



**UNIVERSIDAD ESTATAL
PENÍNSULA DE SANTA ELENA**

**FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES**

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

TRABAJO DE TITULACIÓN

Propuesta Tecnológica, previo a la obtención del Título de:

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

**“Aplicación web para la automatización y gestión de horarios de clases
mediante algoritmos basados en el modelo de programación lineal en
la escuela ‘Lic. Angélica Villón Lindao’”**

AUTOR:

BAQUERIZO GÓMEZ JOSÉ ANTONIO

PROFESOR TUTOR:

ING. IVÁN CORONEL SUÁREZ, MSIA.

LA LIBERTAD - ECUADOR

2021

AGRADECIMIENTO

Agradezco a Dios por permitirme cumplir una de mis metas y mantenerme de pie durante todo este proceso.

Con honores a mis padres y familiares, que estuvieron siempre presentes y apoyándome durante toda mi carrera. A mi abuela y mi hermano por todo su apoyo incondicional.

A mi equipo: Alex, Johnny, Andrés y Katherine que conformamos IRIS, que me motivaron y me presionaron a salir del común denominador, empujándome a aprender cosas nuevas y salir de mi zona de confort. Siempre buscando los resultados más destacados y enfrentar nuevos desafíos y proyectos.

A los buenos docentes, que lograron transmitir enseñanzas importantes durante este tiempo de estudio. También a aquellos que presionaron constantemente para obtener incluso mejores resultados.

Al ing. Jaime Orozco y Marjorie Coronel, que siempre estuvieron prestos a ayudarme dentro y fuera de las actividades académicas y profesionales.

A mi docente tutor, que resolvió mis dudas de manera oportuna para la culminación de este trabajo de titulación.

Y a todas las personas especiales que siempre estuvieron conmigo y me ayudaron a mantener la mejor versión de mi incluso en los peores momentos: Gabriela, Aline, Bryan Holly, y a muchos más.

APROBACIÓN DEL TUTOR

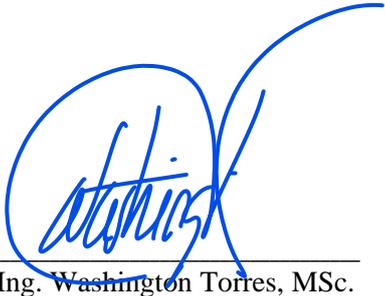
En mi calidad de Tutor/Tutora del trabajo de titulación denominado: **“Aplicación web para la automatización y gestión de horarios de clases mediante algoritmos basados en el modelo de programación lineal en la escuela Lic. Angélica Villón Lindao”**, elaborado por el estudiante **Baquerizo Gómez José Antonio**, de la carrera de **Tecnologías de la información** de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, lo apruebo en todas sus partes y autorizo al estudiante para que inicie los trámites legales correspondientes.

La libertad, agosto del 2021



Ing. Iván Coronel Suárez, MSIA.

TRIBUNAL DE GRADO



Ing. Washington Torres, MSc.
DIRECTOR(E) DE CARRERA



Lsi. Daniel Quirumbay Yagual, MSIA.
DOCENTE ESPECIALISTA



Ing. Iván Coronel Suárez, MSIA.
PROFESOR TUTOR



Ing. Marjorie Coronel Suárez, MGTI.
PROFESOR DE GUÍA

RESUMEN

La escuela de educación básica “Lic. Angélica Villón Lindao”, ubicada en el cantón Santa Elena, se encuentra dentro de un proceso de mejora continua para garantizar la excelencia académica y los servicios educativos que ofrecen. Sin embargo, una de sus mayores limitantes es la carencia de implementación tecnológica que poseen, de manera que, varias de sus actividades son realizadas de forma manual o con herramientas de ofimática. Esto conlleva a la aparición de dificultades en aquellos procesos que se pueden considerar como complejos. Uno de ellos es la planificación y estructuración de los horarios de clases, cuya elaboración puede representar un desafío a la hora de determinar aquellos factores a evaluar para obtener resultados exitosos. El proceso al ser totalmente dependiente del factor humano lo vuelve susceptible a errores que generen un retraso en las labores académicas dentro de las primeras semanas de clases. Por este motivo se propuso la implementación de una aplicación web que tiene como propósito automatizar y sistematizar el proceso de generación de horarios con los siguientes módulos: sesión, administración, gestión y reportes. La construcción de la aplicación web se basó en el modelo incremental de software basados en cada uno de sus módulos, además del planteamiento de un modelo de optimización mediante programación lineal entera con el objetivo de generar horarios de forma automática y reducir el tiempo y esfuerzo que este proceso conlleva. Se aplicó JavaScript en su totalidad mediante el uso del stack de tecnologías MERN donde se incluyen: Mongo DB como base de datos no relacional orientada a documentos, Express JS como marco de trabajo para desarrollar interfaces web, React como librería para el desarrollo de interfaces y Node como entorno de ejecución de JavaScript del lado del servidor. Una vez concluido el desarrollo de la aplicación, el modelo matemático, el servicio web y la implementación de la base de datos se logró demostrar la reducción de tiempo que los usuarios emplearían gracias a que los horarios pueden ser desarrollados de forma automática, respetando todas las restricciones necesarias y mitigando aquellos errores que antes de la aplicación pudieron generarse, aportando valor estratégico y la optimización de las operaciones realizadas en la institución.

ABSTRACT

The basic education school “Lic. Angélica Villón Lindao”, located in the Santa Elena canton, is within a process of continuous improvement to guarantee academic excellence and the educational services they offer. However, one of their biggest limitations is the lack of technological implementation they have, therefore several of their activities are developed manually or with office automation tools. This leads to the appearance of difficulties in processes that can be considered as complex. One of them is the planning and structuring of class schedules, the elaboration of which can represent a challenge when determining the factors to evaluate in order to obtain successful results. The process by being totally dependent on the human factor makes it susceptible to errors that generate a delay in academic work within the first weeks of classes. For this reason, the implementation of a web application was proposed that aims to automate and systematize the process of generating schedules with the following modules: session, administration, management, and reports. The construction of the web application was based on the incremental software architecture model based on each of their modules. In addition to the approach to an optimization model using integer linear programming with the aim of generating schedules automatically and reducing the time and effort that this process entails. JavaScript was applied in its entirety using the MERN stack technologies which includes Mongo DB as a non-relational document-based database, Express JS as a framework to develop web interfaces, React as a library for interface development and Node as server-side JavaScript runtime environment. Once the development of the application, the mathematical model, the web service, and the implementation of the database was completed, it was possible to demonstrate the reduction of time that users would use thanks to the fact that the schedules can be developed automatically, respecting all the necessary restrictions, and mitigating those errors that could be generated before the application, providing strategic values and optimizing the operations developed in the institution.

DECLARACIÓN

El contenido del presente Trabajo de Graduación es de mi responsabilidad; el patrimonio intelectual del mismo pertenece a la Universidad Estatal Península de Santa Elena.



BAQUERIZO G

Baquerizo Gómez José Antonio

TABLA DE CONTENIDOS

ITEM	PÁGINA
AGRADECIMIENTO	2
APROBACIÓN DEL TUTOR	3
TRIBUNAL DE GRADO	4
RESUMEN	5
ABSTRACT	6
DECLARACIÓN	7
TABLA DE CONTENIDOS	8
ÍNDICE DE FIGURAS	12
ÍNDICE DE TABLAS	13
LISTA DE ANEXOS	15
INTRODUCCIÓN	16
CAPÍTULO I	17
1. FUNDAMENTACIÓN	17
1.1. Antecedentes	17
1.2. Descripción del proyecto	19
1.3. Objetivos del proyecto	22
1.3.2. Objetivo general	22
1.3.3. Objetivos Específicos	22
1.4. Justificación	22
1.5. Metodología del proyecto	24
1.5.2. Metodología de la investigación	24
1.5.3. Metodología de recolección de información	24
1.5.4. Variable del proyecto	25
1.5.5. Beneficiarios del proyecto	25
1.5.6. Metodología de desarrollo	25

CAPITULO II	27
2. La propuesta	27
2.1. Marco Contextual	27
2.1.2. Escuela de Educación Básica “Lic. Angélica Villón Lindao”	27
2.1.3. Misión	28
2.1.4. Visión	28
2.1.5. Educación General Básica Ecuatoriana (EGB)	28
2.2. Marco Conceptual	29
2.2.2. Aplicación web	29
2.2.3. Interfaz de programación de aplicaciones (API)	30
2.2.4. Arquitectura MERN	30
2.2.5. Single Page Application (SPA)	31
2.2.6. Base de datos	31
2.2.7. Base de datos no relacional orientada a documentos	31
2.2.8. JavaScript	32
2.2.9. React JS	32
2.2.10. Next JS	32
2.2.11. Node JS	33
2.2.12. Express JS	33
2.2.13. MongoDB	33
2.2.14. Entorno de Ejecución	34
2.2.15. NginX	34
2.2.16. Editor de código	34
2.2.17. Visual Studio Code	34
2.2.18. Sistema de control de versiones	34
2.2.19. Git	35
2.2.20. Github	35

2.2.21.	Postman	35
2.2.22.	Server-side Rendering	35
2.2.23.	JSON web token (JWT)	35
2.2.24.	Diagrams.net	35
2.2.25.	Programación lineal entera (PLE)	36
2.3.	Marco teórico	37
2.3.2.	Calidad de horarios académicos para estudiantes y docentes	37
2.3.3.	La automatización de los procesos en las instituciones educativas	38
2.3.4.	Investigación de operaciones en Analítica de negocios	38
2.3.5.	Optimización por programación lineal	38
2.3.6.	Herramientas de código libre	39
2.4.	Componentes de la propuesta	39
2.4.2.	Módulos del sistema	39
2.4.3.	Requerimientos funcionales	43
2.4.4.	Requerimientos no funcionales	47
2.5.	Diseño de la propuesta	48
2.5.2.	Arquitectura de software	48
2.5.3.	Diagrama de Casos de Uso	49
2.5.4.	Diagrama de caso de uso general de la aplicación	49
2.5.5.	Diagrama de caso de uso Curso y paralelo	50
2.5.6.	Diagrama de caso de uso Docentes	51
2.5.7.	Diagrama de caso de uso Asignaturas	52
2.5.8.	Diagrama de caso de uso Gestión de horarios	53
2.5.9.	Diagrama de caso de uso Solicitud de horarios	54
2.5.10.	Diagrama de caso de uso Obtener reportes	55
2.5.11.	Diagrama de actividades	56
2.5.12.	Modelo de programación lineal	57

2.5.13.	Diagrama Documental	60
2.5.14.	Diseño de interfaces	61
2.5.15.	Diccionario de datos	65
2.6.	Estudio de Factibilidad	72
2.6.1.	Factibilidad técnica	72
2.6.2.	Factibilidad financiera	73
2.7.	Resultados	75
2.7.1.	Pruebas de funcionalidad	75
2.7.2.	Resultados	86
2.7.3.	Resultados finales	87
2.8.	Conclusiones	88
2.9.	Recomendaciones	89
2.10.	Bibliografía	90
	ANEXOS	95

ÍNDICE DE FIGURAS

ITEM	PÁGINA
Ilustración 1 Modelo incremental del software.	26
Ilustración 2 Ubicación de la Escuela Angélica Villón Lindao.	27
Ilustración 3 Esquema arquitectura MERN	30
Ilustración 4 Esquema de base de datos orientada a documentos.	32
Ilustración 5 Ejemplo de Documento JSON en MongoDB.	33
Ilustración 6 Ejemplo de función objetivo.	36
Ilustración 7 Ejemplo de modelización de programación lineal.	36
Ilustración 8 ejemplo de conjunto de restricciones lineales.	37
Ilustración 9 Clasificación de los problemas de optimización.	39
Ilustración 10 Arquitectura del sistema.	48
Ilustración 11 Caso de uso general	49
Ilustración 12 Caso de uso Curso y paralelo	50
Ilustración 13 Caso de uso Docentes	51
Ilustración 14 Caso de uso Asignaturas	52
Ilustración 15 Caso de uso Gestión de horarios	53
Ilustración 16 Caso de uso Solicitud de horarios	54
Ilustración 17 Caso de uso Obtener Reportes	55
Ilustración 18 Diagrama de actividades de la aplicación	56
Ilustración 19 Diagrama de documentos de la aplicación	60
Ilustración 20 Intefaz principal de la aplicación	61
Ilustración 21 Interfaz de listados de la aplicación	61
Ilustración 22 Interfaz de ingreso de datos de la aplicación	62
Ilustración 23 Interfaz de creación de horarios	62
Ilustración 24 Interfaz de gestión de horarios	63
Ilustración 25 Interfaz de visualización de horarios	63
Ilustración 26 Interfaz de reportes	64

ÍNDICE DE TABLAS

ITEM	PÁGINA
Tabla 1 Beneficiarios de la solución	25
Tabla 2 Áreas de conocimiento de la EGB ecuatoriana.	29
Tabla 3 Requerimientos funcionales de la aplicación	46
Tabla 4 Requerimientos no funcionales de la aplicación	47
Tabla 5 Caso de uso Curso y paralelo	50
Tabla 6 Caso de uso Docentes	51
Tabla 7 Caso de uso Asignaturas	52
Tabla 8 Caso de uso Gestión de horarios	53
Tabla 9 Caso de uso Solicitud de horarios	54
Tabla 10 Caso de uso Obtener Reportes	55
Tabla 11 Parámetros del método matemático	57
Tabla 12 Diccionario de datos - Schoolyear	65
Tabla 13 Diccionario de datos – Course	66
Tabla 14 Diccionario de datos – Parallel	67
Tabla 15 Diccionario de datos – Class	68
Tabla 16 Diccionario de datos – Teacher	69
Tabla 17 Diccionario de datos - Schedule	70
Tabla 18 Diccionario de datos - Account	71
Tabla 19 Factibilidad técnica: software	72
Tabla 20 Factibilidad técnica: Hardware	72
Tabla 21 Factibilidad financiera: Hardware	73
Tabla 22 Factibilidad financiera: Software	73
Tabla 23 Factibilidad financiera: RRHH	74
Tabla 24 Factibilidad financiera: Gastos varios	74
Tabla 25 Factibilidad financiera: Costo Total	74
Tabla 26 Prueba: Acceso a la aplicación	76
Tabla 27 Prueba: Ingreso de periodos lectivos	77
Tabla 28 Prueba: Ingreso de cursos	78
Tabla 29 Prueba: Ingreso de paralelos	79
Tabla 30 Prueba: Ingreso de profesores	80
Tabla 31 Prueba: Creación de horarios de clases	82

Tabla 32 Prueba: Creación de horarios de clases automáticos	83
Tabla 33 Prueba: Obtención de reportes	84
Tabla 34 Prueba: Obtención de horarios de clases	85
Tabla 35 Análisis de resultados	86

LISTA DE ANEXOS

ITEM	PÁGINA
Anexo 1 Formato de entrevista	95
Anexo 2 Formato de observación general de la institución	96
Anexo 3 Formato de observación, procesos de gestión de horarios	97
Anexo 4 Entrevista 07 de diciembre del 2020	97
Anexo 5 Pantalla de inicio de sesión	98
Anexo 6 Pantalla principal	98
Anexo 7 Módulo de usuarios	99
Anexo 8 Módulo de administración	99
Anexo 9 Módulo de gestión de horarios	100
Anexo 10 Panel de creación automática	100
Anexo 11 Pantalla de visualización de horarios	101
Anexo 12 Módulo de reportes	101
Anexo 13 Pruebas del modelo matemático en desarrollo	102
Anexo 14 Pruebas exitosas del modelo matemático	102

INTRODUCCIÓN

La escuela de educación básica “Lic. Angélica Villón Lindao” es una institución joven que continuamente se encuentra mejorando sus procesos. Uno de ellos es la generación de horarios de clases, que debe garantizar la correcta estructuración de las jornadas académicas para los estudiantes y profesores de la institución.

Este proceso, que la institución ejecuta mediante herramientas de escritorio comunes, genera ciertas limitantes que lo vuelven tardío y susceptible a errores, generando conflictos y correcciones que pueden derivar en la pérdida de clases durante las primeras semanas haciendo uso innecesario del tiempo de los administrativos de la institución. Del mismo modo, definir aquellos parámetros y patrones a seguir para la correcta estructuración de los horarios de clases puede resultar complicado a simple vista y no mitiga los conflictos antes mencionados.

En base a estas características la presente propuesta tecnológica tiene como finalidad, abordar la gestión de horarios de clases y automatizarlos mediante una aplicación web para la institución de forma que se logre reducir el tiempo empleado en el desarrollo de los horarios y no existan errores inesperados que retrasen las labores académicas. La aplicación web está construida con el stack MERN y consta de los siguientes módulos: sesión, usuario, administración, gestión de horarios y reportes.

Para el desarrollo de la aplicación web se tomó en cuenta el modelo cliente-servidor sobre el cual se ejecutarán la base de datos, la aplicación web y la interfaz web (API) del sistema. Toda esta arquitectura está basada en el lenguaje de programación JavaScript y el uso de bases de datos no relaciones orientadas a documentos MongoDB, por lo que lo vuelve un entorno especializado para el rápido desarrollo de aplicaciones.

La problemática de gestión de horarios de clases cumple con las características necesarias para ser considerada como un problema de optimización y asignación, por lo cual la investigación de operaciones nos ayuda a plantear matemáticamente un modelo de programación lineal entera que sujeto a un conjunto de restricciones nos hace posible hallar una solución óptima para el problema abordado.

La combinación de estas herramientas permitió automatizar la generación de horarios de clases y reducir el tiempo empleado a unos pocos segundos, por lo que sirve de apoyo estratégico fundamental para la institución.

CAPÍTULO I

1. FUNDAMENTACIÓN

1.1. Antecedentes

En el ámbito educativo uno de los problemas más grandes que surgen al comienzo de los periodos académicos es el proceso de gestión de los horarios de clases. La correcta estructuración y organización de estos es una de las condiciones necesarias para mantener la eficiencia de los actores educativos y conseguir mayor rendimiento durante las jornadas académicas [1].

La problemática de la generación de los horarios depende del tipo de institución (escuelas, colegios, universidades), la programación los horarios, los eventos y clases a asignar. Sin embargo, resulta complejo determinar aquellos criterios que definen la sistematización con los que estos se deben realizar [2].

En la escuela “Lic. Angélica Villón Lindao” este proceso no se encuentra normalizado, por lo que es común observar que se programen mediante tanteo, provocando en ciertas ocasiones: conflictos, cruces de asignaturas y una mala distribución de estos recursos. Como consecuencia, se realizan correcciones a lo largo de las primeras semanas de actividades derivando en retrasos al flujo normal de las actividades académicas y pérdida de horas de clases. [Ver anexo 2](#)

La escuela de educación básica “Lic. Angélica Villón Lindao” se encuentra actualmente ubicada en el cantón Santa Elena. Fue fundada el 12 de septiembre del año 2000, y está constituida por un grupo de profesionales del área administrativa como de la docencia. La directiva de esta institución actualmente está conformada por el coordinador Arq. Mauricio Yunda Villao y la directora administrativa, Lcda. Ibelice Tomalá V. Esta institución busca brindar educación inclusiva y de calidad que promueva el constante aprendizaje de los jóvenes de la Península, además de desarrollar su potencial de manera virtual.

Al inicio de un periodo académico, y después de la matriculación la institución realiza varias tareas, tales como: el distributivo de cursos y paralelos, la asignación de las materias correspondientes a cada docente y la programación de los horarios de clases. Para este último, la administración necesita determinar la distribución de horas semanales por asignaturas, docentes encargados, recesos, la cantidad de cursos, paralelos, la duración de las jornadas, los intervalos, etc, de manera que estos sean elaborados tanto para los docentes y estudiantes. [Ver anexo 2](#)

Este proceso se realiza de forma espontánea con herramientas clásicas como el uso de hojas de cálculo y sin un protocolo específico. Esto no contribuye con la eficacia del desarrollo de horarios y su dificultad de gestión incrementa conforme aumenta la cantidad de docentes, asignaturas y cursos a lo largo de los periodos académicos. Esta situación, ligada a la carencia tecnológica que posee la unidad educativa, limita una organización eficaz y la resolución de esta problemática. [Ver anexo 2](#)

En la oficina de la dirección de la Escuela de Educación Básica "Lic. Angélica Villón Lindao" se efectuó una entrevista al Arq. Mauricio Yunda Villao actual coordinador de la institución, quien manifestó cómo se desarrolla el proceso descrito para la planificación de los horarios de clases. Además, recaló los problemas que esto conlleva en el correcto proceder del quehacer educativo, la falta de infraestructura tecnológica, y los recientes cambios que la administración planea para la implementación de activos tecnológicos en la institución, despertando así, el interés por el desarrollo de estos proyectos a través de la plataforma web que beneficien y se alineen con la visión de la institución. [Ver anexo 1](#)

De la misma manera se ha explorado otras soluciones para la administración de los horarios dentro de instituciones educativas. La APLICACIÓN PARA LA GESTIÓN DE HORARIOS ACADÉMICOS desarrollada en la Universidad de Vigo – España, permite la gestión y divulgación de horarios académicos para la Escuela de Ingeniería de Telecomunicaciones de dicha universidad, sin embargo, debido a que este software actualmente es de uso interno no se puede acceder a él. Además, está enfocado en la gestión y divulgación de los horarios hacia los estudiantes y no posee un enfoque para el aporte logístico y la optimización de este proceso [3].

