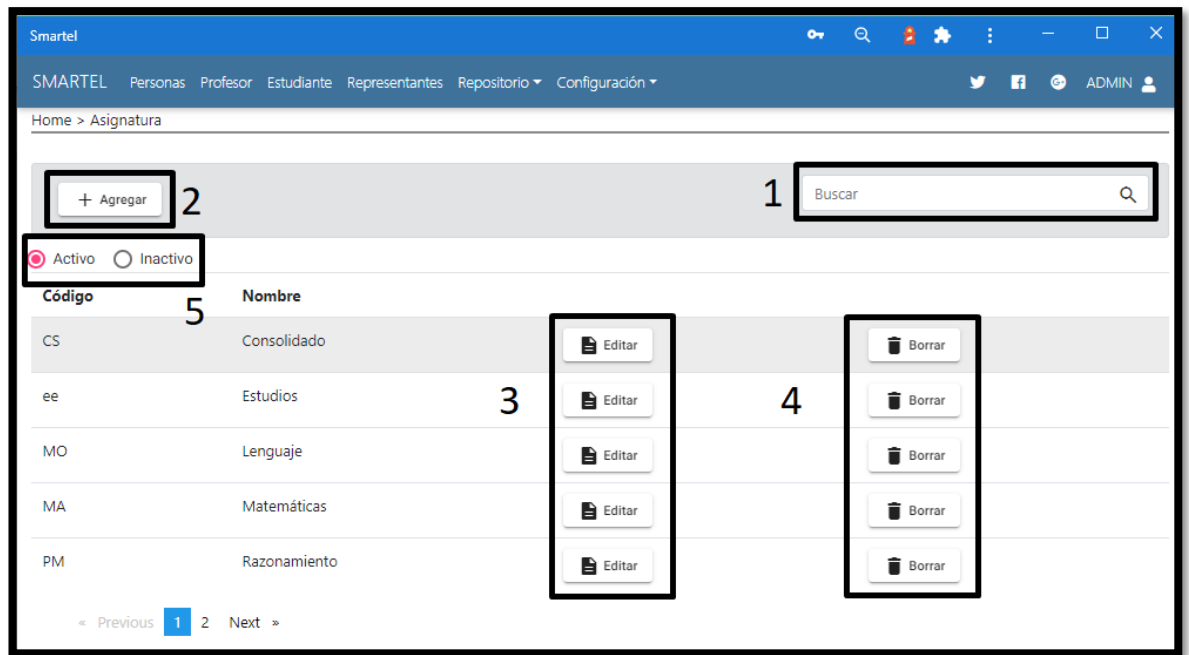
Posee una serie de submenús que se detallan a continuación:



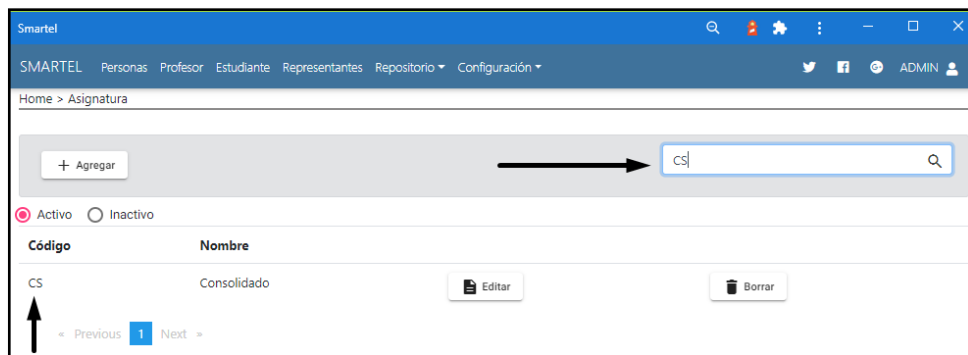
A continuación, se explica las diferentes interfaces para realizar el mantenimiento de una asignatura, no se expondrá las pantallas para los mantenimientos de las demás opciones ya que las pantallas conservan el mismo formato y acciones que Asignatura.

2.2.1. Asignatura

La pantalla de Asignatura permite realizar diferentes acciones que se describen a continuación:



1. Buscar: Permite realizar una búsqueda de las diferentes asignaturas existentes filtradas por el estado seleccionado ya sea activo o inactivo. Para poder realizar la búsqueda se deberá ingresar mínimo 2 caracteres y no distingue de mayúsculas o minúsculas.



2. Agregar: Al presionar el botón de agregar se desplegará una ventana emergente en la cual el usuario ingresará el código y nombre de la respectiva asignatura, se realizará las validaciones correspondientes, posteriormente se activará el botón guardar y el administrador deberá dar clic en la opción de guardar y confirmar que desea guardar una asignatura.

3. Editar: La opción editar desplegará una ventana emergente en la que se podrá editar el nombre de la asignatura el administrador deberá dar clic en la opción de guardar y confirmar que desea guardar una asignatura.

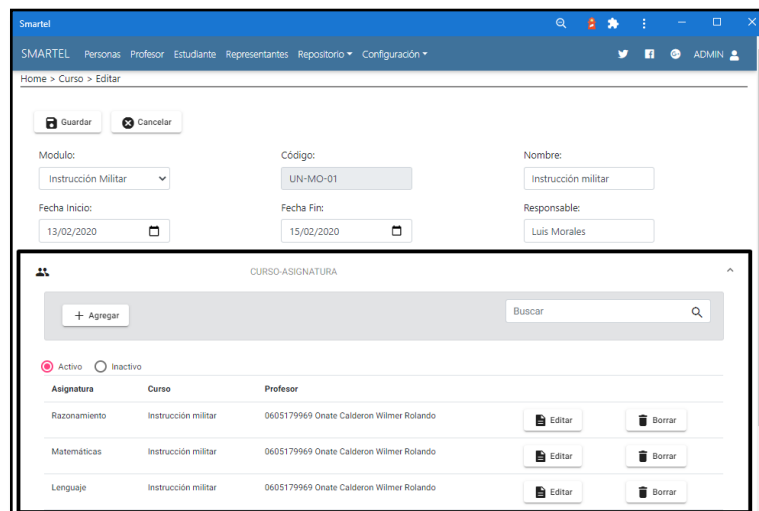
4. Eliminar: Al dar clic en el botón eliminar se desplegará un mensaje de confirmación para eliminar el registro, el administrador deberá confirmara el mensaje y se procederá a eliminar la asignatura, se podrá eliminar una asignatura siempre y cuando esta no se encuentre asociada a un curso.

5. Estado Activo o Inactivo: Esta opción estará presente en los registros que se realice una eliminación lógica. Listara las asignatura activas e inactivas. Al seleccionar la opción inactiva visualizara las asignaturas inactivas las misma podrán ser activadas dando clic en el botón activar de la respectiva asignatura.

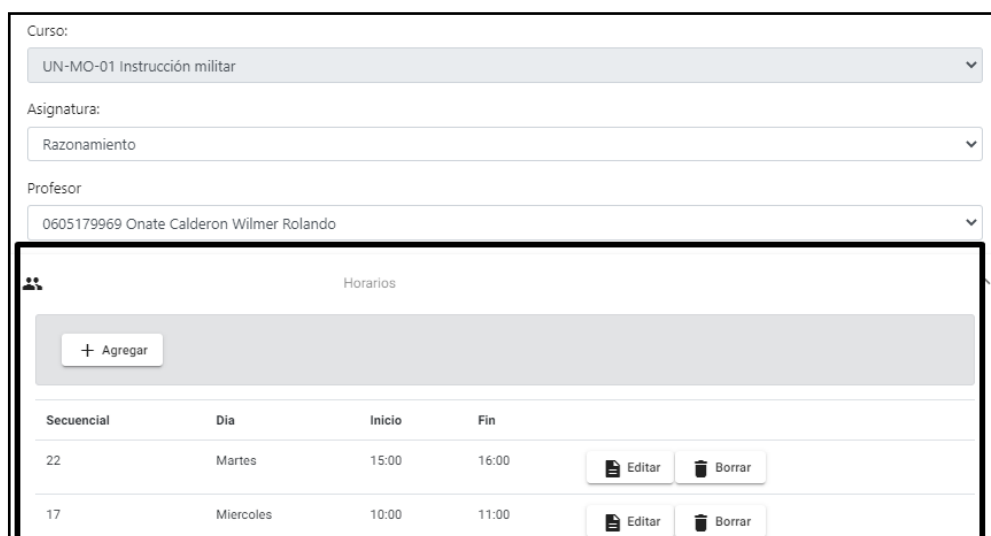


2.2.2. Ingreso de Asignaturas en un Curso

La asignación de asignaturas a un curso se realizará mediante la opción de editar Curso, se mostrará un desplegable donde se podrá realizar las diferentes acciones como buscar, agregar, editar, eliminar.

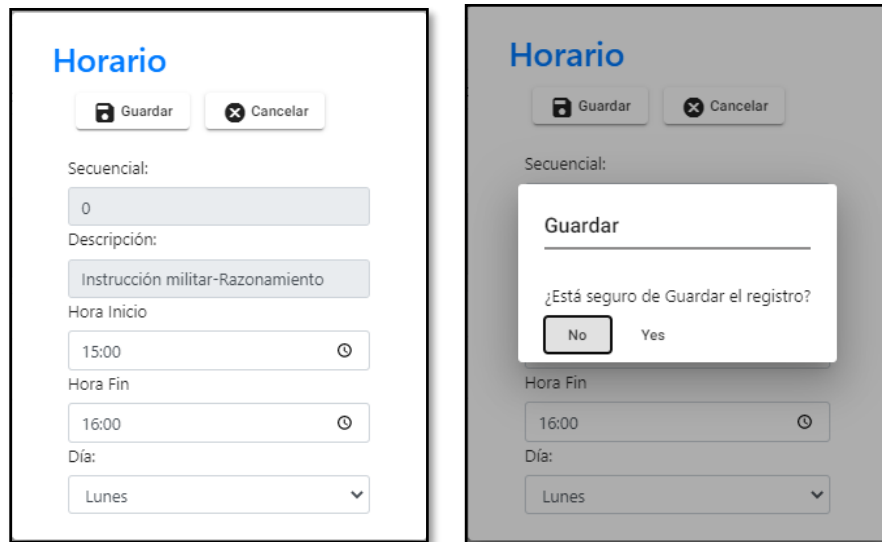


La opción de editar de Curso – Asignatura me permite editar un Curso – Asignatura y realizar las diferentes acciones a horario por Curso Asignatura.



