



**UNIVERSIDAD ESTATAL DE MILAGRO  
FACULTAD DE CIENCIAS E INGENIERÍA**

**TRABAJO DE PROYECTO TÉCNICO  
PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERÍA EN SISTEMAS COMPUTACIONALES.**

**TEMA:** DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA CREACIÓN Y  
GESTIÓN DE HORARIOS EN LA FACULTAD DE CIENCIAS E INGENIERÍA DE LA  
UNIVERSIDAD ESTATAL DE MILAGRO

**Autora:**

ANA CRISTINA DIAZ GINES

**Tutor:**

ING. ÓSCAR DARÍO LEÓN GRANIZO

**Milagro, 2021  
ECUADOR**

## DERECHOS DE AUTOR

Ingeniero.

Fabricio Guevara Viejó, PhD.

**RECTOR**

**Universidad Estatal de Milagro**

Presente.

Yo, ANA CRISTINA DIAZ GINES, en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de integración curricular, modalidad presencial., mediante el presente documento, libre y voluntariamente procedo a hacer entrega de la Cesión de Derecho del Autor, como requisito previo para la obtención de mi Título de Grado, como aporte a la Línea de Investigación Implementación de un sistema para gestión de horarios en el sector educativo, de conformidad con el Art. 114 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, concedo a favor de la Universidad Estatal de Milagro una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservo a mi favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo a la Universidad Estatal de Milagro para que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

El autor declara que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

Milagro, 25 de noviembre del 2021.

---

ANA CRISTINA DIAZ GINES

Autora

CI: 1206718106

## **APROBACIÓN DEL TUTOR DEL TRABAJO DE PROYECTO TÉCNICO**

Yo, ING. LEÓN GRANIZO ÓSCAR DARÍO en mi calidad de tutor del trabajo de proyecto técnico, elaborado por ANA CRISTINA DIAZ GINES, cuyo título es DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA CREACIÓN Y GESTIÓN DE HORARIOS EN LA FACULTAD DE CIENCIAS E INGENIERÍA DE LA UNIVERSIDAD ESTATAL DE MILAGRO, que aporta a la Línea de Investigación Implementación de un sistema para gestión de horarios en el sector educativo previo a la obtención del Título de Grado de Ingeniería en Sistemas Computacionales; considero que el mismo reúne los requisitos y méritos necesarios en el campo metodológico y epistemológico, para ser sometido a la evaluación por parte del tribunal calificador que se designe, por lo que lo APRUEBO, a fin de que el trabajo sea habilitado para continuar con el proceso previa culminación de Trabajo de Proyecto Técnico de la Universidad Estatal de Milagro.

Milagro, 25 de noviembre del 2021.

---

ING. LEÓN GRANIZO ÓSCAR DARÍO

Tutor  
C.I: 0928368513

## APROBACIÓN DEL TRIBUNAL CALIFICADOR

El tribunal calificador constituido por:

Elija un elemento. Haga clic aquí para escribir apellidos y nombres (tutor).

Elija un elemento. Haga clic aquí para escribir apellidos y nombres (Secretario/a).

Elija un elemento. Haga clic aquí para escribir apellidos y nombres (integrante).

Luego de realizar la revisión del Trabajo de Proyecto Técnico, previo a la obtención del título (o grado académico) de Ingeniería en Sistemas Computacionales presentado por ANA CRISTINA DIAZ GINES.

Con el tema de trabajo de DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA CREACIÓN Y GESTION DE HORARIOS EN LA FACULTAD DE CIENCIAS E INGENIERÍA DE LA UNIVERSIDAD ESTATAL DE MILAGRO.

Otorga al presente Trabajo de Proyecto Técnico, las siguientes calificaciones:

Trabajo de Integración Curricular	[    ]
Defensa oral	[    ]
<b>Total</b>	[    ]

Emite el siguiente veredicto: (aprobado/reprobado) \_\_\_\_\_

Fecha: Haga clic aquí para escribir una fecha.

Para constancia de lo actuado firman:

	Nombres y Apellidos	Firma
Presidente	Apellidos y nombres de Presidente.	_____
Secretario /a	Apellidos y nombres de Secretario	_____
Integrante	Apellidos y nombres de Integrante.	_____

## APROBACIÓN DEL TRIBUNAL CALIFICADOR

El tribunal calificador constituido por:

Elija un elemento. Haga clic aquí para escribir apellidos y nombres (tutor).

Elija un elemento. Haga clic aquí para escribir apellidos y nombres (Secretario/a).

Elija un elemento. Haga clic aquí para escribir apellidos y nombres (integrante).

Luego de realizar la revisión del Trabajo de Proyecto Técnico, previo a la obtención del título (o grado académico) de Ingeniería en Sistemas Computacionales presentado por ANA CRISTINA DIAZ GINES.

Con el tema de trabajo de DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA CREACIÓN Y GESTION DE HORARIOS EN LA FACULTAD DE CIENCIAS E INGENIERÍA DE LA UNIVERSIDAD ESTATAL DE MILAGRO.

Otorga al presente Trabajo de Proyecto Técnico, las siguientes calificaciones:

Trabajo de Integración	[    ]
Curricular	
Defensa oral	[    ]
<b>Total</b>	<b>[    ]</b>

Emite el siguiente veredicto: (aprobado/reprobado) \_\_\_\_\_

Fecha: Haga clic aquí para escribir una fecha.

Para constancia de lo actuado firman:

	Nombres y Apellidos	Firma
Presidente	Apellidos y nombres de Presidente.	_____
Secretario /a	Apellidos y nombres de Secretario	_____
Integrante	Apellidos y nombres de Integrante.	_____

## **DEDICATORIA**

Dedico este trabajo principalmente a Dios, por ser mi fortaleza, mi guía, para lograr culminar una etapa más de mis estudios como profesional.

A mis padres Ana y Ángel por brindarme su apoyo incondicional durante cada etapa de mi vida y me han formado en una persona con valores, ética e impulsarme a culminar con éxito mi carrera.

A mis hermanos por siempre creer en mí, por darme fortaleza en momentos difíciles, y compartir mi sueño durante estos años para lograr mi meta.

A mi novio, por sus abrazos, buenas vibras y palabras de aliento.

A cada ángel que Dios colocó en mi camino y aportó en mi crecimiento personal y profesional.

*Ana Cristina Díaz Ginés*

## AGRADECIMIENTO

A Dios, por brindarme salud, sabiduría e inteligencia en cada etapa de mi vida y ayudarme a culminar con éxito mi carrera profesional.

A mis padres Ana y Ángel por brindarme su apoyo incondicional durante cada etapa de mi vida y me han formado en una persona con valores, ética e impulsarme a culminar con éxito mi carrera.

A mis hermanos por siempre creer en mí, por darme fortaleza en momentos difíciles, y compartir mi sueño durante estos años para lograr mi meta.

A mi novio, por sus abrazos, buenas vibras y palabras de aliento.

A cada ángel que Dios colocó en mi camino y aportó en mi crecimiento personal y profesional.

A cada uno de los docentes que formaron parte de este proceso, gracias por compartir sus sólidos conocimientos, experiencia, ser nuestra guía para convertirnos con éxito en profesionales, en especial mi agradecimiento a mi Tutor ingeniero Oscar Darío León por su tiempo, dedicación y dirección para poder desarrollar de la mejor manera este trabajo de investigación.

Finalmente, mi agradecimiento a la Universidad Estatal de Milagro, la cual me permitió incrementar mis conocimientos y me formó como profesional para enfrentarme a una sociedad laboral competitiva.

*Ana Cristina Díaz Gines*

## ÍNDICE GENERAL

DERECHOS DE AUTOR .....	II
APROBACIÓN DEL TUTOR DEL TRABAJO DE PROYECTO TÉCNICO .....	III
APROBACIÓN DEL TRIBUNAL CALIFICADOR .....	IV
APROBACIÓN DEL TRIBUNAL CALIFICADOR .....	V
DEDICATORIA .....	VI
AGRADECIMIENTO .....	VII
ÍNDICE GENERAL .....	VIII
ÍNDICE DE FIGURAS .....	X
ÍNDICE DE TABLAS .....	XII
RESUMEN .....	XIV
ABSTRACT .....	XV
CAPÍTULO 1 .....	1
1. Introducción .....	1
1.1. Planteamiento del problema .....	2
1.2. Objetivos .....	3
1.2.1. Objetivo general .....	3
1.2.2. Objetivos específicos .....	4
1.3. Alcance .....	4
1.4. Estado del arte .....	5
1.4.1. Antecedentes del estudio .....	5
1.4.2. Fundamentación teórica .....	8
1.4.3. Definición .....	21
CAPÍTULO 2 .....	22
2. Metodología .....	22

2.1. Metodología de investigación .....	22
2.1.1. Análisis de datos. ....	23
2.2. Metodología de desarrollo.....	25
2.2.1. Product Backlog (Historias de usuario completas).....	27
2.2.2. Sprint Backlog. ....	33
2.2.3. Sprint Planning. ....	38
2.2.4. Incremento / Sprints, Burndown Chart y demás.....	41
CAPÍTULO 3 .....	44
3. Propuesta de solución .....	44
3.1. Análisis de factibilidad.....	44
3.1.1. Factibilidad técnica .....	44
3.1.2. Factibilidad económica .....	45
3.2. Propuesta .....	47
3.2.1. Diagrama de base de datos.....	47
3.2.2. Diagrama de aplicación .....	53
3.2.3. Desarrollo.....	53
3.3. Validación de la propuesta .....	62
CONCLUSIONES.....	64
RECOMENDACIONES .....	65
REFERENCIAS BIBLIOGRÁFICAS .....	66
ANEXOS .....	70

## ÍNDICE DE FIGURAS

Figura 1 <i>Arquitectura cliente / servidor</i> .....	10
Figura 2 <i>Arquitectura de dos niveles</i> .....	11
Figura 3 <i>Arquitectura de tres niveles</i> .....	11
Figura 4 <i>Arquitectura de cuatro niveles</i> .....	12
Figura 5 <i>Arquitectura de Django</i> .....	18
Figura 6 <i>Arquitectura de PostgreSQL</i> .....	20
Figura 7 <i>Burndown charts para el desarrollo del sistema web - inicial</i> .....	43
Figura 8 <i>Burndown charts para el desarrollo del sistema web – final</i> .....	43
Figura 9 <i>Modelo de entidad relación de la base de datos (parte 1)</i> .....	48
Figura 10 <i>Modelo de entidad relación de la base de datos (parte 2)</i> .....	49
Figura 11 <i>Modelo de entidad relación de la base de datos (parte 3)</i> .....	50
Figura 12 <i>Modelo de entidad relación de la base de datos (parte 4)</i> .....	51
Figura 13 <i>Modelo de entidad relación de la base de datos (parte 5)</i> .....	52
Figura 14 <i>Diagrama de aplicación del sistema web propuesto</i> .....	53
Figura 15 <i>Pantalla de inicio del sistema web para la FACI</i> .....	54
Figura 16 <i>Opciones para la configuración de horarios</i> .....	54
Figura 17 <i>Configuración de periodo – vista inicial</i> .....	55
Figura 18 <i>Configuración en la opción periodo - creación del periodo</i> .....	55
Figura 19 <i>Configuración en la opción asignaturas – vista inicial</i> .....	56
Figura 20 <i>Configuración en la opción asignaturas – creación de asignaturas</i> .....	57
Figura 21 <i>Configuración en la opción cursos – vista inicial</i> .....	57
Figura 22 <i>Configuración en la opción cursos – creación de cursos</i> .....	58
Figura 23 <i>Configuración en la opción cursos – botón gestionar disponibilidad</i> .....	59
Figura 24 <i>Configuración en la opción docente – vista inicial</i> .....	59

Figura 25 <i>Configuración en la opción docente – creación de docente</i> .....	60
Figura 26 <i>Configuración en la opción docente – días libres</i> .....	61
Figura 27 <i>Visualización de horarios generados en el sistema web</i> .....	62

## ÍNDICE DE TABLAS

Tabla 1 <i>Causas y consecuencia del problema</i> .....	3
Tabla 2 <i>Criterios de prioridad y complejidad para las actividades</i> .....	28
Tabla 3 <i>Historia de usuario – Sprint 1</i> .....	28
Tabla 4 <i>Historia de usuario – Sprint 2</i> .....	29
Tabla 5 <i>Historia de usuario – Sprint 3</i> .....	29
Tabla 6 <i>Historia de usuario – Sprint 4</i> .....	30
Tabla 7 <i>Historia de usuario – Sprint 5</i> .....	31
Tabla 8 <i>Historia de usuario – Sprint 6</i> .....	31
Tabla 9 <i>Product Backlog</i> .....	32
Tabla 10 <i>Sprint 1</i> .....	33
Tabla 11 <i>Sprint 2</i> .....	34
Tabla 12 <i>Sprint 3</i> .....	34
Tabla 13 <i>Sprint 4</i> .....	35
Tabla 14 <i>Sprint 5</i> .....	36
Tabla 15 <i>Sprint 6</i> .....	37
Tabla 16 <i>Sprint backlog</i> .....	38
Tabla 17 <i>Sprint planning 1</i> .....	38
Tabla 18 <i>Sprint planning 2</i> .....	39
Tabla 19 <i>Sprint planning 3</i> .....	39
Tabla 20 <i>Sprint planning 4</i> .....	40
Tabla 21 <i>Sprint planning 5</i> .....	40
Tabla 22 <i>Sprint planning 6</i> .....	41
Tabla 23 <i>Incremento / Sprint (parte 1)</i> .....	41
Tabla 24 <i>Incremento / Sprint (parte 2)</i> .....	42

Tabla 25	<i>Requisitos técnicos de hardware para el desarrollo del sistema web</i>	44
Tabla 26	<i>Requisitos técnicos de software para el desarrollo del sistema web</i>	44
Tabla 27	<i>Costos de hardware para el desarrollo del sistema web</i>	45
Tabla 28	<i>Costos de software para el desarrollo del sistema web</i>	46
Tabla 29	<i>Costos de recurso humano para el desarrollo del sistema web</i>	46
Tabla 30	<i>Costo total para el desarrollo del sistema web</i>	46
Tabla 31	<i>Informe de prueba que valida el sistema web</i>	63

**Título de Trabajo de Proyecto Técnico:** DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA CREACIÓN Y GESTIÓN DE HORARIOS EN LA FACULTAD DE CIENCIAS E INGENIERÍA DE LA UNIVERSIDAD ESTATAL DE MILAGRO

**RESUMEN**

El estudio tuvo como objetivo desarrollar un sistema web que facilite la creación y gestión de horarios de los docentes en la Facultad de Ciencias e Ingeniería FACI de la Universidad Estatal de Milagro que aporte a la coordinación del ejercicio profesional de sus educadores, justificando la investigación en la carencia de un sistema automatizado que facilite este procedimiento, evite equivocaciones y conflictos entre actores educativos, es decir estudiantes, docentes y directivos. Para alcanzar este objetivo, se recurrió a una metodología de investigación descriptiva con enfoque cualitativo, aplicando una entrevista al docente encargado de crear los horarios en la FACI, conociendo que la entidad dispone de un sistema web, pero no ajustado a sus necesidades, desencadenando como principal problema el cruce de horarios. Así se respaldó la propuesta de un sistema web que aporte a su asignación efectiva, eligiendo como metodología de desarrollo la SCRUM que permitió dividir el proyecto en seis Sprint o etapas que comprendieron 1) Análisis de requerimientos del usuario; 2) Análisis y diseño de bases de datos; 3) Diseño y maquetación de las pantallas; 4) Almacenamiento de la información; 5) Configuración de disponibilidad de horarios; 6) Generación de horarios. Su cumplimiento demandó una duración total de 30 días, obteniendo un sistema que permite configurar el periodo, asignaturas, cursos y docentes para generar horarios ajustados a las necesidades de la FACI, demostrando ser factible técnica y económicamente, necesitando una inversión de USD 1.300 para hardware y recurso humano, excluyendo software al ser tecnologías de código abierto.

**PALABRAS CLAVE:** Horario, sistema web, software, coordinación.

**Título de Trabajo de Proyecto Técnico:** DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA CREACIÓN Y GESTIÓN DE HORARIOS EN LA FACULTAD DE CIENCIAS E INGENIERÍA DE LA UNIVERSIDAD ESTATAL DE MILAGRO

**ABSTRACT**

The objective of the study was to develop a web system that facilitates the creation and management of teachers' schedules in the Faculty of Sciences and Engineering FACI of the State University of Milagro that contributes to the coordination of the professional practice of its educators, justifying the research in the lack of an automated system that facilitates this procedure, avoids mistakes and conflicts between educational actors, that is, students, teachers and managers. To achieve this objective, a descriptive research methodology with a qualitative approach was used, applying an interview to the teacher in charge of creating the schedules at the FACI, knowing that the entity has a web system but not adjusted to its needs, triggering as the main problem crossing schedules. Thus, the proposal of a web system that contributes to its effective allocation was supported, choosing SCRUM as the development methodology, which allowed dividing the project into six Sprints or stages that included 1) Analysis of user requirements; 2) Analysis and design of databases; 3) Design and layout of the screens; 4) Information storage; 5) Schedule availability configuration; 6) Generation of schedules. Its compliance required a total duration of 30 days, obtaining a system that allows configuring the period, subjects, courses and teachers to generate schedules adjusted to the needs of the FACI, proving to be technically and economically feasible, requiring an investment of USD 1,300 for hardware and human resources, excluding software as they are open source technologies.

**KEY WORDS:** Schedule, web system, software, coordination.

## CAPÍTULO 1

### 1. Introducción

Dentro de cualquier entidad, sin distinción del campo de sus operaciones, debe existir una correcta coordinación entre sus miembros que garantice su funcionamiento óptimo. De acuerdo a Flórez (2017) las organizaciones deben esforzarse porque sus procesos se encuentren coordinados y posibiliten a cada miembro el desarrollo de sus funciones para el alcance del objetivo social. Con lo expuesto, si existe una coordinación deficiente que impide a sus miembros desenvolverse de manera efectiva, esto influirá negativamente en el logro de los objetivos de la institución.

En el plano de las instituciones educativas, su objetivo implica brindar una educación de calidad al estudiante. Respecto a la educación de calidad, en Ecuador es un derecho garantizado en la Constitución de la República por su gran influencia en el desarrollo nacional al formar profesionales competitivos (Asamblea Nacional del Ecuador, 2008). Por ende, todo el sistema educativo y las instituciones que lo componen, incluyendo las de educación superior, deben asegurarse de operar con eficiencia para el logro de este objetivo.

Lo expuesto se refuerza en el art.351 de la Constitución nacional, en donde se menciona que el sistema de educación superior se regirá bajo una serie de principios, entre los cuales destaca la calidad (Asamblea Nacional del Ecuador, 2008). Así, se espera que, en las instituciones, entre ellas las universitarias, operen con eficiencia para garantizar el derecho a la educación de calidad a futuros profesionales.

Sin embargo, dentro de la Facultad de Ciencias e Ingeniería [FACI] de la Universidad Estatal de Milagro, al no contar con un sistema automatizado que permita gestionar los horarios, se crean conflictos entre horas y materias, ocurriendo cruces entre

los horarios asignados a cada docente y creando también un conflicto para los estudiantes, generando confusiones e inconvenientes para tomar las materias.