A nivel nacional y local existen dos soluciones para esta problemática referentes a la gestión de horarios académicos, el primero se enfoca en un software de escritorio para la Facultad de Ciencias Físicas y Matemáticas de la Universidad Central del Ecuador, pero esta se caracteriza por estar enfocada solo a la institución de educación superior, por lo tanto, sus requerimientos varían un poco respecto a las escuelas y colegios [4]. A nivel local se implementó un sistema de gestión de horarios para el colegio UPSE, sin embargo, este último está diseñado solo para usarse de forma local y, aunque cumple con los requerimientos del colegio, fue construido como software de escritorio lo que vuelve difícil su uso desde dispositivos externos, además de contribuir solo en la organización de los horarios, mas no el aporte logístico para la optimización del proceso [5].

Luego de la revisión e investigación de un grupo de soluciones relacionadas con esta temática se ha notado que cumplen parcialmente con los requerimientos actuales, mas no complementan toda la necesidad de esta institución, es decir, que funcione como una aplicación web que permita la gestión y automatización de los horarios y provea métricas que ayuden a mejorar este proceso. Por ello se propone una aplicación web de gestión y automatización del proceso mencionado en la institución “Lic. Angélica Villón Lindao” con la finalidad de mejorar la administración y reducir el tiempo que esto conlleva.

1.2. Descripción del proyecto

Debido a la actual necesidad de la Escuela de Educación básica "Lic. Angélica Villón Lindao" de contar con una solución tecnológica para optimizar el proceso de gestión de los horarios de clases y distribución óptima de la carga del itinerario para los docentes [ver anexo 1](#), se propone el desarrollo de una aplicación web para el soporte y la gestión de los horarios de clases, a partir de la creación de su base de datos, los módulos de inicio de sesión, gestión de docentes, cursos, paralelos y asignaturas, gestión y construcción de los horarios de clases y gestión de reportes que soporten estratégicamente a la correcta distribución horaria.

La aplicación web permitirá acceso a los usuarios con tres roles diferentes: administrador, colaborador, y observador, mediante correo electrónico y contraseña. El nivel de privilegios que tenga cada usuario será determinado por el rol que cumpla en la aplicación.

Todos los usuarios tendrán permisos de visualización en el sistema, donde podrán acceder al listado de asignaturas, el listado de los docentes, y podrán observar y guardar los horarios desarrollados, tanto en su vista para docentes, como para los estudiantes. El administrador podrá incluir nuevos usuarios y asignarles los roles de colaborador u observador, donde registrará su correo electrónico y contraseña. Las cuales el usuario podrá modificar posteriormente.

El usuario administrador y colaborador podrán registrar: los periodos lectivos, cursos, paralelos, y asignaturas correspondientes, siguiendo un patrón de dependencia jerárquica. Además, podrá registrar el listado de docentes general según el periodo lectivo con información de interés tal como: nombres, apellidos, cédula de identidad, correo electrónico que ayude a identificar una mejor postura para su asignación horaria.

El usuario administrador y colaborador podrán asignar y gestionar a cada docente registrado la(s) asignatura(s) o curso(s) en los que se desempeñará durante un periodo lectivo determinado, para su posterior asignación en los horarios de clases.

El usuario administrador y colaborador serán capaces de gestionar los horarios para cada curso, en los que se tendrá en cuenta, los días laborales, asignaturas, los profesores designados, la cantidad de horas semanales, y serán comparados con los mismos parámetros de otros cursos para evitar el choque de asignaturas o una incorrecta distribución de los horarios.

Los horarios de clases también podrán ser generados de forma automática, en el que se buscará la solución óptima y compacta mediante la aplicación del método matemático: Optimización por Programación Lineal Entera (PLE) para garantizar la calidad de los horarios académicos generados y respetar cada una de las restricciones impuestas por las necesidades de la institución.

Todos los usuarios tendrán permisos de visualización en el sistema, donde podrán acceder al listado de asignaturas, el listado de los docentes, y podrán observar y guardar los horarios desarrollados, tanto en su vista para docentes, como para los estudiantes.

Se generarán reportes que brinden soporte logístico para la correcta estructuración y optimización de la carga horaria para los docentes, en la que se establecerán criterios y se los cotejará con variables y alternativas relacionadas que otorguen al administrador una visión clara para la mejora del distributivo en desarrollo. Además, tendrán la posibilidad de revisar alertas cuando existan conflictos entre los horarios y realizar la búsqueda de periodos libres en común durante la jornada de clases.

Para diseño y desarrollo de la aplicación se ha considerado trabajar con MERN Stack y el uso de sus tecnologías claves basadas en JavaScript por representar un entorno totalmente especializado y de código abierto [6]. Las principales herramientas utilizadas son:

- **JavaScript:** Es el lenguaje común para el desarrollo del proyecto. Es utilizado para la creación de la aplicación web y la API mediante el uso de las librerías y marcos de trabajo que componen el Stack MERN [6].
- **ReactJS:** Es la biblioteca de JavaScript para el desarrollo de la aplicación web, componente front-end del proyecto [7]. Es utilizado para integrar la información con la interfaz como una Single Page Application [8].
- **NextJS:** Es el marco de trabajo de ReactJS para generar la renderización del lado del servidor, optimizando el procesamiento y el rendimiento de nuestra aplicación web [9].
- **NodeJS:** Es el entorno de ejecución para el proyecto desde el que se desplegará nuestra API [10].
- **ExpressJS:** Es el marco de trabajo de NodeJS para el desarrollo de interfaces web (API) y comunicar la aplicación con su base de datos, además de manejar las solicitudes y la autenticación [11].
- **MongoDB:** Es el sistema de base de datos no relacional y orientada a documentos, que se adapte perfectamente a las características complejas de la estructura de los horarios de clases [12].

El desarrollo de esta solución tecnológica para la gestión y planificación de los horarios de clases dentro de la Escuela de Educación Básica “Lic. Angélica Villón Lindao”, se acopla a la línea de investigación de gestión del desarrollo de software con la implementación de la aplicación web y la gestión de bases de datos para brindar soporte a la gestión y optimización del proceso de distribución de cargas y planificación de horarios [13].

1.3. Objetivos del proyecto

1.3.2. Objetivo general

Desarrollar una aplicación web para la gestión y planificación de horarios de clases en la Escuela "Lic. Angélica Villón Lindao" mediante métodos matemáticos y las herramientas del MERN Stack.

1.3.3. Objetivos Específicos

1. Recopilar la información referente al proceso de creación y gestión de horarios para contribuir a la óptima resolución de la problemática en la institución.
2. Automatizar el proceso de creación de horarios de clases mediante optimización por Programación Lineal Entera.
3. Disminuir el tiempo invertido en el desarrollo, gestión y corrección de los horarios de clases durante el inicio de los periodos académicos.

1.4. Justificación

El continuo desarrollo de las empresas de todas las áreas impulsa a estas a afrontar la transformación digital, que se refiere al paso hacia el contexto tecnológico, introducción hacia las comunicaciones, aplicaciones, plataformas y diversos contenidos digitales a través de numerosos cambios en su modelo productivo, buscando afianzarse al ritmo acelerado de los crecientes mercados [14].

La incorporación de nuevas tecnologías requiere el correcto planteamiento de los parámetros dentro de un determinado problema, muchas veces, en ellos implican la transformación radical en las instituciones, ayudando a diferenciar los servicios e incrementar la voluntad del usuario, reducir el costo operativo de los procesos en base a saltos de eficiencia e incrementar el valor que retornan cada uno de sus procesos [14].

En base a los resultados tradicionales del proceso de gestión de horarios de clases, la escuela de Educación Básica "Lic. Angélica Villón Lindao" tiene la necesidad de

digitalizar y optimizar este proceso, trayendo consigo la implementación de las TI a su entorno. Por ello, se plantea la implementación de una aplicación web que sirva para la planificación, creación y gestión de los horarios de clases enfocado en la automatización y la mejora de la carga horaria del docente.

La aplicación web brindará hacia los usuarios una plataforma que garantice el correcto almacenamiento, procesamiento e integridad de la información para la gestión de los horarios de clases, agilizando el seguimiento de este proceso a partir de su implementación y eliminando las correcciones tardías cuando las jornadas de estudio ya han comenzado.

La aplicación web busca proporcionar una herramienta en la que se pueda realizar estos distributivos para cada uno de los cursos al inicio de los periodos académicos, incrementando la agilidad con la que se realiza este proceso, reduciendo el tiempo empleado y evitando/corrigiendo aquellos conflictos que surjan de manera oportuna.

Se proporcionarán métricas para la correcta distribución de la carga horaria para los docentes mediante la generación de reportes en los que se podrá realizar comparativas individuales y colectivas de la distribución horaria del profesorado, agrupados por: horarios, jornadas, cursos y profesores; además, obtener los horarios generados en formato PDF listos para impresión, en vistas para cada cursos y docentes.

La propuesta de esta solución, aunque esté destinada a la escuela “Angélica Villón Lindao” se acopla a la realidad de muchas de las instituciones educativas por lo que podría también ser implementada en un futuro hacia otras escuelas que componen el sistema educativo en el país.

De esta forma, la aplicación web para la gestión de horarios de clases incrementará la agilidad con la que se aborda la gestión de horarios de clases, reducirá los contratiempos relacionados al actual procedimiento, y facilitará a los administradores la toma de decisiones y distribución horaria para los docentes y estudiantes en la institución.

El tema propuesto está alineado a los objetivos del Plan Nacional de Desarrollo específicamente al siguiente eje:

Eje 2.- Economía al servicio de la sociedad.

Objetivo 5.- Impulsar la productividad y competitividad para el crecimiento económico sostenible de manera redistributiva y solidaria [15].

Política 5.6.- Promover la investigación, la formación, la capacitación, el desarrollo y la transferencia tecnológica, la innovación y el emprendimiento, la protección de la propiedad intelectual, para impulsar el cambio de la matriz productiva mediante la vinculación entre el sector público, productivo y las universidades [15].

1.5. Metodología del proyecto

1.5.2. Metodología de la investigación

Para el desarrollo de esta propuesta tecnológica se ejecutará el tipo de investigación exploratoria [16], en la que se va a recopilar información de trabajos relacionados con el objetivo de analizar situaciones que aterricen en la línea del proceso de gestión de horarios en diferentes entidades académicas y establecer comparativas que nos permitan tener una visión clara del alcance de cada uno de ellos y la problemática que se va a abordar.

Para conocer el proceso de gestión de horarios en la institución y extraer información que nos permita ejecutar esta propuesta, se ejecutará el tipo de investigación diagnóstica [16], mediante la aplicación de las técnicas de entrevista y observación para determinar aquellos factores que deriven en la resolución de este objeto de estudio.

1.5.3. Metodología de recolección de información

Para la recopilación de datos referentes al proceso en estudio se utilizarán técnicas de recolección de información cualitativas [17]. Se ejecutará la técnica de entrevista hacia el coordinador, Arq. Mauricio Yunda Villao ([Ver anexo 1](#)), donde se determinarán aquellos factores que influyen dentro del proceso y que puedan contribuir a la mejora de este, beneficiando a los actores (Inspectores y coordinadores) que operan esta gestión, e indirectamente a los educadores y alumnos de la entidad educativa.

Además, se aplicará la técnica de observación dentro de la oficina de administración de la entidad educativa, ubicada en la calle 18 de agosto entre Paquisha y, Fausto Fajardo,

Santa Elena, dónde se coordinarán las actividades a realizar y se detallará el paso a paso que conlleva el proceso de gestión de horarios. ([Ver anexo 2](#))

1.5.4. Variable del proyecto

Tiempo: La presente propuesta busca optimizar este proceso de gestión de horarios de clases mediante la reducción del tiempo empleado desde su organización y corrección hasta la publicación de estos durante el inicio de los periodos lectivos. Se evaluará a los actores que intervienen en este proceso antes y después de la implementación de la solución tecnológica.

1.5.5. Beneficiarios del proyecto

La solución tecnológica beneficiará directa e indirectamente a grupos que forman parte de la institución educativa. Detallados mediante la siguiente tabla:

Beneficiarios	Cantidad
Directos	
Administradores	5
Indirectos	
Estudiantes	160
Docentes	12
	Total: 177

Tabla 1 Beneficiarios de la solución

1.5.6. Metodología de desarrollo

Para brindar una solución de calidad se plantea utilizar el modelo de desarrollo de software incremental ya que nos permite realizar una extensión de las fases que implementa el modelo cascada [18] pero basado en iteraciones para conseguir objetivos específicos. Cada iteración está vinculada con el avance de la iteración anterior. [19]

El modelo incremental en cada una de sus iteraciones contiene las siguientes etapas:

- **Análisis:** Se estudian los requerimientos de la etapa y los objetivos que esta abarcará.
- **Diseño:** Se maquetarán los procesos a realizar, los casos de uso, y la interfaz de usuario.

- **Codificación:** Se codificarán cada una de las actividades planteadas y diseñadas en los puntos anteriores.
- **Pruebas:** Se realizarán pruebas según los casos de usos para la prevención y corrección de errores antes de la implementación.

En la presente propuesta, cada incremento se verá representado como un módulo del sistema, iniciando desde el inicio de sesión y finalizando con los reportes. Cada uno de ellos se regirá por las 4 fases que se aplican en esta metodología tal como se muestra en la gráfica a continuación:

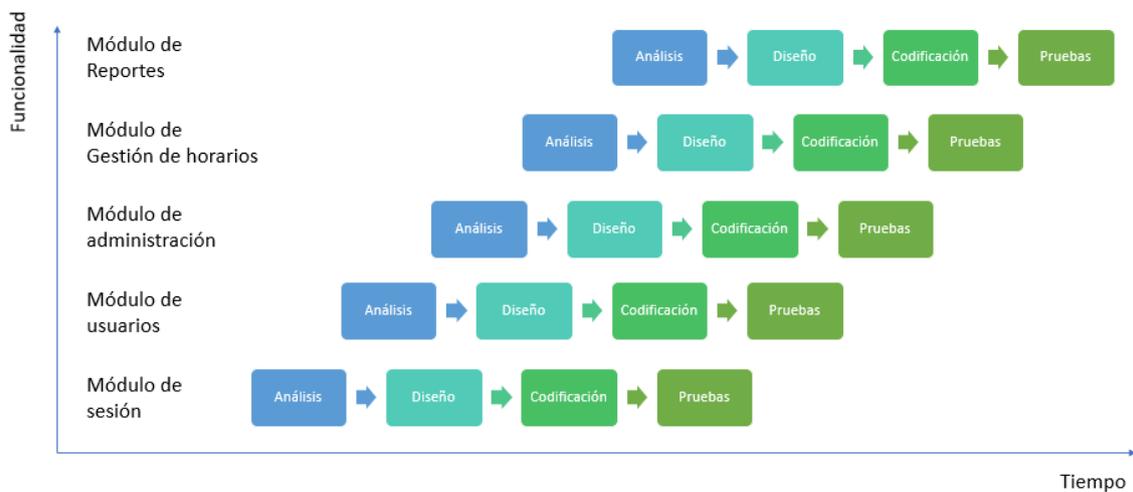


Ilustración 1 Modelo incremental del software.

CAPITULO II

2. La propuesta

2.1. Marco Contextual

2.1.2. Escuela de Educación Básica “Lic. Angélica Villón Lindao”

La escuela de educación básica “Lic. Angélica Villón Lindao” está ubicada en el cantón Santa Elena y con fecha de fundación el 12 de septiembre del año 2000. La institución brinda servicios educativos inclusivos y de calidad, buscando promover el aprendizaje de las habilidades básicas y los valores de cada estudiante. La directiva de esta institución actualmente está conformada por el coordinador Arq. Mauricio Yunda Villao y la directora administrativa, Lcda. Ibélce Tomalá V. [Ver anexo 1](#)

La institución se encuentra ubicada en el cantón Santa Elena de la Provincia Santa Elena.



Ilustración 2 Ubicación de la Escuela Angélica Villón Lindao.

Fuente: Google maps

2.1.3. Misión

La escuela de Educación Básica “Lcda. Angélica Villón Lindao” ofrece un conjunto de talentos humanos técnicos y pedagógicos al servicio de la educación, desarrollando procesos educativos para los estudiantes con o sin necesidades educativas especiales, atendiendo con absoluta responsabilidad y equidad, sin discriminación alguna y con un enfoque en los derechos. [Ver anexo 3](#)

2.1.4. Visión

La escuela de Educación Básica “Lcda. Angélica Villón Lindao” anhela brindar una educación actualizada acorde a las exigencias que la sociedad actual demanda, con la atención y diversidad, con docentes capacitados y comprometidos al cambio, para enrumbar a la formación de talentos humanos que se inserten con facilidad de estudios y a la sociedad, que adquieran un pensamiento crítico y reflexivo a través de la investigación científica con una práctica permanente de valores éticos y morales por medio de la inclusión, permitiendo a las personas con necesidades educativas especiales ser parte de la comunidad educativa. [Ver anexo 3](#)

2.1.5. Educación General Básica Ecuatoriana (EGB)

La Educación General Básica en Ecuador permite que los estudiantes adquieran el conjunto de capacidades y habilidades elementales a partir de los valores del perfil bachiller ecuatoriano [20]. Esta etapa educativa abarca la educación desde primer grado a décimo grado y está dividida en cuatro subniveles:

1. Preparatoria, correspondiente al primer grado de EGB [20].
2. Básica Elemental, correspondiente al segundo, tercer y cuarto grado de EGB [20].
3. Básica Media, correspondiente al quinto, sexto y séptimo grado de EGB [20].
4. Básica Superior, correspondiente a octavo, noveno y décimo grado de EGB [20].

El currículum que engloba la Educación general básica permite desarrollar aprendizajes en las siguientes áreas con sus respectivas asignaturas:

ÁREAS DE CONOCIMIENTO	ASIGNATURAS PARA EGB
Lengua y Literatura	Lengua y Literatura
Lengua Extranjera	Inglés
Matemática	Matemática
Ciencias Naturales	Ciencias Naturales
Ciencias Sociales	Estudios Sociales
Educación Física	Educación Física
Educación Cultural y Artística	Educación Cultural y Artística

Tabla 2 Áreas de conocimiento de la EGB ecuatoriana.

Fuente: Ministerio de educación

Apegándose al reglamento actual de la Ley Orgánica de Educación Intercultural (LOEI). Y el acuerdo ministerial Nro. MINEDUC-ME-2016-00020-A. Que especifica la normativa horaria y de distribución académica actual para el componente básico (EGB) y bachillerato (BGU) [21].

2.2. Marco Conceptual

2.2.2. Aplicación web

Una aplicación web es un sistema informático que no necesariamente necesita estar instalado en el ordenador. Usualmente estas hacen uso de internet por medio de HTTP [22] y se ejecutan sobre un navegador. El término de aplicación web se asocia con el almacenamiento en la nube, por lo que la información puede ser accesible desde cualquier parte del mundo [23].

La evolución de la web ha permitido que el contenido que muestran los navegadores evolucione de texto plano a sistemas complejos que pueden soportar gran cantidad de peticiones continuas y totalmente integradas con bases de datos y servicios de terceros [23].

2.2.3. Interfaz de programación de aplicaciones (API)

Las interfaces de programación de aplicaciones, mayormente conocida por sus siglas en inglés, API, son un conjunto de protocolos y aplicaciones que tienen como objetivo integrar el software de aplicaciones, permitiendo comunicación bilateral mediante un conjunto de reglas entre sistemas aparentemente distintos [24].

2.2.4. Arquitectura MERN

MERN es el acrónimo para referirse a las 4 tecnologías especializadas en JavaScript: MongoDB, Express JS, React JS y Node JS. Estas son utilizadas para construir aplicaciones webs avanzadas mediante un entorno de código libre. MERN provee las herramientas necesarias partiendo desde el lado del cliente, hasta el lado del servidor [8].

En la parte del cliente, a diferencia de MEAN, que utiliza Angular junto a su patrón Modelo-Vista-Controllador, MERN funciona con una estructura SPA (Single Page Application) junto con React JS que se encarga de renderizar la vista conforme a los estímulos que tiene la aplicación web en el momento, sin necesidad de verificar los cambios manualmente [25].

Node JS y Express JS aparecen del lado del servidor como el entorno de ejecución que soportará JavaScript en esta arquitectura y el componente lógico que la aplicación consumirá respectivamente [8].