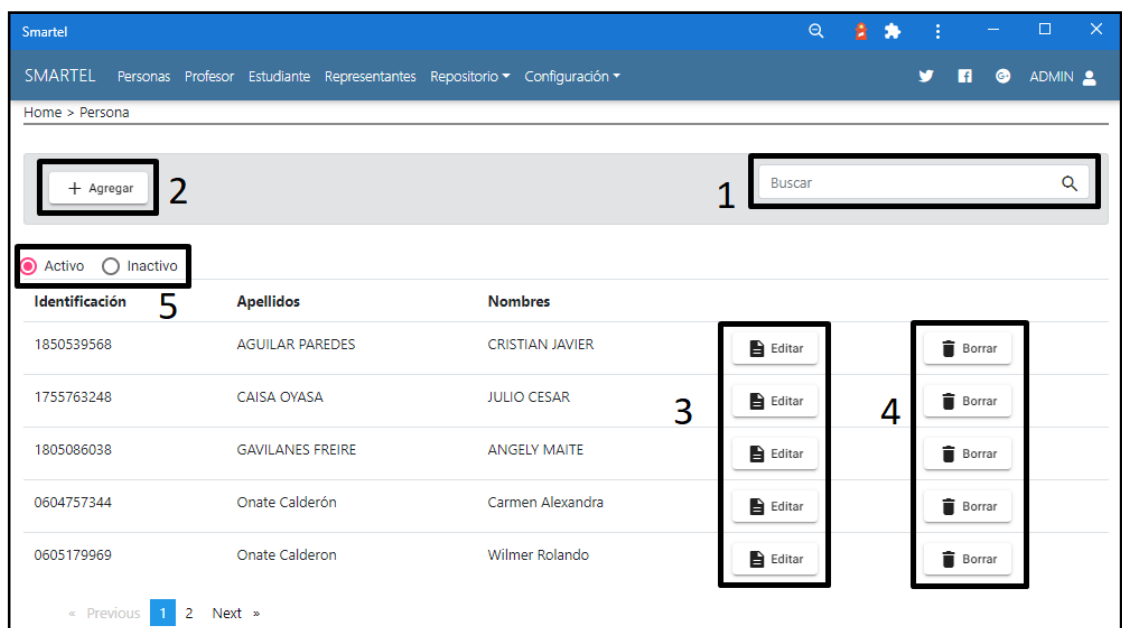
2.2.3. Agregar Horario por Curso – Asignatura

Para ingresar una hora a un Curso – Asignatura el administrador deberá dar clic en el botón agregar y se visualizará una ventana emergente, el administrador ingresará la hora de inicio , la hora fin y el día y posteriormente dará clic en el botón Guardar , se mostrará un mensaje de confirmación al que el administrador deberá dar clic en la opción de Si.

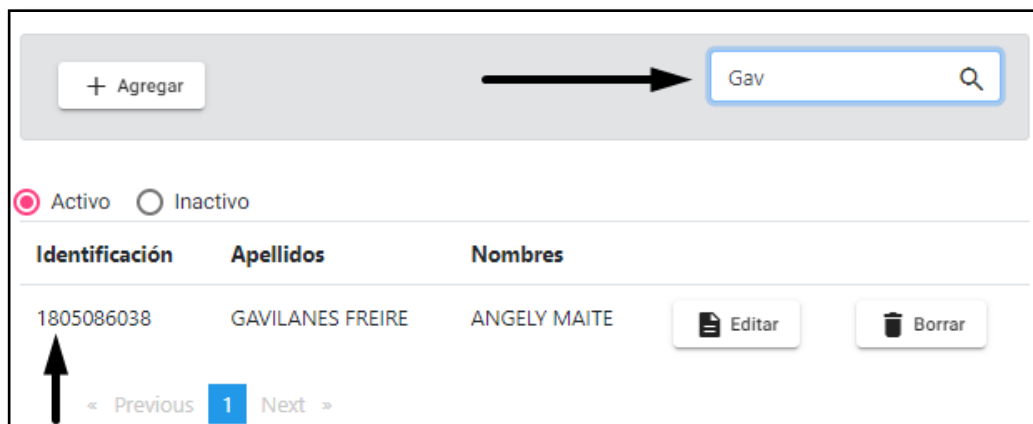


2.3. Personas

El ingreso de una persona es fundamental para la aplicación web progresiva por que contendrá la información básica para la creación de estudiantes, profesores y representantes.



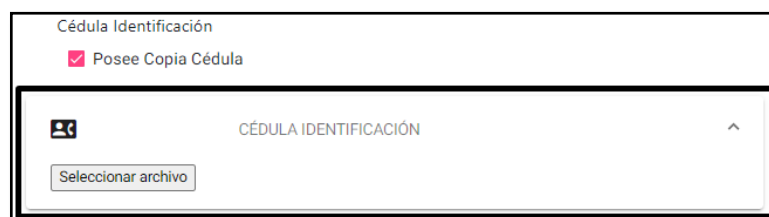
1. Buscar: Permite realizar una búsqueda de las diferentes personas existentes filtradas por el estado seleccionado ya sea activo o inactivo. Para poder realizar la búsqueda se deberá ingresar mínimo 3 caracteres y no distingue de mayúsculas o minúsculas.



2. Agregar: Al presionar el botón de agregar se redireccionará a una ventana donde el administrados podrá ingresar información de persona, se realizará las validaciones correspondientes, posteriormente se activará el botón guardar y el usuario deberá dar clic en la opción de guardar y confirmar que desea guardar una persona.

Los campos de email, contraseña email y copia de cedula de identificación son opcionales, estos dos últimos solo se ingresarán si la persona está dispuesta a proporcionarlos.

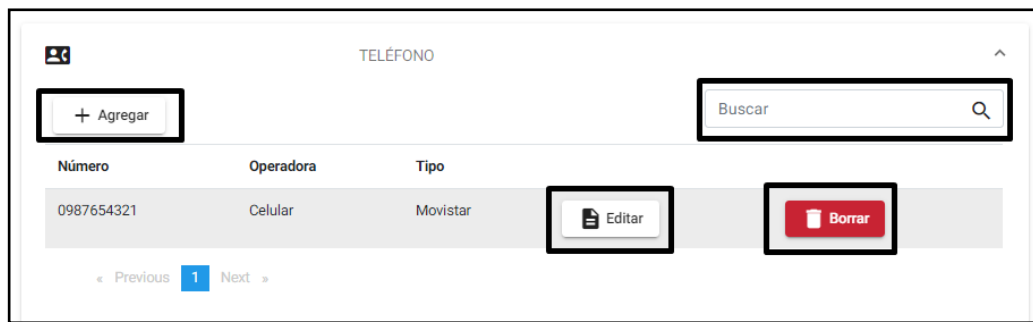
Al dar check en la opción de “Posee Copia Cédula” se mostrará un desplegable donde se podrá seleccionar una imagen desde el equipo.



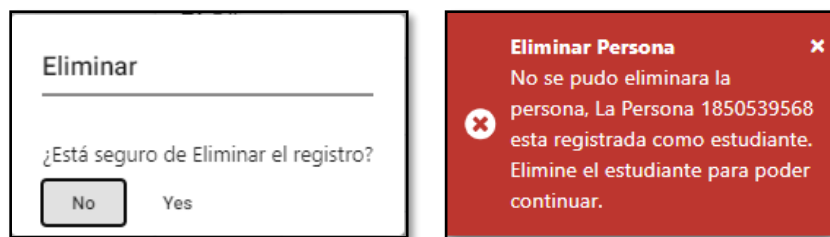
3. Editar: El botón editar redireccionará a una ventana en al que se podrá editar los diferentes capos de una persona a excepción de tipo de identificación e identificación, el administrador deberá dar clic en la opción de guardar y confirmar que desea guardar la persona.

A screenshot of an "Editar" form for a person's identification details. At the top, there are "Guardar" and "Cancelar" buttons. The form is organized into three columns. The first column contains: "Tipo Identificación:" with a dropdown menu set to "CEDULA"; "Apellido Paterno" with a text input field containing "CAISA"; "Fecha Nacimiento" with a date picker set to "10/02/2000"; "Email" with a text input field containing "juliocaisa30@gmail.c"; and "Contraseña Email" with a text input field containing "Ingrese un contraseí". The second column contains: "Identificación:" with a text input field containing "1755763248"; "Apellido Materno" with a text input field containing "OYASA"; "Dirección domiciliaria" with a text input field containing "QUERO"; "Instrucción Académica" with a dropdown menu set to "ESTUDIANTE"; and "Cédula Identificación" with a checkbox labeled "Posee Copia Cédula". The third column contains: "Estado Civil:" with a dropdown menu set to "SOLTERO/A"; "Nombres" with a text input field containing "JULIO CESAR"; "Referencia Domiciliaria" with a text input field containing "QUERO"; and "Género" with radio buttons for "Masculino" (selected) and "Femenino".

Además, la opción de editar nos visualizara un desplegable donde podremos ingresar los números telefónicos que posee la persona. Las diferentes acciones de listar, agregar, editar y eliminar se realizan en la misma estructura antes mencionada en la opción de configuración > Asignatura.



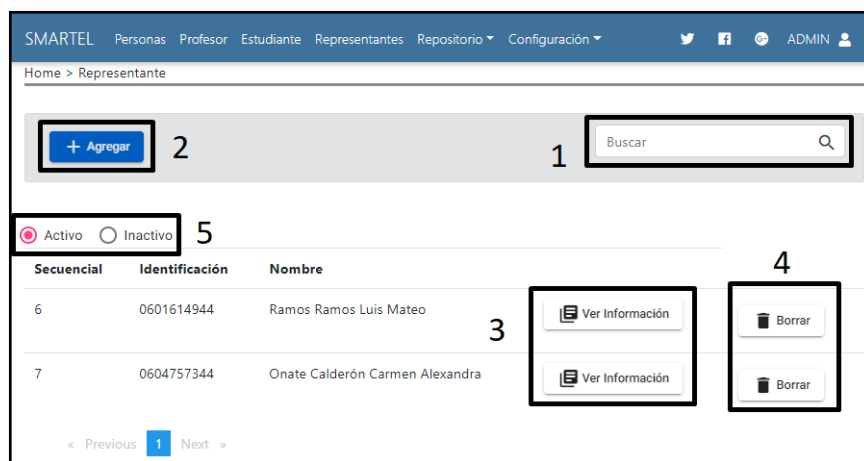
4. Eliminar: Al dar clic en el botón eliminar se desplegará un mensaje de confirmación para eliminar el registro, el usuario confirmara el mensaje y se procederá a eliminar la persona, se podrá eliminar una asignatura siempre y cuando esta no se encuentre asociada un estudiante, profesor o representante.



5. Estado Activo o Inactivo: Listara las personas activas e inactivas. Al seleccionar la opción inactiva visualizara las personas inactivas las misma podrán ser activadas dando clic en el botón activar de la respectiva persona.