Partiendo de lo expuesto, el propósito del proyecto es el desarrollo e implementación de un sistema que brinde una solución viable a este problema, garantizando la coordinación entre los horarios de cada docente, evitando situaciones que afecten negativamente a la calidad de la educación que la facultad debe brindar a sus estudiantes.

### **1.1. Planteamiento del problema**

A pesar del constante desarrollo de tecnologías digitales y la posibilidad de emplearlas como herramientas en el campo académico, dentro de la FACI algunas operaciones se realizan manualmente. Una de estas operaciones es la coordinación de los horarios de los docentes pertenecientes a esta facultad, ocasionando errores que conllevan al cruce de horarios e incumplimiento de ciertas actividades.

El uso de tecnologías en la educación puede favorecer no solo al proceso de enseñanza – aprendizaje, sino también a las actividades administrativas para que las instituciones sean más eficientes. Esta postura es defendida por Melo et al. (2018), indicando que los diferentes actores educativos, involucrando a docentes, estudiantes y personal administrativo, deben apoyarse en tecnologías, aprovechándolas para desempeñar sus funciones con mayor eficiencia, adaptándose a esta era digital.

Si bien, con la pandemia de COVID – 19 ha incrementado la necesidad de aprovechar tecnologías, aún existen entidades que no logran adoptarlas con suficiencia. Es el caso de la FACI, la cual carece de un software que le permita gestionar los horarios para organizar a los docentes y evitar el cruce entre materias, perjudicando así la efectividad con la cual garantizan a los estudiantes una educación suficiente.

Cabe señalar que, el coordinar manualmente los horarios, repercute en equivocaciones y provoca conflictos entre docentes, estudiantes y directivos, creando un entorno que dificulta la formación académica de los futuros profesionales.

**Tabla 1**

*Causas y consecuencia del problema*

<b>Causas</b>	<b>Consecuencias</b>
C1. Falta de software automatizado para la gestión de horarios	E1. Existen confusiones y cruces de horarios entre docentes
C2. Falta de preparación del personal encargado de asignar horarios	E2. Demora en los procesos para la asignación de horarios
C3. Coordinación manual de los horarios	E3. Mayor probabilidad de errores al momento de asignar horarios al docente
C4. Falta de comunicación entre docentes y directivos al momento de asignar horarios	E4. Los horarios asignados no se ajustan a los tiempos del docente.
C5. La FACI dispone de un número importante de docentes.	E5. La asignación de horarios demanda mayor esfuerzo para evitar errores.
C6. La mayoría de docentes realiza requerimientos específicos de horario	E6. No es posible satisfacer los requerimientos de todos los docentes al momento de asignarles horario

*Nota:* Esta tabla muestra las causas y consecuencia del problema estudiado, fuente y elaboración propia

## **1.2. Objetivos**

### **1.2.1. Objetivo general**

Desarrollar un sistema web que facilite la creación y gestión de horarios de los docentes en la Facultad de Ciencias e Ingeniería de la Universidad Estatal de Milagro, aportando a la coordinación del ejercicio profesional de los educadores.

### **1.2.2. Objetivos específicos**

- Evaluar el proceso para la generación de los horarios asignados a los docentes, identificando los aspectos considerados para dicha asignación.
- Desarrollar el sistema web de horarios usando Python.
- Realizar pruebas de efectividad del sistema desarrollado.

### **1.3. Alcance**

El proyecto será desarrollado al año 2021 a fin de entregar a la FACI de la Universidad Estatal de Milagro un sistema web para la correcta coordinación de los horarios de sus docentes. Con ello, se pretende garantizar que impartan sus clases completas sin ningún inconveniente, favoreciendo al aprendizaje y formación de sus estudiantes, los cuales serán futuros profesionales y tienen derecho a una educación de calidad en todos los sentidos.

El sistema web contará con un módulo que permitirá cargar los niveles, los paralelos, las materias, las horas de cada materia, el profesor, su disponibilidad, la asignación de carga horario a cada educador, y con ello generará un horario que ayude a la coordinación efectiva de los educadores, evitando el cruce de horarios que afecte negativamente a la formación de los estudiantes.

Como limitante se considera el distanciamiento social y las medidas aplicadas para prevenir los contagios de COVID – 19, lo cual impide un acercamiento más directo a los informantes claves. Se espera superar esta situación mediante el uso de internet y otras herramientas tecnológicas que permitan consultar y recolectar los datos de la muestra en forma remota.

## 1.4. Estado del arte

### 1.4.1. Antecedentes del estudio

En esta sección se consultaron estudios relacionados al tema planteado, es decir el desarrollo de un sistema o aplicación web que permita la creación y gestión de horarios.

**Título del estudio 1:** Sistema informático para la elaboración y publicación del horario docente de la Universidad de Holguín.

Partiendo de los estudios internacionales se identificó el desarrollado por Hernández et al. (2020) en España, el cual se orientó al desarrollo de un sistema para la asignación y publicación de horarios a docentes universitarios, justificando su desarrollo en la necesidad de que cada profesional tenga un horario adecuado que aporte a su labor como educadores. La propuesta resultó en dos aplicaciones web, la primera destinada a dar asistencia al proceso de elaboración del horario asignado a estos profesionales y otra para que dicho horario pueda publicarse.

El desarrollo se llevó a cabo utilizando el *frameworks* AngularJS, ambas interactuando en una interfaz de Programación de Aplicaciones, la cual fue implementada con los *frameworks* NodeJS y ExpressJS a través del protocolo HTTP y el uso de la Transferencia de Estado Representacional. A su vez, se utilizó MongoDB como sistema gestor de la base de datos, siendo tecnologías de código abierto que no generaron ningún costo. Para el presente proyecto también se considera el uso de tecnologías Open Source, las cuales aporten a la viabilidad de la aplicación web destinada a asignar los horarios a los educadores de la FACI.

**Título del estudio 2:** Aplicación web para la administración y asignación de los horarios en las facultades de la Universidad Santo Tomás Seccional Tunja.

Entre los estudios regionales se ubicó el desarrollado por Fajardo y Soler (2017) en Colombia, el cual se encaminó al diseño de una aplicación web para administrar y asignar

horarios a los docentes en una facultad universitaria, justificando su desarrollado puesto que la gestión se realiza por medio de hojas de cálculo, careciendo de comodidad, agilidad y funcionalidad para esta asignación.

En el desarrollo de la aplicación web se utilizaron como herramientas MySQL, el lenguaje de programación PHP y el frameworks Lavarel, empleando una metodología deductiva de enfoque cualitativo al aplicarse entrevistas que permitieron conocer las perspectivas de los docentes respecto a los problemas para asignar horarios y los aspectos a considerarse.

Las etapas para el desarrollo fueron cinco que involucraron: Identificación de problemas; análisis de requerimientos; diseño de la aplicación; desarrollo de la aplicación; pruebas; e implementación. Como propuesta se desarrolló la aplicación web *Horarios USTA TUNJA* con módulos de iniciar sesión, cambiar contraseña, administración de docentes, administrar grupos, asignación de horario y también la emisión de reportes PDF.

**Título del estudio 3:** Desarrollo de un sistema integrado de gestión de horarios y control de asistencia docente para el centro Preuniversitario de la UNAMAD.

Otro estudio regional fue presentado en Perú por Del Alcázar (2019) para el diseño de un sistema de gestión de horarios y control de la asistencia docente en una institución universitaria. La metodología de desarrollo fue la SCRUM, mientras que la metodología de investigación fue aplicada y pre experimental, realizándose encuestas a docentes para evaluar la asignación actual de horarios.

Los resultados arrojaron que existen deficiencias y se respalda el diseño del sistema, para el cual se usó el entorno de desarrollo integrado NetBeans.IDE 8.1 RC2 y la tecnología Node.js, complementada con el *frameworks* Angular JS y MySQL como gestor de base de datos.

**Título del estudio 4:** Implementación de un Sistema de Gestión de Horarios de Clases para los docentes del colegio península de Santa Elena.

Finalmente, como estudio nacional se ubicó el desarrollado por Barzola (2015) en Santa Elena, presentando como objetivo la implementación de un sistema para la gestión de los horarios de los docentes en un colegio, los cuales son elaborados en Microsoft Excel y restan eficiencia. Se utilizó una metodología lógica deductiva aplicándose entrevistas a docentes y directivos, lo cual permitió conocer en detalle los problemas con la asignación de horarios; mientras que las tecnologías empleadas fueron JAVA Eclipse ZK para el desarrollo del sistema; MySQL Workbench 6.3 para administrar bases de datos; y Jasper Report para elaborar los reportes.

**Título del estudio 5:** Desarrollo de un sistema para la creación de horarios para la Universidad Central del Ecuador.

Un estudio en Quito realizado por Amaya (2017) involucró el desarrollo de un sistema para la creación de horarios en la Universidad Central del Ecuador. La metodología para el desarrollo fue SCRUM por ser la más utilizada, empleándose para el diseño de la aplicación web distintas tecnologías: PostgreSQL como motor de base de datos; Spring STS como entorno de desarrollo; Navicat para administrar bases de datos; Java; Jaspersoft Studio como diseñador visual para JasperReports; Apache Tomcat; Apache CXF; Bootstrap; jQuery; Lombok y distintos frameworks; siendo cada una de licencia libre para el ahorro de costos

Para comprobar su funcionamiento se realizaron pruebas en navegadores web desde computadoras y dispositivos móviles, además de comprobarse la generación de horarios sin ningún problema para los docentes. Con los antecedentes revisados se comprueba la posibilidad de diseñar un sistema web para la creación y gestión de horarios asignados a

los docentes en la FACI, esto a partir de software libres al igual que en la propuesta planteada.

#### **1.4.2. Fundamentación teórica**

**1.4.2.1. Sistema Web.** El Grupo Consultor EFE (2021) define a estos sistemas como aplicaciones de software, mismas que son utilizadas al acceder a un servidor web, esto mediante un navegador conectado a internet o a una intranet. Entre las ventajas de su utilización están la facilidad de uso, ahorro de costos en hardware y software, seguridad de los datos, facilitan el trabajo a distancia y colaborativo, entre otras.

Estos sistemas, también llamados aplicaciones web, reciben su nombre porque se ejecutan en internet; siendo los archivos y datos que utiliza procesados y almacenados dentro de la web (Goodwill Community Foundation, 2020). Cabe señalar que, desde cualquier dispositivo conectado a esta red, podrá accederse a la aplicación web para obtener la información que se requiera en un momento determinado.

Generalmente, también se requieren los datos de acceso, siendo un nombre de usuario y contraseña. Como característica de estas aplicaciones, Molina et al. (2018) sostienen que son aplicaciones cliente/ servidor, en donde el cliente (explorador, navegador o visualizador), el servidor (servidor web) y el protocolo bajo el cual se comunican (http) se encuentran estandarizados y no son creados por quien figura como el programador de la aplicación.

Estas herramientas se han vuelto más populares en el ámbito académico, empresarial, cultural y demás por sus grandes ventajas. Según Molina et al. (2018) entre estas ventajas se encuentra el poder ejecutarlas desde cualquier dispositivo informático conectado a internet, sistema operativo multiplataforma, copias de seguridad almacenadas en los servidores, no requiere instalación de programas, posibilidad de compartir información en forma simultánea a varias personas y la facilidad de uso.

Con lo descrito, es importante tener en cuenta que la propuesta estará alineada al diseño de un sistema web que aporte a la correcta coordinación de los horarios para los docentes de la FACI en la Universidad Estatal de Milagro.

**1.4.2.2. Arquitectura de un servidor web.** Para abordar este tema, es importante definir qué es un servidor web, indicando Gisbert (2016) que es un programa que emplea un modelo cliente / servidor, el cual acepta solicitudes HTTP realizadas por clientes web y responde a dichas solicitudes en forma de página web. Por sus características, también toma el nombre de servidor HTTP, haciendo referencia a un programa informático que se encarga de procesar una aplicación del lado del servidor.

De acuerdo a Ferrer (2017) un servidor web es un programa que permite atender y responder peticiones realizadas desde navegadores, respondiendo a tales solicitudes empleando el protocolo HTTP o HTTPS, siendo esta última la versión autenticada y cifrada del protocolo HTTP. Estos servidores generalmente tienen una arquitectura simple, la cual se basa en ejecutar de manera infinita el siguiente bucle:

- Esperar peticiones.
- Recibir peticiones.
- Buscar los recursos.
- Enviar los recursos al navegador que realizó la petición, empleado la misma conexión.
- Vuelve al primer punto.

Este es el esquema más básico de los servidores HTTP, aunque solo servirá para ficheros estáticos. A continuación, tomando como referencia a Gutiérrez (2018) se expone la arquitectura de un sistema cliente / servidor:

Figura 1

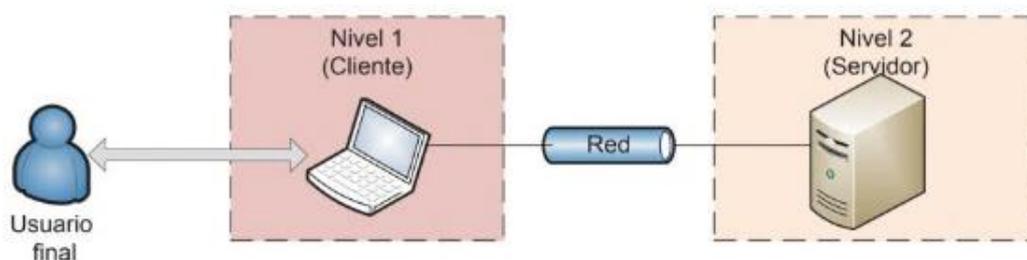
*Arquitectura cliente / servidor*

*Nota:* Se explica el funcionamiento de la arquitectura cliente / servidor empleada por los servidores web, fuente y elaborado por Gutiérrez (2018)

Bajo esta estructura, el cliente es quien envía la solicitud e inicia el requerimiento al equipo servidor mediante su dirección IP, recibiendo el servidor la solicitud y respondiendo con la dirección IP y el puerto del equipo del solicitante. Estos requerimientos pueden ser múltiples a través de redes WAN o LAN.

López (2016) explica que los servidores webs actuales emplean una arquitectura multinivel, los cuales manejan en su mayoría una arquitectura de dos o más niveles. En esta, un dispositivo con acceso a internet, denominado nivel 1, se conecta con un nivel 2 que corresponde al servidor que ejecuta la petición y devolverá la respuesta al cliente web.

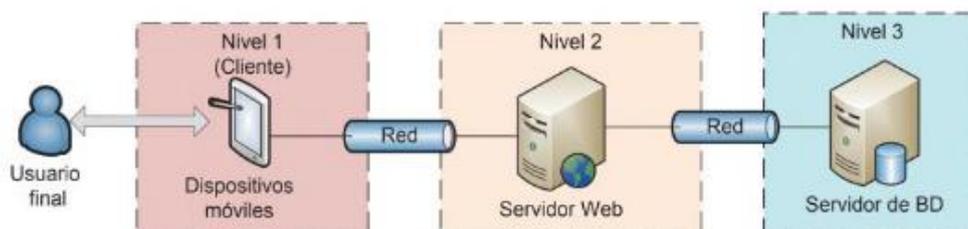
Figura 2

*Arquitectura de dos niveles*

*Nota:* Se explica el funcionamiento de la arquitectura física de dos niveles, fuente y elaborado por López (2016)

Para los demás casos, una arquitectura tendrá más niveles en la medida que intervenga otro elemento hardware, el cual generalmente desdobra las funciones del servidor.

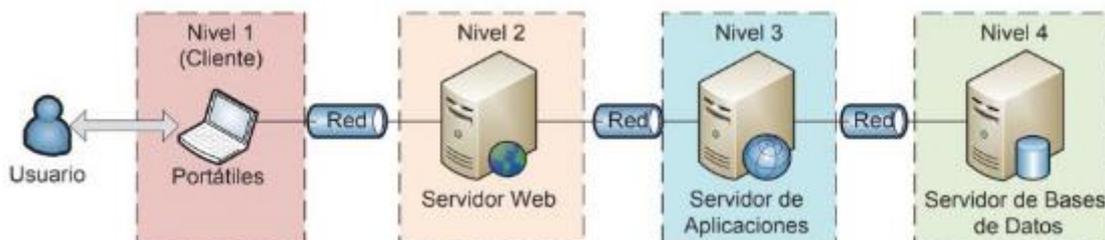
Figura 3

*Arquitectura de tres niveles*

*Nota:* Se explica el funcionamiento de la arquitectura física de tres niveles, fuente y elaborado por López (2016)

En esta ilustración, el servidor se desdobra en dos equipos, siendo uno el servidor web que recibe y atiende las peticiones; mientras el otro es un servidor de base de datos, el cual almacena y gestiona el acceso a la información. También existen arquitecturas de cuatro niveles, presentándose un ejemplo a continuación:

Figura 4

*Arquitectura de cuatro niveles*

*Nota:* Se explica el funcionamiento de la arquitectura física de cuatro niveles, fuente y elaborado por López (2016)

En este caso, se incluye un servidor de aplicaciones, teniendo este servidor una función en particular y creando así un nuevo nivel en la arquitectura. Podría mencionarse que, con el tiempo, se ha hecho necesario que se incluyan servidores de aplicaciones para reducir la carga del servidor de páginas

En la postura de Niño (2018) estos servidores se ejecutan continuamente en el ordenador, atendiendo peticiones que los clientes web realizan. A su vez, también se utiliza el término para referirse al ordenador físico, pero, a lo largo del estudio, se aplicará para el software.

En Windows, el servidor web que más se utiliza es *Internet Information Services IIS*, el cual viene integrado en distintas versiones de este sistema operativo (Niño, 2018). Como tal, el servidor IIS es un conjunto de servicios, lo cuales convierten un ordenador en un servidor web, además incluye otros servicios.

Niño (2018) sostiene que las peticiones realizadas en protocolo HTTP son interpretadas por los servidores y luego, este mismo servidor envía al cliente el contenido HTML, contenido que es interpretado por el navegador. También se envía el contenido adicional para que se incluyan en la página tales como imágenes, hojas de estilo, script y también videos.

Lo más usual es que la respuesta sea un recurso concreto, el cual puede encontrarse en el propio servidor, en otro o en ambos. Si existe un error en la petición, se informa al navegador de un código de error, mismo que se muestra al usuario.

**1.4.2.3. Técnicas de testeo de software.** Berenguel (2016) explica que las pruebas o test tienen como fin asegurar la calidad del software en desarrollo, existiendo distintas pruebas según el ámbito de aplicación y su nivel de abstracción. Centrándose en las pruebas según el nivel de abstracción se encuentran las de caja negra [BBT] o funcionales; y las de caja blanca [WBT].

Las pruebas de caja negra [Black Box Testing BBT] son aquellas en la cual el sistema que se pretende probar es visto como una caja negra, esto porque se desconoce cualquier detalle de su implementación y estructura (Berenguel, 2016). Como tal, el test proporciona un conjunto de datos de entrada al sistema, lo cual permite comprobar si los datos de salida son los esperados.

Por otra parte, las pruebas estructurales o de caja blanca [White Box Testing WBT] se encargan de comprobar la estructura interna del software. De acuerdo a Berenguel (2016) la diferencia entre ambas es que, las pruebas de caja negra revisan si el sistema cumple los requisitos funcionales, que la interfaz no tiene errores, que no hay requisitos incompletos, entre otros aspectos.