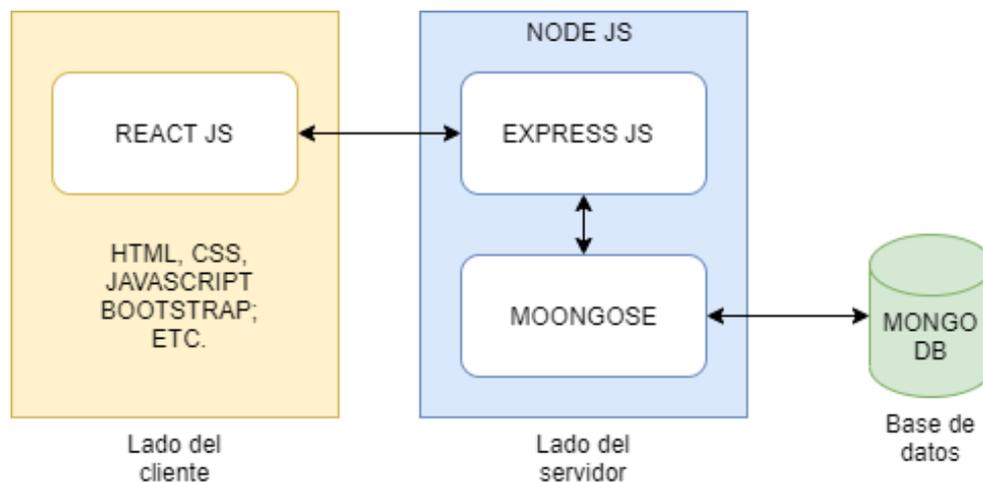


Ilustración 3 Esquema arquitectura MERN

2.2.5. Single Page Application (SPA)

Es una aplicación web que funciona en una sola página, sin necesidad de refrescar el navegador en cada momento que necesitamos trasladarnos de una sección a otra, otorgando una experiencia más fluida y rápida en el entorno web. Aunque una SPA contiene una sola página, maneja varias vistas como si se tratase de una página web normal. Este tipo de aplicaciones web reaccionan dinámicamente a los estímulos del entorno, nuevos datos, y cambios en el navegador, de forma que se sustituyen y actualizan solo los componentes necesarios [26].

2.2.6. Base de datos

Una base de datos es un conjunto de datos estructurados que pertenecen a un mismo contexto y son almacenados de forma centralizada o distribuida (repositorios). Las bases de datos se almacenan generalmente mediante documentos o textos, pero que también han migrado a un entorno totalmente digital desde inicios del siglo XX [27].

2.2.7. Base de datos no relacional orientada a documentos

Las bases de datos no relacionales, también conocidas como bases de datos NoSQL (Not Only SQL) proveen mecanismos que se alejan del modelo relacional de bases de datos, es decir, poseen un paradigma diferente sin dejar de procesar datos estructurados y semiestructurados [28]. Estas bases de datos han incrementado su uso en entornos Big Data y aplicaciones web en tiempo real [28].

Entre los diferentes paradigmas encontrados en un entorno NoSQL encontramos las bases de datos orientadas a documentos. Estos son almacenes que no poseen datos en forma de filas y columnas, debido a que estas proporcionan mucha más flexibilidad para trabajar con elementos que poseen características similares entre ellos. Los conjuntos de registros almacenados en este paradigma se conocen como documentos y sus contenedores como colecciones [29]. Dentro de estos se alojan datos semiestructurados que pueden ser consultados mediante DBMS u otras herramientas [29].

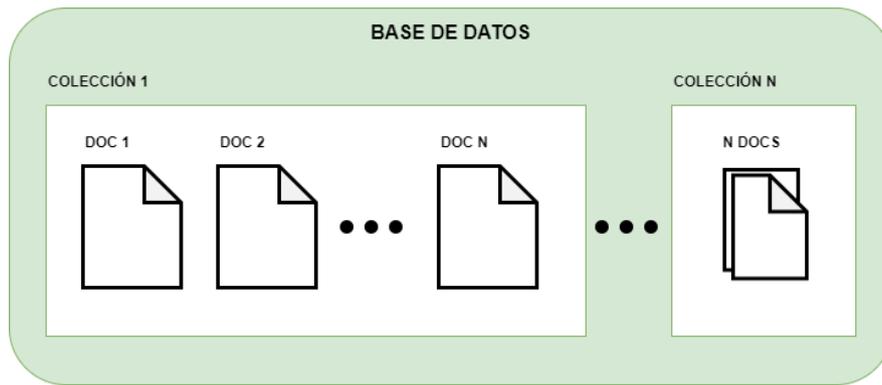


Ilustración 4 Esquema de base de datos orientada a documentos.

2.2.8. JavaScript

Es un lenguaje de programación interpretado basado en prototipos. Tiene soporte multiparadigma por lo que puede ser usado en programación funcional, u orientada a objetos [30]. Desarrollado principalmente para la web, en la actualidad, se desenvuelve en diversos escenarios: móvil, web, escritorio e interfaces API. Actualmente utiliza el estándar ECMAScript [31].

2.2.9. React JS

Es una biblioteca de JavaScript desarrollada por Facebook, como software libre para construir interfaces de usuario, como alternativa a Angular, React JS es declarativo y basado en componentes que se renderiza en el entorno Node [8]. El objetivo de React es integrar los datos con la interfaz y construir el entorno por medio de componentes [8].

2.2.10. Next JS

Es un framework de React JS que permite la renderización híbrida y del lado del servidor para la generación de aplicaciones web basadas en ReactJS [9]. Incluye soporte para optimización de imágenes, pre-renderizado, typescript, enrutamiento de API, CSS a nivel de componentes, Bundling, etc. [9].

2.2.11. Node JS

Es un entorno de ejecución multiplataforma para el lado del servidor, está basado en JavaScript y el motor V8 de Google, este entorno está diseñado para crear aplicaciones asíncronas de forma escalable [10]. El uso del protocolo HTTP permite que Node sea rápido y otorgue baja latencia, especialmente para el entorno web [10].

2.2.12. Express JS

Es un framework de Node JS presente como un software de código abierto para el desarrollo de aplicaciones web y APIs basados en JavaScript [11]. Su flexibilidad otorga características para aplicaciones web y móviles. Permite el uso del protocolo HTTP de forma sólida y eficiente [11].

2.2.13. MongoDB

Es un sistema de base de datos NoSQL, orientado a documentos y de código abierto. En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida [12].

```
1  {
2    _id: "5cf0029caff5056591b0ce7d",
3    firstname: 'Jane',
4    lastname: 'Wu',
5    address: {
6      street: '1 Circle Rd',
7      city: 'Los Angeles',
8      state: 'CA',
9      zip: '90404'
10   }
11 }
```

Ilustración 5 Ejemplo de Documento JSON en MongoDB.

Fuente: <https://mongodb.com>

2.2.14. Entorno de Ejecución

Un entorno de ejecución proporciona toda la funcionalidad necesaria que el código necesita para ser ejecutado, partiendo desde el hardware y el lenguaje base de programación hasta cada una de las interacciones del usuario y los componentes de un software. Un entorno de ejecución carga todas las dependencias necesarias y dispone de recursos para la ejecución del software [32].

2.2.15. NginX

Es un servidor web open source que es utilizado también como “Proxy Reverse”, para web sockets, SSL y balanceador de carga. Este se caracteriza por el bajo uso de memoria con un enfoque asíncrono [33].

2.2.16. Editor de código

Un editor de código es un software editor de texto plano que específicamente está destinado a tratar con código fuente y lenguajes de programación. Los editores de código poseen características especiales destinadas a asistir a los desarrolladores de software, entre ellas se encuentran: velocidad de ejecución, compiladores, intérpretes y analizadores de código [34].

2.2.17. Visual Studio Code

Visual Studio Code es un software editor de código fuente de escritorio disponible para Windows, MacOS y Linux. Incluye principal soporte para Javascript, Typescript y NodeJS, con un amplio ecosistema de extensiones y personalización para otros lenguajes tales como C, C++, C#, Java, Python, PHP, Golang, etc. Además de integraciones con entornos como Unity y .NET [34].

2.2.18. Sistema de control de versiones

Los sistemas de control de versiones son aplicaciones que permiten gestionar y documentar el proceso de desarrollo de software. Estos sistemas permiten que múltiples desarrolladores trabajen colaborativamente y realicen respaldos del código para su recuperación sin ocasionar pérdidas o errores catastróficos de código [35].

2.2.19. Git

Git es un sistema de control de versiones distribuido de código abierto y gratuito diseñado para manejar todo, desde proyectos pequeños a muy grandes, con velocidad y eficiencia. Posee características como bifurcaciones locales económicas, staging áreas y múltiples flujos de trabajo. [36]

2.2.20. Github

Github es una interfaz web para el desarrollo colaborativo. Permite contribuir en diferentes proyectos mediante un entorno automático [37]. Permite la revisión de código, la generación de ramas de desarrollo, importar y exportar desde el sistema de control de versiones de Git, discusiones en equipo, y respaldos de código y repositorios públicos [37].

2.2.21. Postman

Postman es una herramienta colaborativa para el desarrollo de APIs para simplificar el paso a paso de la construcción de interfaces de aplicaciones y proveer testing de forma automática. Permite generar y publicar documentación atractiva y legible para que las APIs sean más fácil de consumir [38].

2.2.22. Server-side Rendering

El server-side Rendering, o Renderización en parte del servidor permite generar HTML y cada uno de los componentes, procesándolos en el servidor, en vez de hacerlo en la parte del cliente, que demandaría más consumo de recursos especialmente para el desarrollo de aplicaciones web complejas [39].

2.2.23. JSON web token (JWT)

JSON web tokens son credenciales de código abierto que permiten el acceso a recursos mediante dos participantes de forma segura basadas en el estándar RFC 7519 [40]. Las peticiones en JWT son codificadas mediante objetos JSON como carga, y encriptación basada en diferentes algoritmos soportados [40].

2.2.24. Diagrams.net

Diagrams.net es una herramienta en línea colaborativa. Permite la creación de diagramas UML, mapas conceptuales, diagramas de red, etc. Basada en la nube, es de acceso libre y permite integraciones con Git, Google Drive, OneDrive y nextCloud [41].

2.2.25. Programación lineal entera (PLE)

En términos generales, aquellos fenómenos en los que intervienen variables positivas y que se puedan relacionar mediante relaciones de igualdades o desigualdades que reflejen aquellas limitaciones del problema a resolver puede plantearse como un modelo de programación matemática. Además, si estas restricciones y la función objetivo se definen todas como lineales, entonces estaremos frente al campo de este modelo denominado “Programación Lineal” [42].

La programación lineal es un modelo matemático que se dedica a optimizar una función objetivo. Esta función está sujeta a un conjunto de restricciones mediante un sistema de ecuaciones e inecuaciones igualmente lineales, siempre y cuando ninguna de estas pueda ser negativa [42].

El rendimiento dentro de la programación lineal se refiere a la optimización de la función objetivo y se puede expresar de dos formas:

- Maximización: Cuando se busca ampliar el máximo número o utilidad de un resultado [43].
- Minimización: Cuando se busca reducir el número de recursos utilizados para un proceso [43].

$$Z = C_1 X_1 + C_2 X_2 + \dots + C_n X_n$$

$$Z = \sum_{j=1}^n C_j X_j$$

Ilustración 6 Ejemplo de función objetivo.

Fuente: Dialnet

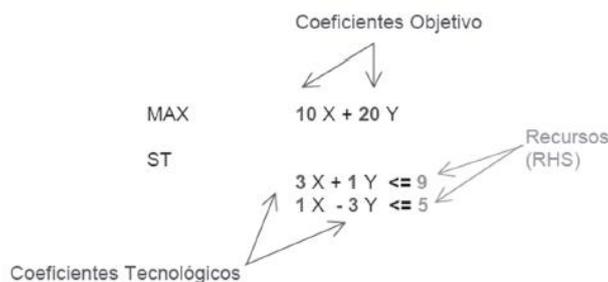


Ilustración 7 Ejemplo de modelización de programación lineal.

Fuente: Dialnet

Las restricciones expresadas por igualdades y desigualdades están compuestas por los coeficientes técnicos, los procesos o actividades a medir, y las limitaciones que conllevan cada uno de ellos. Estos componentes son todos tomados en cuenta dentro de la función objetivo. [44]

$$\begin{aligned}
 &A_{11} X_1 + A_{12} X_2 + \dots + A_{1n} X_n \leq B_1 \\
 &A_{21} X_1 + A_{22} X_2 + \dots + A_{2m} X_n \geq B_2 \\
 &\dots \\
 &A_{m1} X_1 + A_{m2} X_2 + \dots + A_{mn} X_n = B_m \\
 &X_1, X_2, \dots, X_n \geq 0
 \end{aligned}$$

Ilustración 8 ejemplo de conjunto de restricciones lineales.

Fuente: Dialnet

Estas restricciones pueden ser constituir 3 tipos básicos: “Mayor que”, “Menor que”, “Igual que”. Ellas constituyen la existencia de regiones factibles de las cuales existirá al menos una solución óptima [44].

La programación lineal al considerarse un campo amplio puede aplicarse a muchos problemas prácticos en el entorno real donde aquellos problemas pueden formularse estrictamente por una serie de algoritmos especializados hacia su solución [44].

2.3. Marco teórico

2.3.2. Calidad de horarios académicos para estudiantes y docentes

La problemática de los horarios escolares radica en que resulta complicado determinar aquellos factores que intervienen de manera objetiva para resolver la asignación de turnos hacia los estudiantes y los docentes [1]. Este campo provee de enfoques interesantes para soluciones de alta calidad. Las asignaciones de horarios óptimas son sumamente importantes para prevenir los conflictos entre cursos, horarios y docentes [1].

La solución ideal es obtener horarios libres de conflictos, completos y compactos, sin embargo, en la realidad, estos resultados se obtienen después de varias correcciones,

muchas de ellas ya sobre la marcha de las jornadas académicas [1]. Bajo este panorama, se prioriza la calidad del desarrollo de los horarios académicos como parte fundamental del quehacer estudiantil y de la docencia para una correcta ejecución de los planes institucionales. [1]

2.3.3. La automatización de los procesos en las instituciones educativas

La academia constantemente se encuentra frente a cambios sustanciales dentro de sus parámetros educativos a través de las tecnologías de la información [45]. Es importante que cada institución posea la flexibilidad para adaptarse a estos cambios y adoptar nueva tecnología [45]. La automatización de los procesos académicos corresponder a la evolución de las instituciones, volviéndolas más productivas y optimizando sus recursos de una forma segura y eficaz.

2.3.4. Investigación de operaciones en Analítica de negocios

El rápido desarrollo de las industrias ha vuelto significativamente importante la toma de decisiones, en las que se aplican herramientas para el análisis descriptivo, predictivo, y prescriptivo destacando el análisis de datos, la inteligencia de negocios y la investigación de operaciones respectivamente [46].

Con estos avances y las nuevas implementaciones tecnológicas, las empresas vuelven complejas sus operaciones, y por ello se hacen presentes nuevos retos donde el principal problema suele ser la optimización de los recursos de la empresa. Entiéndase por recursos al factor económico, activos físicos, activos humanos, etc.

La investigación de operaciones logra abarcar esta necesidad de manera eficiente gracias al uso de modelos matemáticos exactos y heurísticos; uno de los modelos más populares es la programación lineal para la asignación de recursos y resolución de problemas.

2.3.5. Optimización por programación lineal

La programación Lineal plantea dar respuestas a situaciones en las que se requiere que el problema esté sujeto a restricciones específicas. Mediante este método matemático, se busca maximizar o minimizar (Optimizar) una función objetivo en base a las restricciones planteadas. Su empleo y utilidad es frecuente aplicaciones hacia pequeñas y medianas empresas, la academia, el sector industrial, la economía y la estrategia militar [42]. Esta puede ser aplicada en diferentes contextos dependiendo de la naturaleza del problema.

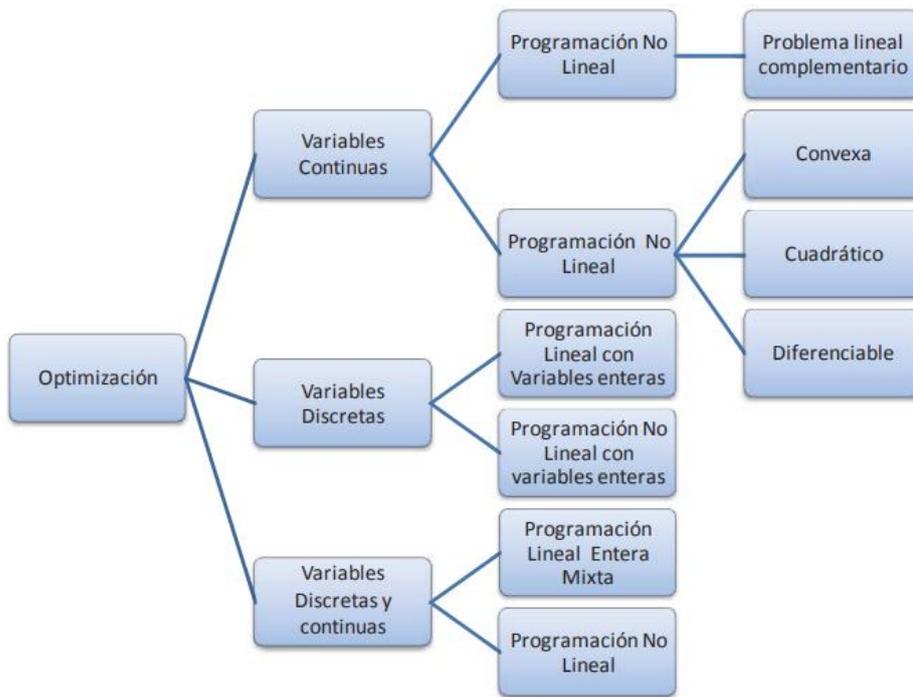


Ilustración 9 Clasificación de los problemas de optimización.

Fuente: Bemudez Yeicy, *Applications of linear, mixed and integer programming*

2.3.6. Herramientas de código libre

El éxito del software de código abierto ha dado la posibilidad a los desarrolladores enfrentar los desafíos tal y como lo harían las grandes empresas. Las herramientas de código abierto siempre se encuentran disponibles y adaptables al contexto donde se la utilice.

Su importancia radica en que ha cambiado el paradigma del consumo de software y lo ha vuelto más accesible a usuario en general. Estas herramientas son desarrolladas y mantenidas por la comunidad, por lo que el software propietario se está viendo desplazado por este movimiento y su principal finalidad es mover a la comunidad de las tecnologías de la información y el conocimiento sin ánimos de lucro abierta a las instituciones emergentes. [47]

2.4. Componentes de la propuesta

2.4.2. Módulos del sistema

El desarrollo de esta aplicación web permitirá a los participantes de la Escuela de Educación básica "Lic. Angélica Villón Lindao" mejorar el tiempo que se emplea en la estructuración y ejecución de este proceso a través de los siguientes módulos:

- **Módulo de sesión:** Permitirá el inicio de sesión por parte de los usuarios del sistema, mediante el correo electrónico y contraseña. Se contemplará implementar mecanismos de seguridad para el acceso de los usuarios. No se contemplará la verificación de cuentas mediante correo electrónico. El nivel de privilegios que tenga cada usuario será determinado por el rol que cumpla en la aplicación. (Administrador, colaborador, observador)
- **Módulo de usuarios:** Permitirá al administrador la gestión de los usuarios del sistema.
 - **Creación de usuarios:** El administrador podrá incluir nuevos usuarios y asignarles los roles de colaborador u observador, donde registrará su correo electrónico y contraseña. Las cuales el usuario podrá modificar posteriormente.
 - **Actualización de usuarios:** Permitirá la actualización de los datos de los usuarios, tales como el nombre, correo electrónico o una nueva contraseña.
 - **Eliminación de usuarios:** Excluirá al usuario seleccionado el acceso al sistema.
- **Módulo de administración:** Permitirá la gestión de los periodos lectivos, cursos, paralelos, asignaturas y docentes correspondientes, siguiendo un patrón de dependencia jerárquica.
 - **Administración de periodos académicos:** Permite registrar, modificar o eliminar un periodo académico. Este es el elemento del que parte todo el entorno de trabajo.
 - **Administración de docentes:** Permite registrar, modificar o eliminar a los docentes que conforman un periodo lectivo. Permite registrar también la información de cada docente, tales como: Nombres, apellidos, cédula de identidad, correo electrónico que ayude a identificarlo para su asignación horaria y asignaturas que impartirá.
 - **Administración de asignaturas:** Permite registrar, modificar o eliminar las asignaturas en el periodo lectivo, además se define cuántas horas semanales se deberán impartir en dicha asignatura y se selecciona el o los docentes que la impartirán, en uno o más cursos.
 - **Administración de cursos y paralelos:** Permite registrar, modificar o eliminar los cursos y paralelos dentro de un periodo lectivo.

- **Módulo de gestión de horarios:** En este módulo el usuario administrador y colaborador podrán distribuir las materias para cada curso, tomando en cuenta la duración de las jornadas académicas, la disponibilidad del docente y el espacio entre asignaturas. La creación de los horarios de clases se realiza individualmente para cada curso y el sistema asistirá al usuario cuando un horario entre en conflictos, y le advertirá de tomar cambios para corregirlos.