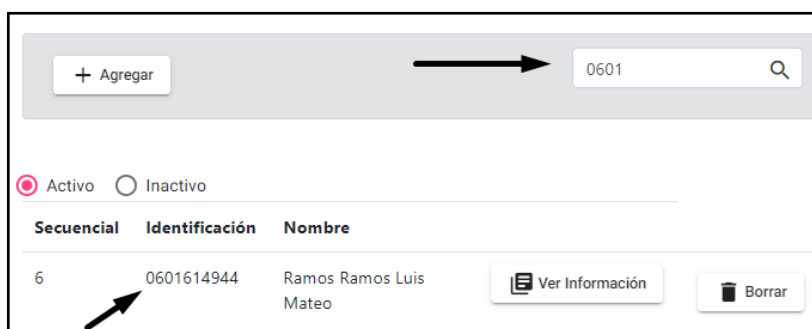
2.4. Representantes

Los representantes poseen las siguientes opciones que se describen a contunuacion:

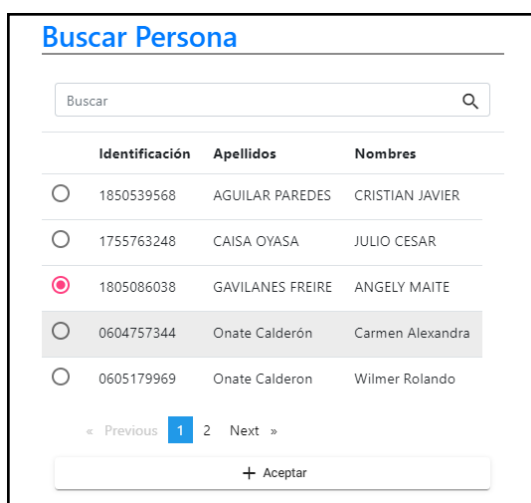


1. Buscar: Permite realizar una búsqueda de las diferentes representantes existentes filtradas por el estado seleccionado ya sea activo o inactivo. Para poder

realizar la búsqueda se deberá ingresar mínimo 3 caracteres y no distingue de mayúsculas o minúsculas.



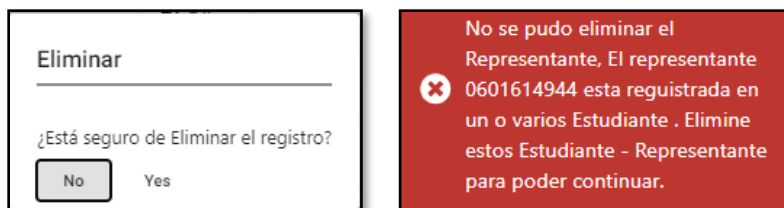
2. Agregar: Al presionar el botón de agregar se mostrará una ventana emergente donde el administrados seleccionara la persona que desea convertir en representante, se realizará las validaciones correspondientes para que no se dupliquen datos, posteriormente se activará el botón Aceptar y el usuario deberá dar clic en la opción de guardar y confirmar que desea guardar un representante.



3. Ver Información: El botón “Ver Información” mostrara a una ventana emergente en al que se podrá visualizar un resumen de la información que posee el representante.



4. Eliminar: Al dar clic en el botón eliminar se desplegará un mensaje de confirmación para eliminar el registro, el administrador confirmara el mensaje y se procederá a eliminar el representante, se podrá eliminar un representante siempre y cuando esta no se encuentre asociada un estudiante.

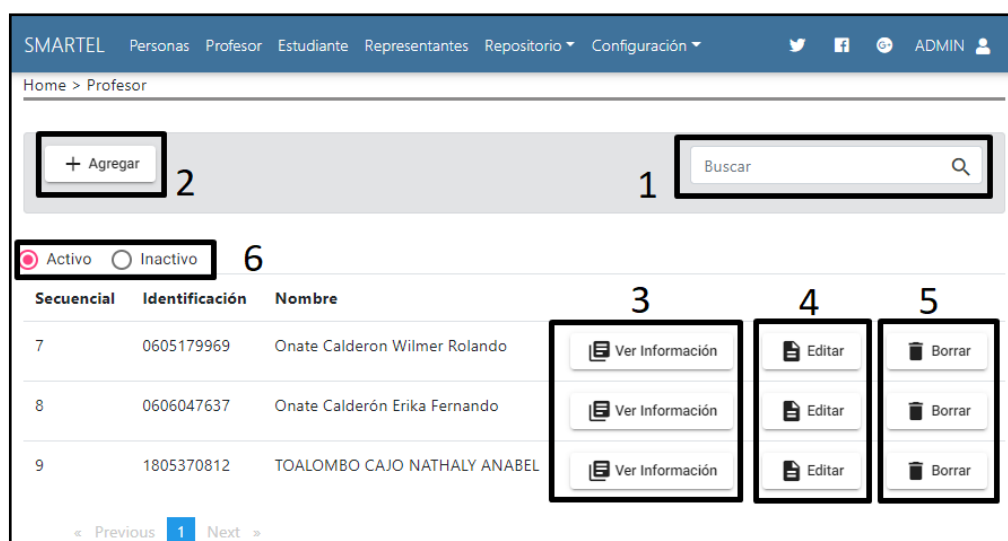


5. Estado Activo o Inactivo: Listara las personas activas e inactivas. Al seleccionar la opción inactiva visualizara las personas inactivas las misma podrán ser activadas dando clic en el botón activar de la respectiva representante.



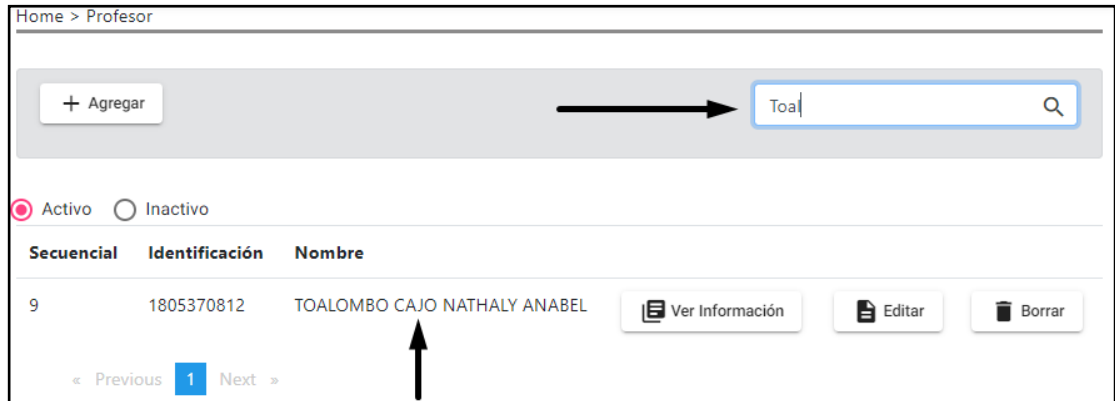
2.5. Profesor

La pantalla de profesores posee las siguientes acciones que se describen a continuación:

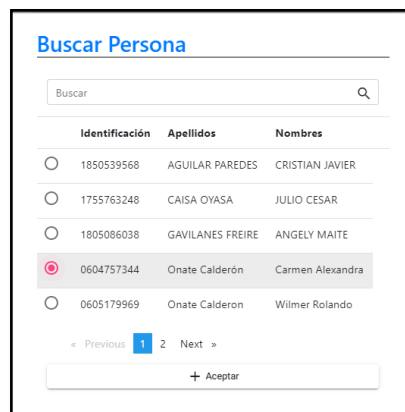


1. Buscar: Permite realizar una búsqueda de los diferentes profesores existentes filtradas por el estado seleccionado ya sea activo o inactivo. Para poder realizar la

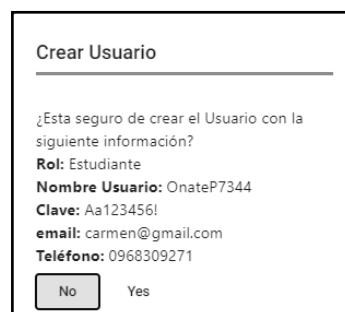
búsqueda se deberá ingresar mínimo 3 caracteres y no distingue de mayúsculas o minúsculas.



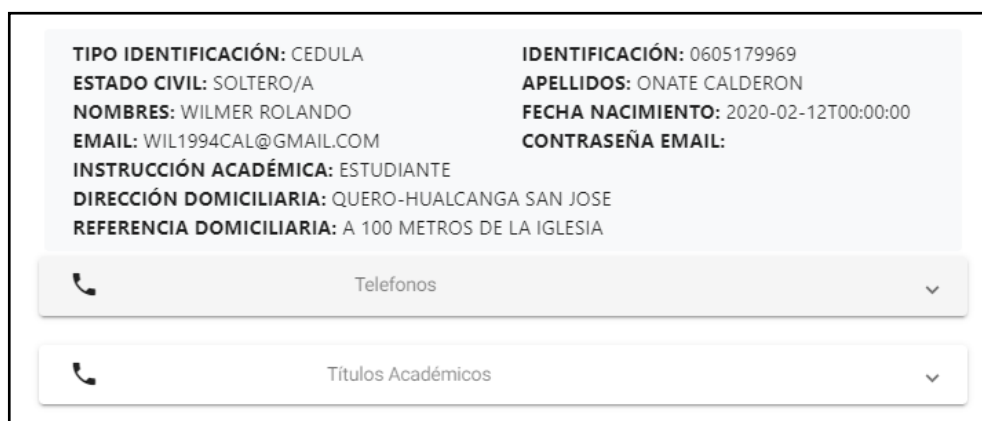
2. Agregar: Al presionar el botón de agregar se mostrará una ventana emergente donde el administrados seleccionara la persona que desea convertir en profesor, se realizará las validaciones correspondientes para que no se dupliquen datos, posteriormente se activará el botón Aceptar y el usuario deberá dar clic en la opción de guardar y confirmar que desea guardar un profesor.



Al realizar el proceso antes mencionado la Aplicación web progresiva creara automáticamente el usuario con el rol profesor. Se visualizará una ventana emergente donde se muestra la información con la que se creará el respectivo usuario, el administrador deberá confirmar que desea crear el respectivo usuario.



3. Ver Información: El botón “Ver Información” mostrara a una ventana emergente en al que se podrá visualizar un resumen de la información que posee el profesor.