Por tal motivo, suelen realizarse después de las pruebas de caja blanca, las cuales están más cercanas al código. No son excluyentes porque existe probabilidad de que un error no detectado por una prueba, pueda ser detectado por la otra (Berenguel, 2016). Así, las pruebas de caja blanca son la de unidad y de integración; mientras que las de caja negra que analizan el sistema en su conjunto suelen ser las de validación y de sistemas.

Respecto a las pruebas de caja blanca mencionadas, según Lee (2020) son descritas a continuación:

- Unitarias: Están enfocadas en probar unidades/piezas individuales de un sistema. Así, cualquier método, función, procedimiento o módulo puede ser el que se someta a esta prueba, siendo generalmente la primera realizada por los desarrolladores en la fase de desarrollo.
- Prueba de integración: Se prueban distintos módulos de un sistema como un grupo, teniendo la finalidad de validar la integración de estos módulos e identificar errores cuando trabajan juntos en la aplicación.

Sobre las pruebas de caja negra, de acuerdo a Pauta y Moscoso (2017) se menciona lo siguiente:

- Pruebas del sistema: Se enfocan en los requisitos para el uso del sistema, verificando el ingreso, procesamiento, recuperación de los datos, e incluso la implementación, comprobando si cubre las necesidades del negocio
- Prueba de validación: Es aquella que verifica si el software cumple las especificaciones y permite lograr su finalidad, comprobando si está alineado a lo que esperaba el usuario.

**1.4.2.4. Tecnologías para el desarrollo del sistema.** En esta sección se presentan las tecnologías empleadas para el desarrollo del sistema web que se presentará a la FACI de la Universidad Estatal de Milagro:

**1.4.2.4.1. Python.** Chazallet (2016) indica que es un lenguaje de programación multiparadigma, interpretado, de alto nivel, dotado de una gestión automática de los recursos, tipado dinámico fuerte y un sistema de gestión de excepciones. Apareció en los años 1990, es gratuito y puede funcionar en todas las plataformas, teniendo además distintas implementaciones como Jython, CPython, PyPy e IronPython.

Python abarca todos los segmentos, lo cual va desde la programación de un sistema, involucrando incluso la programación web, pasando por la programación de videojuegos,

aplicaciones sin interfaz gráfica, cálculo científico, videojuegos, redes o el software embebido (Chazallet, 2016). Para los programadores que son principiantes, Python hace posible producir rápidamente un código funcional y simple; mientras que aquellos que se han especializado en otros, les facilita la adaptación.

Por otra parte, si el programador domina bien la base del lenguaje, ofrece un gran modularidad y una gran paleta funcional que permanece simple, aunque los problemas que se gestionen sean complejos. Además, dispone de una librería estándar que cubre un espectro amplio de funcionalidades y, para funcionalidades adicionales, también se dispone de una cantidad grande de proyectos externos, tales como frameworks webs (Chazallet, 2016). Por tal motivo, actualmente la mayoría de empresas innovadoras lo selecciona para el desarrollo de aplicaciones, teniendo en cuenta que el lenguaje permite llevar a cabo desarrollos profesionales rápidos y con buen rendimiento.

Según Chazallet (2016) la sintaxis de su lenguaje es clara, minimalista, sencilla, explícita y cercana al lenguaje natural, lo cual permite que se comprenda un algoritmo desde su primera lectura, especialmente si el lector domina el inglés. Sin embargo, debe destacarse que Python es un lenguaje que ha sido diseñado para ser simple, permaneciendo extremadamente flexible y orientado a una filosofía que brinda directrices claras. Además, permite a los desarrolladores llevar a cabo sus proyectos sin imponerles una forma específica de hacer las cosas, dándoles espacio para llevarlos a cabo.

Para el sistema web propuesto, se empleará Python 3.9.6 como entorno de desarrollo integrado

*1.4.2.4.2. Arquitectura de Python 3.2.6.* De acuerdo a Chazallet (2016) el núcleo de Python es su lenguaje, proporcionando una gramática original y que posee amplias posibilidades. Python permite escribir diccionarios, listas y generadores, disponiendo de 33 palabras reservadas que poseen un significado preciso y claro.

Tudor (2019) explica que Python fue creado por Guido Van Rossum, con una filosofía de diseño que hacen énfasis en la legibilidad del código, útil porque permite el desarrollo rápido de aplicaciones y compatible con múltiples paradigmas de programación, incluyendo imperativo, orientado a objetivo, con una gran biblioteca completa y funcional.

Los programas que son escritos en Python generalmente se denominan como scripts, los cuales son pequeños programas ejecutados en un entorno de ejecución y que, en el caso de Python, dicho entorno es el intérprete de Python. La mayoría de estos scripts son los de procesamiento de texto generalmente y, en el desarrollo web, Python se utiliza como un lenguaje de scripting del lado del servidor.

Hay que destacar que las secuencias de comandos del lado del servidor es una técnica que se emplea para el desarrollo web e involucra el empleo de secuencias de comandos en un servicio, las cuales producen una respuesta personalizada para las solicitudes que los usuarios realicen al sitio web. Puede emplearse para crear programas con una interfaz gráfica de usuario, también para desarrollar aplicaciones, servidores web y procesar grandes datos. Como tal, es un lenguaje de programación orientado a objetivos, lo cual hace referencia a aquella en la cual el programador no solo establece el tipo de datos de una estructura de datos, sino también el tipo de funciones u operaciones que pueden aplicarse a la estructura de dichos datos.

Así, esta estructura se convierte en un objeto que contiene tanto funciones como datos. De acuerdo a Tudor (2019), Python 2.0 se lanzó el año 2000, el cual permitió que el proceso de desarrollo sea transparente, además de la participación de la comunidad. Luego, al año 2008, se lanzó Python 3.0, el cual tuvo cambios importantes que le quitaban compatibilidad, no funcionado adecuadamente con programas escritos con versiones anteriores de Python.

Ambas son versiones oficiales y populares de Python, continuando el uso de Python 2 aunque exista Python 3. Esto se debe a que la última versión tiene cambios importantes en la sintaxis de los lenguajes de programación y el comportamiento; por ende los scripts y programas creados con Python 2 deben recodificarse para su compatibilidad con la versión reciente (Tudor, 2019). Por ende, en programas con miles de líneas, la migración del código puede convertirse en un gran problema

*1.4.2.4.3. Django 3.2.6.* Camino (2018) indica que es un framework gratuito de aplicaciones web y de código abierto, el cual está escrito en Python, teniendo una serie de ventajas como las siguientes:

- Muy rápido, puesto que permite desarrollar una aplicación muy buena en un tiempo reducido.
- Viene bien cargado, ya que algunas cosas que deben realizarse para desarrollar una aplicación web ya se encuentran implementadas, solo hay que adaptarla a las necesidades
- Es muy seguro, encontrándose ya implementadas algunas medidas de seguridad por defecto.

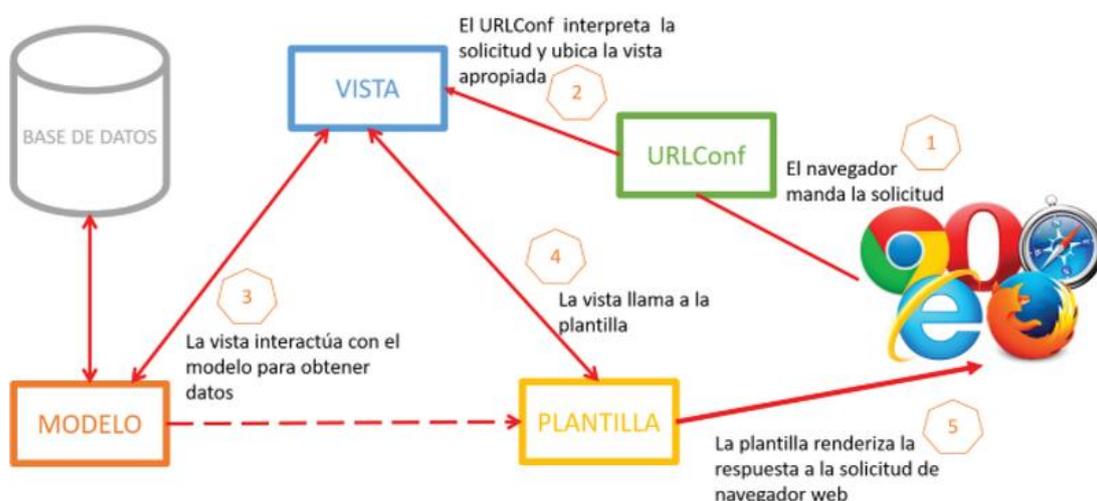
Cabe señalar que un framework hace referencia a un conjunto de componentes que facilitan en desarrollo de aplicaciones, ya que están listos para utilizarse. En el sistema web a desarrollar, se utilizará Django 3.2.6 como frameworks de ambiente web de Python.

*1.4.2.4.4. Arquitectura Django.* Tomando como referencia a MDN Web Docs Mozilla (2020) una aplicación o sistema web tradicional espera peticiones HTTP del usuario y, en el momento que se reciben, el sistema elabora lo requerido según la URL. Dependiendo de aquello que se requiere podrá leerse o escribirse información contenida en una base de datos o realizar cualquier otra tarea que satisfaga la petición. Así, el sistema devolverá la respuesta al explorador web.

Generalmente, las aplicaciones web de Django agrupan el código que gestiona estos pasos, pero en ficheros separados. Cabe señalar que utiliza la arquitectura MTV (Modelo, vista, plantilla), siendo una variación de la arquitectura MVC (Modelo, vista, controlador), siendo presentado a continuación de acuerdo a MDN Web Docs Mozilla (2020):

Figura 5

*Arquitectura de Django*



*Nota:* Detalle de la arquitectura de Django, fuente y elaborado por Hidalgo et al. (2020)

URL: Es posible procesar las peticiones de las URL individualmente mediante una función individual; sin embargo, se considera más sostenible que se escriba una función de visualización separada para cada recurso. Para ello, se emplea un mapeador de URL, esto para que las peticiones HTTP se redirijan a la vista apropiada según la URL de la petición

Vista (view): Como tal, es una función para la gestión de peticiones, la cual recibe peticiones y respuestas HTTP. Para ello, las vistas acceden a datos que son necesarios para cubrir las peticiones por medio de modelos, las cuales delegan el formateo de cada respuesta a las plantillas (templates).

Modelos (Models): Se denominan de esta manera a los objetos Python, mismos que definen la estructura que tendrán los datos dentro de la aplicación, además de proporcionar mecanismos para añadir, borrar, modificar y consultar registros dentro de la base de datos.

Plantillas (templates): Son ficheros de textos, mismos que definen el diagrama o la estructura que tendrá otro fichero, tal como una página HTML. Una vista podrá crear una página empleando una plantilla, misma que se rellenará de datos del modelo. Así, la plantilla podrá emplearse para definir la estructura que tendrá cualquier tipo de fichero.

Así, podría determinarse que el funcionamiento de Django implica que: El navegador o explorador web realiza la petición, la cual es interpretada por el URLConf y busca la vista apropiada. La vista interactúa con el modelo para la obtención de los datos, teniendo en cuenta que dicha vista se denomina plantilla. Así, la plantilla renderiza la respuesta al navegador que realizó la petición.

*1.4.2.4.5. PostgreSQL 12.* Zea et al. (2017) indicaron que es un sistema para la gestión de base de datos objeto - relacional, siendo libre y el gestor de código abierto más potente que existe en el mercado. Como tal, emplea el modelo cliente/ servidor, siendo desarrollado por Michael Stonebraker en los años 80.

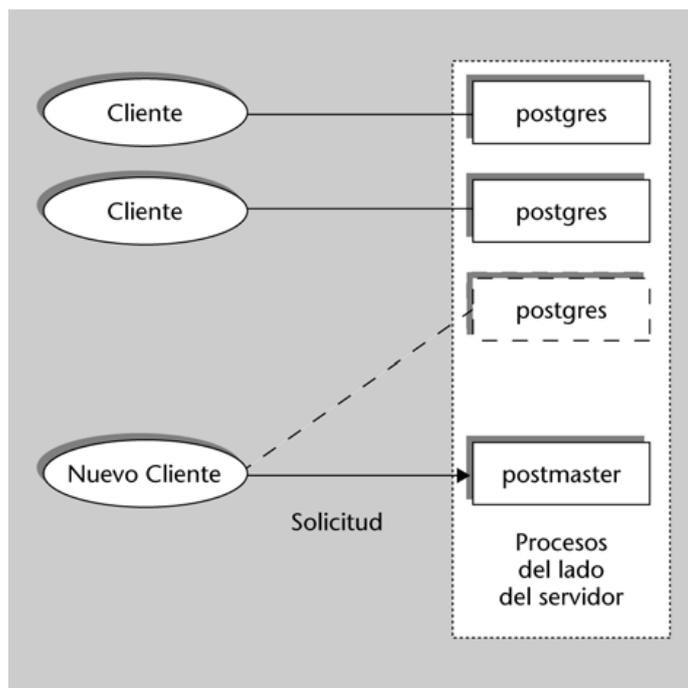
Adicionalmente, García (2017) explica que, entre las características principales, además de ser Open Source, está el aceptar una alta concurrencia; ser multiusuario y multiplataforma; puede utilizarse en una amplia variedad de lenguajes de programación; y ser adaptable a las necesidades de los usuarios. Para el desarrollo de la aplicación propuesta se utilizará PostgreSQL 12 como gestor de la base de datos.

*1.4.2.4.6. Arquitectura PostgreSQL 12.* Respecto a las bases de datos objeto – relacional que se diseñan en este software, sus características son el tamaño ilimitado de la base de datos; máximo tamaño de la tabla de 32 terabytes; 1,6 terabytes como tamaño máximo de un registro; 1GB como tamaño máximo de una celda; pueden agregarse entre

250 a 1600 columnas según el tipo de datos; mientras que las filas y número de índices también se consideran ilimitados (Digital Guide IONOS, 2019). A continuación, se presenta la arquitectura PostgreSQL:

Figura 6

*Arquitectura de PostgreSQL*



*Nota:* Detalle de la arquitectura de PostgreSQL, fuente y elaborado por Gilbert y Pérez (2016)

Según Gilbert y Pérez (2016) la arquitectura de PostgreSQL es cliente – servidor, siendo postgres el programa servidor y, entre muchos programas clientes se encuentra pgaccess y psql que se refiere a un cliente gráfico o a uno de texto respectivamente. El proceso servidor postgres puede atender a un cliente exclusivamente, lo cual implica que habrá tantos procesos de este tipo en función del número de clientes, siendo el proceso postmaster el que ejecutará un nuevo servidor para clientes que soliciten conexión.

Al equipo anfitrión o host que almacena la base de datos en conjunto se denomina sitio, el cual se ejecuta solo por un proceso el postmaster y múltiples postgres.

### 1.4.3. Definición

**1.4.3.1. *Http*.** Son las siglas del protocolo de hipertexto empleado en la transmisión de datos, derivándose del inglés *HyperText Transfer Protocol* que traducido al español significa *Protocolo de Transferencia de Hipertexto*, normalmente utilizado por los sitios web (Castaño & Jurado, 2016). Fue diseñado para la comunicación entre servidores y navegadores web siguiendo el modelo clásico cliente – servidor, en donde el primero establece una conexión y realiza una petición a un servidor, para luego recibir una respuesta de dicho servidor.

**1.4.3.2. *Internet*.** Sánchez (2016) lo define como una malla mundial de redes de computadores y ordenadores interconectados, denominándose red de redes. Está diseñado para que funcione sobre cualquier sistema operativo y hardware, haciendo posible el intercambio de información entre sus usuarios

**1.4.3.3. *TIC's*.** Heinze et al. (2017) ubican a las tecnologías de la información y la comunicación TIC's como el conjunto de herramientas, recursos y programas empleados para almacenar, recuperar, procesar y comunicar información, esto mediante distintos soportes tecnológicos. Podría considerarse que el sistema web funcionará gracias al empleo de TIC's.

## CAPÍTULO 2

### 2. Metodología

En este apartado se presentan los aspectos metodológicos para el desarrollo del proyecto, dejando en claro la metodología de investigación y de desarrollo seleccionadas.

#### 2.1. Metodología de investigación

Respecto a la metodología de investigación, se planteó como descriptiva, indicando Hernández et al. (2018) que se encuentra orientada a describir o caracterizar una situación o fenómeno de interés, detectando las causas y condiciones en las cuales este produce.

Dentro del proyecto se aplicó para conocer y describir cómo se crean y gestionan actualmente los horarios para los docentes, evidenciando los aspectos considerados para su asignación y cómo surge el problema con el cruce de dichos horarios, entre otros puntos importantes. Esto permitiría diseñar un sistema web que brinde una solución viable al problema, el cual afecta actualmente a la coordinación de la labor docente.

En relación al enfoque de la investigación, se consideró el cualitativo, el cual implica la recolección de datos mediante procedimientos no estadísticos, esto a fin de describir una realidad de interés en forma explicativa, subjetiva e interpretativa, aplicándose cuando se abordan situaciones muy específicas que necesitan analizarse de manera profunda (Martínez, 2019). En este caso, se consideró como esencial su aplicación para profundizar en el problema a partir de opiniones y criterios de los profesionales encargados de la creación de horarios en la FACI de la Universidad Estatal de Milagro.

Con lo descrito, la población de estudio se conformó por el cuerpo docente la FACI, aplicándose un muestreo no probabilístico para la selección de los participantes. Este tipo de muestreo se aplica cuando el número de individuos que puede aportar con información útil es limitado, seleccionándolos según criterios que establece el investigador sin recurrir a procedimientos estadísticos (Otzen & Manterola, 2017).

Por tal motivo, se consideró idóneo escoger como muestra al docente encargado de la creación de los horarios, siendo el Ing. Denis Darío Mendoza Cabrera, quien expuso los criterios aplicados para asignarlos a cada educador dentro de la FACI, además de profundizar en los problemas que influyen en la descoordinación de los educadores.

Habiendo descrito la muestra, se seleccionó como técnica para la recolección de datos a la entrevista. Dos Santos (2017) explica que esta técnica cualitativa se desarrolla como un diálogo entre dos o más personas para recopilar información subjetiva, orientándose a un tema o situación en particular, estando compuesta de preguntas abiertas que posibilitan al consultado una respuesta acorde a sus opiniones o experiencias. Dicho esto, el instrumento empleado fue la entrevista estructurada, preparándose con anticipación el cuestionario de preguntas para luego ser aplicada al informante clave.

### **2.1.1. Análisis de datos.**

En esta sección se presentan los resultados de la entrevista realizada al docente encargado de la creación de los horarios, el Ing. Denis Darío Mendoza Cabrera, exponiendo cómo se realiza esta gestión dentro de la FACI, proporcionando las siguientes respuestas:

#### **1. ¿Cómo funciona el sistema actual para asignar horarios?**

El sistema actual tiene desde funciones básicas hasta avanzadas. Parte de opciones como nuevo, abrir, guardar y vista previa, pero la parte esencial comprende opciones como asignaturas, clases, aulas y profesores lo cual permite asignar, organizar y cuadrar los horarios. Sin embargo, hay gestiones que deben realizarse manualmente y esto resta competitividad.

#### **2. ¿Cuáles son las gestiones que realiza en el sistema?**

Lo primero que se hace es alimentar el sistema con archivos de Excel, los cuales contienen todo el listado de los profesores respecto a materias, horas que va a trabajar y

otros datos. También cuenta con una opción de prueba para verificar que no existan problemas.