Los horarios académicos también pueden ser creados automáticamente gracias a la aplicación de programación lineal entera (PLE). [42] Este modelo matemático se dedica a maximizar o minimizar una función objetivo, de tal manera que esta es sujeta a creación y optimización basada en un conjunto de restricciones mediante un sistema de inecuaciones lineales adaptadas al problema a resolver; con miras a una solución libre de conflictos, compacta y balanceada para estudiantes y docentes. [44] Se tomará en cuenta:

1. Días escolares y periodos de clases.
2. Número de clases en cada curso.
3. Profesores disponibles.
4. La cantidad de horas que una asignatura debe cumplir semanalmente.
5. La duración máxima asignaturas de manera continua.
6. La disponibilidad horaria entre cursos.
7. La disponibilidad horaria del docente.
8. Sincronización con horarios existentes.

Los horarios de clases generados por el algoritmo PLE [42] pueden ser objeto de ajustes basados en las necesidades por factores humanos, por ello, estarán disponibles en todo momento para una posible modificación, rigiéndose a las restricciones bases para evitar cualquier tipo de conflictos.

Todos los horarios se pueden guardar para cada curso, y también estarán disponibles automáticamente en su versión para los docentes.

- **Módulo de reportes:** En este módulo se generarán reportes que brinden soporte logístico para la correcta estructuración y optimización de la carga horaria para los docentes, en la que se establecerán criterios y se los cotejará con variables y alternativas relacionadas que otorguen al administrador una visión clara para la mejora del distributivo en desarrollo.
 - **Reporte de análisis de carga horaria por docente:** Este reporte está destinado al análisis individual por docente, y toma como criterios de evaluación: Los días laborales en la institución, la cantidad de horas asignadas cada día y la cantidad de horas invertidas en una asignatura por día. Y ofrece como alternativas el análisis general por semana, por modalidad (Vespertina/Diurna) o por cursos específicos.
 - **Reporte de comparativas horarias por grupos de docentes:** Este reporte está destinado al análisis grupal de docentes, y toma como criterios de evaluación: El listado de docentes y la cantidad horas totales asignadas por cada docente (Dependiendo de la alternativa seleccionada). Y ofrece como alternativas el análisis semanal, por día específico, por asignatura o por curso específico.
 - **Reporte de horarios de clases para cursos:** Este reporte obtiene los horarios creados visibles para los estudiantes.
 - **Reporte de horarios de clases para docentes:** Este reporte obtiene cada horario visible para los docentes.

Esta aplicación web no abarcará la gestión de estudiantes para distribuirlos en cursos y paralelos al inicio de los periodos lectivos.

2.4.3. Requerimientos funcionales

Código	Requerimiento
RF-01	<p>El sistema interactuará con 3 tipos de usuario:</p> <ul style="list-style-type: none">a. Administrador: El usuario tendrá acceso completo al sistema.b. Colaborador: El usuario tendrá acceso solo al módulo de gestión de horarios de clases y la visualización de los reportes.c. Observador: El usuario solo tendrá acceso a la visualización de los reportes y los horarios de clases ya creados
RF-02	<p>El inicio de sesión se manejará mediante nombre de usuario y contraseña, si la contraseña es correcta el usuario es autenticado, caso contrario, el acceso es denegado.</p>
RF-03	<p>El usuario administrador podrá crear nuevos usuarios y otorgarles los privilegios que crea conveniente (Colaborador u Observador).</p>
RF-04	<p>La solución requiere de una aplicación web para garantizar su disponibilidad en un ambiente virtual.</p>
RF-05	<p>La aplicación web se basará en la siguiente composición: Un encabezado que contendrá el acceso a la configuración de usuario y el cierre de la sesión; un menú vertical izquierdo, que contendrá la navegación principal; el área de trabajo de cada una de las vistas y un pie de página con la información de autoría.</p>
RF-06	<p>El encabezado poseerá el título de la página, información de usuario y el acceso hacia el panel de configuración y perfil del usuario y cierre de sesión</p>
RF-07	<p>En el menú principal se incluirá el acceso hacia cada uno de los módulos (Administración, Gestión y Reportes) y sus secciones en la aplicación.</p>
RF-08	<p>En el encabezado del área de trabajo incluirá la ruta de navegación en la que se encuentra el usuario y el título de la sección.</p>
RF-09	<p>En el área de trabajo se contendrán los formularios correspondientes la sección activa según corresponda.</p>

RF-10	La aplicación deberá mostrar una alerta en el área de trabajo en caso de que los formularios se encuentren incompletos o incorrectos.
RF-11	Cada registro ingresado deberá contar con la opción respectiva para actualizar su información.
RF-12	Cada registro ingresado deberá contar con la opción respectiva para eliminar su información.
RF-13	La aplicación deberá mostrar un aviso en el área de trabajo cuando un formulario se envíe correctamente.
RF-14	Los datos almacenados deben ser recuperados en cada sección y mostrados en pantalla mediante tablas.
RF-15	La aplicación deberá almacenar los datos referentes al periodo lectivo, su descripción, y duración para posterior uso.
RF-16	En el menú principal se deberá seleccionar uno de los periodos lectivos para habilitar el acceso a cada módulo.
RF-17	La aplicación deberá almacenar cada curso con su(s) paralelo(s) en el periodo lectivo que se encuentre activo.
RF-18	La aplicación deberá almacenar cada profesor por periodo lectivo. Los profesores deben ingresarse con datos únicos como: Número de cédula y correo electrónico.
RF-19	La aplicación deberá almacenar cada asignatura en el periodo lectivo que se encuentre activo, a su vez se le podrá asignar uno o varios cursos y/o uno o varios profesores.
RF-20	Cada asignatura tendrá un número de horas totales y semanales por periodo lectivo.
RF-21	Los horarios de clases serán divididos en intervalos de 20 minutos para determinar el inicio y el fin de las horas de clases.
RF-22	Los horarios de clases tendrán dos modalidades: <ul style="list-style-type: none"> a. Matutina, comprendida de Lunes a Viernes desde las 8:00 a.m. a las 10:00 a.m. y de 10:40 a.m. a 12:00 p.m b. Vespertina, comprendida de Lunes a Viernes desde la 1:00 p.m. a las 3:00 p.m. y de 3:40 a 5p.m

RF-23	La creación de los horarios de clases se realiza individualmente para curso. El usuario podrá seleccionar la modalidad que crea conveniente.
RF-24	Las asignaturas por crear no deben exceder los 4 intervalos seguidos, tampoco deben superar su cantidad asignada de horas semanales
RF-25	Para insertar una asignatura dentro del horario de clases, el usuario seleccionará la asignatura, el profesor correspondiente, el día de la semana, la hora de inicio y la hora de fin.
RF-26	La aplicación deberá generar una alerta en caso de que la asignatura por crear se cruce en día y horas con una de las asignaturas ya existentes.
RF-27	La aplicación deberá generar una alerta en caso de que el profesor ya tenga una asignatura creada en otro curso y esta se cruce en día y horas con el nuevo registro.
RF-28	Los horarios de clases de pueden generar de forma automática mediante aplicación del modelo matemático de programación lineal entera (PLE)
RF-29	Los horarios de clases generados automáticamente pueden ser actualizados de forma manual, siguiendo los requerimientos especiales de la administración
RF-30	La aplicación deberá almacenar los horarios de clases para su posterior visualización y validación.
RF-31	La aplicación generará reportes para el análisis de la carga horarias de los académicos, estos reportes serán individuales y para grupos de docentes.
RF-32	Los reportes grupales serán mostrados mediante gráficos de barras y se tomarán en cuenta el listado de docentes y la cantidad horas totales asignadas por cada docente, el análisis puede ser diario, semanal, por asignaturas o cursos.
RF-33	Los reportes individuales están destinados por docente y mediante barras apiladas se toman los siguientes criterios de evaluación: Los

	días laborales en la institución, La cantidad de horas asignadas cada día y la cantidad de horas invertidas en una asignatura por día. Y ofrece como alternativas el análisis general por semana, por modalidad (Vespertina/Diurna) o por cursos específicos.
RF-34	Los horarios de clases elaborados y finalizados deben estar disponibles para su descarga por cada curso en formato PDF.
RF-35	Los horarios de clases elaborados y finalizados deben estar disponibles para su descarga por cada docente en formato PDF.

Tabla 3 Requerimientos funcionales de la aplicación

2.4.4. Requerimientos no funcionales

Código	Requerimiento
RNF-01	La información será almacenada en el sistema de Base de Datos NoSQL y Orientada a documentos MongoDB.
RNF-02	Las contraseñas de los usuarios serán encriptadas durante el registro y autenticación mediante el algoritmo SHA-256 con salteado de 5 pasos.
RNF-03	La autenticación de los usuarios será manejada mediante JSON web token.
RNF-04	Se verificará que la existencia de los campos necesarios sea única (Identificadores)
RNF-05	El token de una sesión activa se almacenará en Local Storage.
RNF-06	El tiempo de una sesión activa es de un máximo de 24 horas.
RNF-07	La carga enviada a la API será verificada del lado del cliente y también del servidor
RNF-08	La información será almacenada en el sistema de Base de Datos NoSQL y Orientada a documentos MongoDB.
RNF-09	La aplicación web se implementará y ejecutará sobre un servidor web con sistema operativo Linux - Ubuntu 16.04.
RNF-10	El esquema de base de datos se definirá en la API y regirá toda la estructura en formato: Colección - Documentos
RNF-11	La información se almacenará en formato BSON siguiendo la estructura de colecciones – documentos del esquema de base de datos.

Tabla 4 Requerimientos no funcionales de la aplicación

2.5. Diseño de la propuesta

2.5.2. Arquitectura de software

La arquitectura del software es la estructura fundamental sobre la que se soporta un sistema informático. Esta hace referencia a cómo se relacionan cada uno de sus componentes, cuál es la relación que existe entre ellos, cómo se comunican para dar vida al proyecto relacionado. Todos estos componentes permiten orientar su diseño y evolución hacia el resultado deseado [48].

Al trabajar con la pila de desarrollo MERN se toman en cuenta dos componentes importantes, el lado del cliente y del servidor. Esta arquitectura permite la construcción de aplicaciones web complejas donde se requiere separar las capas con las que el usuario interactúa y el componente lógico de la propuesta siguiendo el siguiente esquema de comunicación [6].

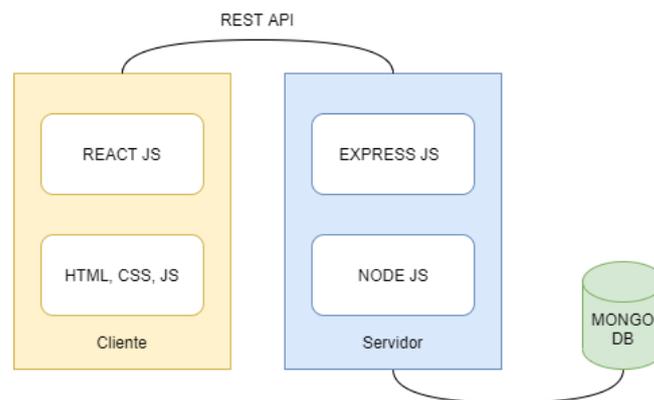


Ilustración 10 Arquitectura del sistema.

2.5.3. Diagrama de Casos de Uso

Un caso de uso es un conjunto de escenarios que conllevan un objetivo en común ejecutado por el usuario en una actividad particular de un sistema. El diagrama de caso de uso es la descripción de aquellas acciones que se ejecutan en un sistema desde el punto de vista del usuario. Conocido como “Actor”, este ejecuta un rol específico dentro del sistema y cada uno de ellos puede conllevar a instancias de caso de uso. [49]

2.5.4. Diagrama de caso de uso general de la aplicación

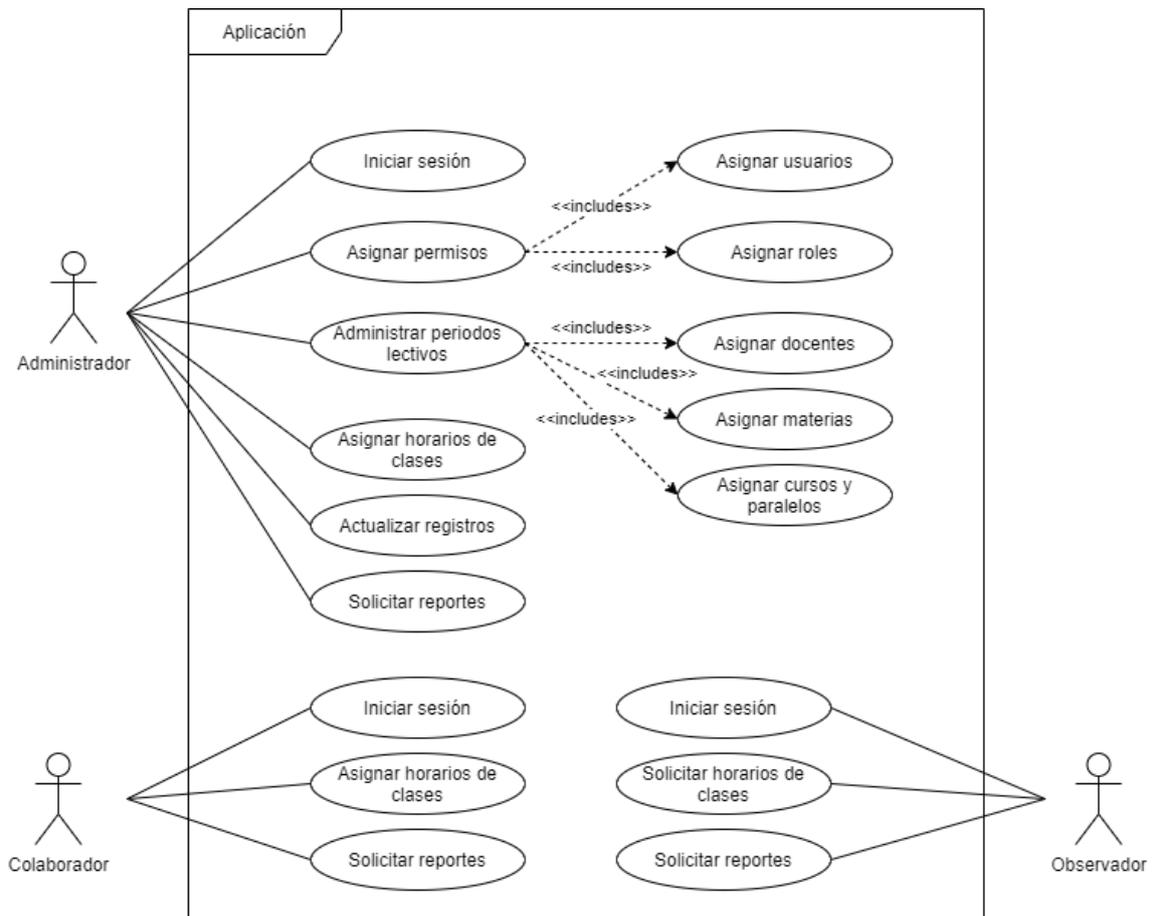


Ilustración 11 Caso de uso general

2.5.5. Diagrama de caso de uso Curso y paralelo

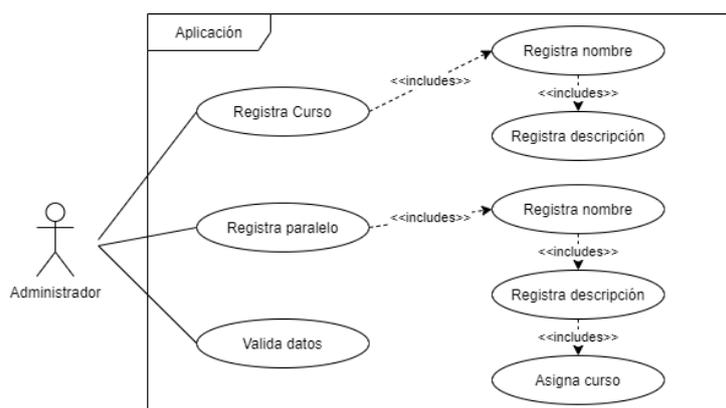


Ilustración 12 Caso de uso Curso y paralelo

Caso de uso:	Curso y Paralelos
Área:	Administración
Actores:	Administrador
Descripción:	Permite al administrador la creación cursos y asignar sus paralelos con los datos correspondientes.
Evento Desencadenador:	El usuario ingresa al sistema y se dirige al módulo de administración en la pestaña de cursos seguido de la pestaña paralelos.
Pasos	
	<ol style="list-style-type: none"> 1. El usuario selecciona el periodo lectivo para iniciar su espacio de trabajo. 2. El usuario se dirige a la sección Administración del menú lateral. 3. El usuario selecciona la pestaña “Cursos” 4. El usuario visualiza los cursos creados anteriormente. 5. El usuario se dirige al formulario de creación de cursos 6. El usuario registra el nombre y descripción del curso. 7. El usuario valida los datos y guarda la información. 8. El usuario selecciona la pestaña “Paralelos” 9. El usuario visualiza los paralelos creados anteriormente. 10. El usuario se dirige al formulario de creación de paralelos 11. El usuario registra el nombre y descripción del paralelo. 12. El usuario asigna un curso del listado al paralelo. 13. El usuario valida los datos y guarda la información.
Pre-condiciones:	Haber iniciado sesión con privilegios de administrador. Haber seleccionado un periodo lectivo.
Post-condiciones:	Asignar al menos un curso o docente para registrar la asignatura y continuar el proceso.
Requerimientos cumplidos:	RF-17: Creación de cursos y paralelos.

Tabla 5 Caso de uso Curso y paralelo

2.5.6. Diagrama de caso de uso Docentes

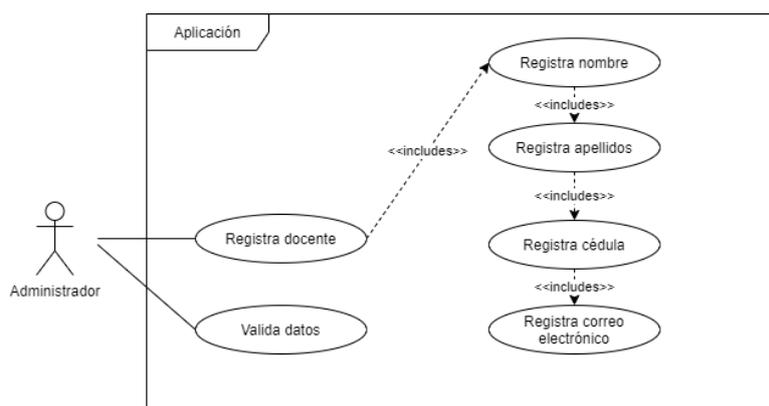


Ilustración 13 Caso de uso Docentes

Caso de uso:	Docentes
Área:	Administración
Actores:	Administrador
Descripción:	Permite al administrador la creación de docentes con sus respectivos datos básicos: nombres, apellidos, cédula y correo electrónico.
Evento Desencadenador:	El usuario ingresa al sistema y se dirige al módulo de administración en la pestaña de Docentes.
Pasos	
	<ol style="list-style-type: none"> 1. El usuario inicia sesión en la aplicación web. 2. El usuario selecciona el periodo lectivo para iniciar su espacio de trabajo. 3. El usuario se dirige a la sección Administración del menú lateral. 4. El usuario selecciona la pestaña “Docentes” 5. El usuario visualiza los docentes creados anteriormente. 6. El usuario se dirige al formulario de creación de docentes 7. El usuario registra el nombre y apellido del docente. 8. El usuario registra la cédula del docente. 9. El usuario valida que la cédula esté correcta y no se repita. 10. El usuario registra el correo electrónico del docente. 11. El usuario valida que el correo electrónico esté correcto y no se repita. 12. El usuario valida los datos y guarda la información. 13. El listado se actualiza y el usuario visualiza el nuevo campo creado.
Pre-condiciones:	Haber iniciado sesión con privilegios de administrador. Haber seleccionado un periodo lectivo.
Post-condiciones:	Asignar al menos un curso o docente para registrar la asignatura y continuar el proceso.
Requerimientos cumplidos:	RF-18: Creación de docentes.