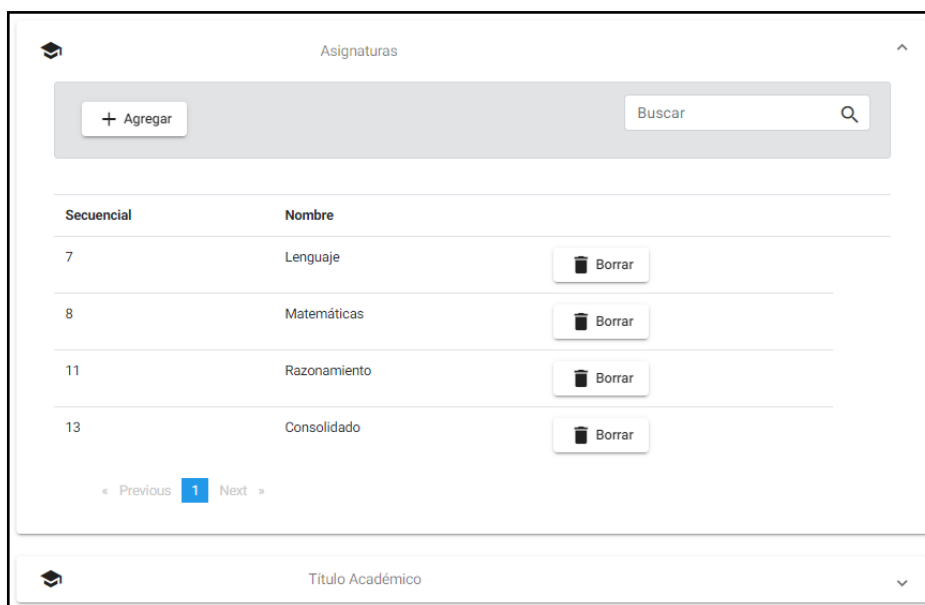
The screenshot shows a user information card with the following details:

TIPO IDENTIFICACIÓN: CEDULA	IDENTIFICACIÓN: 0605179969
ESTADO CIVIL: SOLTERO/A	APELLIDOS: ONATE CALDERON
NOMBRES: WILMER ROLANDO	FECHA NACIMIENTO: 2020-02-12T00:00:00
EMAIL: WIL1994CAL@GMAIL.COM	CONTRASEÑA EMAIL:
INSTRUCCIÓN ACADÉMICA: ESTUDIANTE	
DIRECCIÓN DOMICILIARIA: QUERO-HUALCANGA SAN JOSE	
REFERENCIA DOMICILIARIA: A 100 METROS DE LA IGLESIA	

Below the information card, there are two expandable sections:

- Telefonos** (with a phone icon and a dropdown arrow)
- Títulos Académicos** (with a phone icon and a dropdown arrow)

4. Editar: El botón editar redireccionará a una ventana en al que se visualizara la información principal del profesor como es identificación y nombres completos, esta opción podremos ingresar las asignaturas y títulos académicos que posee el profesor. Las diferentes acciones de listar, agregar, editar y eliminar se realizan en la misma estructura antes mencionada en la opción de configuración > Asignatura.

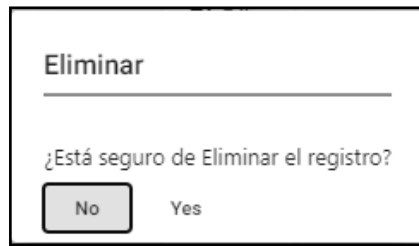


The screenshot shows the 'Asignaturas' (Subjects) management interface. At the top, there is a header with a graduation cap icon, the title 'Asignaturas', and a search bar with the text 'Buscar' and a magnifying glass icon. Below the header, there is a '+ Agregar' button. The main content area is a table with the following columns: 'Secuencial', 'Nombre', and 'Borrar'.

Secuencial	Nombre	Borrar
7	Lenguaje	Borrar
8	Matemáticas	Borrar
11	Razonamiento	Borrar
13	Consolidado	Borrar

At the bottom of the table, there is a pagination control: '< Previous 1 Next >'. Below the table, there is a footer with a graduation cap icon and the text 'Titulo Académico' with a dropdown arrow.

5. Eliminar: Al dar clic en el botón eliminar se desplegará un mensaje de confirmación para eliminar el registro, el administrador confirmará el mensaje y se procederá a eliminar el profesor.

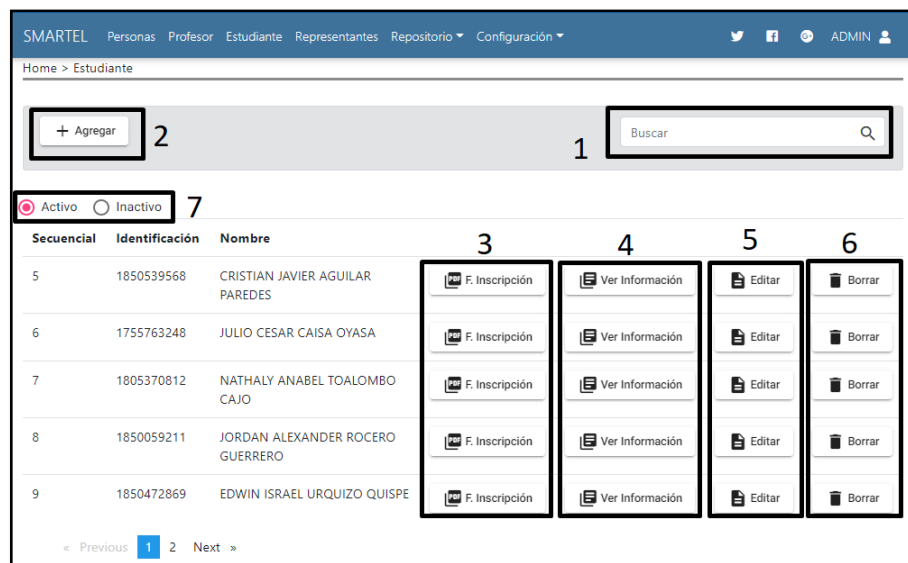


6. Estado Activo o Inactivo: Listara las personas activas e inactivas. Al seleccionar la opción inactiva visualizara las personas inactivas las misma podrán ser activadas dando clic en el botón activar del respectivo profesor.



2.6. Estudiantes

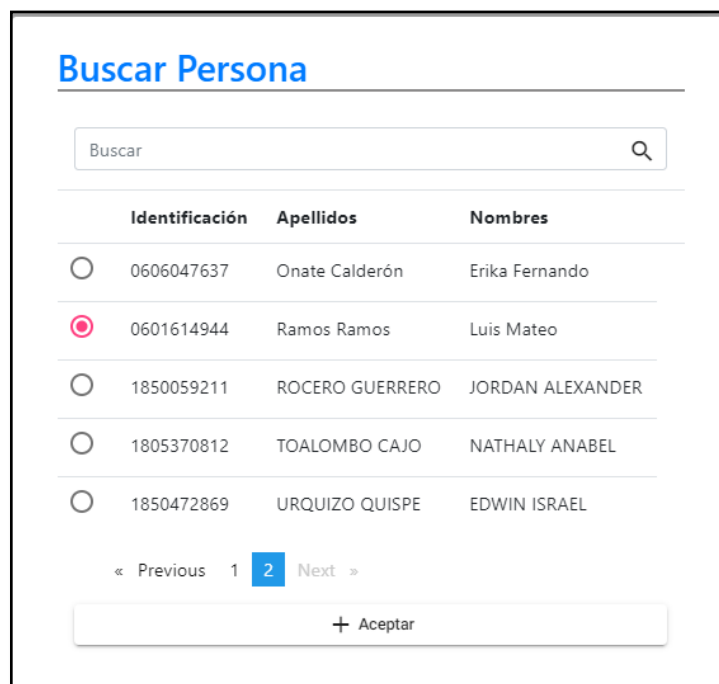
La pantalla de estudiantes posee las siguientes acciones que se describen a continuación:



1. Buscar: Permite realizar una búsqueda de las diferentes estudiantes existentes filtradas por el estado seleccionado ya sea activo o inactivo. Para poder realizar la búsqueda se deberá ingresar mínimo 3 caracteres y no distingue de mayúsculas o minúsculas.



2. Agregar: Al presionar el botón de agregar se mostrará una ventana emergente donde el administrados seleccionara la persona que desea convertir en estudiante, se realizará las validaciones correspondientes para que no se dupliquen datos, posteriormente se activará el botón Aceptar y el usuario deberá dar clic en la opción de guardar y confirmar que desea guardar el estudiante.



Al realizar el proceso antes mencionado la Aplicación web progresiva creara automáticamente el usuario con el rol estudiante. Se visualizará una ventana emergente donde se muestra la información con la que se creará el respectivo usuario, el administrador deberá confirmar que desea crear el respectivo usuario.

Crear Usuario

¿Esta seguro de crear el Usuario con la siguiente información?

Rol: Estudiante
Nombre Usuario: RamosE4944
Clave: [REDACTED]
email: ramos@gmail.com
Teléfono: S/N

3. Ficha de Inscripción: El botón “Ficha de Inscripción” generara un reporte de inscripción del estudiante. Se mostrará una venta donde se ingresará el valor y el costo mensual por cual se realizó la inscripción y daremos clic en generar el respectivo reporte.

Reporte Ficha de Inscripción

Costo del Curso

Valor Mensual

SMARTEL
Centro de Capacitación y Nivelación Académica

1 de Agosto del 2020

FICHA DE INSCRIPCIÓN

DATOS DEL ALUMNO:		DATOS DEL REPRESENTANTE:	
Nombres	CRISTIAN JAVIER	Nombres	Luis Mateo
Apellidos	AGUILAR PAREDES	Apellidos	Ramos Ramos
Cédula	1850539568	Cédula	0601614944
Dirección	Hualcanga Chico Zona 1	Dirección	Ambato
Teléfono	0991172256	Teléfono	Ninguno
Correo	aguilarchristian105@gmail.com	Correo	ramos@gmail.com
Costo	\$ 300		

Obligaciones

El estudiante deberá asistir a clases de acuerdo al horario establecido en esta ficha.

El estudiante deberá repasar fuera del horario de clases lo aprendido en clases, con el fin de afianzar sus conocimientos.

El representante legal deberá cancelar la mensualidad de \$80 dentro de los primeros quince días de cada mes.

Smartel, está obligado a impartir clases de calidad en las asignaturas estipuladas en el horario adjunto, a fin de que el estudiante esté preparado para rendir las diferentes evaluaciones.

Para constancia de la inscripción firma el representante legal.

.....
Ramos Ramos Luis Mateo
CI:0601614944

Contactos: 0999099219 smartel smartel_ec

PORQUE CREENOS EN TUS SUEÑOS, TRABAJAMOS POR TU ÉXITO

4. Ver Información: El botón “Ver Información” mostrara a una ventana emergente en la que se podrá visualizar un resumen de la información que posee el profesor.

TIPO IDENTIFICACIÓN: CEDULA	IDENTIFICACIÓN: 1850539568
ESTADO CIVIL: SOLTERO/A	APellidos: AGUILAR PAREDES
NOMBRES: CRISTIAN JAVIER	FECHA NACIMIENTO: 1999-03-17T00:00:00
EMAIL: AGUILARCHRISTIAN105@GMAIL.COM	CONTRASEÑA EMAIL: 333
CONTRASEÑA SER BACHILLER: SDSDW	INSTRUCCIÓN ACADÉMICA: BACHILLER
DIRECCIÓN DOMICILIARIA: HUALCANGA CHICO ZONA 1	
REFERENCIA DOMICILIARIA: HUALCANGA CHICO ZONA 1	
Telefonos	▼
Instituciones	▼
Representante	▼
Asignaturas	▼

5. Editar: El botón editar redireccionará a una ventana en al que se visualizara la información principal del estudiante como es identificación y nombres completos, esta opción podremos ingresar las asignaturas representantes y títulos académicos que posee el estudiante. Las diferentes acciones de listar, agregar, editar y eliminar se realizan en la misma estructura antes mencionada en la opción de configuración > Asignatura.