Cabe señalar que es común que existan errores para corregirse una vez se termina de cuadrar toda la información.

### **3. ¿Cuáles suelen ser los errores generalmente con la asignación de horarios?**

Esto se debe al cruce de horarios entre docentes y suele producirse porque el sistema actual no ha sido alimentado debidamente y porque el sistema empleado como tal no resulta del todo adecuado. Además, también se debe a que no todos los docentes pueden ser organizados según deseen y se producen conflictos, lo cual provoca que deban llegarse a acuerdos con cada profesional.

### **4. ¿Qué consideraciones deberían existir para un sistema de distribución de horarios?**

Lo primero es que se alimente la información de contacto para cada profesor y así sea posible ubicarlo rápidamente sin necesidad de revisar otras fuentes. Adicional a esto, actualmente no se consideran las aulas disponibles dentro del sistema, ya que se maneja una enseñanza online, pero una vez se retome la presencial tendría que tomarse en cuenta para evitar superar la capacidad de la facultad respecto al número de paralelos disponibles.

Finalmente, es importante que se incluya la opción de "prueba" para verificar que los horarios estén correctamente asignados y no surjan conflictos.

**Principales hallazgos:** Si bien, la FACI sí dispone de un sistema, el mismo no resulta óptimo según lo indica el consultado, siendo una de las limitantes que señala. Adicionalmente, las solicitudes de cada docente para que el horario se asigne según su conveniencia es otra barrera que dificulta la tarea de coordinar a cada docente sin que se produzcan conflictos.

Cabe señalar que también se obvia información como los medios de contacto con cada docente, lo cual podría facilitar la comunicación en la FACI, encontrándose rápidamente su número de teléfono o correo electrónico sin recurrir a otras fuentes. El consultado hizo hincapié en algunos aspectos al momento de diseñar el sistema, en donde señaló el incluir la opción *prueba* que verifique si cada profesor tiene correctamente asignado su horario, y considerar el número de aulas una vez se retome la enseñanza presencial, detectando y corrigiendo algún problema respecto a los salones, materias y docentes disponibles.

## **2.2. Metodología de desarrollo**

Una vez descrita la metodología de investigación se plantea la de desarrollo, seleccionándose para el presente proyecto la metodología SCRUM. De acuerdo a Subra y Vannieuwenhuyse (2018) es un método ágil de desarrollo que se basa en la cohesión del equipo el cual, en conjunto, trabajarán para el logro del mismo objetivo a través de varias interacciones. Así, el equipo será capaz de revisar los avances e identificar cambios que sean necesarios.

Este equipo debe ser auto - organizado y se construye alrededor de tres roles que son el SCRUM Master, el cliente o Product Owner, y el equipo de desarrollo. Aquí también se organizan ceremonias, tales como reuniones diarias y las revisiones del *Sprint*, teniendo en cuenta que un Sprint son mini proyectos con ciclos de ejecución muy cortos y que tienen como finalidad incrementar el valor del producto que se construye (Subra & Vannieuwenhuyse, 2018). Para esto, cada Sprint dispondrá de una planificación en donde se determinará qué se entregará y cómo se va cumplir con la entrega.

Además, la metodología SCRUM considera la presencia constante del cliente. A fin de profundizar en el tema, se presentan a continuación los roles:

- Product Owner, llamando comúnmente PO, es quien tiene la visión del producto y su función principal es redactar las *User Stories*, siendo listas de necesidades cuyo conjunto conforma el *Product Backlog*.
- El SCRUM Master es el facilitador, quien se encargará de eliminar obstáculos que puedan surgir ante los miembros del equipo, vigilando que SCRUM se aplique correctamente.
- El equipo de desarrollo, siendo aquel conformado por profesionales con la misión de desarrollar las *User Stories* dentro del *Product Backlog* y así presentar un entregable de calidad.

Una vez descrito los roles, se presenta el marco de trabajo de SCRUM, el cual está compuesto por los siguientes eventos según Narváez et al. (2019):

- *Sprint*: Hace referencia al tiempo que se asigna para un grupo de tareas cuyo objetivo es la entrega parcial del producto. Es la base de la metodología y se recomiendan periodos máximos de un mes.
- *Sprint Planning* o planeación del Sprint: Es el evento en el cual se reúnen los tres roles antes mencionados para definir las tareas a ejecutarse dentro del Sprint, incluyendo su duración en horas y días.
- *Sprint Goal* u objetivo del Sprint: Son las metas que se establecen para el Sprint, lo cual dará dirección al equipo de desarrollo en la construcción del incremento.
- *Daily SCRUM* o SCRUM diario: Son reuniones diarias en donde los miembros del equipo indican sus actividades actuales y posteriores, además de los inconvenientes que podrían suscitarse para que el equipo se apoye entre sí. Se propone una duración de 15 minutos.
- *Sprint Review* o revisión del sprint: Se lleva a cabo al finalizar el Sprint y su propósito es evaluar los incrementos y realizar adaptaciones si es necesario. Se

estima que tiene una duración de cuatro horas y el equipo expone todo lo desarrollado en el sprint.

- *Sprint Retrospective* o Retrospectiva de Sprint: Permite al equipo de desarrollo diseñar un plan de mejoras para implementar en el sprint siguiente. Ocurre después la Revisión del Sprint, antes de la siguiente Planeación del Sprint.

Estos eventos dan como resultado artefactos, indicando Narváez et al. (2019) que son tres, exponiéndose a continuación:

- *Product Backlog* o lista de productos, siendo descripciones de los requisitos y las funciones que se esperan dentro del proyecto.
- *Sprint Backlog* o lista de sprint, siendo una parte del Product Backlog que obedece a un sprint y cuyo objetivo es desarrollar un incremento del producto.
- Incremento: Son los ítems que se completan en la lista del Sprint y el conjunto de los todos los incrementos logrados con los sprint anteriores.

### **2.2.1. Product Backlog (Historias de usuario completas).**

Previo al diseño del Product Backlog es importante dejar claro quiénes ocuparán los roles dentro del equipo SCRUM:

- Product Owner o cliente, siendo en este caso el director de la FACI como principal beneficiario del sistema para la gestión de los horarios.
- El SCRUM Master, ocupando este rol la persona encargada de dirigir al equipo encargado del diseño del sistema web, es decir el Ing. Óscar León.
- El equipo de desarrollo, encabezado por Ana Cristina Díaz Gines.

Una vez descritos los roles, se procede a la presentación del Product Backlog, pero antes indicando que la prioridad y complejidad de cada actividad se estableció según el siguiente criterio:

**Tabla 2**

*Criterios de prioridad y complejidad para las actividades*

<b>Número</b>	<b>Prioridad</b>	<b>Complejidad</b>
1	Baja	Fácil
2	Media	Moderada
3	Alta	Complejo
4	Muy alta	Muy complejo

*Nota:* En esta tabla se expone la valoración a utilizar según la prioridad y complejidad de cada actividad, fuente y elaboración propia

Una vez expuestos los criterios, se presentan las historias de usuario según los Sprints considerados:

**Tabla 3**

*Historia de usuario – Sprint 1*

<b>Historia de usuario</b>			
<b>Nombre de usuario: HU1 Análisis de requerimientos del usuario</b>			
Prioridad	4	Complejidad	3
Descripción	En esta etapa se establece todo lo requerido para el diseño del sistema, planificando su desarrollo y las características que tendría para entrar a funcionamiento.		
Criterios de aceptación	<ul style="list-style-type: none"> <li>• Que los procesos estén estructurados</li> <li>• La planificación del desarrollo del sistema elaborada</li> <li>• Características del sistema y las tecnologías de desarrollo bien definidas.</li> </ul>		

*Nota:* En la tabla se describe lo que incluirá el Sprint 1 para el desarrollo del sistema web, fuente y elaboración propia

El primer Sprint comprende el análisis de requerimientos del usuario, dejando claro qué características tendría el sistema y qué tecnologías se utilizarían para su desarrollo. Para este Sprint, la prioridad fue muy alta (4) y su complejidad de nivel tres.

**Tabla 4***Historia de usuario – Sprint 2*

<b>Historia de usuario</b>			
<b>Nombre de usuario: HU2 Análisis y diseño de bases de datos</b>			
Prioridad	4	Complejidad	3
Descripción	En esta etapa se plantea la estructura de lo que se quiere almacenar en la base de datos. Además, se procederá a la creación de tablas y modelos.		
Criterios de aceptación	<ul style="list-style-type: none"> <li>• Tablas y modelos creados dentro de la base de datos</li> </ul>		

*Nota:* En la tabla se describe lo que incluirá el Sprint 2 para el desarrollo del sistema web, fuente y elaboración propia

El segundo Sprint corresponde al análisis y diseño de la base de datos, en donde se plantea la estructura de la base de datos, diseñando las tablas y modelos en donde se almacenarán. Para este Sprint, la prioridad fue muy alta (4) y su complejidad de nivel tres.

**Tabla 5***Historia de usuario – Sprint 3*

<b>Historia de usuario</b>			
<b>Nombre de usuario: HU3 Diseño y maquetación de las pantallas</b>			
Prioridad	4	Complejidad	4
Descripción	En esta etapa se crean los controles y vistas o pantallas que corresponden al sistema.		
Criterios de aceptación	<ul style="list-style-type: none"> <li>• Controles y Pantallas diseñadas y funcionales.</li> </ul>		

*Nota:* En la tabla se describe lo que incluirá el Sprint 3 para el desarrollo del sistema web, fuente y elaboración propia

El tercer Sprint tiene como propósito el diseño de los controles y vistas dentro del sistema web, los cuales permitirán la visualización de los datos y la navegación por parte

de los usuarios. Para este Sprint, la prioridad fue muy alta (4) y su complejidad de nivel cuatro.

**Tabla 6**

*Historia de usuario – Sprint 4*

<b>Historia de usuario</b>			
<b>Nombre de usuario: HU4 Almacenamiento de la información.</b>			
Prioridad	4	Complejidad	3
Descripción	Esta etapa se centra en comprobar que toda la información pueda guardarse correctamente, además de configurar los datos sobre periodos, asignaturas, cursos y docentes, terminando con el testeado del sistema y ajustes que se requieren.		
Criterios de aceptación	<ul style="list-style-type: none"> <li>• Sistema funcional, capaz de almacenar correctamente la información.</li> </ul>		

*Nota:* En la tabla se describe lo que incluirá el Sprint 4 para el desarrollo del sistema web, fuente y elaboración propia

En el cuarto Sprint se comprueba que la información pueda ser almacenada correctamente, además de iniciar con la configuración de todos los detalles requeridos para lograr la asignación correcta de los horarios. Aquí ya se realiza el testeado del sistema y, de ser requerido, también ajustes necesarios para su funcionamiento óptimo. En este caso, la prioridad fue muy alta (4) y la complejidad de tercer nivel.

**Tabla 7***Historia de usuario – Sprint 5*

<b>Historia de usuario</b>			
<b>Nombre de usuario: HU5 Configuración de disponibilidad de horarios</b>			
Prioridad	4	Complejidad	3
Descripción	Esta etapa ya se centra en lograr la asignación de horarios, involucrando la disponibilidad de los cursos y docentes.		
Criterios de aceptación	<ul style="list-style-type: none"> <li>• Cursos y docentes configurados</li> </ul>		

*Nota:* En la tabla se describe lo que incluirá el Sprint 5 para el desarrollo del sistema web, fuente y elaboración propia

En el quinto Sprint se llevó a cabo la configuración de disponibilidad de cursos y docentes, centrándose en la asignación correcta de los horarios. Para este caso, la prioridad fue muy alta (4) y la complejidad de tercer nivel.

**Tabla 8***Historia de usuario – Sprint 6*

<b>Historia de usuario</b>			
<b>Nombre de usuario: HU6 Generación de horarios</b>			
Prioridad	4	Complejidad	3
Descripción	Comprende el visualizar los horarios generados automáticamente por el sistema, verificando que no existan conflictos. Se incluye la generación de reportes.		
Criterios de aceptación	<ul style="list-style-type: none"> <li>• Horarios generados sin conflictos</li> <li>• Reporte generado de horarios a docentes.</li> </ul>		

*Nota:* En la tabla se describe lo que incluirá el Sprint 6 para el desarrollo del sistema web, fuente y elaboración propia

Como último Sprint, su finalidad es lograr la visualización de los horarios sin ningún conflicto, además de la emisión de los reportes en donde constan los horarios

asignados a los educadores. Para este caso, la prioridad fue muy alta (4) y la complejidad de tercer nivel. A fin de explicar de mejor manera el contenido de cada sprint, se presenta a continuación el Product Backlog:

**Tabla 9**

*Product Backlog*

Ítem	Sprint	Prioridad	Complejidad	Estimación
HU1	Análisis de requerimientos del usuario	4	2	2d
HU2	Análisis y diseño de bases de datos	4	3	5d
HU3	Diseño y maquetación de las pantallas	4	4	5d
HU4	Almacenamiento de la información.	4	3	8d
HU5	Configuración de disponibilidad de horarios	4	3	5d
HU6	Generación de horarios	4	3	3d
<b>SISTEMA WEB</b>		4	3	28d

*Nota:* En la tabla se presenta el Product Backlog, detallando cada Sprint, su prioridad, complejidad y estimación en días, fuente y elaboración propia

Considerando que el desarrollo del sistema web consta de cuatro etapas o sprint, a las cuales se asignó un nivel de prioridad y complejidad, logra observarse que cada una de ellas presenta una calificación similar, con una prioridad muy alta (4) y complejidad de tercer nivel. Por otra parte, en lo que respecta a su estimación, el tercer sprint tendrá una mayor duración al ser la parte medular del proyecto, tardando 10 días y centrándose en asignar los horarios a los docentes.

El segundo sprint con mayor duración es el tercero que implica la generación y visualización de reportes con 7 días; mientras los restantes tendrán 5 días. Así, se estima que el desarrollo del sistema web tardará 27 días.

### 2.2.2. Sprint Backlog.

En esta sección se detalla cómo se encuentra estructurado cada Sprint, exponiendo las actividades, su prioridad y complejidad y estimación. Cabe señalar que el promedio de las prioridades y complejidades en cada etapa da como resultado la prioridad y complejidad del sprint; mientras que la suma de las estimaciones, arrojó el tiempo que tardaría en ejecutarse cada uno.

**Tabla 10**

*Sprint 1*

<b>SPRINT 1: Análisis de requerimientos del usuario</b>				
<b>Ítem</b>	<b>Descripción</b>	<b>Prioridad</b>	<b>Complejidad</b>	<b>Estimación</b>
1.1.	Levantamiento de información	4	2	1d
1.2.	Estructuración de procesos	4	4	1d
1.3.	Planificación	4	2	1d
1.4.	Análisis de las características del sistema	4	4	1d
<b>SPRINT 1</b>		<b>4</b>	<b>3</b>	<b>4d</b>

*Nota:* En la tabla se presenta la composición del Sprint 1, su prioridad, complejidad y estimación en días, fuente y elaboración propia

El Sprint 1 consta de cuatro actividades, cada una con una prioridad muy alta (4); sin embargo, su complejidad varía entre moderada y muy alta. Las que resultan de complejidad alta o de cuarto nivel son el levantamiento de información y el análisis de la característica del sistema, planteándose así el esquema de las operaciones y el esquema de operaciones. Por otra parte, las de complejidad moderada o de segundo nivel involucran el levantamiento de información y la planificación del diseño.

**Tabla 11***Sprint 2*

<b>SPRINT 2: Análisis y diseño de bases de datos</b>				
<b>Ítem</b>	<b>Descripción</b>	<b>Prioridad</b>	<b>Complejidad</b>	<b>Estimación</b>
2.1.	Diseño conceptual	4	3	2d
2.2.	Análisis y creación del modelo entidad relación.	4	3	3d
<b>SPRINT 2</b>		<b>4</b>	<b>3</b>	<b>5d</b>

*Nota:* En la tabla se presenta la composición del Sprint 2, su prioridad, complejidad y estimación en días, fuente y elaboración propia

El Sprint 2 consta de un total de dos actividades, cada una con una prioridad muy alta (4) y complejidad de tercer nivel, dando una prioridad promedio de 4 y una complejidad de 3 para esta etapa. Entre ambas, la mayor duración correspondió a la segunda actividad con tres días; mientras la primera actividad presenta una estimación de dos días. Así, se estimó que el sprint 2 terminaría en cinco días.

**Tabla 12***Sprint 3*

<b>SPRINT 3: Diseño y maquetación de las pantallas</b>				
<b>Ítem</b>	<b>Descripción</b>	<b>Prioridad</b>	<b>Complejidad</b>	<b>Estimación</b>
3.1.	Creación de los controles y funciones dentro del sistema	4	4	2d
3.2.	Creación de las vistas y validaciones de usuarios	4	4	3d
<b>SPRINT 3</b>		<b>4</b>	<b>4</b>	<b>5d</b>

*Nota:* En la tabla se presenta la composición del Sprint 3, su prioridad, complejidad y estimación en días, fuente y elaboración propia

El Sprint 3 consta de dos actividades, cada una con una prioridad muy alta (4) y complejidad de cuarto nivel por ser una parte esencial en el desarrollo del sistema. Esto da una prioridad promedio de 4 y una complejidad muy alta para esta etapa. Entre ambas, la mayor duración correspondió a la segunda actividad con tres días; mientras la primera actividad presenta una estimación de dos días. Así, se estimó que el sprint 3 terminaría en cinco días.

**Tabla 13**

*Sprint 4*

<b>SPRINT 4: Almacenamiento de la información.</b>				
<b>Ítem</b>	<b>Descripción</b>	<b>Prioridad</b>	<b>Complejidad</b>	<b>Estimación</b>
4.1.	Realización de pruebas	4	4	2d
4.2.	Configuración de periodo en cantidad de horas y días	4	4	1d
4.3.	Configuración de asignaturas	4	4	1d
4.4.	Configuración de cursos	4	4	1d
4.5.	Configuración de docentes	4	4	1d
4.6.	Testeo del sistema web	4	2	1d
4.7.	Desarrollo de ajustes requeridos	4	2	1d
<b>SPRINT 4</b>		<b>4</b>	<b>3</b>	<b>8d</b>

*Nota:* En la tabla se presenta la composición del Sprint 4, su prioridad, complejidad y estimación en días, fuente y elaboración propia

El Sprint 4 presenta un total de siete actividades, siendo el más largo dentro del SCRUM. Considerando que consiste en el almacenamiento de la información, tiene actividades como la realización de pruebas para verificar que toda la información pueda guardarse correctamente, para luego configurar el periodo, asignaturas, cursos y docentes, lo cual es esencial para asignar horarios. Para estas cinco primeras actividades la prioridad se ubica en muy alta (4) y complejidad de cuarto nivel.

Mientras tanto, las dos actividades restantes reflejan una prioridad muy alta (4) y complejidad de segundo nivel o moderada. Esto implica el testeo del sistema y el desarrollo de posibles ajustes que se requieren, arrojando una prioridad promedio de cuatro y una complejidad de tres, estimándose que el sprint 4 terminaría en ocho días.