Tabla 6 Caso de uso Docentes

2.5.7. Diagrama de caso de uso Asignaturas

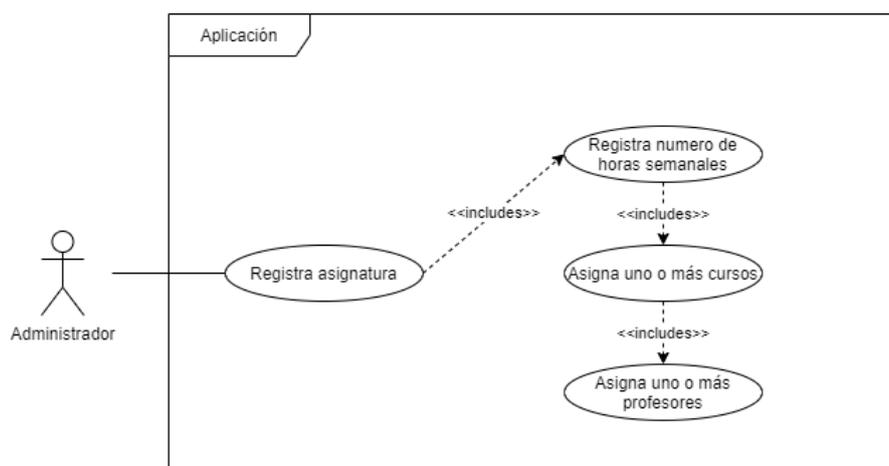


Ilustración 14 Caso de uso Asignaturas

Caso de uso:	Asignaturas
Área:	Administración
Actores:	Administrador
Descripción:	Permite al administrador la creación de asignaturas con su respectivo número de horas, cursos y docentes correspondientes.
Evento Desencadenador:	El usuario ingresa al sistema y se dirige al módulo de administración en la pestaña de Asignaturas.
Pasos	
	<ol style="list-style-type: none"> 1. El usuario inicia sesión en la aplicación web. 2. El usuario selecciona el periodo lectivo para iniciar su espacio de trabajo. 3. El usuario se dirige a la sección Administración del menú lateral. 4. El usuario selecciona la pestaña “Asignaturas” 5. El usuario visualiza las asignaturas creadas anteriormente. 6. El usuario se dirige al formulario de creación de asignaturas 7. El usuario registra el nombre de la asignatura. 8. El usuario registra la cantidad de horas semanales de la asignatura. 9. El usuario selecciona uno o más docentes para dicha asignatura. 10. El usuario selecciona uno o más cursos donde se impartirá dicha asignatura. 11. El usuario valida los datos y guarda la información. 12. El listado se actualiza y el usuario visualiza el nuevo campo creado.
Pre-condiciones:	Haber iniciado sesión con privilegios de administrador. Haber seleccionado un periodo lectivo Tener registros de cursos y docentes.
Post-condiciones:	Asignar al menos un curso o docente para registrar la asignatura y continuar el proceso.
Requerimientos cumplidos:	RF-19: Crear asignaturas RF-20: Asignar carga horaria

Tabla 7 Caso de uso Asignaturas

2.5.8. Diagrama de caso de uso Gestión de horarios

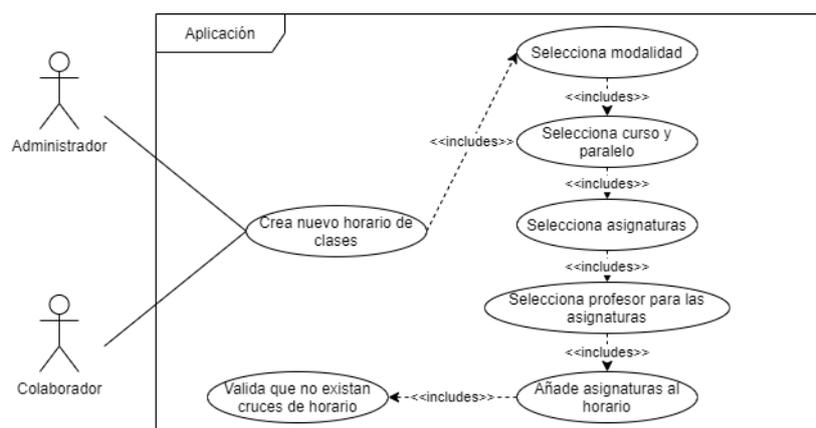


Ilustración 15 Caso de uso Gestión de horarios

Caso de uso:	Gestión de horarios
Área:	Administración
Actores:	Administrador, colaborador
Descripción:	Permite al administrador y colaborador la creación de horarios de clases para cada paralelo.
Evento Desencadenador:	El usuario ingresa al sistema y se dirige al módulo de gestión en la pestaña de horarios.
Pasos	
	<ol style="list-style-type: none"> 1. El usuario inicia sesión en la aplicación web. 2. El usuario selecciona el periodo lectivo para iniciar su espacio de trabajo. 3. El usuario se dirige a la sección Gestión del menú lateral. 4. El usuario selecciona la pestaña “Horarios”. 5. El usuario se dirige al formulario de creación de horarios. 6. El usuario selecciona la modalidad, el curso y el paralelo a trabajar. 7. El usuario selecciona el botón “Crear horario”. 8. El usuario se dirige al formulario de gestión de horario. 9. El usuario selecciona la asignatura y le asigna un profesor. 10. El usuario selecciona el día donde se añadirá la asignatura. 11. El usuario selecciona las horas de inicio y fin de la jornada en dicha asignatura. 12. El usuario selecciona el botón agregar. 13. El usuario repite el proceso las veces necesarias. 14. El usuario valida que no existan conflictos. 15. El usuario selecciona el botón guardar y finaliza los cambios.
Pre-condiciones:	<p>Haber iniciado sesión como administrador o colaborador.</p> <p>Haber seleccionado un periodo lectivo.</p> <p>Tener registros de cursos, paralelos, docentes, asignatura.</p>
Post-condiciones:	Registrar al menos una asignatura y continuar el proceso.
Requerimientos cumplidos:	RF-21 a RF-27: correspondientes al a gestión de horarios de clases

Tabla 8 Caso de uso Gestión de horarios

2.5.9. Diagrama de caso de uso Solicitud de horarios

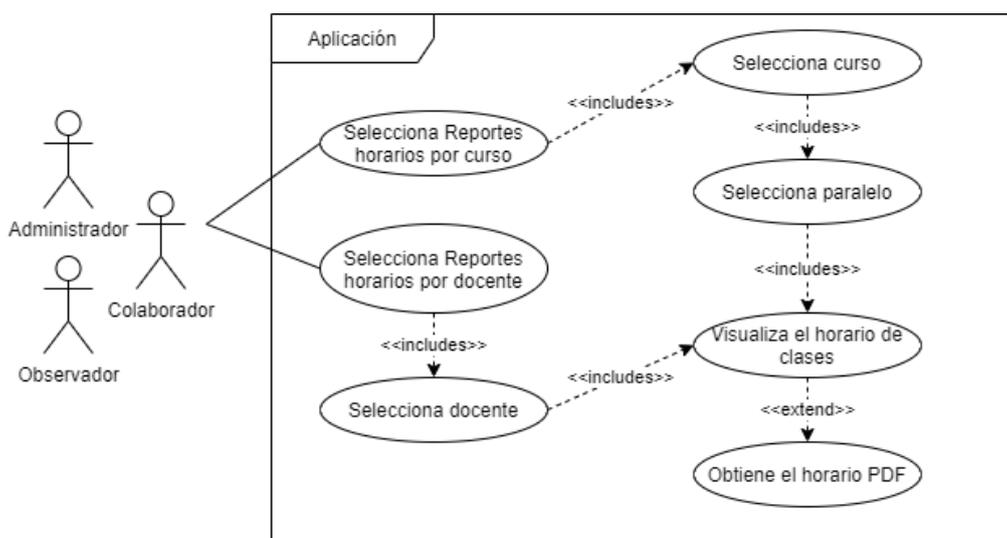


Ilustración 16 Caso de uso Solicitud de horarios

Caso de uso:	Gestión de horarios
Área:	Administración
Actores:	Administrador, colaborador, observador
Descripción:	Permite al administrador, colaborador y observador la visualización y descarga de horarios de clases para cursos y docentes
Evento Desencadenador:	El usuario ingresa al sistema y se dirige al módulo de reportes en la pestaña de horarios.
Pasos	
<ol style="list-style-type: none"> 1. El usuario inicia sesión en la aplicación web. 2. El usuario selecciona el periodo lectivo para iniciar su espacio de trabajo. 3. El usuario se dirige a la sección Reportes del menú lateral. 	
<ol style="list-style-type: none"> 4.a. El usuario selecciona la pestaña “Horarios Cursos”. 5.a. El usuario selecciona un curso del listado. 6.a. El usuario selecciona un paralelo del listado. 7.a. El usuario selecciona el botón “Buscar horario”. 	
<ol style="list-style-type: none"> 4.b. El usuario selecciona la pestaña “Horarios Docente”. 5.b. El usuario selecciona un docente del listado. 6.b. El usuario selecciona el botón “Buscar horario”. 	
<ol style="list-style-type: none"> 8. El usuario visualiza el horario 9. El usuario descarga el horario obtenido 	
Pre-condiciones:	Sesión como Administrador, colaborador u observador. Haber seleccionado un periodo lectivo. Tener registros de horarios de clases.
Post-condiciones:	N/A.
Requerimientos cumplidos:	RF-34: Obtener reportes de horarios por curso. RF-35: Obtener reportes de horarios por docente.

Tabla 9 Caso de uso Solicitud de horarios

2.5.10. Diagrama de caso de uso Obtener reportes

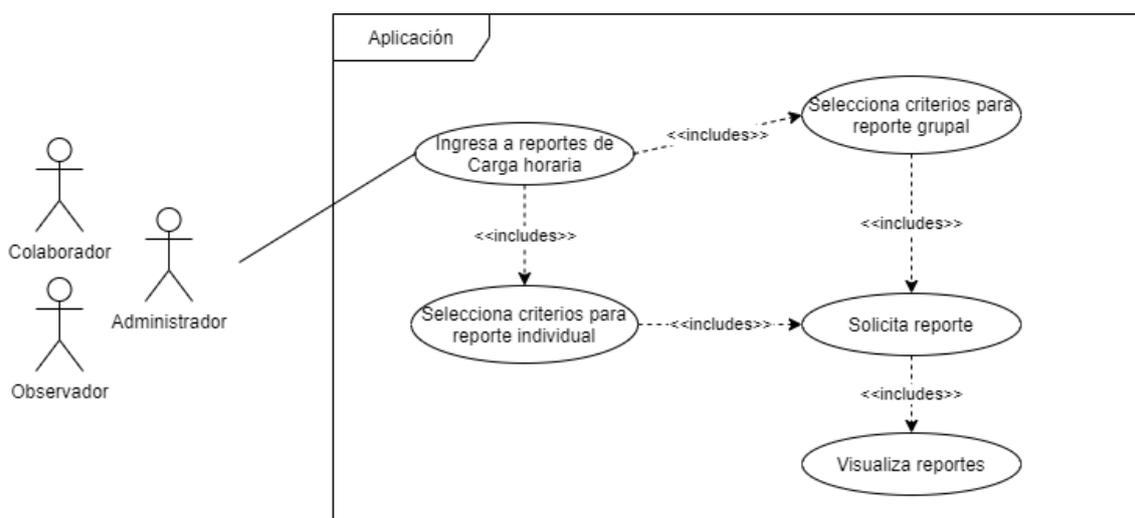


Ilustración 17 Caso de uso Obtener Reportes

Caso de uso:	Obtener reportes
Área:	Administración
Actores:	Administrador, colaborador, observador
Descripción:	Permite al administrador, colaborador y observador la visualización de reportes relacionados a las cargas horarias.
Evento Desencadenador:	El usuario ingresa al sistema y se dirige al módulo de reportes en la pestaña de Carga Horaria.
Pasos	
<ol style="list-style-type: none"> 1. El usuario inicia sesión en la aplicación web. 2. El usuario selecciona el periodo lectivo para iniciar su espacio de trabajo. 3. El usuario se dirige a la sección Reportes del menú lateral. 4. El usuario selecciona la pestaña “Carga horaria”. <ol style="list-style-type: none"> 5.a. El usuario selecciona un curso del listado. 6.a. El usuario selecciona un paralelo del listado. 7.a. El usuario selecciona el botón “Buscar horario”. 4.b. El usuario selecciona la pestaña “Horarios Docente”. <ol style="list-style-type: none"> 5.b. El usuario selecciona un docente del listado. 6.b. El usuario selecciona el botón “Buscar horario”. 8. El usuario visualiza el horario 9. El usuario descarga el horario obtenido 	
Pre-condiciones:	Sesión como Administrador, colaborador u observador. Haber seleccionado un periodo lectivo. Tener registros de horarios de clases.
Post-condiciones:	N/A.
Requerimientos cumplidos:	RF-34: Obtener reportes de horarios por curso. RF-35: Obtener reportes de horarios por docente.

Tabla 10 Caso de uso Obtener Reportes

2.5.11. Diagrama de actividades

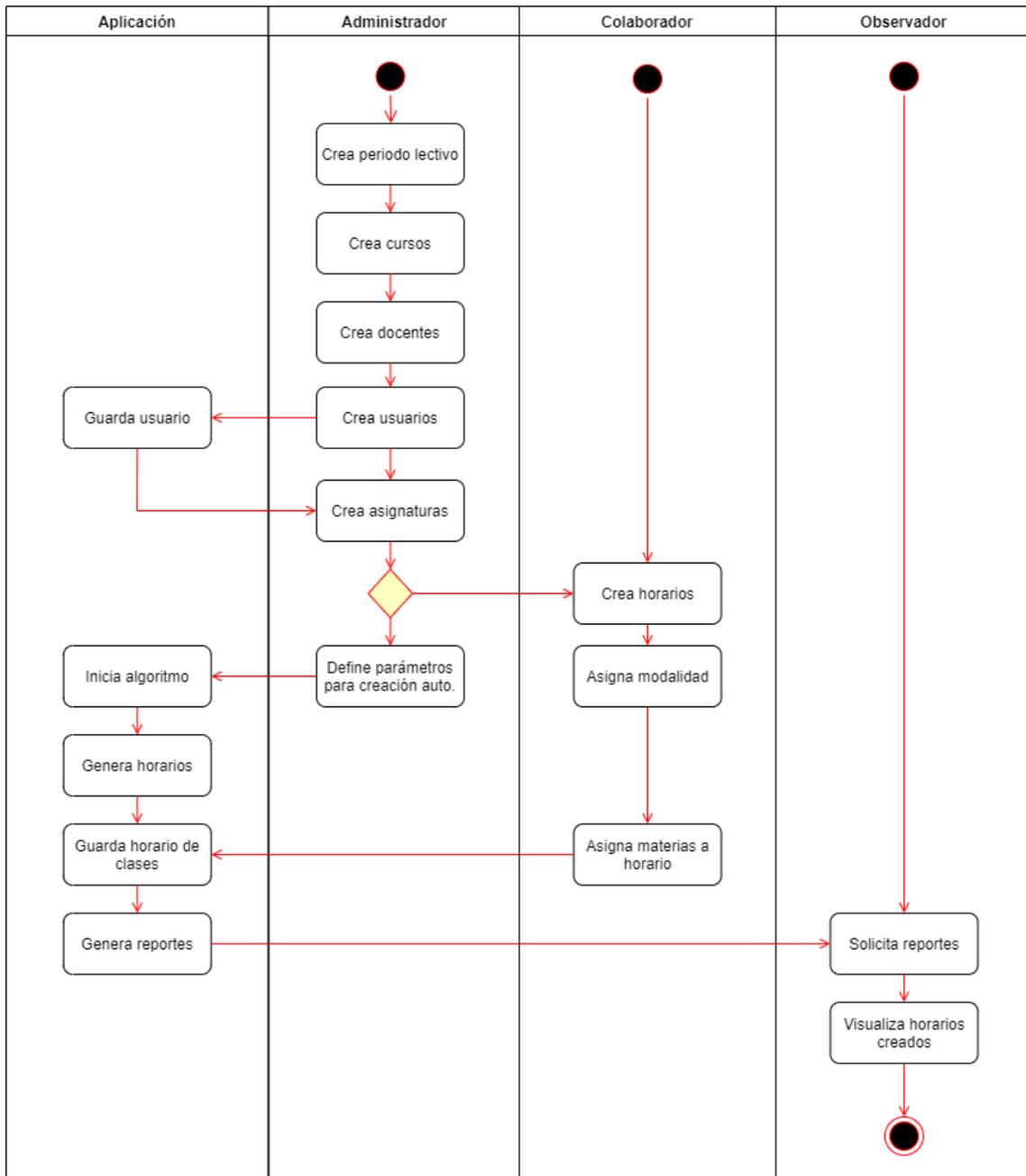


Ilustración 18 Diagrama de actividades de la aplicación

2.5.12. Modelo de programación lineal

Para automatizar el proceso de creación de horarios se ha seleccionado trabajar con algoritmos de optimización pertenecientes al área de investigación de operaciones en analítica de negocios.

Antes de plantear el modelo de programación lineal, debemos definir cuáles son nuestras variables por considerar, estas variables están sujetas a las características que la institución plantea para su problemática, conocidos como parámetros del negocio:

Parámetros del negocio		
Variable	Nomenclatura	Definición
A	$A = \{1, \dots, AT\}$	Conjunto de asignaturas.
C	$C = \{1, \dots, CT\}$	Conjunto de cursos-paralelos.
H	$H = \{1, \dots, HT\}$	Conjunto de horarios disponibles en la semana.
D	$D = \{1, \dots, DT\}$	Conjunto de días en la semana escolar.
P	$P = \{1, \dots, PT\}$	Conjunto de profesores asignados.
AT	$AT = \mathbb{Z}^+$	Número de asignatura totales.
CT	$CT = \mathbb{Z}^+$	Número de cursos-paralelos totales.
HT	$HT = \mathbb{Z}^+$	Número de horarios totales en la semana.
DT	$DT = \mathbb{Z}^+$	Número de días totales en la semana.
PT	$PT = \mathbb{Z}^+$	Número total de profesores disponibles
Ta	T_a donde $a \in A$	Número de horas semanales en una asignatura.
Td	T_d donde $d \in D$	Número de horas máximas que se puede impartir una asignatura en un día.
Hd	$Hd \subseteq H$	Subconjunto de horarios en un día de la semana.
Had	$Had \subseteq H$	Subconjunto de horarios por cada asignatura en un día de la semana.
Xach	$Xach = \{0, 1\}$	Conjunto de asignaturas en un curso y paralelo específico dentro de un horario de la semana.

Tabla 11 Parámetros del método matemático

Función objetivo

Una vez definido los parámetros del negocio se presenta el modelo de programación lineal para solucionar la asignación automática de horarios de clases. Se establece como función objetivo la maximización de cada asignación dentro de cada horario disponible, por asignatura en cada curso – paralelo.

$$\text{Max } Z = \sum_{a \in A} \sum_{p \in P} \sum_{c \in C} \sum_{h \in H} X_{apch}$$

Ecuación 1 Función objetivo de la solución

Restricciones

Es necesario que la función objetivo esté sujeta a un conjunto de restricciones que permita garantizar el cumplimiento de cada requerimiento dentro de la asignación de horarios de clases. Las restricciones del modelo son presentadas en conjuntos por cada requerimiento a cumplir dentro del modelo de programación lineal:

1. El primer conjunto de restricciones permite que cada asignatura dentro de cada curso-paralelo sea asignada un número determinado de horas por semana.

$$\sum_{h \in H} X_{apch} = T_a \quad \forall a \in A, c \in C$$

Ecuación 2 Conjunto de restricciones 1

2. El segundo conjunto de restricciones evita que existan cruces de horarios por asignatura, es decir, evita que se asigne una misma asignatura en un mismo horario en cursos diferentes.

$$\sum_{c \in C} X_{apch} \leq 1 \quad \forall a \in A, h \in H$$

Ecuación 3 Conjunto de restricciones 2

3. El tercer conjunto de restricciones permite que una asignatura sea asignada un número máximo de horas dentro de cada día de la semana.

$$\sum_{h \in H_d} X_{apch} \leq T_d \quad \forall a \in A, c \in C, d \in D$$

Ecuación 4 Conjunto de restricciones 3

4. El cuarto conjunto de restricciones evita que existan cruces de horarios por docente, es decir, evita que se asigne a un mismo curso, en un mismo horario varias asignaturas con el mismo docente.

$$\sum_{c \in C} X_{apch} \leq 1 \quad \forall p \in P, h \in H$$

Ecuación 5 Conjunto de restricciones 4

5. El quinto conjunto de restricciones indica asignación en un horario de clases sea una variable binaria, es decir: 1 si una asignatura es asignada a un curso dentro de un horario específico, caso contrario 0.

$$X_{apch} \in \{0,1\} \quad \forall a \in A, c \in C, h \in H, p \in P$$

Ecuación 6 Conjunto de restricciones 5

La solución e interpretación del programa lineal entero es desarrollada en el servicio web de la aplicación, del lado del servidor mediante la librería GLPK (GNU Linear programming kit).

Los parámetros de los horarios de clases son enviados en formato JSON, son interpretados y transformados en sus respectivos conjuntos de variables y restricciones según las ecuaciones mostradas anteriormente.

El programa lineal es resuelto mediante relajación lineal usando GLPK y posteriormente su resolución es restringida al conjunto de variables enteras positivas y los conjuntos de restricciones. Una vez encontrada la solución óptima, se interpretan las variables y se transforman nuevamente en objeto JSON y son guardados en la base de datos.