Home > Estudiante > Editar

Secuencial: 5 Identificación: 1850539568 Nombre: AGUILAR PAREDES CRISTIAN J/

Clave Ser Bachiller: sdsdw

Representantes

Instituciones

Asignaturas

6. Eliminar: Al dar clic en el botón eliminar se desplegará un mensaje de confirmación para eliminar el registro, el administrador confirmará el mensaje y se procederá a eliminar el estudiante.

Eliminar

¿Está seguro de Eliminar el registro?

7. Estado Activo o Inactivo: Listara las personas activas e inactivas. Al seleccionar la opción inactiva visualizara las personas inactivas las misma podrán ser activadas dando clic en el botón activar del respectivo estudiante.

Activo Inactivo

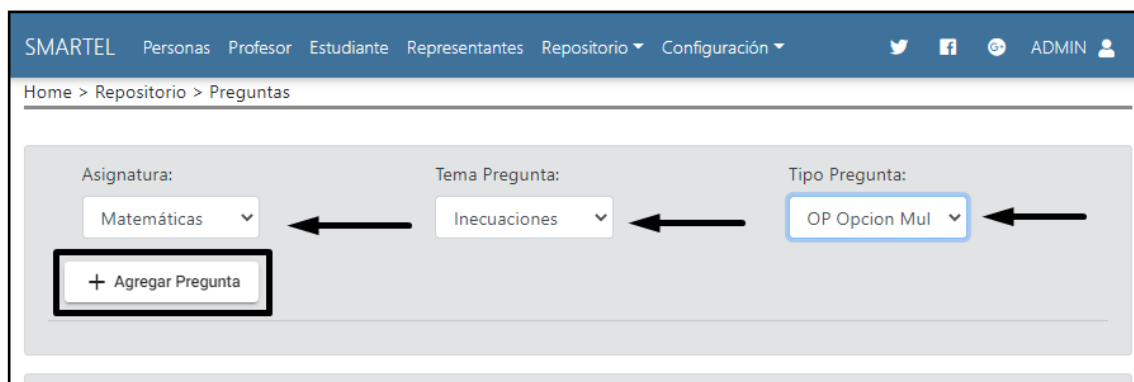
Secuencial	Identificación	Nombre	
8	1805086038	GAVILANES FREIRE ANGELY MAITE	<input type="button" value="Ver Información"/> <input type="button" value="Activar"/>

< Previous 1 Next >

2.2.7. Repositorio

Agregar Pregunta

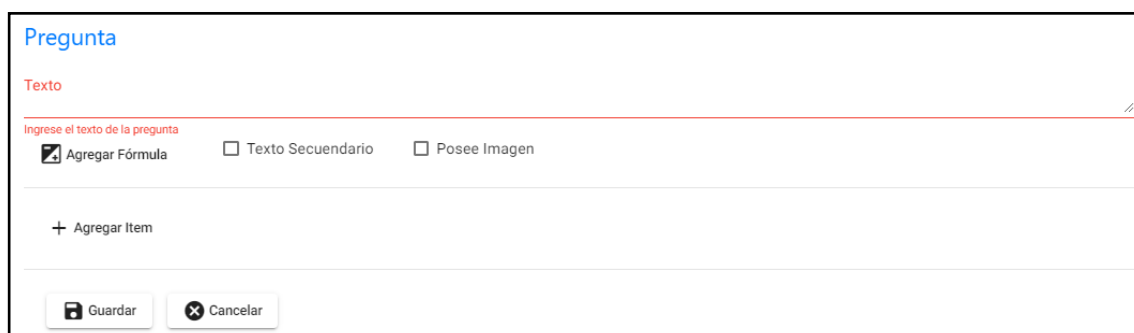
Para agregar preguntas el administrador o profesor deberá seleccionar la asignatura, tema de pregunta y tipo de pregunta, posteriormente el usuario deberá dar clic en el botón Agregar Pregunta.



The screenshot shows the SMARTEL interface with the following elements:

- Navigation bar: SMARTEL, Personas, Profesor, Estudiante, Representantes, Repositorio, Configuración, ADMIN.
- Breadcrumbs: Home > Repositorio > Preguntas
- Form fields:
 - Asignatura: Matemáticas (dropdown)
 - Tema Pregunta: Inecuaciones (dropdown)
 - Tipo Pregunta: OP Opcion Mul (dropdown)
- Buttons: + Agregar Pregunta (highlighted with a red box)

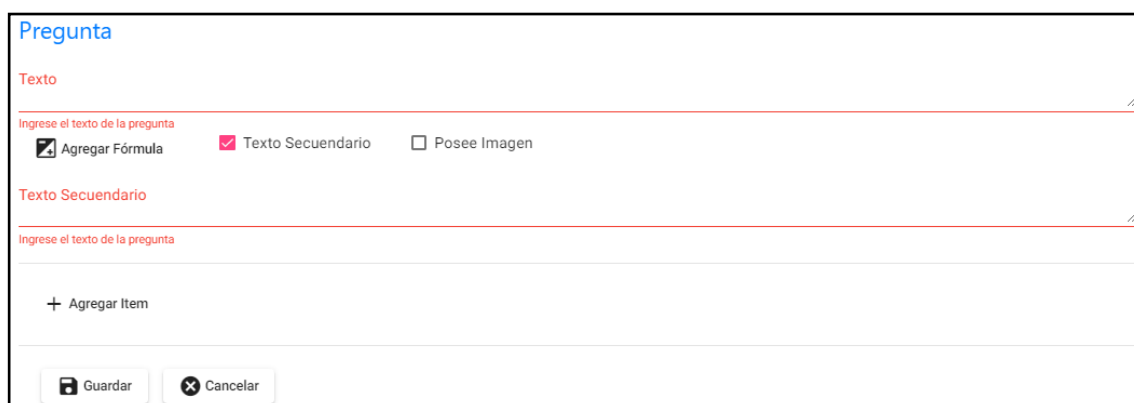
Posteriormente se visualizará una ventana emergente donde se deberá ingresar el texto de la pregunta y los respectivos ítems.



The 'Pregunta' modal form contains the following elements:

- Section: Pregunta
- Section: Texto
- Text input field: Ingrese el texto de la pregunta (with a red error message)
- Form elements:
 - Agregar Fórmula
 - Texto Secundario
 - Posee Imagen
- Button: + Agregar Item
- Buttons: Guardar, Cancelar

Si el texto posee un texto secundario se seleccionará la opción de “Texto Secundario” y se visualizará un segundo cuadro de texto para ingresar el respectivo texto.



The 'Pregunta' modal form is updated with the following elements:

- Section: Pregunta
- Section: Texto
- Text input field: Ingrese el texto de la pregunta (with a red error message)
- Form elements:
 - Agregar Fórmula
 - Texto Secundario
 - Posee Imagen
- Section: Texto Secundario
- Text input field: Ingrese el texto de la pregunta (with a red error message)
- Button: + Agregar Item
- Buttons: Guardar, Cancelar

Si el texto de la pregunta posee una fórmula matemática se seleccionará la opción de “Agregar Formula” se desplegará una segunda ventana emergente donde el usuario ingresará la fórmula matemática en formato Latex. El usuario deberá seleccionar si desea que la formula se presente en un nuevo párrafo o en el mismo.

El usuario dará clic sobre el botón “Agregar Item” se agregará una sección donde el usuario ingresará un ítem de respuesta.

El usuario ingresara los ítems que considere necesarios, posteriormente seleccionara un ítem ingresado y se activaran dos botones: para seleccionar el ítem como respuesta o eliminar el ítem ingresado.

Al seleccionar el ítem correspondiente como respuesta de la pregunta, el ítem seleccionado se resaltará de un color diferente(verde). Posteriormente el usuario deberá dar clic el botón “Guardar” para ingresar la pregunta al repositorio.

Item 2 Posee Imagen

Agregar Fórmula

Item 6 Posee Imagen

Agregar Fórmula

Item 10 Posee Imagen

Agregar Fórmula

Item 22 Posee Imagen

Agregar Fórmula

Después de ingresar la respectiva pregunta se visualizará de la siguiente manera.

Matemáticas - Inecuaciones

Una micro empresa dedicada a la reparación y mantenimiento de equipos de refrigeración industrial ofrece sus servicios a empresas mediante dos tarifas que se cobran en dólares. La primera tarifa, que corresponde a días laborables, se representa como $L = 2(x + 8)$, mientras que la tarifa en la que se consideran los fines de semana es $F = 3(x + 2)$, donde x representa el número de horas de cada servicio. Si una empresa desea realizar un mantenimiento preventivo a todos sus equipos, determine el número de horas que puede costear, de manera que el valor que pagará en los días laborables sea equivalente al costo que pagará durante los fines de semana.