**Tabla 14**

*Sprint 5*

<b>SPRINT 5: Configuración de disponibilidad de horarios</b>				
<b>Ítem</b>	<b>Descripción</b>	<b>Prioridad</b>	<b>Complejidad</b>	<b>Estimación</b>
5.1.	Configuración de disponibilidad de cursos	4	4	2d
5.2.	Configuración de disponibilidad de docentes	4	4	2d
5.3.	Asignación de horarios	4	2	1d
<b>SPRINT 5</b>		<b>4</b>	<b>3</b>	<b>5d</b>

*Nota:* En la tabla se presenta la composición del Sprint 5, su prioridad, complejidad y estimación en días, fuente y elaboración propia

El Sprint 5 consta de tres actividades, teniendo las dos primeras una prioridad muy alta (4) y complejidad de cuarto nivel; mientras la tercera registra la misma prioridad pero una complejidad de segundo nivel. Esta valoración arrojó una prioridad promedio de cuatro y una complejidad de tres.

Entre las actividades señaladas, la mayor duración correspondió a la primera y la segunda con dos días, mientras la tercera se estima en un día. Así, se determinó que el sprint 5 terminaría en cinco días.

**Tabla 15***Sprint 6*

<b>SPRINT 6: Generación de horarios</b>				
<b>Ítem</b>	<b>Descripción</b>	<b>Prioridad</b>	<b>Complejidad</b>	<b>Estimación</b>
6.1.	Generación automática de horarios	4	4	1d
6.2.	Configuración manual de horarios en conflicto	4	4	1d
6.3.	Exportación o generación de reportes	4	2	1d
<b>SPRINT 6</b>		<b>4</b>	<b>3</b>	<b>3d</b>

*Nota:* En la tabla se presenta la composición del Sprint 6, su prioridad, complejidad y estimación en días, fuente y elaboración propia

El Sprint 6 se compone de tres actividades, teniendo las dos primeras una prioridad muy alta (4) y complejidad de cuarto nivel; mientras la última registra la misma prioridad, pero una complejidad de segundo nivel. Esta valoración arrojó una prioridad promedio de cuatro y una complejidad de tres, considerando que cada actividad tendría una estimación de un día. Así, para cumplir el sprint 6 se plantea un plazo de tres días.

A continuación, se presenta el sprint backlog, detallando las fechas de inicio y fin para cada etapa:

**Tabla 16***Sprint backlog*

<b>SPRINT</b>	<b>Días</b>	<b>Fecha inicio</b>	<b>Fecha fin</b>
Análisis de requerimientos del usuario	4	1/10/2021	5/10/2021
Análisis y diseño de bases de datos	5	7/10/2021	12/10/2021
Diseño y maquetación de las pantallas	5	14/10/2021	19/10/2021
Almacenamiento de la información	8	21/10/2021	29/10/2021
Configuración de disponibilidad de horarios	5	31/10/2021	5/11/2021
Generación de horarios	3	7/11/2021	10/11/2021
<b>SISTEMA WEB</b>	30	1/10/2021	10/11/2021

*Nota:* En la tabla se presenta la duración de cada Sprint con su fecha de inicio y fin, fuente y elaboración propia

**2.2.3. Sprint Planning.**

En esta sección se presenta el Sprint Planning, incluyendo las fechas para la revisión de cada etapa:

**Tabla 17***Sprint planning 1*

<b>SPRINT 1: Análisis de requerimientos del usuario</b>	
<b>Fecha de inicio</b>	1/10/2021
<b>Fecha de fin</b>	5/10/2021
<b>Revisión de los avances</b>	6/10/2021
<b>Tareas a desarrollar</b>	1.1. Levantamiento de información 1.2. Estructuración de procesos 1.3. Planificación 1.4. Análisis de las características del sistema

*Nota:* En la tabla se presenta la fecha de inicio y fin del Sprint 1 y la revisión de los avances, fuente y elaboración propia

**Tabla 18***Sprint planning 2*

<b>SPRINT 2: Análisis y diseño de bases de datos</b>	
<b>Fecha de inicio</b>	7/10/2021
<b>Fecha de fin</b>	12/10/2021
<b>Revisión de los avances</b>	13/10/2021
<b>Tareas a desarrollar</b>	2.1. Diseño conceptual 2.2. Análisis y creación del modelo entidad relación.

*Nota:* En la tabla se presenta la fecha de inicio y fin del Sprint 2 y la revisión de los avances, fuente y elaboración propia

**Tabla 19***Sprint planning 3*

<b>SPRINT 3: Diseño y maquetación de las pantallas</b>	
<b>Fecha de inicio</b>	14/10/2021
<b>Fecha de fin</b>	19/10/2021
<b>Revisión de los avances</b>	20/10/2021
<b>Tareas a desarrollar</b>	3.1. Creación de los controles y funciones dentro del sistema 3.2. Creación de las vistas y validaciones de usuarios

*Nota:* En la tabla se presenta la fecha de inicio y fin del Sprint 3 y la revisión de los avances, fuente y elaboración propia

**Tabla 20***Sprint planning 4*

<b>SPRINT 4: Almacenamiento de la información</b>	
<b>Fecha de inicio</b>	21/10/2021
<b>Fecha de fin</b>	29/10/2021
<b>Revisión de los avances</b>	27/10/2021 30/10/2021
<b>Tareas a desarrollar</b>	4.1. Realización de pruebas 4.2. Configuración de periodo en cantidad de horas y días 4.3. Configuración de asignaturas 4.4. Configuración de cursos 4.5 Configuración de docentes 4.6. Testeo del sistema web 4.7. Desarrollo de ajustes requeridos

*Nota:* En la tabla se presenta la fecha de inicio y fin del Sprint 4 y la revisión de los avances, fuente y elaboración propia

**Tabla 21***Sprint planning 5*

<b>SPRINT 5: Configuración de disponibilidad de horarios</b>	
<b>Fecha de inicio</b>	31/10/2021
<b>Fecha de fin</b>	5/11/2021
<b>Revisión de los avances</b>	6/11/2021
<b>Tareas a desarrollar</b>	5.1. Configuración de disponibilidad de cursos 5.2. Configuración de disponibilidad de docentes 5.3. Asignación de horarios

*Nota:* En la tabla se presenta la fecha de inicio y fin del Sprint 5 y la revisión de los avances, fuente y elaboración propia

**Tabla 22***Sprint planning 6*

<b>SPRINT 6: Generación de horarios</b>	
<b>Fecha de inicio</b>	7/11/2021
<b>Fecha de fin</b>	10/11/2021
<b>Revisión de los avances</b>	11/11/2021
<b>Tareas a desarrollar</b>	6.1. Generación automática de horarios 6.2. Configuración manual de horarios en conflicto 6.3. Exportación o generación de reportes

*Nota:* En la tabla se presenta la fecha de inicio y fin del Sprint 6 y la revisión de los avances, fuente y elaboración propia

**2.2.4. Incremento / Sprints, Burndown Chart y demás.**

En esta sección se muestra la tabla que permitió el registro de los incrementos en la medida que se cumplían los Sprint y las actividades dentro de cada uno, cuyo desarrollo permitió la obtención del sistema web para la creación y gestión de horarios en la FACI.

**Tabla 23***Incremento / Sprint (parte 1)*

<b>INICIO: 3/10/2021</b>				
<b>FIN: 10/11/2021</b>				
<b>HISTORIA DE USUARIO</b>		<b>PENDIENTE</b>	<b>EN CURSO</b>	<b>HECHO</b>
SPRINT 1	1.1. Levantamiento de información			X
	1.2. Estructuración de procesos			X
	1.3. Planificación			X
	1.4. Análisis de las características del sistema			X
SPRINT 2	2.1. Diseño conceptual			X
	2.2. Análisis y creación del modelo entidad relación			X

*Nota:* En la tabla se presenta el estado en el cual se encuentra el desarrollo del sistema web, fuente y elaboración propia

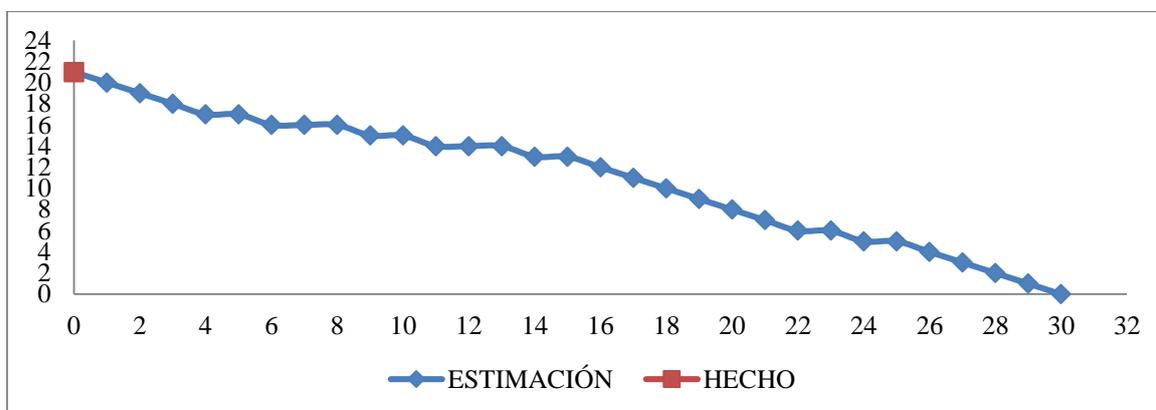
**Tabla 24***Incremento / Sprint (parte 2)*

<b>INICIO:</b> 3/10/2021				
<b>FIN:</b> 10/11/2021				
<b>HISTORIA DE USUARIO</b>		<b>PENDIENTE</b>	<b>EN CURSO</b>	<b>HECHO</b>
SPRINT 3	3.1. Creación de los controles y funciones dentro del sistema			X
	3.2. Creación de las vistas y validaciones de usuarios			X
SPRINT 4	4.1. Realización de pruebas			X
	4.2. Configuración de periodo en cantidad de horas y días			X
	4.3. Configuración de asignaturas			X
	4.4. Configuración de cursos			X
	4.5 Configuración de docentes			X
	4.6. Testeo del sistema web			X
	4.7. Desarrollo de ajustes requeridos			X
SPRINT 3	5.1. Configuración de disponibilidad de cursos			X
	5.2. Configuración de disponibilidad de docentes			X
	5.3. Asignación de horarios			X
SPRINT 6	6.1. Generación automática de horarios			X
	6.2. Configuración manual de horarios en conflicto			X
	6.3. Exportación o generación de reportes			X

*Nota:* En la tabla se presenta el estado en el cual se encuentra el desarrollo del sistema web, fuente y elaboración propia

Al iniciar, todas las actividades se encontraban en estado pendiente, pero, en la medida que fueron completadas, se empezaron a marcar en la columna *hecho*. Con lo descrito, se presenta a continuación el *Burndown chart* o diagrama de quemado para el desarrollo del sistema web:

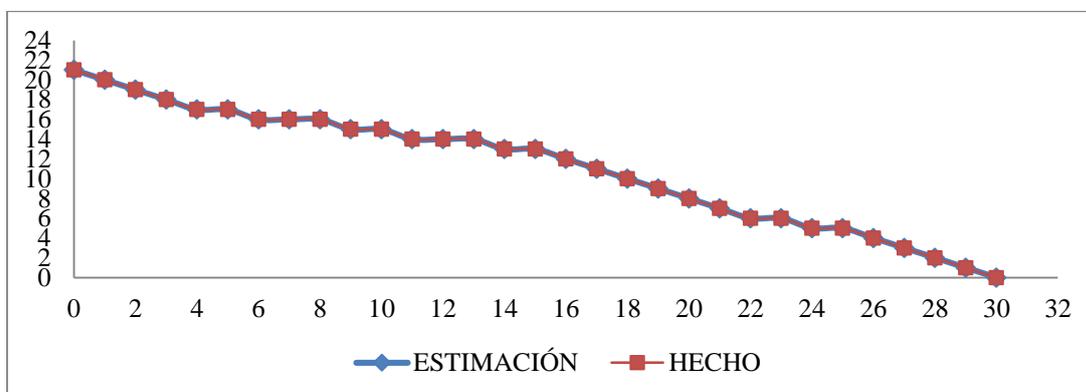
Figura 7

*Burndown charts para el desarrollo del sistema web - inicial*

*Nota:* Estimación para el desarrollo del sistema web, fuente y elaboración propia

En la figura 7, el eje horizontal correspondió a los días y el vertical al número de actividades que tiene el proyecto, siendo 21 al sumar las establecidas en cada sprint. Así, la línea de estimación inicia en la actividad 21 y, mientras se avanza en el desarrollo del sistema, las actividades van disminuyendo según lo programado, hasta el día 30 cuando las actividades restantes son cero. Por otro lado, también se grafica el avance del sistema mismo que, al iniciar, se encontraba en 21 actividades y cero días transcurridos, terminando en 0 actividades con 30 días transcurridos.

Figura 8

*Burndown charts para el desarrollo del sistema web – final*

*Nota:* Cumplimiento del desarrollo del sistema web, fuente y elaboración propia

## CAPÍTULO 3

### 3. Propuesta de solución

#### 3.1. Análisis de factibilidad

En esta sección se presentan las posibilidades de éxito del sistema web para la creación y gestión de horarios de los docentes en la FACI, abordando tanto la factibilidad técnica como la económica.

##### 3.1.1. Factibilidad técnica

Dentro de la factibilidad técnica se pretende conocer las características del Hardware y Software utilizados para el desarrollo del sistema web y, de esta manera, determinar su grado de disponibilidad en el mercado según los requerimientos técnicos. A continuación, se detallan cada uno de ellos:

**Tabla 25**

*Requisitos técnicos de hardware para el desarrollo del sistema web*

Hardware	Descripción
Computadora	Procesador: Intel Core i3 Sistema operativo: Windows 10 Disco Duro: 512 GB Memoria RAM: 4GB

*Nota:* Se presenta el hardware utilizado para el desarrollo del sistema, fuente y elaboración propia

**Tabla 26**

*Requisitos técnicos de software para el desarrollo del sistema web*

Software	Descripción
Entorno de Desarrollo Integrador	Python 3.9.6
Framework	Django 3.2.6
Gestor de base de datos	PostgreSQL 12

*Nota:* Se presenta el software utilizado para el desarrollo del sistema, fuente y elaboración propia

Con lo descrito, el desarrollo del sistema web presenta una factibilidad técnica porque hace uso de un hardware de gran disponibilidad y común en el mercado, sin necesidad de recurrir a modelos de última generación con una elevada capacidad de procesamiento. Esto garantiza que el desarrollo pueda ser realizado de manera ágil, sin retrasos por la búsqueda de un equipo que cumpla estándares muy altos respecto a requisitos técnicos.

Por otra parte, el software se encuentra conformado principalmente por tecnologías Open Source o de código abierto, en versiones que están disponibles al público e incluso, debido a su popularidad, existe una amplia comunidad de usuarios, casos de uso y demás información que ayudó al desarrollo del sistema web propuesto.

### 3.1.2. Factibilidad económica

En este punto se pretende demostrar si el sistema web propuesto es factible económicamente, analizando los costos de su desarrollo y el beneficio que se obtendría tras su implementación. A continuación, se presentan los costos que demandaría el desarrollo:

**Tabla 27**

*Costos de hardware para el desarrollo del sistema web*

<b>COSTOS DE HARDWARE</b>		
1	Computadora	USD 500,00
	<b>Total</b>	<b>USD 500,00</b>

*Nota:* Se presenta el costo del hardware utilizado para el desarrollo del sistema web, fuente y elaboración propia

**Tabla 28***Costos de software para el desarrollo del sistema web*

<b>COSTOS DE SOFTWARE</b>			
1	Python 3.9.6	USD	0,00
1	Django 3.2.6	USD	0,00
1	PostgreSQL 12	USD	0,00
	<b>Total</b>	<b>USD</b>	<b>0,00</b>

*Nota:* Se presenta el costo del software empleado para el desarrollo del sistema web, fuente y elaboración propia

Al utilizar tecnologías Open Source y gratuitas, el software no registra ningún costo, lo cual es una ventaja en términos económicos.

**Tabla 29***Costos de recurso humano para el desarrollo del sistema web*

<b>COSTOS DE RECURSO HUMANO</b>		
1	Programador (honorario por desarrollo del sistema)	USD 800,00
	<b>Total</b>	<b>USD 800,00</b>

*Nota:* Se presenta el costo del hardware empleado para el desarrollo del sistema web, fuente y elaboración propia

**Tabla 30***Costo total para el desarrollo del sistema web*

<b>COSTO TOTAL DEL SISTEMA WEB</b>	
Costos de hardware	USD 500,00
Costos de software	USD 00,00
Costos de recurso humano	USD 800,00
<b>Total</b>	<b>USD 1300,00</b>

*Nota:* En la tabla se presentan los costos totales para el desarrollo del sistema web, fuente y elaboración propia

El desarrollo del sistema para la creación y gestión de horarios a los docentes de la FACI tendría un costo total de USD 1.300. Así, la institución dispondrá de un sistema web funcional que se ajuste a sus necesidades, permitiendo una asignación correcta y eficiente de horarios para los educadores, reduciendo el malestar de los profesionales y garantizando que los estudiantes reciban una formación suficiente, acorde a los tiempos que maneja cada materia.

Además, su uso no demandaría costos por concepto de licencias, lo cual es otro beneficio del sistema propuesto. Considerando la ausencia de costos futuros por licencias y la obtención de un sistema web funciones que responda a las exigencias de la FACI, la propuesta demuestra su factibilidad económica.

### **3.2. Propuesta**

La propuesta involucra el desarrollo de un sistema web que aporte a la creación y gestión de horarios de los docentes en la FACI, permitiendo la coordinación efectiva del ejercicio profesional de los educadores. De esta manera se evitaría el cruce de horarios y las confusiones entre los docentes, además de los estudiantes que, ante estos conflictos, no reciben una enseñanza acorde a los tiempos que exige el programa académico.