2.5.13. Diagrama Documental

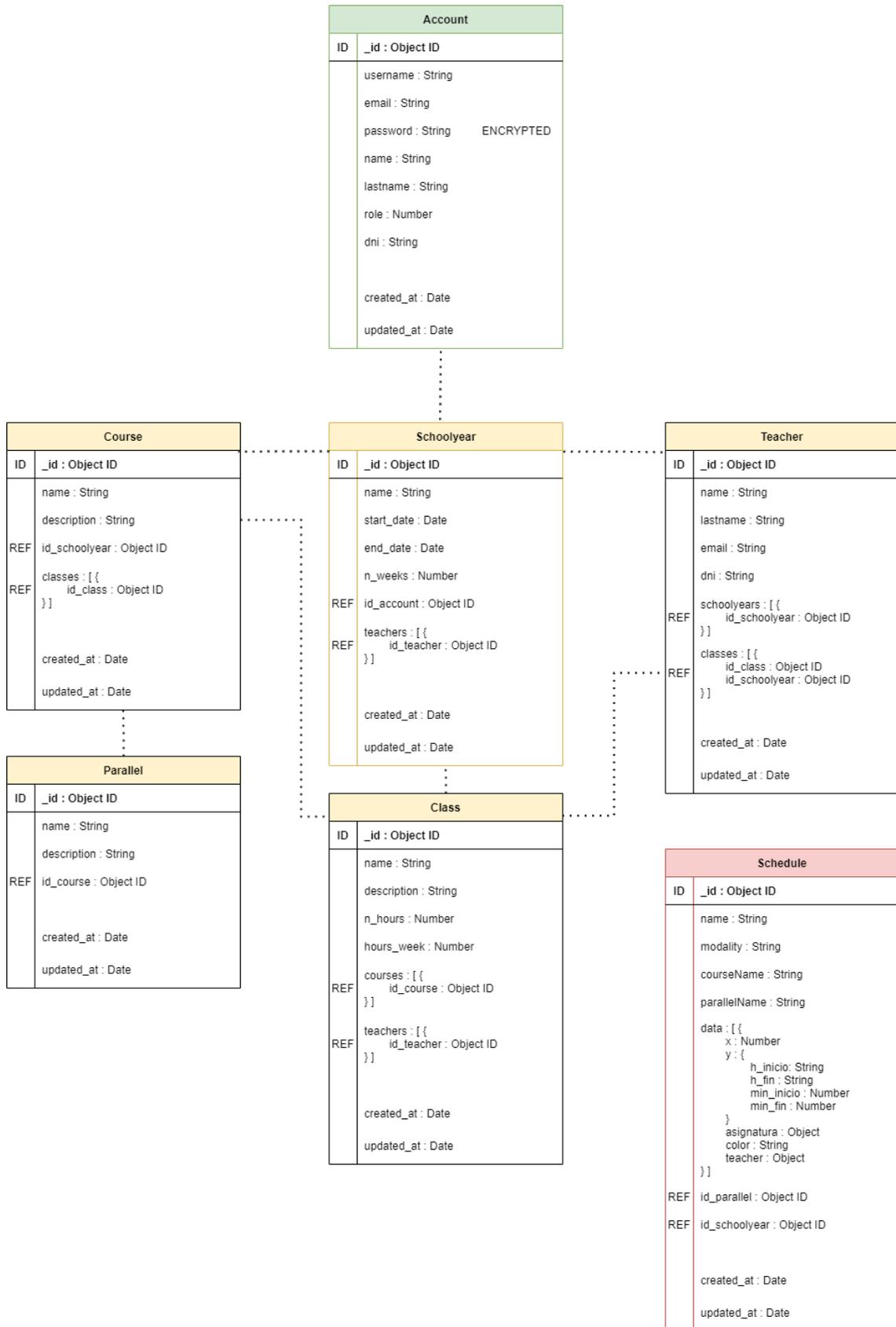


Ilustración 19 Diagrama de documentos de la aplicación

2.5.14. Diseño de interfaces

Interfaz principal



Ilustración 20 Intefaz principal de la aplicación

La interfaz principal del usuario se divide en 4 secciones: Sección de título (Superior), Sección de menú (Lateral Izquierdo), Sección de pie de página (Inferior) y área de trabajo. Esta interfaz actúa como contenedora de cada uno de los módulos del sistema, a excepción del módulo de autenticación. Contiene acceso a las acciones principales de configuración y cerras sesión, así como la navegación de menú hacia cada módulo del sistema.

Interfaz de listados

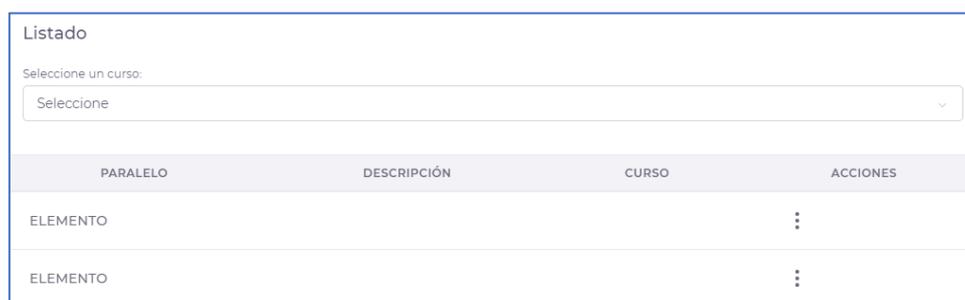


Ilustración 21 Interfaz de listados de la aplicación

La interfaz de listados aplica para todo el módulo de administración, y contiene un filtro que variará según la sección en la que el usuario se encuentre. Cuando se seleccione una opción en el filtro, automáticamente se desplegará el listado de elementos correspondientes. Cada elemento tendrá la opción individual de Editar o Eliminar según las necesidades del usuario.

Interfaz de ingreso de datos

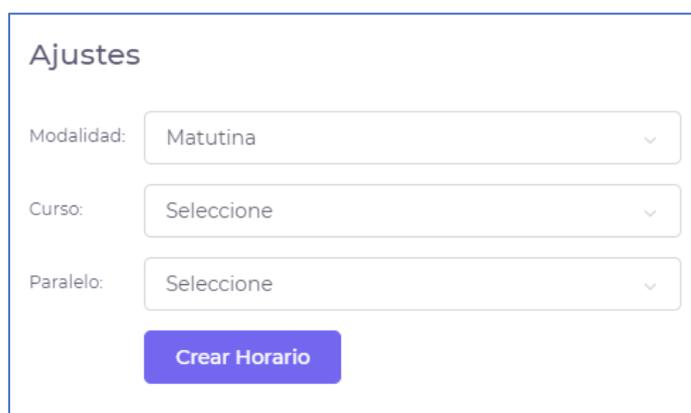


The screenshot shows a form titled "INGRESO DE DATOS". It contains several input fields: a text field with "Nombres" and a lock icon, a text area with "Descripción", a dropdown menu with "OPCION", and two "Select..." dropdown menus. There are also two numeric input fields with "0". At the bottom, there are two buttons: "Guardar" (blue) and "Reiniciar" (white).

Ilustración 22 Interfaz de ingreso de datos de la aplicación

La interfaz de ingreso de datos aplica para todo el módulo de administración, y posee los campos en formatos texto, numérico, fecha y opción múltiple. La opción reiniciar permite que todos los campos del formulario sean limpiados y la opción guardar envía los registros a la API y guarda los datos.

Interfaz de creación de horarios



The screenshot shows a form titled "Ajustes". It contains three dropdown menus: "Modalidad" with "Matutina", "Curso" with "Seleccione", and "Paralelo" with "Seleccione". At the bottom, there is a blue button labeled "Crear Horario".

Ilustración 23 Interfaz de creación de horarios

La interfaz de creación de horarios estará presente en el módulo de gestión, esta contiene las opciones en formato de combo para crear un nuevo horario de clases, los campos necesarios para esta interfaz son: Modalidad, curso y paralelo.

Interfaz de gestión de horarios



Crear horario

Asignatura:

Profesor:

Día:

Hora inicio: Hora fin:

Ilustración 24 Interfaz de gestión de horarios

La interfaz de gestión de horarios permitirá organizar los horarios de clases, añadiendo nuevas asignaturas al espacio de visualización de horarios. Aquí se seleccionarán los campos necesarios, todos en formato de opción múltiple, siguiendo los datos que hayan sido ingresados en el módulo de administración. La opción agregar permitirá añadir una nueva asignatura al horario activo, y la opción guardar permite almacenar el horario de clases en la base de datos.

Área de visualización de horarios



	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
13:00 - 13:20					
13:20 - 13:40		Física Juan Baquerizo	Ciencias Naturales Jose Baquerizo	Ciencias Naturales Jose Baquerizo	Sociales Jose Baquerizo
13:40 - 14:00					
14:00 - 14:20					
14:20 - 14:40					Ciencias Naturales Jose Baquerizo
14:40 - 15:00					

Ilustración 25 Interfaz de visualización de horarios

El área de visualización de horarios se compone de una rejilla, donde cada columna es representada por los días de la semana, y cada fila por el horario de inicio y fin de la sección. Dentro de las intersecciones entre filas y columnas se mostrarán las asignaturas creadas al momento de gestionar el horario. Cada asignatura es representada por un color y sus delimitadores, además de poseer la opción de modificarlas individualmente.

Interfaz de visualización de reportes

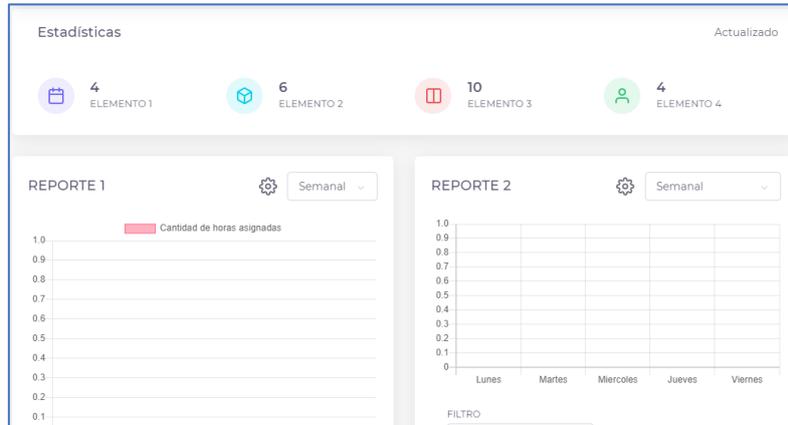


Ilustración 26 Interfaz de reportes

La interfaz de visualización de reportes permite mostrar cuadros estadísticos de los reportes que se hayan configurado, estos reportes manejan el formato de barras apiladas y poseen un conjunto de filtros predefinidos para su visualización personalizada.

2.5.15. Diccionario de datos

El diccionario de datos es una herramienta que nos permite identificar mediante metadatos, las características lógicas que va a usar un sistema de información. En este se define de manera rigurosa características tales como: nombre, tipo de contenido y la descripción de cada campo en una base de datos, de forma que funcione como un elemento común para entender el funcionamiento del sistema.

DICcionario DE DATOS #1				
Colección:	Schoolyear	Fecha de creación:	02-08-2021	
Descripción:	Guarda los datos de los periodos lectivos.			
Campo	Tipo de dato	Requerido	Referencia	Descripción
_id	ObjectId	✓		Identificador
name	String	✓		Nombre del periodo lectivo
start_date	Date	✓		Fecha de inicio del periodo lectivo
end_date	Date	✓		Fecha de fin del periodo lectivo
id_account	ObjectId	✓	✓	Identificador de la cuenta asociada
teachers	Array	✓	✓	Guarda las referencias de los profesores asociados
created_at	Date			Fecha en la que se creó el documento
updated_at	Date			Fecha en la que modificó el documento por última vez

Tabla 12 Diccionario de datos - Schoolyear

DICcionario DE DATOS #2				
Colección:	Course	Fecha de creación:	02-08-2021	
Descripción:	Guarda los datos de cada curso con su información			
Campo	Tipo de dato	Requerido	Referencia	Descripción
_id	ObjectId	✓		Identificador
name	String	✓		Nombre del curso
description	String			Descripción y detalle del curso
id_schoolyear	ObjectId	✓	✓	Identificador
classes	Array	✓	✓	Guarda las referencias de las asignaturas asociadas
created_at	Date			Fecha en la que se creó el documento
updated_at	Date			Fecha en la que modificó el documento por última vez

Tabla 13 Diccionario de datos – Course

DICCIONARIO DE DATOS #3				
Colección:	Parallel	Fecha de creación:		02-08-2021
Descripción:	Guarda los datos de cada paralelo con su información			
Campo	Tipo de dato	Requerido	Referencia	Descripción
_id	ObjectId	✓		Identificador
name	String	✓		Nombre del paralelo
description	String			Descripción y detalle del paralelo
id_course	ObjectId	✓	✓	Identificador
created_at	Date			Fecha en la que se creó el documento
updated_at	Date			Fecha en la que modificó el documento por última vez

Tabla 14 Diccionario de datos – Parallel

DICcionario DE DATOS #4				
Colección:	Class	Fecha de creación:		02-08-2021
Descripción:	Guarda los datos de cada asignatura con su información			
Campo	Tipo de dato	Requerido	Referencia	Descripción
_id	ObjectId	✓		Identificador
name	String	✓		Nombre de la asignatura
description	String			Descripción y detalle de la asignatura
hours_week	Number	✓		Cantidad de horas máximas semanales
courses	Array	✓	✓	Guarda las referencias de los cursos asociados
teachers	Array	✓	✓	Guarda las referencias de los profesores asociados
id_schoolyear	ObjectId	✓	✓	Identificador
created_at	Date			Fecha en la que se creó el documento
updated_at	Date			Fecha en la que modificó el documento por última vez

Tabla 15 Diccionario de datos – Class

DICcionario DE DATOS #5					
Colección:	Teacher	Fecha de creación:		02-08-2021	
Descripción:	Guarda los datos de cada docente con su información				
Campo	Tipo de dato	Requerido	Referencia	Único	Descripción
_id	ObjectId	✓		✓	Identificador
name	String	✓			Nombre del docente
lastname	String	✓			Apellidos del docente
email	String	✓		✓	Correo electrónico del docente
dni	String	✓		✓	Cédula del docente
classes	Array	✓	✓		Guarda las referencias de las asignaturas asociadas
schoolyears	Array	✓	✓		Guarda las referencias de los periodos lectivos asociados
created_at	Date				Fecha en la que se creó el documento
Updated_at	Date				Fecha en la que modificó el documento por última vez

Tabla 16 Diccionario de datos – Teacher

DICCIONARIO DE DATOS #6				
Colección:	Schedule	Fecha de creación:	02-08-2021	
Descripción:	Guarda los datos de cada horario con su información			
Campo	Tipo de dato	Requerido	Referencia	Descripción
_id	ObjectId	✓		Identificador
name	String	✓		Nombre de cada horario de clases
data	Array	✓		Contiene la información y arquitectura de cada horario de clases
modality	Number	✓		Tipo de modalidad del horario de clases
id_teacher	ObjectId		✓	Identificador
id_parallel	ObjectId		✓	Identificador
created_at	Date			Fecha en la que se creó el documento
updated_at	Date			Fecha en la que modificó el documento por última vez

Tabla 17 Diccionario de datos - Schedule

DICCIONARIO DE DATOS #7				
Colección:	Account	Fecha de creación:	02-08-2021	
Descripción:	Guarda los datos de cada cuenta con su información			
Campo	Tipo de dato	Requerido	Único	Descripción
_id	ObjectId	✓		Identificador
email	String	✓		Correo electrónico del usuario
password	String	✓		Contraseña encriptada del usuario
Name	String	✓		Nombres del usuario
lastname	String	✓		Apellidos del usuario
role	Number	✓		Rol del usuario en el sistema
dni	String	✓	✓	Cédula del usuario
created_at	Date			Fecha en la que se creó el documento
updated_at	Date			Fecha en la que modificó el documento por última vez

Tabla 18 Diccionario de datos - Account

2.6. Estudio de Factibilidad

En la presente sección se detallarán los estudios que corroboran la viabilidad del proyecto, se tomará en detalle el estudio de factibilidad Técnica y económica para llevar a cabo el desarrollo de la aplicación web.

2.6.1. Factibilidad técnica

Software

COMPONENTE	CANTIDAD
React JS	1
Next JS	1
MongoDB	1
Express JS	1
Node JS	1
NPM	1
React libraries	1
Node libraries	1
Diagrams.net	1
MongoDB compass	1
Postman	1
Visual Studio Code	1

Tabla 19 Factibilidad técnica: software

Hardware

COMPONENTE	CANTIDAD	DETALLE
Laptop Lenovo	1	Procesador Core i5 12GB RAM SSD 250GB

Tabla 20 Factibilidad técnica: Hardware

El estudio de factibilidad técnica nos muestra que el desarrollo de la aplicación web es viable gracias a que todas sus herramientas son libres y no suponen costo alguno además del equipo utilizado para el desarrollo del proyecto.

2.6.2. Factibilidad financiera

Hardware

COMPONENTE	DETALLE	CANT.	COSTO	SUBTOTAL
Laptop	Procesador Core i5 12GB RAM SSD 250GB	1	\$ 800.00	\$ 800.00
			TOTAL:	\$ 800.00

Tabla 21 Factibilidad financiera: Hardware

El costo de hardware para el desarrollo de este proyecto es de \$ 800.00. Se asume que el autor ya dispone del hardware requerido.

Software

COMPONENTE	CANT.	COSTO	SUBTOTAL
React JS	1	\$ 0.00	\$ 0.00
Next JS	1	\$ 0.00	\$ 0.00
MongoDB	1	\$ 0.00	\$ 0.00
Express JS	1	\$ 0.00	\$ 0.00
Node JS	1	\$ 0.00	\$ 0.00
NPM	1	\$ 0.00	\$ 0.00
React libraries	1	\$ 0.00	\$ 0.00
Node libraries	1	\$ 0.00	\$ 0.00
Diagrams.net	1	\$ 0.00	\$ 0.00
MongoDB compass	1	\$ 0.00	\$ 0.00
Postman	1	\$ 0.00	\$ 0.00
Visual Studio Code	1	\$ 0.00	\$ 0.00
		TOTAL:	\$ 0.00

Tabla 22 Factibilidad financiera: Software

Todas las herramientas de software necesarias para el desarrollo de este proyecto son de código libre, por lo tanto, no suponen ningún costo de uso.

Recursos humanos

COMPONENTE	CANT.	MESES	COSTO/MES	SUBTOTAL
Analista	1	1	\$ 800.00	\$ 800.00
Desarrollador front-end	1	2	\$ 1,000.00	\$ 2,000.00
Desarrollador back-end	1	2	\$ 1,000.00	\$ 2,000.00
Ingeniero en TI	1	1	\$ 1,000.00	\$ 1,000.00
Documentador	1	1	\$ 800,00	\$ 800.00
			TOTAL:	\$ 6,600.00

Tabla 23 Factibilidad financiera: RRHH

El costo de los recursos humanos para el desarrollo de este proyecto es de \$ 6,600.00. En este proyecto el autor asume las actividades de cada componente, por lo tanto, el costo total de este recurso es \$0.00.

Gastos varios

COMPONENTE	CANT.	MESES	COSTO/MES	SUBTOTAL
SERVICIOS BÁSICOS				
Energía eléctrica	1	6	\$ 20.00	\$ 120.00
Transporte	1	6	\$ 20.00	\$ 120.00
Internet	1	6	\$ 30.00	\$ 180.00
OTROS				
Suministros de oficina	1	1	\$ 30.00	\$ 30.00
			TOTAL:	\$ 450.00

Tabla 24 Factibilidad financiera: Gastos varios

El costo de los servicios básicos y demás para el desarrollo de este proyecto es de \$450.00. En este proyecto el autor asume los gastos de cada componente, por lo tanto, el costo total de este recurso es \$0.00.

Costo total

DESCRIPCIÓN	COSTO	
Hardware	\$ 800.00	
Software	\$ 0.00	
Recursos humanos	\$ 6,600.00	
Gastos varios	\$ 450.00	
TOTAL:		\$ 7,850.00

Tabla 25 Factibilidad financiera: Costo Total

El costo total del desarrollo de la aplicación es de \$7,850.00. Sin embargo, se reducen a \$0.00 los costos del recurso humano, hardware y gastos varios por ser asumidos por el autor. El costo total para el desarrollo de este proyecto en la escuela de educación básica “Lcda. Angélica Villón Lindao” es de \$0.00.

Los estudios de factibilidad técnica y financiera demuestran la viabilidad de desarrollo de este proyecto, representando ventajas futuras para la institución que lo adquiere mejorando sus procesos de forma eficiente y ayudándola a conseguir sus objetivos.

2.7. Resultados

2.7.1. Pruebas de funcionalidad

La calidad y funcionalidad es clave determinante para el éxito de una aplicación web, la ejecución de pruebas de funcionalidad permite garantizar el correcto funcionamiento de cada componente en la aplicación, y de ser necesario, corregir dichos errores a tiempo para su implementación oportuna.