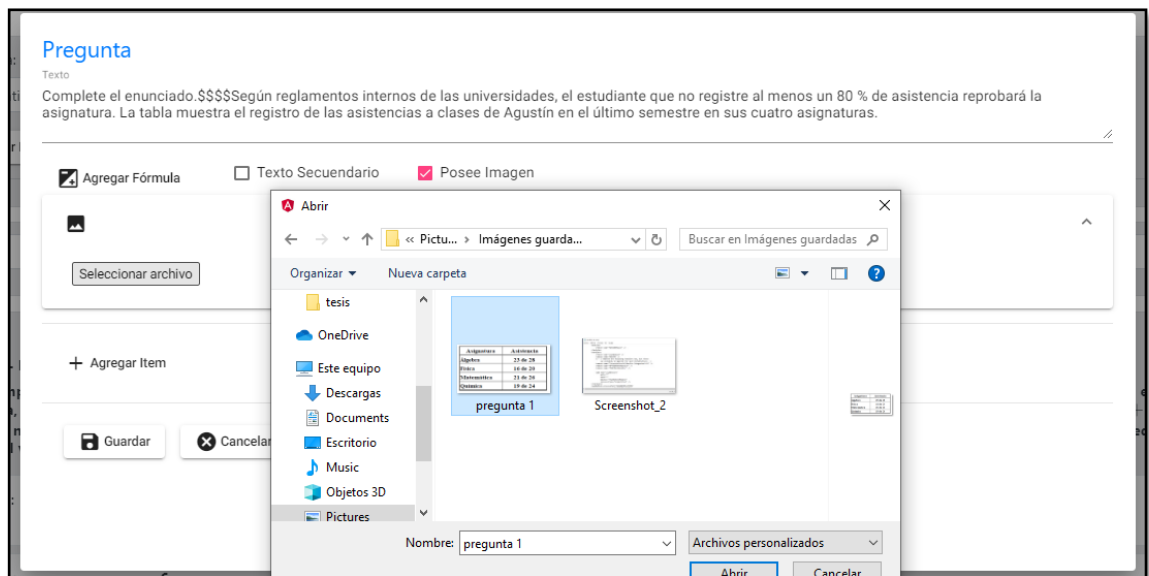
Opción 1 : **2**

Opción 2 : **6**

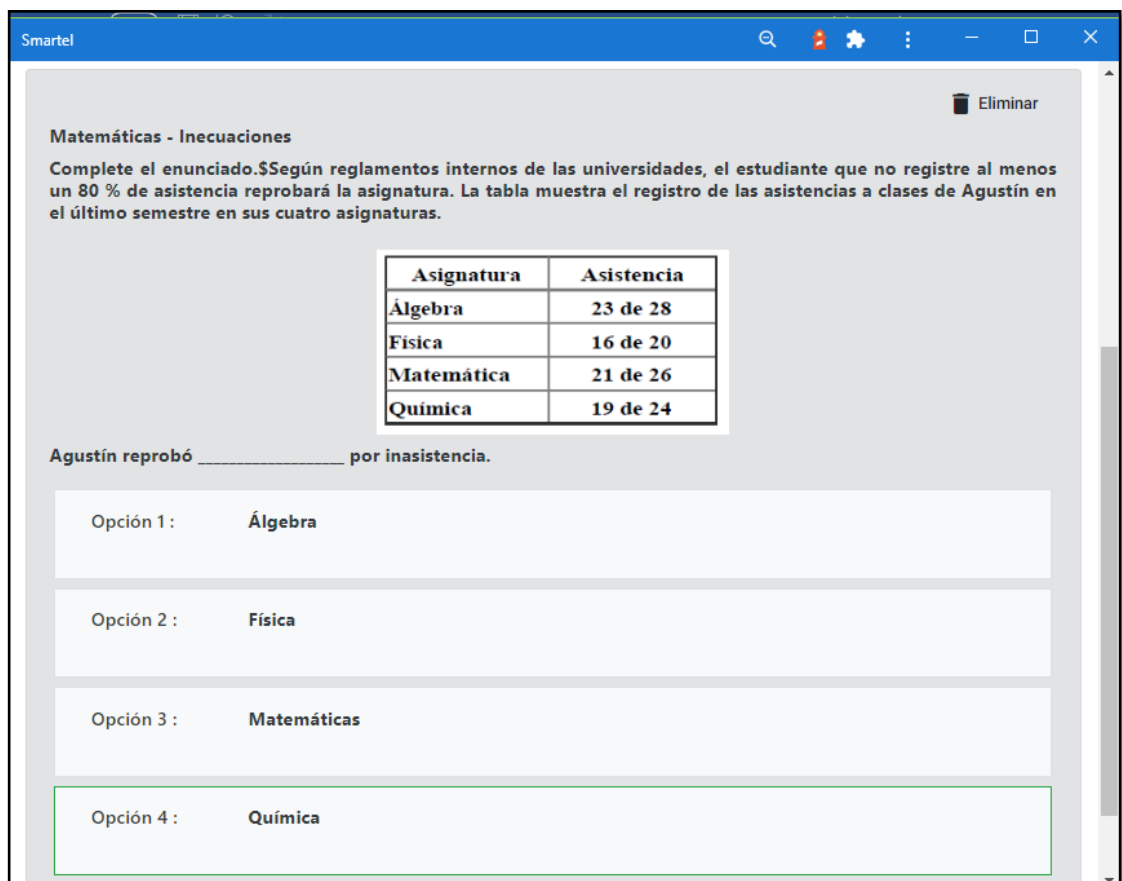
Opción 3 : 10

Opción 4 : **22**

Para el ingreso de imágenes en las respectivas preguntas se deberá seleccionar la opción “Posee Imagen”, se visualizará una desplegable en el formulario donde seleccionará la imagen que sea ingresar. Hay que tener en cuenta que el usuario podrá ingresar mas de una imagen de ser necesario.



Después de realizar el proceso respectivo para ingresar la pregunta se visualizará de la siguiente manera.



Eliminar Pregunta

Para eliminar una pregunta se lo podrá realizar desde Repositorio > Buscar dando clic sobre el botón eliminar de la respectiva pregunta, el usuario deberá confirmar que sea borrar la respectiva pregunta seleccionada.

Asignatura: Matemátic. Tema Pregunta: Inecuacion. Filtrar Por: Activos

Buscar Pregunta Filtrar

Eliminar

Matemáticas - Inecuaciones

Complete el enunciado. Según reglamentos internos de las universidades, el estudiante que no registre al menos un 80 % de asistencia reprobó la asignatura. La tabla muestra el registro de las asistencias a clases de Agustín en el último semestre en sus cuatro asignaturas.

Asignatura	Asistencia
Álgebra	23 de 28
Física	16 de 20
Matemática	21 de 26
Química	19 de 24

Agustín reprobó _____ por inasistencia.

Activar Pregunta

Para activar una pregunta el usuario deberá dirigirse a Repositorio > Buscar y deberá filtrar las preguntas por inactivas, posteriormente el usuario deberá dar clic en el botón “Activar” y confirmar que desea activar la pregunta seleccionada.

Asignatura: Matemátic. Tema Pregunta: Inecuacion. Filtrar Por: Activos

Buscar Pregunta Filtrar

+ Activar

Matemáticas - Inecuaciones

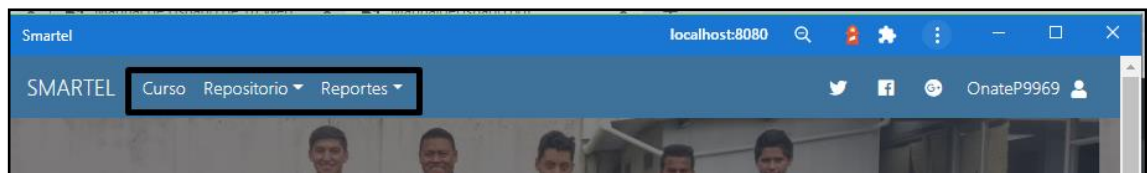
Complete el enunciado. Según reglamentos internos de las universidades, el estudiante que no registre al menos un 80 % de asistencia reprobó la asignatura. La tabla muestra el registro de las asistencias a clases de Agustín en el último semestre en sus cuatro asignaturas.

Asignatura	Asistencia
Álgebra	23 de 28
Física	16 de 20
Matemática	21 de 26
Química	19 de 24

Agustín reprobó _____ por inasistencia.

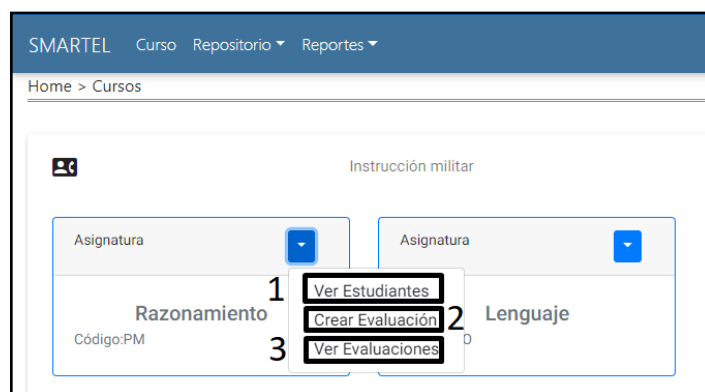
3. ROL PROFESORES

Al ingresar al sistema con el rol de “Profesor” se mostrará las siguientes opciones en el menú horizontal: Curso, Repositorio y Reportes.



3.1. Curso

Al dar clic en la opción curso de menú se mostrará todos los módulos y cursos que posee el profesor. Las acciones que podrá realizar el Profesor en esta interfaz se describen a continuación.

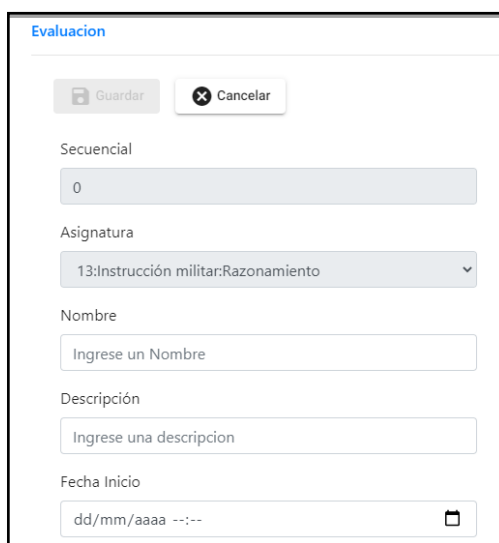


1. Ver estudiantes: El usuario al dar clic en el botón “Ver Estudiantes” se desplegará una ventana emergente que mostrara los estudiantes que están registrados en la respectiva asignatura.



2. Crear Evaluaciones: El usuario al dar clic en el botón “Crear Evaluación” se desplegará una ventana emergente donde el usuario ingresa el nombre, descripción,

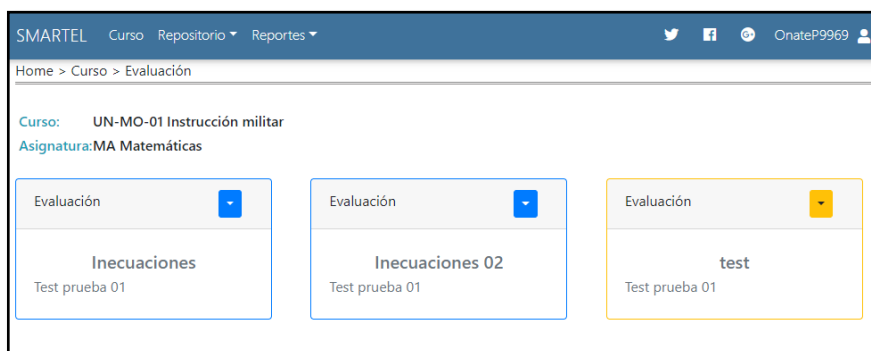
fecha inicio, fecha fin, si se puede ver las respuestas y si se puede ver la calificación por parte del estudiante



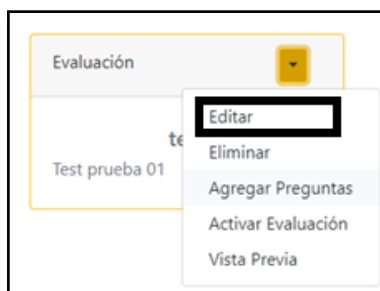
The screenshot shows a form titled "Evaluación" with the following fields and controls:

- Buttons: "Guardar" (Save) and "Cancelar" (Cancel).
- Field: "Secuencial" with a value of "0".
- Field: "Asignatura" with a dropdown menu showing "13:Instrucción militar:Razonamiento".
- Field: "Nombre" with the placeholder text "Ingrese un Nombre".
- Field: "Descripción" with the placeholder text "Ingrese una descripcion".
- Field: "Fecha Inicio" with a date picker showing "dd/mm/aaaa --:--".