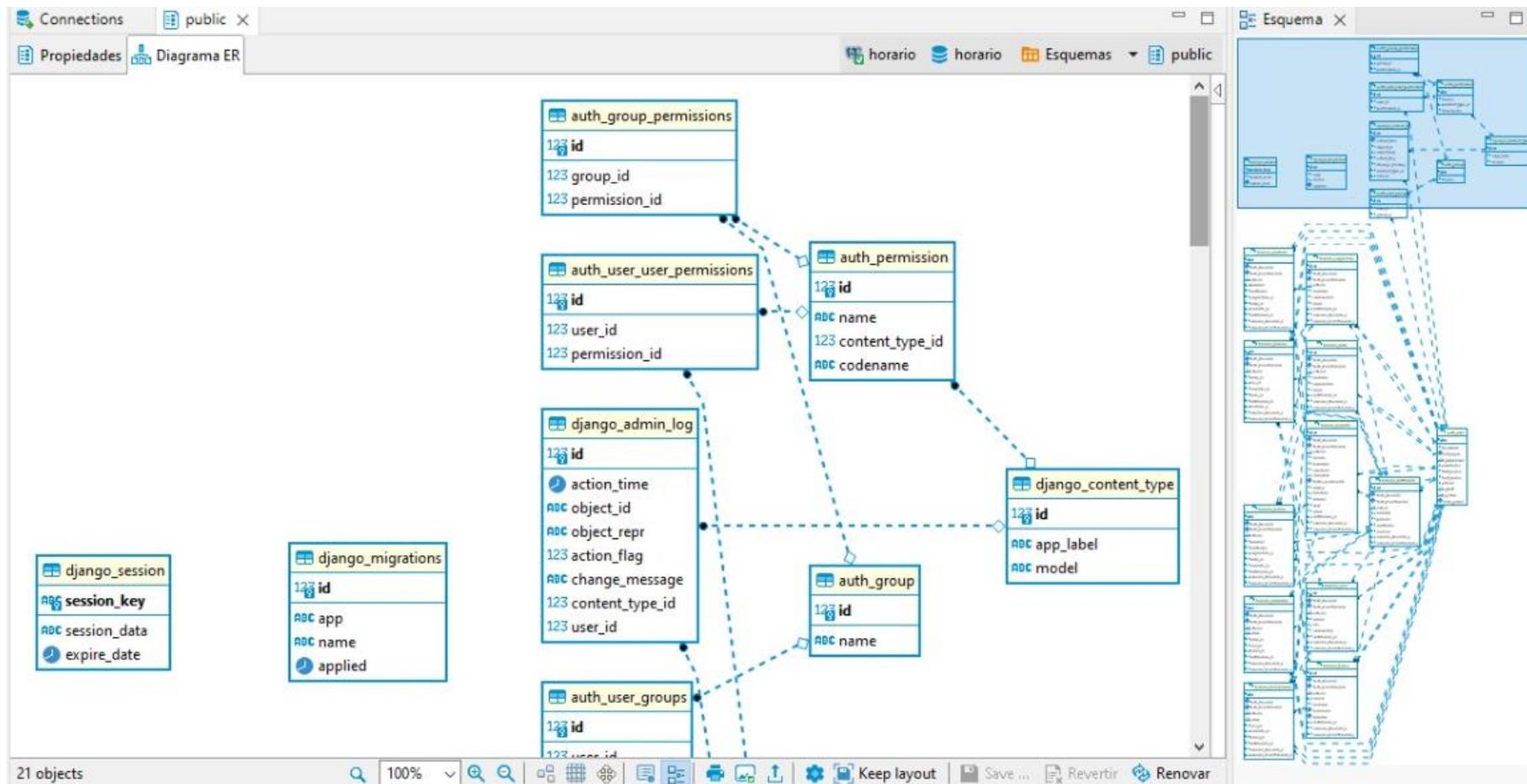
#### **3.2.1. Diagrama de base de datos**

En este punto se presenta el diseño conceptual de la base de datos a través del modelo entidad-relación, el cual expone cómo las entidades se relacionan dentro del sistema web. Cabe señalar que el sistema corresponde al propuesto, el cual se encuentra enfocado en la creación y gestión de horarios a los docentes de la FACI.

Al ser un diagrama extenso, se presenta su estructura por partes utilizando el software DBBeaver 21.2.5:

Figura 9

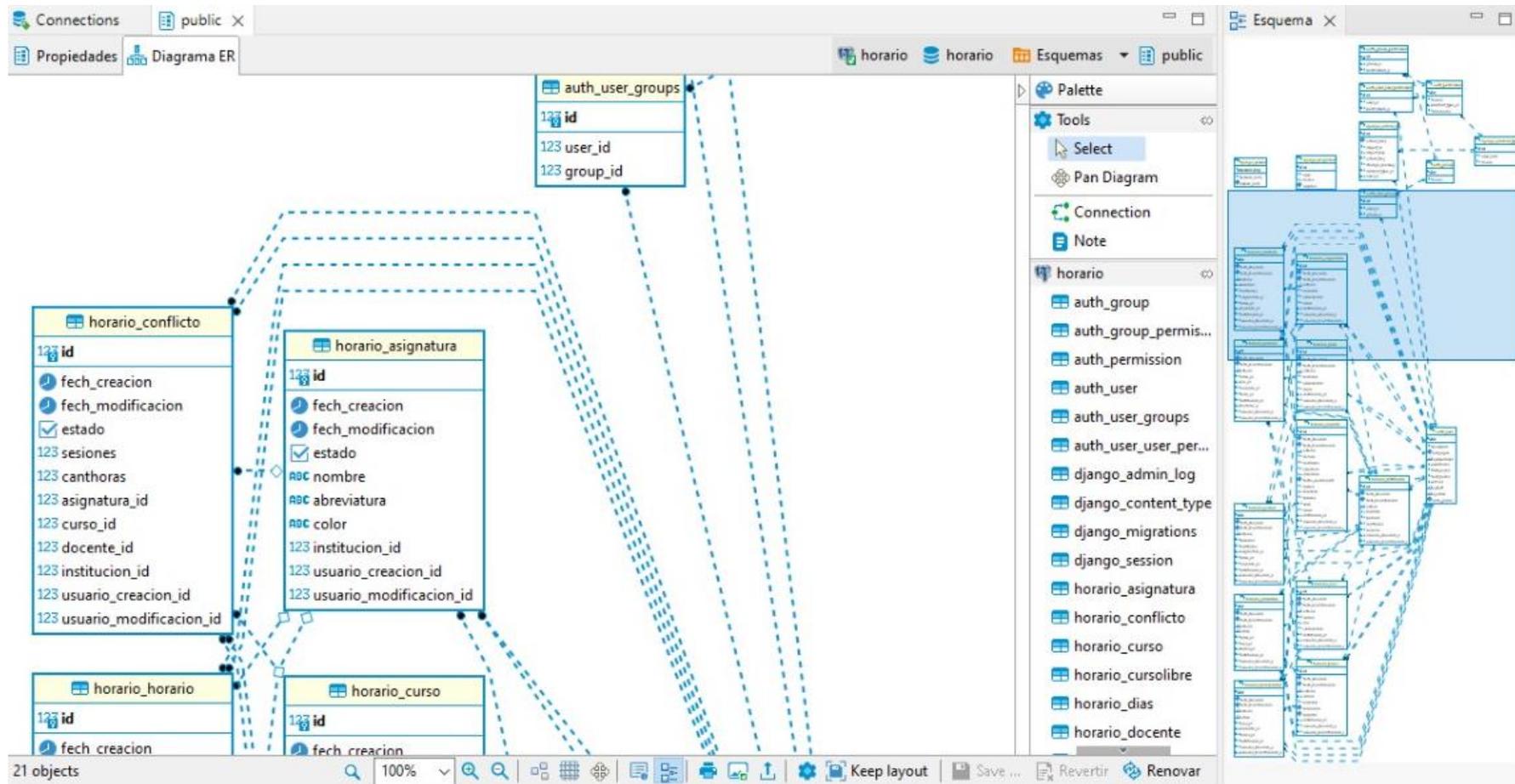
*Modelo de entidad relación de la base de datos (parte 1)*



*Nota:* Se presenta el modelo entidad-relación sacada del Sistema gestor de bases de datos, fuente y elaboración propia

Figura 10

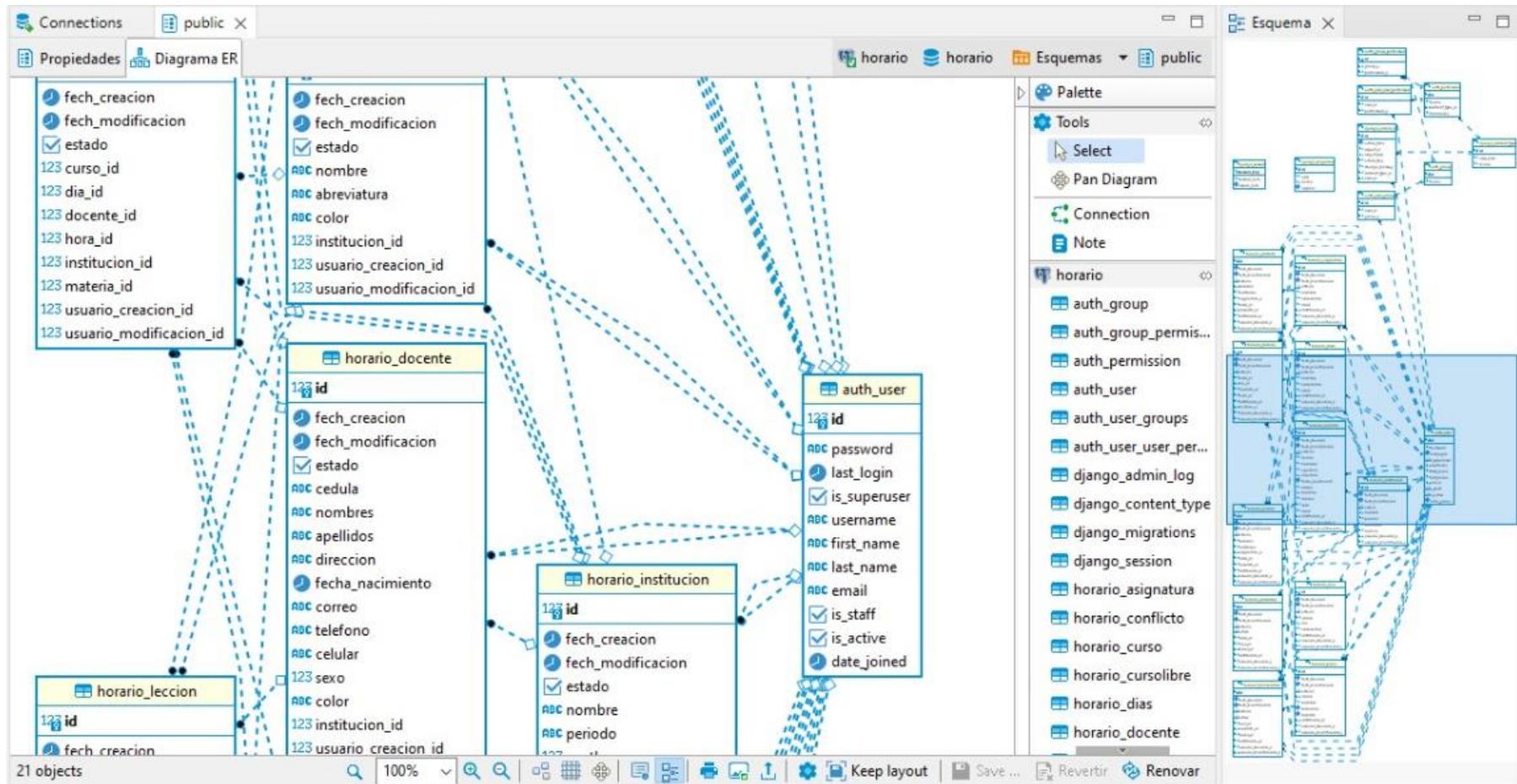
Modelo de entidad relación de la base de datos (parte 2)



Nota: Se presenta el modelo entidad-relación sacada del Sistema gestor de bases de datos, fuente y elaboración propia

Figura 11

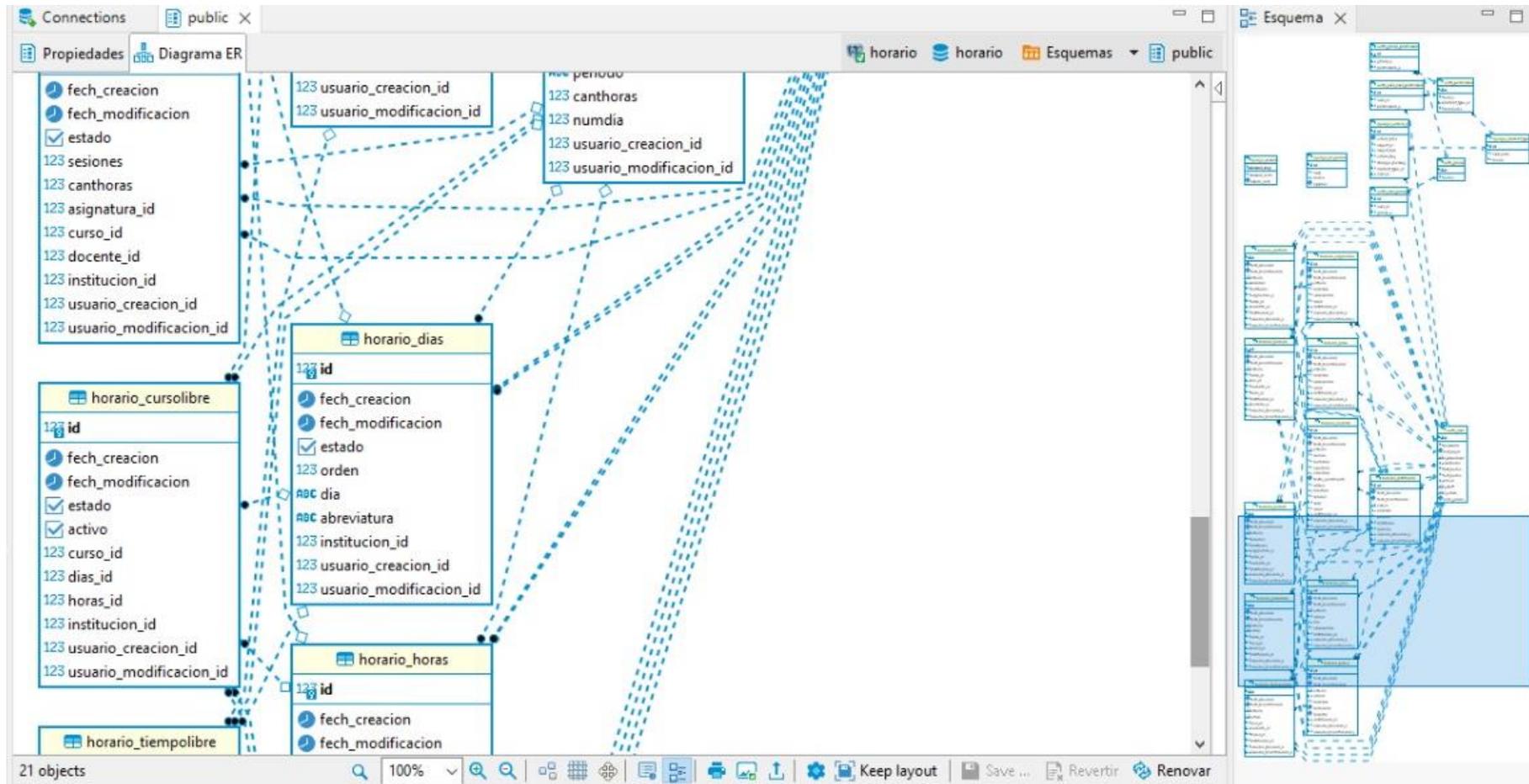
*Modelo de entidad relación de la base de datos (parte 3)*



*Nota:* Se presenta el modelo entidad-relación sacada del Sistema gestor de bases de datos, fuente y elaboración propia

Figura 12

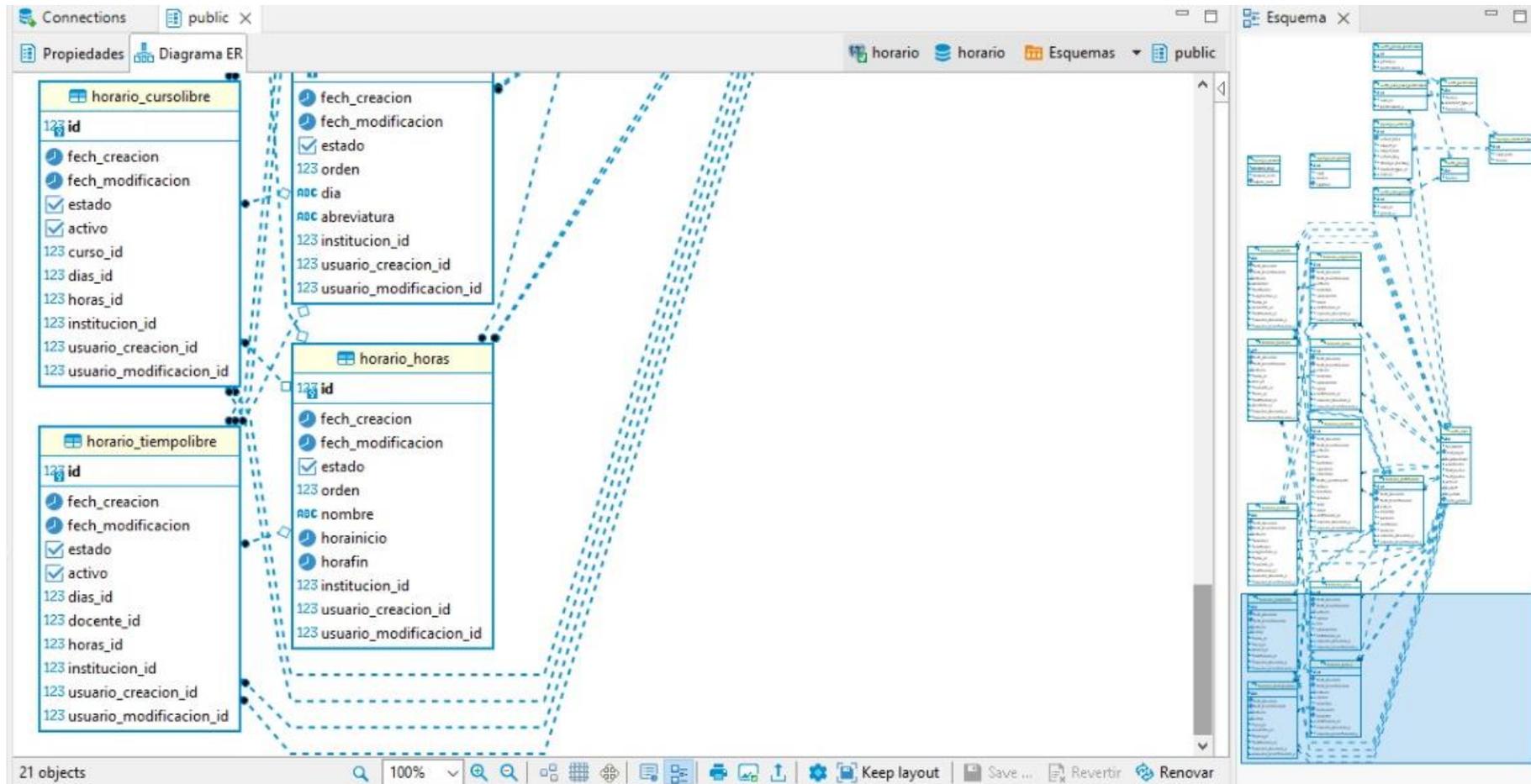
*Modelo de entidad relación de la base de datos (parte 4)*



*Nota:* Se presenta el modelo entidad-relación sacada del Sistema gestor de bases de datos, fuente y elaboración propia

Figura 13

Modelo de entidad relación de la base de datos (parte 5)