Para verificar el correcto funcionamiento de la aplicación se ejecutaron las siguientes pruebas:

- Validación de ingreso de datos para garantizar la integridad de la información procesada.
- Verificación de la funcionalidad de los módulos de: sesión, administración, gestión y reportes.
- Verificación de cumplimiento de restricciones durante la gestión de horarios académicos.
- Verificación del correcto funcionamiento del modelo matemático para la generación de horarios automáticos.
- Comprobación de mejora en los tiempos de desarrollo de horarios antes y después del uso de la aplicación
- Verificación de la validez de los reportes generados en la aplicación.
- Verificación de la correcta obtención y descarga de horarios de clases.

PRUEBA DE FUNCIONALIDAD			
Nombre:	Acceso a la aplicación	#:	1
Objetivo:	Comprobar el correcto inicio de sesión de los usuarios según su rol asignado.		
Usuarios:	Administrador, Colaborador, Observador		
Condiciones:	El usuario debe utilizar sus credenciales exactamente como se encuentran en la base de datos		
Pasos			
<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Ingresar correo electrónico y contraseñas 3. Clic en el botón ingresar 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario digita los datos correctamente, y la aplicación lo redirige hacia la interfaz principal, dando acceso a los módulos según el rol que cumpla en la aplicación.	Exitoso	x	Fallido

Tabla 26 Prueba: Acceso a la aplicación

PRUEBA DE FUNCIONALIDAD			
Nombre:	Ingreso de periodos lectivos	#:	2
Objetivo:	Comprobar el correcto ingreso de nuevos periodos lectivos en la aplicación		
Usuarios:	Administrador		
Condiciones:	El usuario debe rellenar al menos los campos obligatorios. El formulario debe estar correcto.		
Pasos			
<ol style="list-style-type: none"> 1. Ingresar a la aplicación. 2. Dirigirse a la sección de periodos lectivos, en el menú lateral. 3. Clic en el botón (+). 4. Se muestra el formulario de periodos lectivos. 5. Ingresa los campos respectivos. 6. Da clic en el botón guardar. 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario digita los datos correctamente, y los datos del nuevo periodo lectivo son agregados a la base de datos, como consecuencia el usuario puede ver los nuevos datos en la tabla de visualización de registros.	Exitoso	x	Fallido

Tabla 27 Prueba: Ingreso de periodos lectivos

PRUEBA DE FUNCIONALIDAD			
Nombre:	Ingreso de cursos	#:	3
Objetivo:	Comprobar el correcto ingreso de nuevos cursos en un periodo lectivo seleccionado.		
Usuarios:	Administrador		
Condiciones:	El usuario debe rellenar al menos los campos obligatorios. El formulario debe estar correcto.		
Pasos			
<ol style="list-style-type: none"> 1. Ingresar a la aplicación. 2. Dirigirse a la sección de “Administración” pestaña “Cursos” en el menú lateral. 4. Se muestra el formulario de ingreso de cursos. 5. Ingresa los campos respectivos. 6. Da clic en el botón guardar. 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario digita los datos correctamente, y los datos del nuevo curso son agregados a la base de datos, como consecuencia el usuario puede ver los nuevos datos en la tabla de visualización de registros.	Exitoso	x	Fallido

Tabla 28 Prueba: Ingreso de cursos

PRUEBA DE FUNCIONALIDAD			
Nombre:	Ingreso de paralelos	#:	4
Objetivo:	Comprobar el correcto ingreso de nuevos paralelos en un curso seleccionado.		
Usuarios:	Administrador		
Condiciones:	El usuario debe rellenar al menos los campos obligatorios. El formulario debe estar correcto.		
Pasos			
<ol style="list-style-type: none"> 1. Ingresar a la aplicación. 2. Dirigirse a la sección de “Administración” pestaña “Paralelos” en el menú lateral. 4. Se muestra el formulario de ingreso de paralelos. 5. Ingresar los campos respectivos. 6. Selecciona el curso al que se asignará el paralelo. 7. Da clic en el botón guardar. 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario digita los datos correctamente, y los datos del nuevo paralelo son agregados a la base de datos, como consecuencia el usuario puede ver los nuevos datos en la tabla de visualización de registros.	Exitoso	x	Fallido

Tabla 29 Prueba: Ingreso de paralelos

PRUEBA DE FUNCIONALIDAD			
Nombre:	Ingreso de profesores	#:	5
Objetivo:	Comprobar el correcto ingreso de nuevos profesores en un periodo lectivo seleccionado.		
Usuarios:	Administrador		
Condiciones:	El usuario debe rellenar al menos los campos obligatorios. El formulario debe estar correcto.		
Pasos			
<ol style="list-style-type: none"> 1. Ingresar a la aplicación. 2. Dirigirse a la sección de “Administración” pestaña “Profesores” en el menú lateral. 4. Se muestra el formulario de ingreso de profesores. 5. Ingresar los datos del docente. 6. Dar clic en el botón guardar. 7. La aplicación valida los datos antes de ingresarlos a la base de datos 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario digita los datos correctamente, y los datos del nuevo docente son agregados a la base de datos, como consecuencia el usuario puede ver los nuevos datos en la tabla de visualización de registros.	Exitoso	x	Fallido

Tabla 30 Prueba: Ingreso de profesores

PRUEBA DE FUNCIONALIDAD			
Nombre:	Ingreso de asignaturas	#:	6
Objetivo:	Comprobar el correcto ingreso de nuevas asignaturas en un periodo lectivo seleccionado.		
Usuarios:	Administrador		
Condiciones:	El usuario debe rellenar al menos los campos obligatorios. El formulario debe estar correcto.		
Pasos			
<ol style="list-style-type: none"> 1. Ingresar a la aplicación. 2. Dirigirse a la sección de “Administración” pestaña “Asignaturas” en el menú lat. 4. Se muestra el formulario de ingreso de asignaturas. 5. Ingresa los datos básicos de la asignatura. 6. Define cantidad de horas semanales a impartirse. 7. Selecciona uno o más cursos donde se impartirá la asignatura. 8. Selecciona uno o más docentes a cargo de la asignatura. 6. Da clic en el botón guardar. 7. La aplicación valida los datos antes de ingresarlos a la base de datos 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario digita los datos correctamente, y los datos de la nueva asignatura son agregados a la base de datos, como consecuencia el usuario puede ver los nuevos datos en la tabla de visualización de registros.	Exitoso	x	Fallido

Tabla 31 Prueba: Ingreso de asignaturas

PRUEBA DE FUNCIONALIDAD			
Nombre:	Creación de horarios de clases	#:	7
Objetivo:	Comprobar el correcto ingreso y gestión de los horarios de clases en un periodo lectivo seleccionado.		
Usuarios:	Administrador, Colaborador		
Condiciones:	El usuario debe seleccionar los campos necesarios para añadir información al horario de clases. El formulario debe estar correcto.		
Pasos			
<ol style="list-style-type: none"> 1. Ingresar a la aplicación. 2. Dirigirse a la sección de “Gestión” pestaña “Horarios” en el menú lateral. 4. Se muestra el formulario de ajustes del horario de clase. 5. Selecciona la modalidad, el curso y paralelo donde se creará el horario. 6. Da clic en el botón “Crear horario” 7. Se dirige al formulario de creación de horarios. 8. Selecciona la asignatura y el profesor correspondientes. 9. Selecciona el día donde se añadirá el campo 10. Selecciona las horas de inicio y fin del campo 11. Da clic en el botón agregar. 12. Repite los pasos los pasos [7 - 10] según sea necesario. 13. Da clic en el botón guardar. 14. La aplicación valida los datos antes de ingresarlos a la base de datos. 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario selecciona los datos correctamente, los horarios de clases son llenados con cada campo de asignatura, respetando que no existan conflictos entre docentes y cruce de horas. Una vez finaliza la creación del horario, este es guardado exitosamente en la base de datos	Exitoso	x	Fallido

Tabla 31 Prueba: Creación de horarios de clases

PRUEBA DE FUNCIONALIDAD			
Nombre:	Creación de horarios de clases automáticos	#:	8
Objetivo:	Comprobar la correcta creación de los horarios de clases automáticos mediante la ejecución del algoritmo.		
Usuarios:	Administrador, Colaborador		
Condiciones:	El usuario debe seleccionar los parámetros necesarios para iniciar la ejecución del algoritmo.		
Pasos			
<ol style="list-style-type: none"> 1. Ingresar a la aplicación. 2. Dirigirse a la sección de “Gestión” pestaña “Horarios” en el menú lateral. 3. Seleccionar la opción de creación automática de horarios de clases. 4. Seleccionar los parámetros para la creación de horarios. 5. Ejecutar la creación de horarios automáticos. 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario selecciona los datos correctamente, e inicia el algoritmo de creación de horarios automáticos. El algoritmo respeta las restricciones necesarias, obtiene una solución óptima del horario de clases para cada curso y paralelo. Los resultados son obtenidos y guardados en la base de datos en menos de 1 minuto.	Exitoso	x	Fallido

Tabla 32 Prueba: Creación de horarios de clases automáticos

PRUEBA DE FUNCIONALIDAD			
Nombre:	Obtención de reportes	#:	9
Objetivo:	Comprobar la funcionalidad de módulo de reportes grupales e individuales para el análisis de cargas horarias		
Usuarios:	Administrador, Colaborador, observador		
Condiciones:	El usuario debe seleccionar los parámetros necesarios para obtener un reporte.		
Pasos			
<ol style="list-style-type: none"> 1. Ingresar a la aplicación. 2. Dirigirse a la sección de “Reportes” pestaña “Cargas horarias” en el menú lateral. 3. Se muestran los formularios de obtención de reportes grupales e individuales. 4. El usuario selecciona el filtro necesario para cada reporte. 5. El usuario visualiza los reportes actualizados del sistema. 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario selecciona los filtros establecidos, y la aplicación muestra los gráficos estadísticos de cada reporte solicitado.	Exitoso	x	Fallido

Tabla 33 Prueba: Obtención de reportes

PRUEBA DE FUNCIONALIDAD			
Nombre:	Obtención de horarios de clases	#:	10
Objetivo:	Comprobar la funcionalidad de módulo de obtención y descarga de horarios de clases para cursos y docentes.		
Usuarios:	Administrador, Colaborador, observador		
Condiciones:	El usuario debe seleccionar los parámetros necesarios para obtener un horario de clases.		
Pasos			
<ol style="list-style-type: none"> 1. Ingresar a la aplicación. 2. Dirigirse a la sección de “Reportes” pestaña “Horarios Cursos” en el menú lateral. 2. Dirigirse a la sección de “Reportes” pestaña “Horarios Docente” en el menú lat. 3. Se muestran los formularios de obtención de horarios según se haya seleccionado. 4. El usuario selecciona el curso/paralelo o el docente a obtener respectivamente. 5. El usuario visualiza los horarios de clases guardados. 6. El usuario hace clic en el botón descargar en la parte superior del horario. 7. El usuario obtiene el horario de clases en formato .PDF listo para su descarga. 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario selecciona los parámetros necesarios y el sistema obtiene los horarios de clases en su vista para cursos o para docentes respectivamente. Adicionalmente se genera un documento con dicho horario listo para su descarga	Exitoso	x	Fallido

Tabla 34 Prueba: Obtención de horarios de clases

2.7.2. Resultados

Una vez determinada la problemática en la institución de educación básica “Lcda. Angélica Villón Lindao” se realizó la definición de los requerimientos y el planteamiento de la solución. Se realizaron las pruebas necesarias para garantizar el funcionamiento de la aplicación, obteniendo los siguientes resultados:

ANÁLISIS DE RESULTADOS	
DIFICULTADES	SOLUCIONES
Ausencia de un repositorio que permita guardar la información referente a la creación de horarios de clases	Creación de una base de datos NoSQL, que a través de la aplicación web puede obtener todos los datos ingresados para la gestión de horarios.
Dificultad para organizar los horarios de clases para cada: curso, paralelo y docente dentro de la institución.	Creación de un gestor de horarios que permita a los administradores distribuir las asignaturas de forma ágil y sencilla en cada curso la institución.
La creación de horarios de clases se realiza de forma espontánea sin un conjunto de pasos totalmente definidos.	Cada paso del proceso de gestión de horarios se encuentra en ordenadamente para su estructuración, dentro de cada módulo de la aplicación web.
Dificultad para detectar cruces de horarios entre cursos y docentes.	La aplicación web valida internamente que todas las restricciones de un horario de clases se cumplan, detectándolas y previniéndolas inmediatamente.
El proceso de creación de horarios de clases podría extenderse hasta 1 semana en desarrollo y correcciones.	El gestor de horarios permite desarrollarlos hasta en 1 minuto si se usa la creación automática y hasta un total aproximado de 60 minutos incluyendo todo el proceso.
La gestión de horarios se realiza de forma manual sin el uso de herramientas automáticas.	La aplicación web utiliza un algoritmo de creación de horarios de forma automática basados en un modelo de programación lineal en un máximo de 1 minuto.

Tabla 35 Análisis de resultados

2.7.3. Resultados finales

La problemática presente en la institución permitió el desarrollo de una solución tecnológica, basada en herramientas de código abierto y herramientas de optimización de operaciones obteniendo los siguientes resultados:

1. Se recopiló la información necesaria en el proceso de gestión de horarios mediante la aplicación de técnicas de recolección de información, permitiendo levantar los requerimientos y sistematizar cada paso con las restricciones necesarias dentro de cada módulo la aplicación web.
2. La aplicación web permite que los usuarios finales logren planificar, crear, automatizar y gestionar los horarios de clases mediante interfaces limpias y que mejoren el flujo de trabajo durante todo el proceso.
3. La aplicación web permite el registro de periodos lectivos, cursos, paralelos y docentes para asignarlos de forma manual o automática dentro de cada horario de clases, además de mantenerlos disponibles para su gestión oportuna conforme se el usuario lo requiera.
4. La aplicación de métodos matemáticos de optimización permite que los horarios de clases sean creados de forma automática, obteniendo una solución óptima que respete las cargas horarias y evite conflictos dentro de las asignaturas, cursos o docentes en base las restricciones propuestas durante el planteamiento del algoritmo.
5. La solución permite también una reducción significativa del tiempo empleado durante el proceso de creación y gestión de horarios de clases, aportando a los administradores de la institución optimizar sus procesos y agilizar el inicio de las labores académicas para los estudiantes.

2.8. Conclusiones

- En este proyecto se logró desarrollar una aplicación web para la gestión y planificación de horarios de clases en la Escuela "Lic. Angélica Villón Lindao" gracias al uso del stack de tecnologías MERN y la aplicación de métodos matemáticos para optimizar la asignación de recursos en la institución.
- Se recopiló con éxito la información necesaria referente al proceso que llevan los administradores para la planificación, estructuración y gestión de horarios de clases mediante el uso de técnicas cualitativas de recolección de información. Gracias a esto se lograron levantar los requerimientos de la aplicación web y se establecieron claramente los parámetros y restricciones que emplea la creación de los horarios de clases que posteriormente fueron utilizados para plantear el modelo matemático en cuestión.
- La creación de una base de datos orientada a documentos otorgó flexibilidad para estructurar los objetos de Horarios de clases de forma satisfactoria, apegándose a un esquema real y soportando las operaciones realizadas por el sistema de forma independiente y concurrente.
- Poseer los parámetros claros para la estructuración de los horarios de clases permitió el planteamiento de un algoritmo de optimización basado en programación lineal entera, donde sus resultados fueron una solución óptima apegada a todas las restricciones establecidas en cuestión de segundos. La interpretación de estas variables generadas fue pieza clave para estructurar los horarios de clases resultantes de forma que pudieran ser almacenados en la base de datos.
- Se realizaron las pruebas de la aplicación web y se demostró la agilidad con la que se gestionan los horarios de clases. Esta solución permite reducir el tiempo que emplean los administradores en crear horarios de clases de forma significativa y segura.
- El proceso de gestión de horarios de clases era susceptible a errores y conflictos que no se detectaban fácilmente, en algunos casos, retrasando las actividades académicas por las correcciones dentro de las primeras semanas de clases. Con la solución propuesta, estos errores fueron mitigados y el tiempo total empleado para gestionar los horarios de clases pudo ser reducido a 60 minutos o menos, significando una contundente reducción del esfuerzo empleado.

2.9. Recomendaciones

- Considerar la arquitectura con la que se crea esta solución para futuras actualizaciones, el código obedece a un patrón de componentes y con un estado global sobre el que se rige el flujo de datos dentro de la aplicación.
- Usar las versiones con las que fue desplegada la aplicación para prevenir problemas de compatibilidad en sus librerías y paquetes debido a que son de código abierto y están en constante actualización.
- Otorgar las capacitaciones a los usuarios de la aplicación, de forma que se les facilite su interacción con el sistema y obtengan los resultados óptimos en el menor tiempo posible.
- La base de datos creada obedece a la estructura BSON propia de Mongo DB y permite crear registros en forma de documentos de forma flexible, apegándose de forma beneficiosa a estructuras de datos complejas como los horarios de clases. La estructura de base de datos debe estar totalmente validada para garantizar la integridad de la base de datos.

2.10. Bibliografía

- [1] O. E. Fuentes Sordo, «La organización escolar. Fundamentos e importancia para la dirección en la educación.,» Varona, La Habana - Cuba, 2015.
- [2] R. Hernandez, J. Miranda y P. Rey, Programación de Horarios de Clases y Asignación de Salas para la Facultad de Ingeniería de la Universidad Diego Portales Mediante un Enfoque de Programación Entera, Santiago - Chile: Revista Ingeniería de Sistemas, 2008.
- [3] I. Carballo Rodríguez, «Desarrollo de una aplicación para la gestión de horarios académicos,» Universidad de Vigo, Vigo - España, 2020.
- [4] H. O. Campoverde Ramos, «SISTEMA DE GESTIÓN DE HORARIOS ACADÉMICOS PARA LA UNIVERSIDAD CENTRAL DEL ECUADOR,» Quito, 2015.
- [5] A. L. Barzola de la Cruz, «Implementación de un Sistema de Gestión de Horarios de Clases para los docentes del colegio Península de Santa Elena,» Santa Elena, 2016.
- [6] MongoDB, «MongoDB - MERN Stack,» MongoDB, [En línea]. Available: <https://www.mongodb.com/mern-stack>. [Último acceso: 17 Diciembre 2020].
- [7] H. M. Abdullah y A. M. Zeki, Frontend and Backend Web Technologies in Social Networking Sites: Facebook as an Example, California: IEEE , 2014.
- [8] ReactJS ORG, «React,» ReactJS, [En línea]. Available: <https://es.reactjs.org/>. [Último acceso: 17 Diciembre 2020].
- [9] NextJS, «NextJS,» Vercel, [En línea]. Available: <https://nextjs.org/>. [Último acceso: 17 Diciembre 2020].
- [10] Node, «NodeJS,» OpenJs foundation, [En línea]. Available: <https://nodejs.org/es/>. [Último acceso: 17 Diciembre 2020].

- [11 Express, «ExpressJS,» OpenJS Foundation, [En línea]. Available:
] <https://expressjs.com/es/>. [Último acceso: 17 Diciembre 2020].
- [12 MongoDB, «MongoDB,» MongoDB, [En línea]. Available:
] <https://www.mongodb.com/1>. [Último acceso: 17 Diciembre 2020].
- [13 FACSISTEL, «Líneas de Investigación,» FACSISTEL - UPSE, [En línea].
] Available:
http://facsistel.upse.edu.ec/index.php?option=com_content&view=article&id=58&Itemid=463. [Último acceso: 17 Diciembre 2020].
- [14 R. Katz y F. Callorda, «Digitalización de procesos productivos en América Latina,»
] SSRN, México, 2016.
- [15 Secretaría Técnica Planifica Ecuador , «Plan Nacional de Desarrollo 2017 – 2021
] Toda una Vida,» [En línea]. Available: <https://www.planificacion.gob.ec/plan-nacional-de-desarrollo-2017-2021-toda-una-vida/>. [Último acceso: 26 Diciembre 2020].
- [16 N. Mohammad, Metodología de la investigación, Coyoacán Ciudad de México:
] Noriega Editorial, 2005.
- [17 P. Bernal, La investigación en ciencias sociales, Bogotá: Universidad Piloto de
] Colombia, 2017.
- [18 J. Zumba Gamboa, «Evolución de las Metodologías y Modelos utilizados en el
] Desarrollo de Software,» *INNOVA Research Journal*, vol. III, nº 10, pp. 20-33, 2018.
- [19 G. Maida y P. Julián, Metodologías de desarrollo de software, Buenos Aires:
] PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA , 2015.
- [20 Ministerio de educación, «Educación General Básica,» Gobierno de la República de
] Ecuador, [En línea]. Available:
https://educacion.gob.ec/educacion_general_basica/. [Último acceso: 24 Julio 2021].