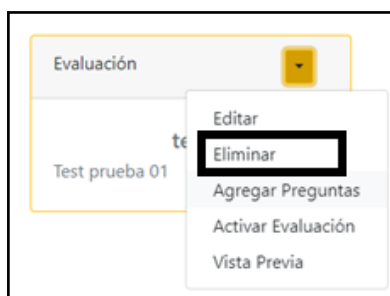
3. Ver Evaluaciones: Al dar clic en el botón “Ver Evaluaciones” la aplicación nos redireccionará a una pantalla donde se mostrará las evaluaciones creadas activas e inactivas. La evaluación activas e inactivas se reconocerán por color de borde: amarillo inactivas y azul activas.



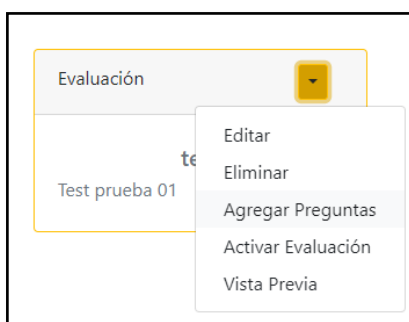
Editar Evaluación: Al dar clic en el botón “Editar” se visualizará una ventana emergente donde el usuario podrá modificar la información de la evaluación.



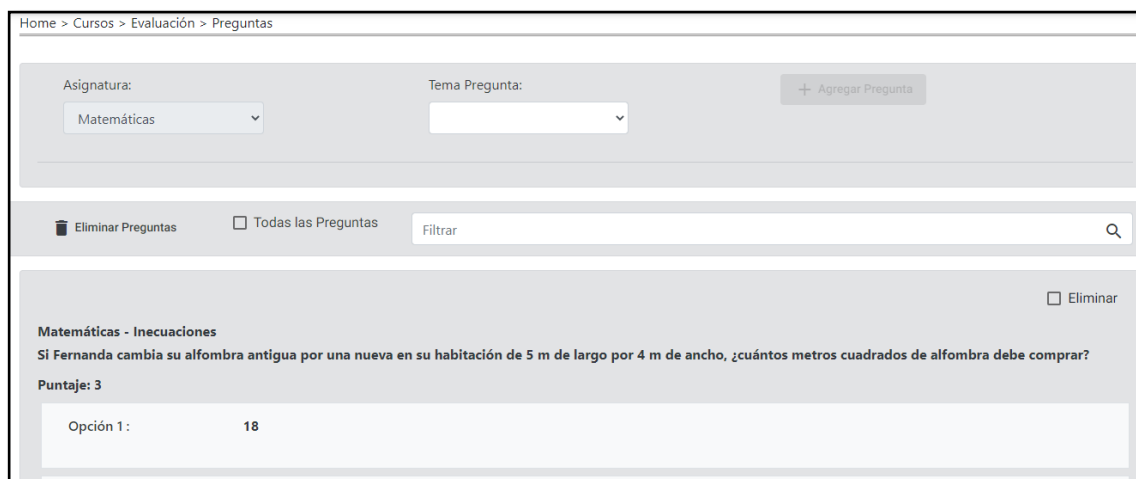
Eliminar Evaluación: Al dar clic en el botón “Eliminar” se eliminará la evaluación, si la evaluación esta activa se controlará que no haya sido realizada por ningún estudiante antes de poder eliminarla.



Agregar preguntas a una evaluación: Esta opción estará activa únicamente en las evaluaciones inactivas.

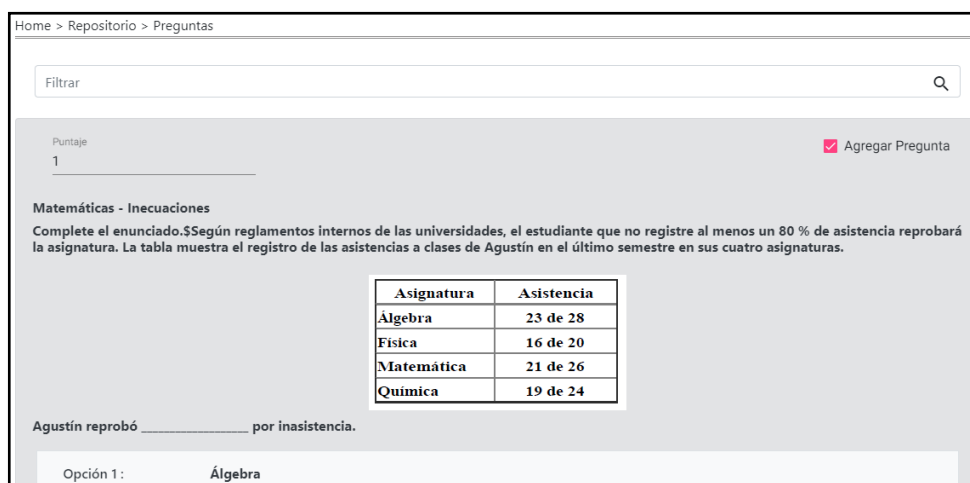


Al dar clic en el botón “Agregar Preguntas” se redireccionará a una pantalla donde el usuario podrá ingresar las preguntas desde el repositorio. Además, se muestra las preguntas ya agregadas al a la evaluación.



Se seleccionará el tema de la pregunta y se dará clic en el botón “Agregar Pregunta” y se visualizará las preguntas no ingresadas según el tema seleccionado, se ingresará el puntaje que la pregunta tendrá en la evaluación y se seleccionar la

pregunta. Se podrá seleccionar varias preguntas para poder ingresarlas. Después de realizar todo el proceso el usuario deberá dar clic en el botón “Aceptar”.

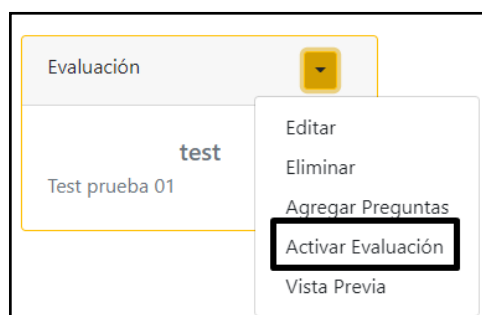


Eliminar Preguntas de una evaluación: El usuario después de dar clic en “Ver Evaluaciones” podrá eliminar una pregunta de una evaluación, el usuario deberá seleccionar la opción “Eliminar” de las preguntas a eliminar y dar clic en el botón “Eliminar preguntas” y confirmar que desea eliminar las preguntas. Las preguntas se eliminarán permanentemente de la evaluación.



Activar Evaluación

Al dar clic y “Activar Evaluación” y confirmar que desea activar la evaluación, se cambiará el estado de la evaluación de activa e inactiva y el estudiante podrá visualizar la evaluación.



Vista Previa: Al dar clic en el botón “Vista Previa” se mostrar al profesor como el estudiante visualizara la evaluación. Hay que tener en cuenta el profesor observara las preguntas en el orden que las ingreso y con la respuesta correcta seleccionada a diferencia del estudiante.

Matemáticas - Inecuaciones
Si Fernanda cambia su alfombra antigua por una nueva en su habitación de 5 m de largo por 4 m de ancho, ¿cuántos metros cuadrados de alfombra debe comprar?

Puntaje: 3

Opción 1 :	18
Opción 2 :	20
Opción 3 :	41
Opción 4 :	81

Justificación Respuesta:
Porque sí:))

< Previous **1** 2 Next >

Evaluación por estudiante: Se redireccionará a un formulario donde se mostrará los estudiantes que rindieron la evaluación. Se mostrará la calificación, la respuesta correcta y la respuesta seleccionada por el estudiante al no.

Estudiante: 1850059211 - ROCERO GUERRERO JORDAN ALEXANDER
Fecha Inicio: 2020-04-19 11:09 **Fecha Fin:** 2020-04-19 11:13 **Calificación:** 2/3

Pregunta:1
Después de un tiro libre en un partido de fútbol, la pelota sale de la cancha y cae por una pendiente. En el primer segundo recorre 5 m, en el segundo recorre 10 m, en el tercer segundo recorre 15 m y así sucesivamente. ¿Cuántos metros recorrerá la pelota al quinto segundo?

(Valoración : 1 / 1)

Opción 1 :	13
Opción 2 :	25
Opción 3 :	30
Opción 4 :	35

< Previous **1** 2 3 Next >

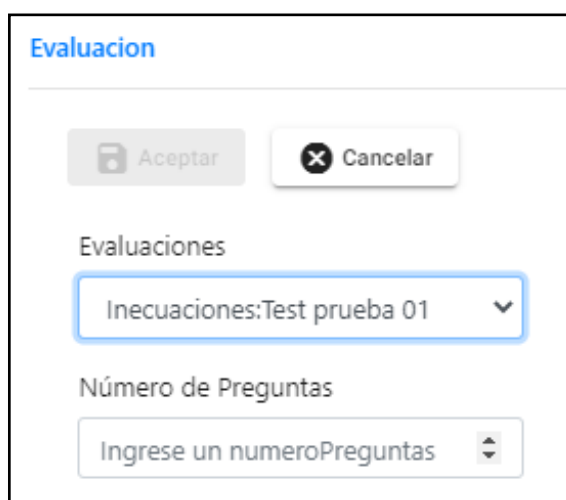
Evaluación Consolidado: La asignatura evaluación consolidado tiene una diferencia de las demás asignaturas que es la forma en que agregar las preguntas a una evaluación la cual se describe a continuación:

Las preguntas de una evaluación consolidada se generarán aleatoriamente de las evaluaciones de las demás asignaturas creadas.



El formulario muestra los campos de configuración para una evaluación consolidada. En la parte superior hay botones para 'Guardar' y 'Cancelar'. Los campos incluyen: 'Secuencial' (input con el valor 0), 'Asignatura' (lista desplegable con '19:Instrucción militar:Consolidad...', 'Nombre' (input con el texto 'Ingrese un Nombre'), 'Descripción' (input con el texto 'Ingrese una descripción'), 'Fecha Inicio' (input con el formato 'dd/mm/aaaa --:--' y un ícono de calendario), 'Fecha Fin' (input con el formato 'dd/mm/aaaa --:--' y un ícono de calendario), 'Duración' (input con el formato '--:--' y un ícono de reloj), 'Ver Respuestas' (checkbox desactivado) y 'Ver Calificación' (checkbox desactivado). Debajo de los campos hay un botón azul '+ Agregar Evaluaciones'. En la parte inferior hay una tabla con columnas 'Código' y 'Nombre', y botones '< Previous' y 'Next >'.