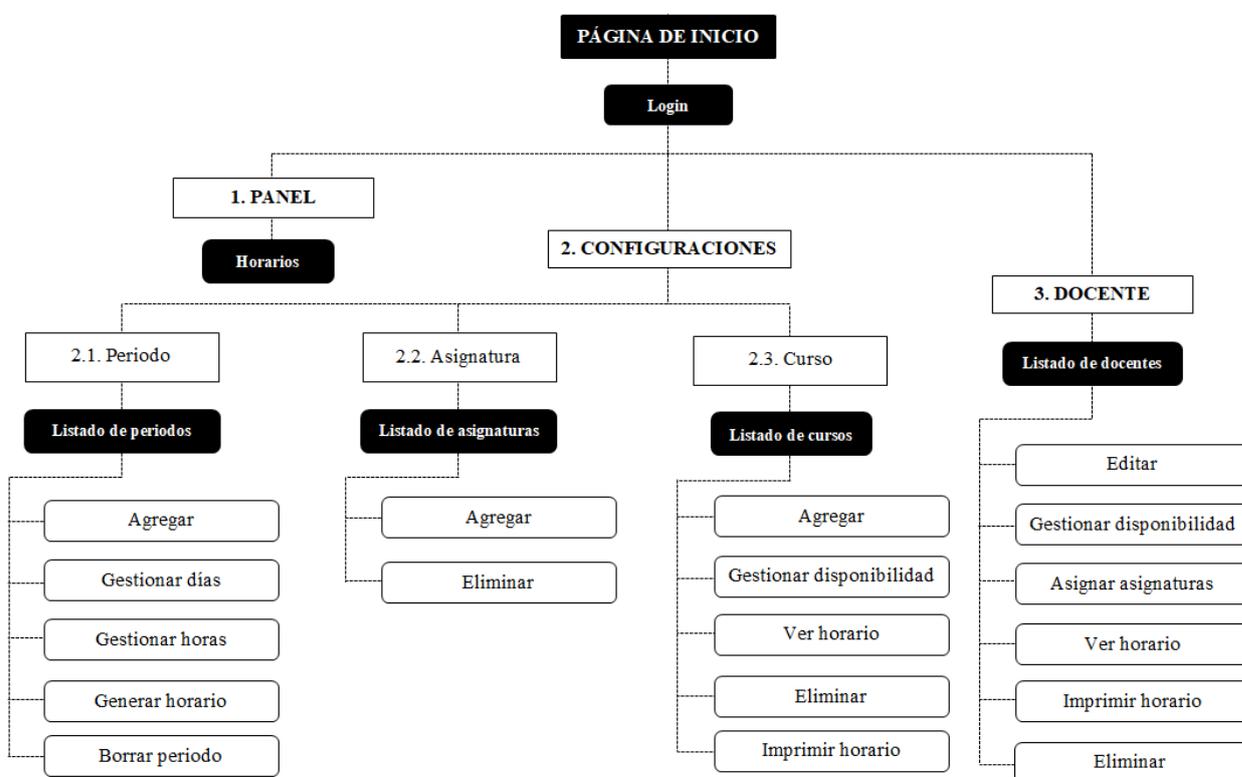
*Nota:* Se presenta el modelo entidad-relación sacada del Sistema gestor de bases de datos, fuente y elaboración propia

### 3.2.2. Diagrama de aplicación

Este diagrama plasma la estructura de la navegación dentro del sistema web, utilizándose en el presente estudio para describir las pantallas o pestañas diseñadas, las opciones y botones que permitirán que el sistema cumpla el objetivo de su diseño. A continuación, se presenta dicho diagrama:

Figura 14

*Diagrama de aplicación del sistema web propuesto*



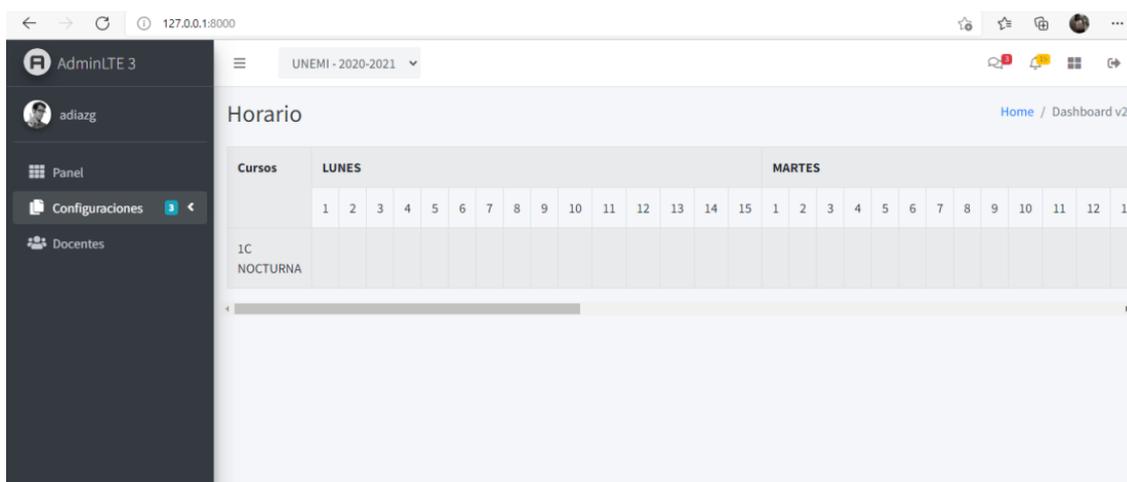
*Nota:* Se presenta la estructura de la navegación dentro del sistema web que seguirá cada usuario, fuente y elaboración propia

### 3.2.3. Desarrollo

Este sistema web fue desarrollado en Python 3.9.6, presentando a continuación cómo se encuentra diseñado:

Figura 15

*Pantalla de inicio del sistema web para la FACI*

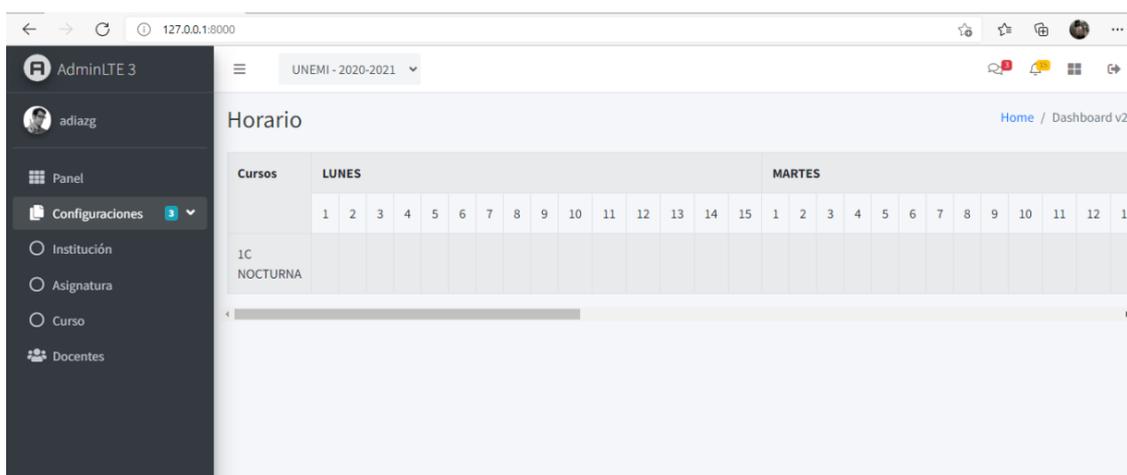


*Nota:* Pantalla inicial o principal del sistema, en la cual puede accederse a todas las funciones, fuente y elaboración propia

En la Figura 15 se muestra la pantalla principal que el usuario observará una vez ingrese al sistema web, desde la cual podrá seleccionar las opciones que ofrece para la creación y coordinación de los horarios a los docentes de la FACI.

Figura 16

*Opciones para la configuración de horarios*



*Nota:* Se presentan las opciones para crear los horarios al seleccionar la pestaña de configuración, fuente y elaboración propia

Al seleccionar la pestaña de *configuraciones* aparecen tres opciones que involucran: periodo, asignaturas y cursos, lo cual se observa en la Figura 16. Entre estas opciones, en *periodo* se ingresa el ciclo que corresponde para la FACI.

Figura 17

*Configuración de periodo – vista inicial*

Nombre	Período	Días	Horas	Acciones
ES CPRU	2021-2022	5	6	[Iconos de acciones]

*Nota:* Se presenta el listado de los periodos que han sido configurados, fuente y elaboración propia

Figura 18

*Configuración en la opción periodo - creación del periodo*

*Nota:* Se presentan los parámetros bajo los cuales se configura el periodo en el sistema web, fuente y elaboración propia

Cuando se ingresa a la opción de *periodo*, debe seleccionarse el botón *agregar*, con lo cual se desplegará un menú para ingresar el código o nombre del periodo, el periodo que corresponde en años, horas y número de días que se ha trabajará, lo cual se puede observar en la Figura 18. Respecto a esto último, no todas las unidades académicas trabajan todos los días, considerando que la Unidad de Educación trabaja de lunes a domingo y la Unidad de Ingeniería en Sistema trabaja solamente de lunes a viernes.

También se deberá ingresar el número de horas que se trabajará al día, ya que algunas laboran solamente en la tarde y otras en la noche. Al ingresar todos los datos, se habrá creado el periodo y se presentará en el listado de periodos.

Una vez ingresado el periodo, se procede a crear el listado de asignaturas, las cuales deben ser ingresadas bajo determinados parámetros expuestos en la Figura 18. Por ejemplo, en la FACI hay materias como Programas en Java, entre otras, que deben ser colocadas de una a una.

Figura 19

*Configuración en la opción asignaturas – vista inicial*

Nombre	Abreviatura	Color	Acciones
CIENCIAS NATURALES	CN	Verde	[Icono de editar] [Icono de eliminar]
ESTUDIOS SOCIALES	ES	Marrón	[Icono de editar] [Icono de eliminar]
HISTORIA	hs	Rosa	[Icono de editar] [Icono de eliminar]

*Nota:* Se presenta el listado de las asignaturas que han sido configuradas, fuente y elaboración propia

Figura 20

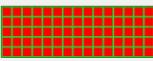
*Configuración en la opción asignaturas – creación de asignaturas*

*Nota:* Se presentan los parámetros bajo los cuales se configuran las asignaturas en el sistema web, fuente y elaboración propia

Para acceder al menú presentado en la Figura 20 se debe seleccionar el botón *agregar*, apareciendo la pantalla que permitirá incluir el nombre, abreviatura y color de cada asignatura. Esto permitirá una rápida identificación de cada materia, facilitando la creación y gestión de los horarios. Luego de haber ingresado el periodo y configurado las asignaturas, se procede a la configuración del curso, lo cual se muestra en la Figura 21.

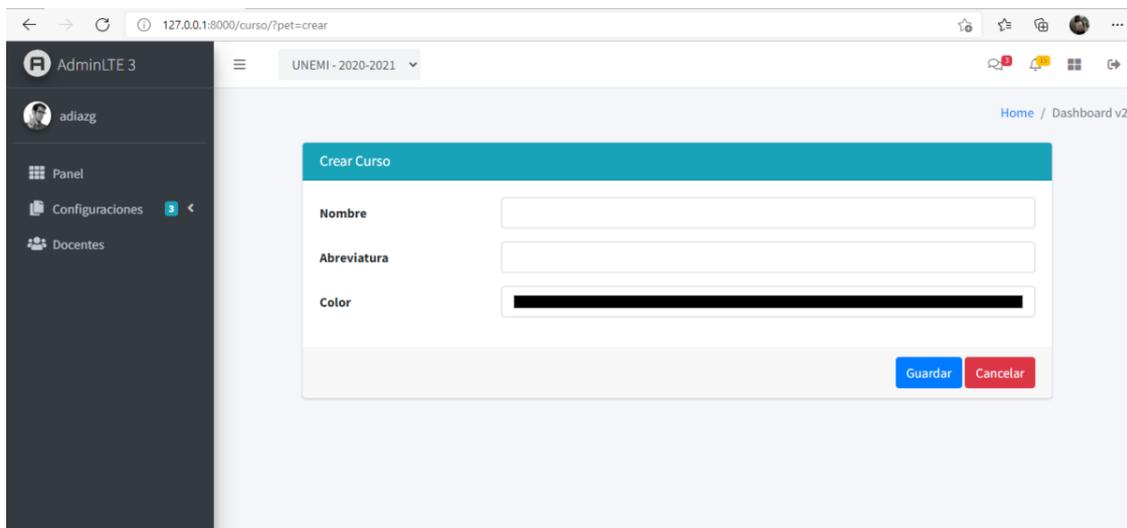
Figura 21

*Configuración en la opción cursos – vista inicial*

Nombre	Abreviatura	Color	Disponibilidad	Acciones
1C NOCTURNA	1cnoc	<span style="background-color: green; width: 20px; height: 15px; display: inline-block;"></span>		

*Nota:* Se presenta el listado de cursos que han sido configurados, fuente y elaboración propia

Figura 22

*Configuración en la opción cursos – creación de cursos*

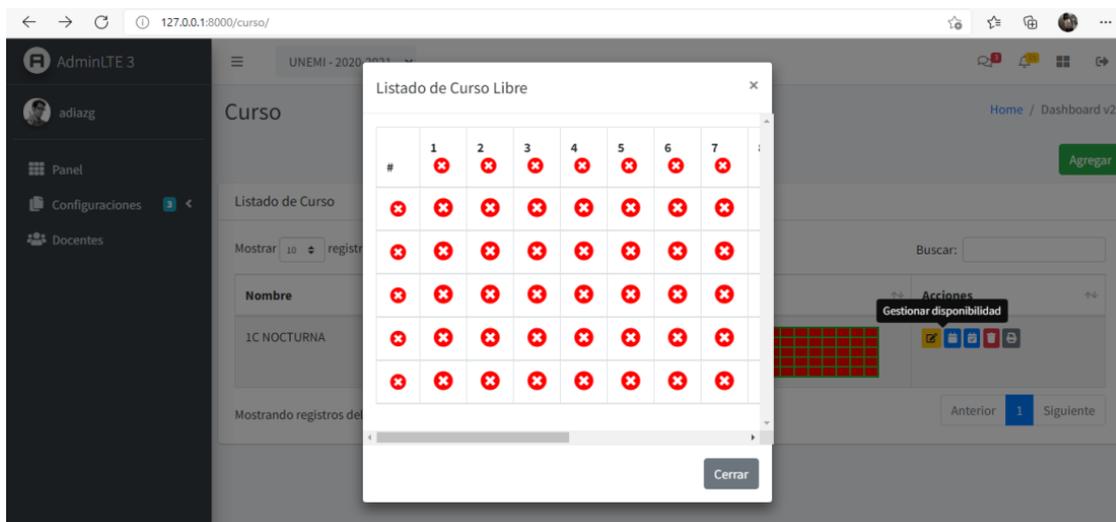
*Nota:* Se presentan los parámetros bajo los cuales se configura cada curso en el sistema web, fuente y elaboración propia

Para ingresar al menú presentado en la Figura 22 debe seleccionarse el botón *agregar* al igual que en los casos anteriores, colocándose el nombre y abreviatura de cada curso, además de asignarse un color.

La configuración de los cursos es la opción más interesante, existiendo más botones que permitirán la creación y gestión óptima de los horarios para los docentes. Uno de estos botones es *Gestionar Disponibilidad*, cuyo color es azul y permite seleccionar los días y horas libres que va a tener el curso, lo cual se muestra en la Figura 23.

Figura 23

Configuración en la opción cursos – botón gestionar disponibilidad

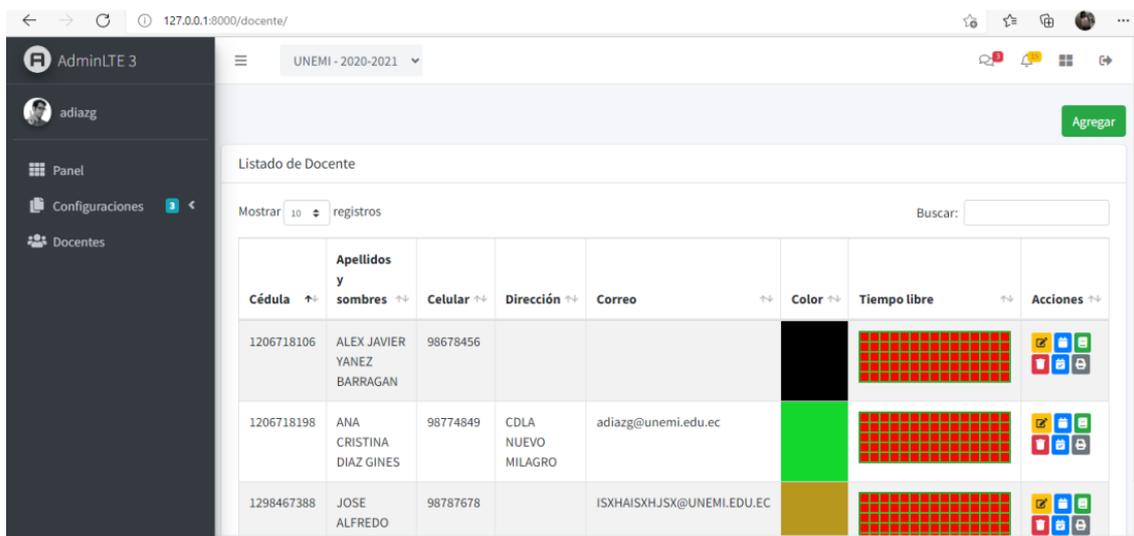


*Nota:* Se presenta el listado de cursos libre, fuente y elaboración propia

Por ejemplo, si una persona estudia de lunes a viernes en la noche, tendrá libre sábado y domingo, además de iniciar sus clases desde 18:00, lo cual debe configurarse en esta opción. Antes de poder generar el horario, también debe configurarse la opción *docente* como se presenta en la Figura 24.

Figura 24

Configuración en la opción docente – vista inicial



*Nota:* Se presenta el listado de docentes que han sido configurados, fuente y elaboración propia

Figura 25

*Configuración en la opción docente – creación de docente*

The screenshot displays a web browser window with the URL `127.0.0.1:8000/docente/?pet=crear`. The page title is 'UNEMI - 2020-2021' and the main heading is 'Crear Docente'. The form contains the following fields:

- Cédula:
- Nombres:
- Apellidos:
- Sexo:
- Fecha Nacimiento:
- Celular:
- Télefono:
- Correo Electrónico:
- Dirección:

The footer of the page includes the copyright notice: 'Copyright © 2014-2019 AdminLTE.io. All rights reserved.' and the version number: 'Version 3.0.1'.

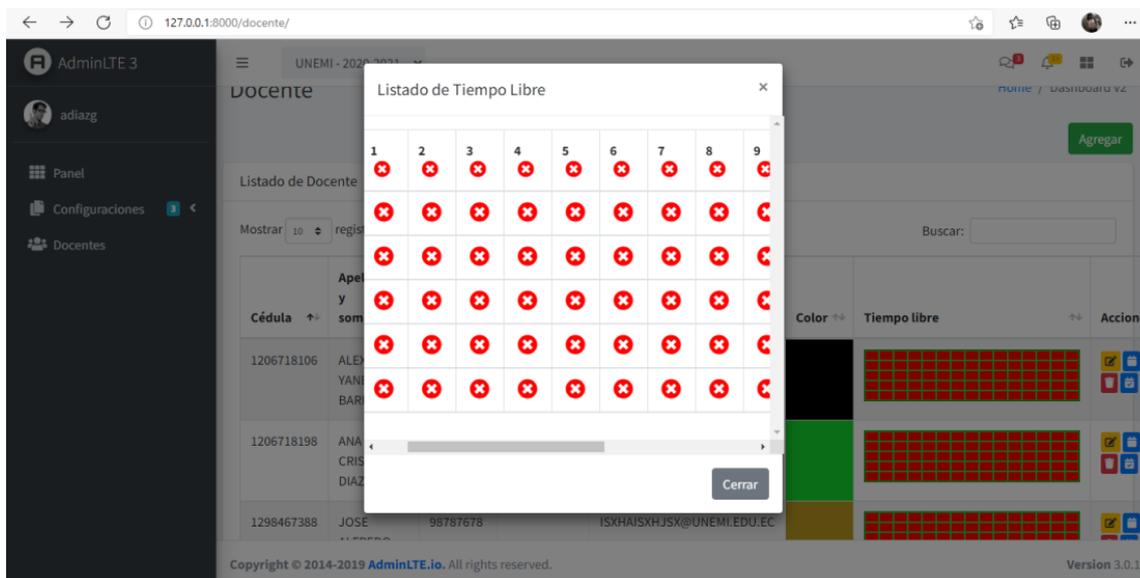
*Nota:* Se presentan los parámetros bajo los cuales se crean los docentes en el sistema web, fuente y elaboración propia

Tal y como se observa en la Figura 25, para crear cada docente se deben colocar los nombres, apellidos, número de cédula, el sexo, la fecha de nacimiento, dirección, número de celular y correo electrónico, lo cual permitirá tener un contacto más directo con el docente. De esta manera, en caso de que exista algún inconveniente, puede ubicarse al educador desde su correo electrónico, celular o su dirección.