- [21 Ministerio de educación, «ACUERDO Nro. MINEDUC-ME-2016-00020-A,»
] Gobierno de la República de Ecuador, Quito, 2016.
- [22 D. Gourley y B. Totty, HTTP: The definitve guide, California: O'Reilly Media, Inc.,
] 2003.
- [23 S. Luján Mora, Programación de aplicaciones web: historia, principios básicos y
] clientes web, Alicante: Editorial Club Universitario, 2002.
- [24 J. Ofoeda, R. Boateng y J. Effah, Application Programming Interface (API)
] Research: A Review of the Past to Inform the Future, Ghana: University of
Professional Studies Edition, 2019.
- [25 React JS, «Reactjs.org,» Facebook Open Source, 2021. [En línea]. Available:
] <https://reactjs.org/blog/2013/06/05/why-react.html>. [Último acceso: 14 17 2021].
- [26 E. Scott, SPA Design and Architecture: Understanding Single-page Web
] Applications, Manning Publications, 2015.
- [27 M. Marqués Andrés, Bases de Datos, Castellón de la Plana: Universitat Jaume I,
] 2011.
- [28 P. Andliger, «DB - Engines,» Solid IT, 21 Noviembre 2013. [En línea]. Available:
] https://db-engines.com/en/blog_post/23. [Último acceso: 4 Julio 2021].
- [29 J. Pokorny, NoSQL databases: a step to database scalability in web environment,
] Emerald Group Publishing Limited, 2013.
- [30 Mozilla, «MDN web docs,» Mozilla, [En línea]. Available:
] <https://developer.mozilla.org/es/docs/Web/JavaScript>. [Último acceso: 17
Diciembre 2020].
- [31 Mozilla, «MDN web docs,» Mozilla, [En línea]. Available:
] https://developer.mozilla.org/es/docs/Web/JavaScript/Language_Resources.
[Último acceso: 17 Diciembre 2020].
- [32 M. Heller, «Node.js The JavaScript runtime explained,» InfoWorld Editor, Boston,
] 2020.

- [33 NginX, «NginX | High performance balancer, Web Server and reverse proxy,» F5
] Inc, [En línea]. Available: <https://www.nginx.com>. [Último acceso: 18 Junio 2021].
- [34 Microsoft, «Visual Studio Code - Code Editing, Refined,» Microsoft, 2021. [En
] línea]. Available: <https://code.visualstudio.com/>. [Último acceso: 05 Julio 2021].
- [35 D. Tello Leal, E. Tello Leal y C. Sosa, «Revisión de los sistemas de control de
] versiones utilizados en el desarrollo de software,» *Revista de Ingenierías USBMED*,
vol. III, nº 1, pp. 15 - 17, 2012.
- [36 Git, «Git - Distributed even if your workflow isn't,» Software Freedom
] Conservancy, 2021. [En línea]. Available: <https://git-scm.com/>. [Último acceso: 8
Julio 2021].
- [37 Github, Inc., «Github - The tools you need to build what you want.,» Github, 2021.
] [En línea]. Available: <https://github.com/features>. [Último acceso: 06 Julio 2021].
- [38 Postman, «Postman - The Collaboration Platform for API Development,» Postman,
] 2021. [En línea]. Available: The Collaboration Platform for API Development.
[Último acceso: 8 Julio 2021].
- [39 T. Fadhilah Iskandar, «Comparison between client-side and server-side,» IOP
] publishing, 2020.
- [40 IETF, «JSON web token,» Mayo 2015. [En línea]. Available:
] <https://datatracker.ietf.org/doc/html/rfc7519>. [Último acceso: 6 Julio 2021].
- [41 Diagrams corporation, «Diagrams - Security-first diagramming for teams.,» 2021.
] [En línea]. Available: <https://www.diagrams.net/>. [Último acceso: 6 Julio 2021].
- [42 Y. Bermudez Colina, «Applications of linear, mixed and integer programming,»
] *Revista científica Universidad de Carabobo*, vol. II, nº 12, pp. 85-104, 2011.
- [43 H. Guerrero Salas, PROGRAMACIÓN LINEAL APLICADA, Bogotá: 2017, 2017.
]
- [44 J. A. Boirivant, «LA PROGRAMACIÓN LINEAL APLICACIÓN DE LA
] PEQUEÑAS,» *Revista Reflexiones UCR*, vol. 1, pp. 89 - 105, 2009.

- [45 M. Ruiz Pillajo, «AUTOMATIZACION DE LOS PROCESOS ACADÉMICOS DE NIVEL BÁSICO UNIFICADO,» Quito, 2015.
- [46 H. Taha, *Operations Research An Introduction*, Columbus: Pearson Prentice Hall, 2007.
- [47 A. Fugetta, «Open source software—an evaluation,» *Journal of Systems and software*, vol. 66, n° 1, pp. 77-90, 2003.
- [48 C. B. Reinoso, *Introducción a la Arquitectura de Software*, Buenos Aires: Universidad de buenos aires, 2004.
- [49 A. Alvarez Cruz, *Casos de Uso - Diagramas de Casos de Uso*, Mérida: Publicaciones de Universidad de los Andes, 2011.
- [50 Adobe, «AdobeXD,» Adobe Inc, [En línea]. Available: <https://www.adobe.com/es/products/xd.html#>. [Último acceso: 17 Diciembre 2020].

ANEXOS

ENTREVISTA	
Entrevistador:	José Baquerizo
Entrevistado:	Arq. Mauricio Yunda Villao
Objetivo:	Determinar aspectos básicos de la institución y proponer el tema del proyecto
Fecha:	11/12/2020
Ubicación:	Calle 18 de agosto entre Paquisha y, Fausto Fajardo, Santa Elena
<ol style="list-style-type: none">1. Describa brevemente la historia de la institución y como está formada.2. ¿Qué objetivo tiene la institución actualmente?3. ¿Quiénes son los miembros de la directiva?4. ¿Cada cuánto tiempo se elige nueva directiva?5. ¿Cómo afrontan la gestión de sus horarios de clases?6. ¿Cuál es el proceso para seguir al inicio del periodo académico?7. ¿Cómo distribuyen las asignaturas a cada docente?8. ¿Cómo monitorean la carga horaria de los mismos?9. ¿Considera óptimo el tiempo que emplea en la elaboración de sus horarios?10. ¿Considera implementar proyectos que ayuden a resolver esta problemática?	

Anexo 1 Formato de entrevista

OBSERVACIÓN

Ente objetivo:	Escuela de Educación Básica “Lic. Angélica Villón Lindao”
Duración:	1 semana
Tipo de observación:	Preliminar Directa
Ubicación:	Calle 18 de agosto entre Paquisha y, Fausto Fajardo, Santa Elena

Descripción:

Se acudirá a la institución para obtener información acerca de la historia de la escuela y obtener sus datos generales como Misión y Visión.

Conclusiones:

Se obtuvo información acerca de la historia de la institución, su misión y su visión.



Anexo 2 Formato de observación general de la institución

OBSERVACIÓN

Ente objetivo:	Escuela de Educación Básica “Lic. Angélica Villón Lindao”
Duración:	1 día
Tipo de observación:	Preliminar Directa
Ubicación:	Calle 18 de agosto entre Paquisha y, Fausto Fajardo, Santa Elena

Descripción:

Se observará el proceso de desarrollo de horarios de clases en la escuela de educación básica “Lic. Angélica Villón Lindao” mediante la mentoría del coordinador de la institución, Arq. Mauricio Yunda Villao.

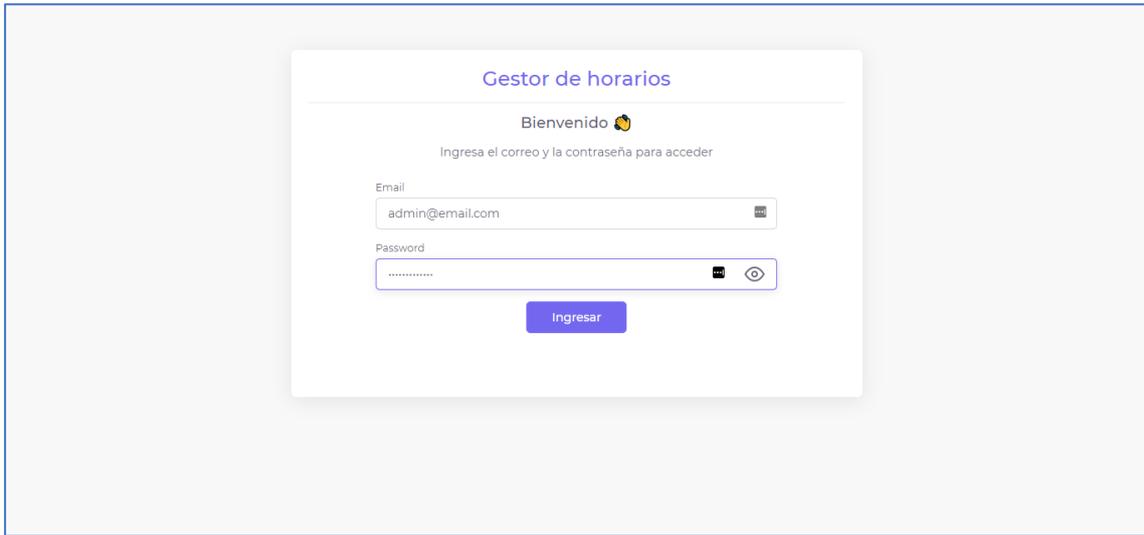
Se evaluará el proceso de desarrollo de horarios de clases.

Conclusiones:

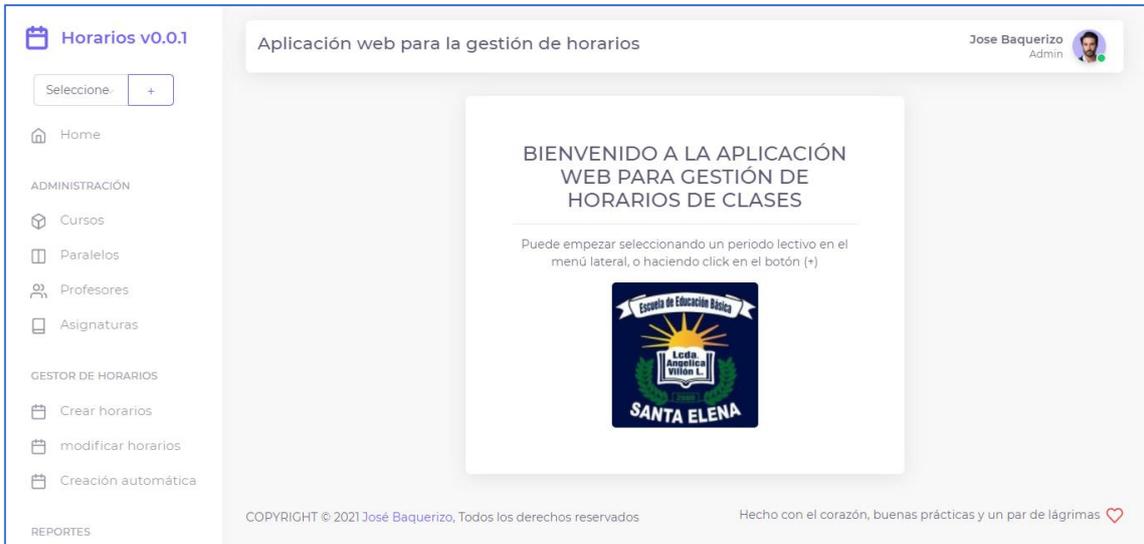
La institución maneja el proceso de gestión de horarios de forma manual, y corrigiendo los errores a medida que se van presentando, algunas veces cuando ya inician las jornadas de clases y existe algún conflicto.

Recomendaciones:

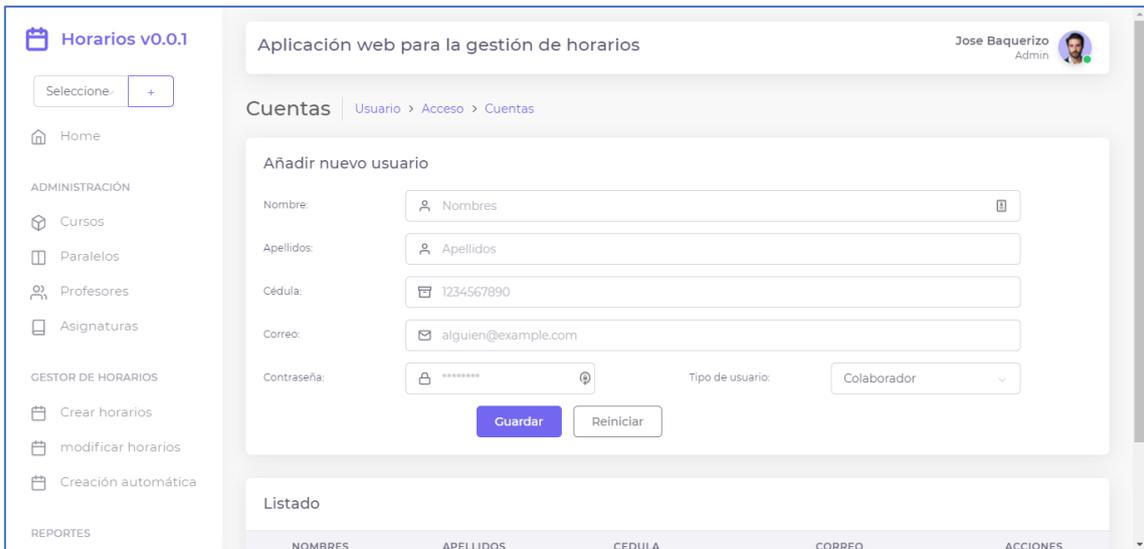
Se recomienda implementar mecanismos que brinden soporte al proceso de gestión de horarios desde su planificación y logística hasta su desarrollo y publicación.



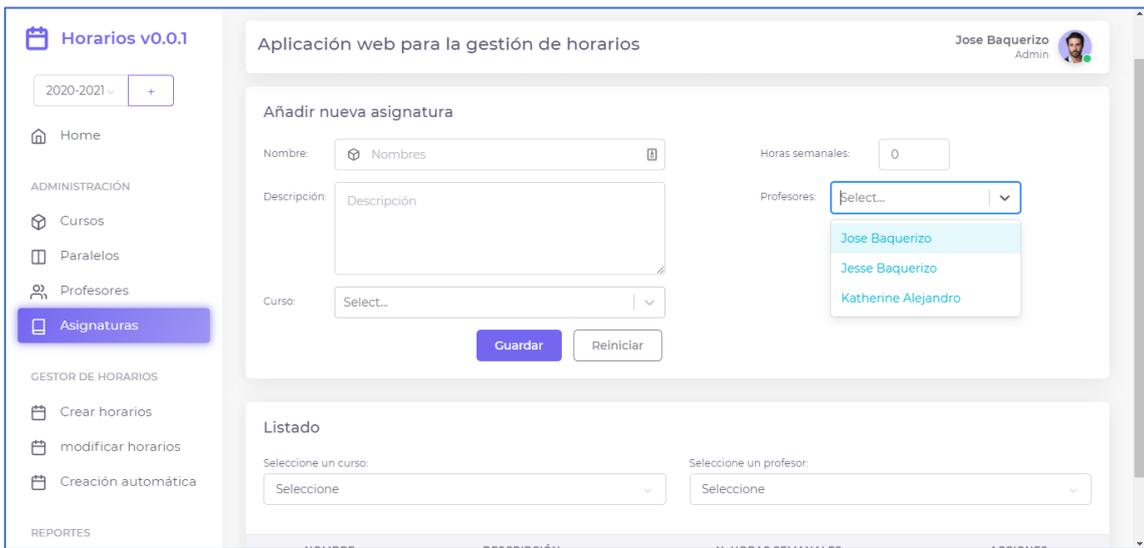
Anexo 5 Pantalla de inicio de sesión



Anexo 6 Pantalla principal



Anexo 7 Módulo de usuarios



Anexo 8 Módulo de administración

Horarios v0.0.1

Aplicación web para la gestión de horarios Jose Baquerizo Admin

ADMINISTRACIÓN

- Cursos
- Paralelos
- Profesores
- Asignaturas

GESTOR DE HORARIOS

- Crear horarios**
- modificar horarios
- Creación automática

REPORTES

- Cargas horarias
- Horarios Cursos

Configuración

Modalidad: Matutina

Curso: Seleccione

Paralelo: Seleccione

Crear Horario

Añadir elementos

Asignatura: Seleccione

Profesor: Seleccione

Día: Seleccione

Hora inicio: Seleccione Hora fin: Seleccione

Agregar **Guardar**

	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES
8:00 - 8:20					
8:20 - 8:40					

Anexo 9 Módulo de gestión de horarios

Horarios v0.0.1

Aplicación web para la gestión de horarios Jose Baquerizo Admin

ADMINISTRACIÓN

- Cursos
- Paralelos
- Profesores
- Asignaturas

GESTOR DE HORARIOS

- Crear horarios
- modificar horarios
- Creación automática**

REPORTES

- Cargas horarias
- Horarios Cursos

Creación de horarios auto. [Home](#) > [Gestión](#) > [Creación de horarios auto.](#)

Configuración

Selección modalidad: Vespertino Máximo de incrementos por día: 1

Selección días: Select...

Cursos - Asignaturas

Selección cursos: Cuarto Paralel... x Quinto Paralel... x Seleccionar

Cuarto Paralelo A

Asignatura H. semanales Profesor

Anexo 10 Panel de creación automática



Anexo 11 Pantalla de visualización de horarios



Anexo 12 Módulo de reportes

FUNCION OBJETIVO:

$$Max Z = \sum_{a \in A} \sum_{p \in P} \sum_{c \in C} \sum_{h \in H} X_{apch}$$

RESTRICCIONES:

$$\sum_{h \in H} X_{apch} = T_a \quad \forall a \in A, c \in C \quad \text{Asignar horas semanales}$$

$$\sum_{c \in C} X_{apch} \leq 1 \quad \forall a \in A, h \in H, p \in P \quad \text{Evita cruces de horarios (Docentes y cursos)}$$

$$\sum_{h \in H} X_{apch} \leq \tau_c \quad \forall a \in A, c \in C, d \in D \quad \text{Que se asigne un maximo diario}$$

$$\sum_{c \in C} X_{apch} \leq 1 \quad \forall p \in P, h \in H \quad \text{Evita cruces de docentes}$$

$$X_{apch} \in \{0,1\} \quad \forall a \in A, c \in C, h \in H, p \in P \quad \text{Asignacion binaria}$$

Quinto A					Quinto B					Quinto C				
d = 1 LUNES = 2 MARTES = 3 MIERCOLES = 4 JUEVES = 5 VIERNES					d = 1 LUNES = 2 MARTES = 3 MIERCOLES = 4 JUEVES = 5 VIERNES					d = 1 LUNES = 2 MARTES = 3 MIERCOLES = 4 JUEVES = 5 VIERNES				
8:00 - 8:20	INGLES 1	INGLES 1			8:00 - 8:20	LENGUA 1	CIENCIAS 2			8:00 - 8:20	INGLES 3	INGLES 3		
8:20 - 8:40	INGLES 1	INGLES 1			8:20 - 8:40	LENGUA 1	CIENCIAS 2			8:20 - 8:40	INGLES 3	INGLES 3		
8:40 - 9:00	LENGUA 1	CIENCIAS 2			8:40 - 9:00	INGLES 1	INGLES 1			8:40 - 9:00	CIENCIAS 2	LENGUA 1		
9:00 - 9:20	LENGUA 1				9:00 - 9:20	INGLES 1	INGLES 1			9:00 - 9:20	CIENCIAS 2			
9:20 - 9:40	CIENCIAS 2				9:20 - 9:40	INGLES 1	INGLES 1			9:20 - 9:40	LENGUA 1			

Anexo 13 Pruebas del modelo matemático en desarrollo

Quinto A					Quinto B					Quinto C				
d = 1 LUNES = 2 MARTES = 3 MIERCOLES = 4 JUEVES = 5 VIERNES					d = 1 LUNES = 2 MARTES = 3 MIERCOLES = 4 JUEVES = 5 VIERNES					d = 1 LUNES = 2 MARTES = 3 MIERCOLES = 4 JUEVES = 5 VIERNES				
8:00 - 8:20	INGLES 1	INGLES 1			8:00 - 8:20	CIENCIAS 2				8:00 - 8:20	INGLES 3	INGLES 3		
8:20 - 8:40	INGLES 1	INGLES 1			8:20 - 8:40	CIENCIAS 2				8:20 - 8:40	INGLES 3	INGLES 3		
8:40 - 9:00	LENGUA 1	CIENCIAS 2			8:40 - 9:00	INGLES 1	LENGUA 1			8:40 - 9:00	CIENCIAS 2	LENGUA 1		
9:00 - 9:20	LENGUA 1				9:00 - 9:20	INGLES 1				9:00 - 9:20	CIENCIAS 2	LENGUA 1		
9:20 - 9:40	CIENCIAS 2				9:20 - 9:40	INGLES 1	LENGUA 1			9:20 - 9:40				

Anexo 14 Pruebas exitosas del modelo matemático