Al dar clic en el botón “Agregar Evaluación” se desplegará una segunda ventana emergente en la cual el usuario seleccionará la evaluación y el número de preguntas a seleccionar, la aplicación web progresiva seleccionará aleatoriamente el número de preguntas ingresadas por el usuario.



La ventana emergente, titulada 'Evaluacion', contiene los botones 'Aceptar' y 'Cancelar'. Incluye un campo 'Evaluaciones' con una lista desplegable que muestra 'Inecuaciones:Test prueba 01' y un campo 'Número de Preguntas' con un input que contiene el texto 'Ingrese un numeroPreguntas'.

Al agregar las preguntas por evaluación se activarán dos botones “Ver Preguntas” y “Eliminar”.

Secuencial 0	Asignatura 19:Instrucción militar:Consolidad	Nombre Ingrese un Nombre
Descripción Ingrese una descripción	Fecha Inicio dd/mm/aaaa ---	Fecha Fin dd/mm/aaaa ---
Duración ---	Ver Respuestas <input type="checkbox"/>	Ver Calificación <input type="checkbox"/>

[+ Agregar Evaluaciones](#)

Código	Nombre	Ver Preguntas	Eliminar
Inecuaciones	Test prueba 01	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

< Previous **1** Next >

Ver preguntas: Ver preguntas visualizara una ventana segunda ventana emergente que mostrara las preguntas seleccionadas aleatoriamente por la aplicación web progresiva.

Matemáticas - Inecuaciones

Si Fernanda cambia su alfombra antigua por una nueva en su habitación de 5 m de largo por 4 m de ancho, ¿cuántos metros cuadrados de alfombra debe comprar?

Opción 1 :	18
Opción 2 :	20
Opción 3 :	41
Opción 4 :	81

Justificación Respuesta:
Porque si:))

< Previous **1** **2** Next >

Eliminar: Al dar clic en el botón “Eliminar” se eliminará las evaluación y preguntas agregadas de la evaluación consolidado.

3.2. Reportes

Los diferentes reportes generados se presentarán en formato PDF. La opción de reportes para el rol profesor tiene las siguientes opciones:

Reportes ▾

- Evaluación por Asignatura
- Evaluación Consolidado
- Horario de Clases

Evaluación por asignatura: Presenta una venta emergente donde se seleccionará el curso, asignatura y la evaluación que desea generar el reporte.


Reporte Evaluaciones por Asignatura

Modalidad:

Curso:

Asignatura- Profesor:

Evaluaciones:


SMARTEL
 Centro de Capacitación y Nivelación Académica

REPORTE DE EVALUACIÓN POR ASIGNATURA

Curso: Instrucción militar **Asignatura:** Matemáticas
Profesor: Onate Calderon Wilmer Rolando **Evaluación:** Inecuaciones 02
Puntaje: 2

Identificación	Nombre Estudiante	Calificación	Correctas	Incorrectas	No Contestadas
1850059211	ROCERO GUERRERO JORDAN ALEXANDER	1	1	1	0
1805086038	GAVILANES FREIRE ANGELY MAITE	1	1	1	0
	PROMEDIO	1	1	1	0

Centro de Capacitación y Nivelación Académica

Evaluación Consolidado: Presenta una venta emergente donde se seleccionará el curso, asignatura y la evaluación consolidadas que desea generar el reporte.


Reporte Evaluaciones por Asignatura

Modalidad:

Curso:

Asignatura- Profesor:

Evaluaciones:



REPORTE DE EVALUACIÓN CONSOLIDADO

Curso: Instrucción militar **Asignatura:** Consolidado
Profesor: Onate Calderon Wilmer Rolando **Evaluación:** Consolidado
Puntaje: 2

Identificación	Nombre Estudiante	Calificación	Correctas	Incorrectas	No Contestadas
1850059211	ROCERO GUERRERO JORDAN ALEXANDER	0	0	2	0
1805370812	TOALOMBO CAJO NATHALY ANABEL	0	0	2	0
1755763248	CAISA OYASA JULIO CESAR	0	0	2	0
1805086038	GAVILANES FREIRE ANGELY MAITE	2	2	0	0
	PROMEDIO	0.5	0.5	1.5	0

Horario Clases: Al seleccionar a la opción de horario de clases se generará un reporte en el cual se indica el día, hora, asignatura que el profeso dicta clases.



HORARIO DE CLASES

Identificación: 0605179969
Profesor: Onate Calderon Wilmer Rolando

	08:00 - 09:00	09:00 - 10:00	10:00 - 11:00	15:00 - 16:00	16:00 - 17:00	17:00 - 18:00
Lunes			Matemáticas	Lenguaje	Lenguaje	
Martes	Matemáticas	Matemáticas		Razonamiento		
Miercoles			Razonamiento		Lenguaje	
Viernes						Matemáticas

4. Estudiantes

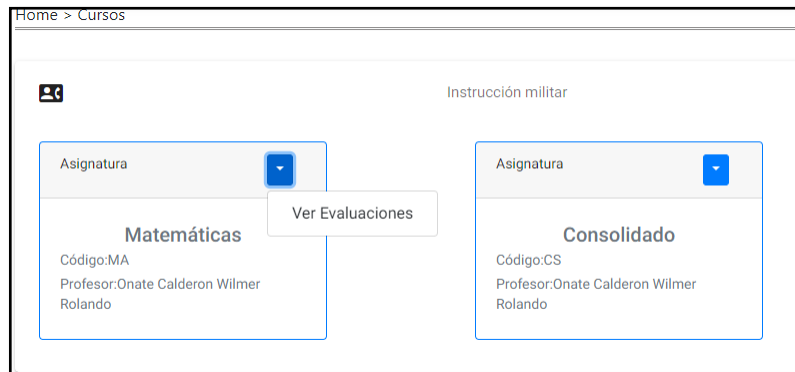
Al ingresar al sistema con el rol de “Estudiante” se mostrará las siguientes opciones en el menú horizontal: Curso y Reportes



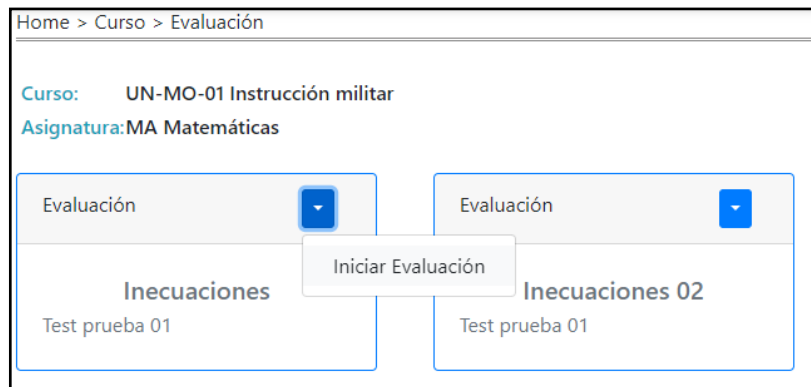
4.1. Cursos:

El usuario al seleccionar la opción de cursos visualizara una pantalla con los diferentes cursos y asignaturas que el estudiante este asignado. Cada asignatura

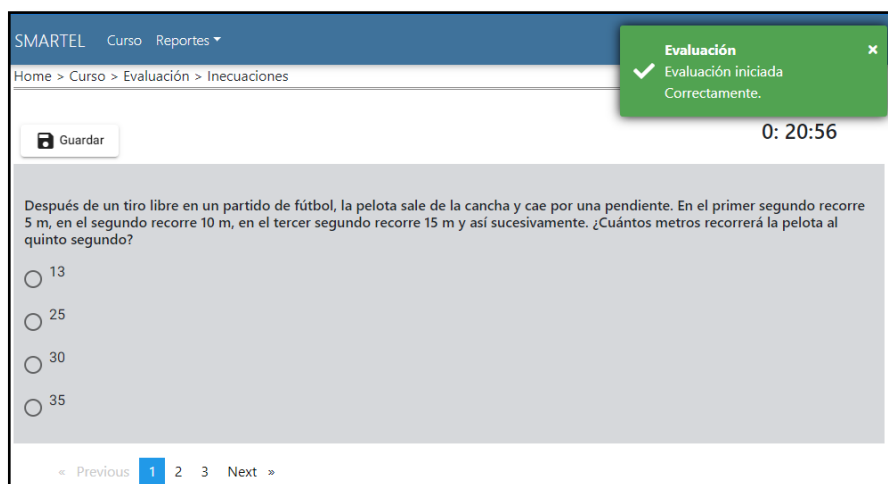
tendrá la opción “Ver Evaluaciones” donde se mostrará las evaluaciones activas que posee la asignatura.



Cada asignatura presentará la opción de “Iniciar Evaluación”, el estudiante al dar clic en el botón ”Iniciar Evaluación” y deberá confirmar que desea iniciar la evaluación.



Una vez iniciada la evaluación, se presentará una pantalla en la que el estudiante podrá contestar cada una de las preguntas. Al finalizar de contestar las preguntas el estudiante deberá dar clic en el botón “Guardar.”



Revisar Evaluación: El estudiante al dar clic en el botón “Revisar Evaluación” se le mostrará una pantalla donde podrá revisar el puntaje y las respuestas correctas e incorrectas de la evaluación. Esta opción estará habilitada dependiendo de la configuración de la evaluación realizada por el profesor.

Fecha Inicio: 2020-08-02 13:44 Fecha Fin: 2020-08-02 13:45 Calificación: 1/3

Pregunta 1:
Después de un tiro libre en un partido de fútbol, la pelota sale de la cancha y cae por una pendiente. En el primer segundo recorre 5 m, en el segundo recorre 10 m, en el tercer segundo recorre 15 m y así sucesivamente. ¿Cuántos metros recorrerá la pelota al quinto segundo?

appBlockCopyPaste (Valoración : 0 / 1)

Opción 1 : 13

Opción 2 : 25


Opción 3 : 30

Opción 4 : 35

« Previous **1** 2 3 Next »

4.2. Reportes

Horario de clases: Al seleccionar la opción de horario de clases se generará un reporte en el cual se indica el día, hora, asignatura que el estudiante recibirá clases.

 **SMARTEL**
Centro de Capacitación y Nivelación Académica

HORARIO DE CLASES

Identificación: 1805086038
Estudiante: GAVILANES FREIRE ANGELY MAITE

	08:00 - 09:00	09:00 - 10:00	10:00 - 11:00	15:00 - 16:00	16:00 - 17:00	17:00 - 18:00
Lunes			Matemáticas Profesor: Onate Calderon Wilmer Rolando			
Martes	Matemáticas Profesor: Onate Calderon Wilmer Rolando	Matemáticas Profesor: Onate Calderon Wilmer Rolando				
Viernes						Matemáticas Profesor: Onate Calderon Wilmer Rolando