Una vez ingresado todos esos datos, se reflejan los mismos botones que se muestran en la opción *curso*, en donde puede manejarse el horario de los docentes y sus horarios libres, tal y como se muestra en la Figura 24. Por ejemplo, si el docente desea tener libre los días viernes, todo el día se selecciona para que tenga el día libre.

Figura 26

### Configuración en la opción docente – días libres



*Nota:* Se presenta cómo se configuran los días libres para los docentes, fuente y elaboración propia

Tomando como referencia la Figura 24, en la parte que dice tiempo libre, el docente no tiene asignado nada de tiempo libre, trabajando los 5 días a la semana en las 15 horas. Sin embargo, una vez seleccionado, se pondrá en color verde los días libres y, en la cuadrícula presentada en la Figura 26, saldrán señalados estos libres también con color verde.

De igual manera que en los casos anteriores, podrá asignarse un color a cada docente. Entonces, una vez ya creado el docente y también sus días libres, puede comenzarse con la asignación de las asignaturas.

En la Figura 24, el botón junto al *tiempo libre*, es el botón de *asignar asignaturas*, en donde se pueden ir asignando las asignaturas disponibles y verificar si concuerda con el horario del docente, mientras se evita el choque de horarios. A continuación, se muestra un horario totalmente generado dentro del sistema:

Figura 27

Visualización de horarios generados en el sistema web

Horas	LU	MA	MI	JU	VI
1 07:00 - 08:00	MATEMATICAS	MATEMATICAS	CIENCIAS NATURALES	CIENCIAS NATURALES	SOCIALES
2 08:00 - 09:00	MATEMATICAS	MATEMATICAS	CIENCIAS NATURALES	CIENCIAS NATURALES	SOCIALES
3 09:00 - 10:00	ESPAÑOL	ESPAÑOL	ETICA	ARITMETICA	SOCIALES
4 10:00 - 11:00	ESPAÑOL	ESPAÑOL	CIENCIAS NATURALES	CIENCIAS NATURALES	SOCIALES
5 11:00 - 12:00	INGLES	TECNOLOGIA	CIENCIAS NATURALES	CIENCIAS NATURALES	ARITMETICA
6 12:00 - 13:00	INGLES	TECNOLOGIA	ETICA	ARITMETICA	ARITMETICA

*Nota:* Se presentan horarios configurados dentro del sistema web, fuente y elaboración propia

Una vez creado, también asignados los docentes y las materias a cada curso, se utilizará el botón *ver horario*, el cual permitirá consultar e imprimir horarios.

### 3.3. Validación de la propuesta

La validación involucró el desarrollo de un informe de pruebas considerando una serie de parámetros que permitieron verificar el correcto funcionamiento del sistema web, presentando a continuación una tabla con dichos parámetros:

**Tabla 31***Informe de prueba que valida el sistema web*

<b>Parámetros</b>	<b>Cumple</b>	<b>No cumple</b>
El sistema es responsive	x	
El sistema carga correctamente	x	
Funciona con todos los navegadores	x	
La comunicación con la base de datos es idónea	x	
Permite cargar los datos correctamente	x	
La información se puede descargar de manera rápida una vez generado los horarios	x	
Permite la visualización y corrección de los datos ingresados	x	
Cumple con el objetivo por el cual fue desarrollado	x	

*Nota:* En la tabla se presentan los parámetros considerar para validar el sistema web, fuente y elaboración propia

El informe de las pruebas realizadas al sistema web permite constatar que funciones correctamente, funcionando en distintos dispositivos y navegadores sin problemas, además de cargar la información correctamente y mantener una comunicación idónea con la base de datos. A su vez, la información que genera el software para la creación y gestión de horarios se puede descargar rápidamente, mostrando cómo se encuentran distribuidos los educadores, permitiendo también su corrección cuando sea necesario.

Por tales motivos, se determina que el sistema web cumple el objetivo por el cual fue diseñado, contribuyendo de manera eficiente a la asignación de horarios a los docentes dentro de la FACI.

## CONCLUSIONES

1. Atendiendo al primer objetivo específico orientado a evaluar el proceso para la generación de los horarios asignados a los docentes, se recurrió a una metodología de investigación descriptiva con enfoque cualitativo que soportó la aplicación de una entrevista, misma que fue dirigida al docente encargado de crear los horarios. Los datos recopilados permitieron concluir que la FACI dispone actualmente de un sistema para la creación y gestión de horarios, al cual se debe ingresar información respecto a las asignaturas, clases, aulas y profesores. Sin embargo, el sistema no se ajusta totalmente a las necesidades de la institución, impidiendo que se asignen efectivamente y provocando el cruce de horarios.
2. Como segundo objetivo específico se encuentra el desarrollo del sistema web, el cual se diseñó utilizando Python 3.9.6 como Entorno de Desarrollo Integrador, el Framework Django 3.2.6 y PostgreSQL 12 como gestor de base de datos. El sistema permite la configuración del periodo, asignaturas y cursos, además de docentes, esto a fin de crear y asignar los horarios en la FACI de manera efectiva, sin confusiones o problemas que afecten a la coordinación académica.
3. El tercer objetivo específico involucró el realizar pruebas de efectividad del sistema desarrollado, lo cual se cumplió tras la presentación de un informe de prueba que evaluó una serie de parámetros, mostrando que el sistema funciona correctamente en distintos dispositivos y navegadores, permite cargar y descargar información sin problemas, visualizarla y corregirla de ser el caso, culminando que efectivamente se logra el objetivo por el cual fue diseñado.

## RECOMENDACIONES

1. Que se evalúen los equipos de computación disponibles en la FACI a fin de verificar si funcionan correctamente, especialmente aquellos utilizados para la asignación de horarios. Es importante que, de existir problemas, se ordene su reparación inmediata o renovación, garantizando que no surjan problemas de hardware cuando se utilice el sistema.
2. Monitorear el funcionamiento del sistema web con una frecuencia mensual, lo cual permita detectar posibles errores o fallos que deban corregirse inmediatamente. A su vez, previo a cada ciclo académico, la revisión efectuada al sistema web debe ser intensiva para evitar problemas técnicos en la creación y gestión de los horarios a docentes en la FACI.
3. Que se evalúe la efectividad de la creación y asignación de horarios a los docentes en otras facultades de la UNEMI, proponiendo el uso de este sistema web para esta laboral. Además, se debe permitir que en futuros estudios se analice el sistema propuesto e incluyan de ser posible, nuevas funcionalidades que contribuyan a la coordinación académica en la FACI.

## REFERENCIAS BIBLIOGRÁFICAS

- Amaya, D. (2017). *Desarrollo de un sistema para la creación de horarios para la Universidad Central del Ecuador* [Tesis de pregrado, Universidad Central del Ecuador]. <http://www.dspace.uce.edu.ec/bitstream/25000/9438/1/T-UCE-0011-320.pdf>
- Asamblea Nacional del Ecuador (2008). *Constitución de la República del Ecuador*. [https://www.oas.org/juridico/pdfs/mesicic4\\_ecu\\_const.pdf](https://www.oas.org/juridico/pdfs/mesicic4_ecu_const.pdf)
- Barzola, A. (2015). *Implementación de un Sistema de Gestión de Horarios de Clases para los docentes del colegio península de Santa Elena* [Tesis de pregrado, Universidad Estatal Península de Santa Elena]. <https://repositorio.upse.edu.ec/handle/46000/3677>
- Berenguel, J. (2016). *Desarrollo de aplicaciones web en el entorno servidor*. Paraninfo.
- Camino, P. (3 de Agosto de 2018). *Qué es Django y por qué usarlo*. <https://openwebinars.net/blog/que-es-django-y-por-que-usarlo/>
- Castaño, J., & Jurado, S. (2016). *Comercio electrónico*. Editex.
- Chazallet, S. (2016). *Python 3: los fundamentos del lenguaje*. Editions ENI.
- Del Alcázar, L. (2019). *Desarrollo de un sistema integrado de gestión de horarios y control de asistencia docente para el centro Preuniversitario de la UNAMAD* [Tesis de pregrado, Universidad Nacional Amazónica de Madre de Dios]. <http://repositorio.unamad.edu.pe/handle/UNAMAD/581>
- Digital Guide IONOS (Marzo de 2019). *PostgreSQL: el gestor de bases de datos a fondo*. <https://www.ionos.es/digitalguide/servidores/know-how/postgresql/>
- Dos Santos, M. (2017). *Investigación de Mercados: Manual universitario*. Diaz de Santos.
- Fajardo, E., & Soler, Ó. (2017). *Aplicación web para la administración y asignación de los horarios en las facultades de la Universidad Santo Tomás Seccional Tunja* [Tesis

de pregrado, Universidad Santo Tomás].

<https://repository.usta.edu.co/handle/11634/22021>

Ferrer, J. (2017). *Aplicaciones web*. Ra - MA.

Flórez, J. (2017). *Plan de negocio: para pequeñas empresas*. Ediciones de la U.

García, A. (2017). *UF2175 - Diseño de bases de datos relacionales*. Elearning.

Gibert, Marc; Pérez, Óscar. (2016). *Bases de datos en PostgreSQL*.

[https://dgvs.mspbs.gov.py/files/ficha\\_formulario/13\\_06\\_2016\\_20\\_45\\_34\\_P06\\_M2109\\_02152.pdf](https://dgvs.mspbs.gov.py/files/ficha_formulario/13_06_2016_20_45_34_P06_M2109_02152.pdf)

Gisbert, B. (2016). *UF1272 - Administración y auditoría de los servicios web*. Elearning.

Goodwill Community Foundation. (2020). *¿Qué son las aplicaciones web?*

<https://edu.gcfglobal.org/es/informatica-basica/que-son-las-aplicaciones-web/1/>

Grupo Consultar EFE. (2021). *Sistemas web*.

<https://grupoconsultorefe.com/servicio/tecnologias-de-la-informacion/sistemas-web>

Gutiérrez, J. (2018). *UF1271 - Instalación y configuración del software de servidor web*.

Elearning.

Heinze, G., Olmedo, V., & Andoney, J. (2017). Uso de las tecnologías de la información y comunicación (TIC) en las residencias médicas en México. *Acta médica Grupo Ángeles*, 15(2), 150-153.

Hernández, A., Ramos, M., Placencia, B., Indacochea, B., Quimís, A., & Moreno, L.

(2018). *Metodología de la investigación científica*. Editorial Área de Innovación y Desarrollo.

Hernández, L., Cleger, S., & Urquiza, R. (2020). Sistema informático para la elaboración y publicación del horario docente de la Universidad de Holguín. *Revista de*

*Investigación Latinoamericana en Competitividad Organizacional RILCO*, 1(5), 4 - 11.

- Hidalgo, B., Guaiña, J., Ramos, V. Y., & Gutiérrez, C. (2020). Aplicación de la metodología DSDM y del framework Django para el sistema de registro de pacientes del servicio de emergencia del hospital pediátrico Alfonso Villagómez. *International Congress on Technology Education and Knowledge Management, KnE Engineering*, 1(1), 60 - 75.
- Lee, G. (2020). *Tipos de pruebas de software: diferencias y ejemplos*.  
<https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/>
- López, M. (2016). *Programación Web en el Entorno Servidor. (MF0492\_3)*. Ra - Ma.
- Martínez, I. (2019). *UF2121 - Programación del trabajo de campo de la investigación*. Madrid: Elearning.
- MDN Web Docs Mozilla (2020). *Introducción a Django*.  
<https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>
- Melo, D., Díaz, A., Vega, O., & Serna, C. (2018). Situación Digital para Instituciones de Educación Superior: Modelo y Herramienta. *Información Tecnológica*, 29(6), 163 - 174.
- Molina, J., Zea, M., Contento, M., & García, F. (2018). Comparación de metodologías en aplicaciones web. *3Ciencias*, 7(1), 1 - 9.
- Narvéz, C., Acosta, S., & Montealegre, J. (2019). *Aplicación de la metodología SCRUM e implementación de la estrategia de sdervicio con base al marco de referencia ITIL V 3.0 en el desarrollo de una aplicación web para una empresa de transporte público* [Tesis de pregrado, Universidad Cooperativa de Colombia].  
[https://repository.ucc.edu.co/bitstream/20.500.12494/15742/1/2019-Aplicacion\\_Scrum\\_Itil.pdf](https://repository.ucc.edu.co/bitstream/20.500.12494/15742/1/2019-Aplicacion_Scrum_Itil.pdf)
- Niño, J. (2018). *Servidores de aplicaciones web*. EDITEX.

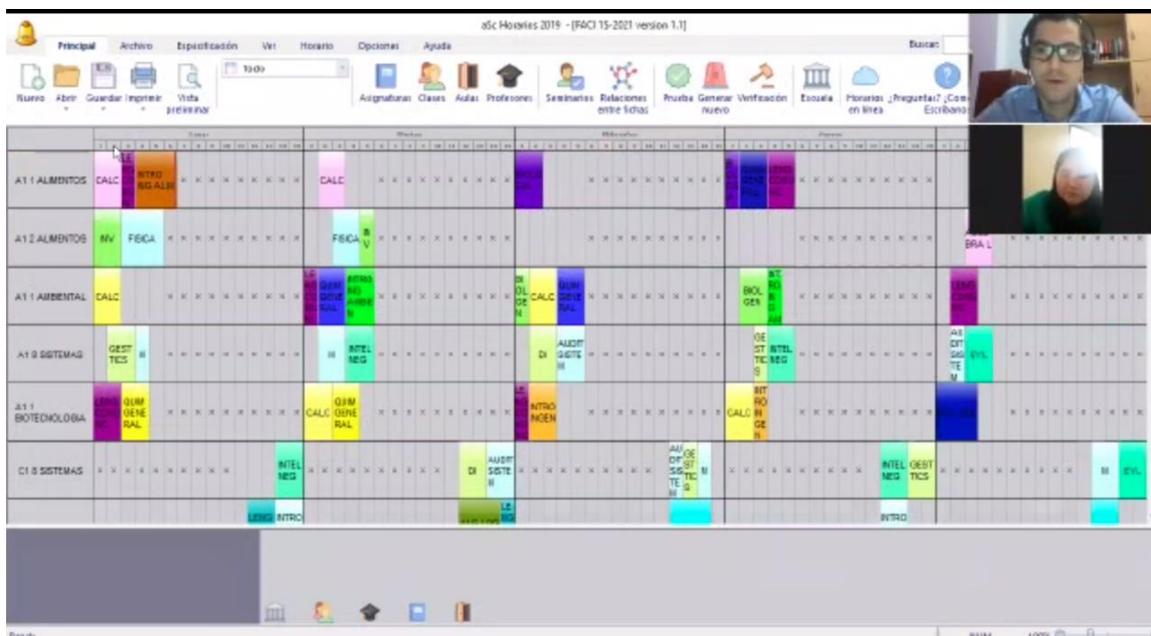
- Otzen, T., & Manterola, C. (2017). Técnicas de Muestreo sobre una Población a Estudio. *International Journal of Morphology*, 35(1), 227-232.
- Pauta, L., & Moscoso, S. (2017). Verificación y validación de software. *Revista Killkana Técnica*, 1(3), 25 - 32.
- Sánchez, Ó. (2016). *Sistema Operativo, Búsqueda de la Información: Internet/Intranet y Correo*. Paraninfo.
- Subra, J., & Vannieuwenhuysse, A. (2018). *Scrum: un método ágil para sus proyectos*. Ediciones ENI.
- Tudor, J. (2019). *Python para principiantes: Aprenda Python en 5 días con orientación paso a paso*. Scholastic Books.
- Zea, M., Molina, J., & Redrován, F. (2017). *Administración de bases de datos con PostgreSQL*. 3Ciencias.

## ANEXOS

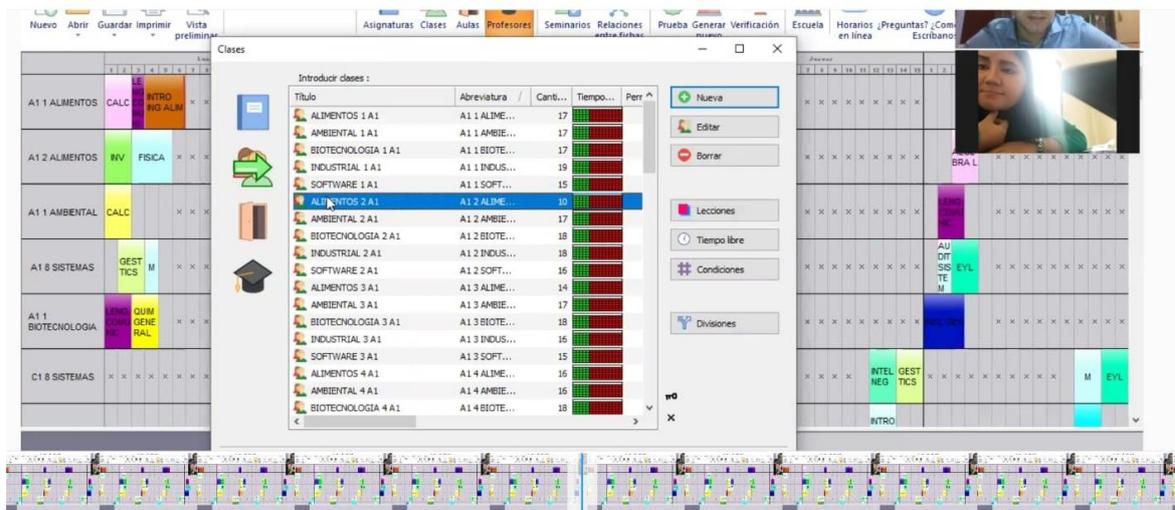
### Anexo 1. Modelo de entrevista realizada a docente encargado de crear horarios

1. ¿Cómo funciona el sistema actual para asignar horarios?
2. ¿Cuáles son las gestiones que realiza en el sistema?
3. ¿Cuáles suelen ser los errores generalmente con la asignación de horarios?
4. ¿Qué consideraciones deberían existir para un sistema de distribución de horarios?

### Anexo 2. Soporte de entrevista realizada - Ing. Denis Darío Mendoza Cabrera



← 16 Septiembre 2021



Anexo 3. Diagrama de espina de pescado

