



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y  
TELECOMUNICACIONES**

**TÍTULO DEL TRABAJO DE TITULACIÓN**

Optimización de gestión de horarios académicos para la Universidad  
Estatad Península de Santa Elena (UPSE).

**AUTOR**

**Ricardo Reyes, Carlos Steven**

**PROYECTO DE UNIDAD DE INTEGRACIÓN CURRICULAR**

Previo a la obtención del grado académico en  
**INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN**

**TUTOR**

**Ing. Jaramillo Infante Mónica Karina Mgt.**

**Santa Elena, Ecuador**

**Año 2025**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y  
TELECOMUNICACIONES**

**TRIBUNAL DE SUSTENTACIÓN**

Ing. José Sánchez Aquino, Mgt.  
**DIRECTOR DE LA CARRERA**

Ing. Mónica Jaramillo Infante, Mgt.  
**TUTOR**

Ing. Alfredo Tumbaco Reyes, Mgt.  
**DOCENTE ESPECIALISTA**

Ing. Marjorie Coronel Suárez, Mgt.  
**DOCENTE GUÍA UIC**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y  
TELECOMUNICACIONES**

**CERTIFICACIÓN**

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por **Ricardo Reyes Carlos Steven**, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 11 días del mes de diciembre del año 2025

**TUTOR**

---

**Ing. Jaramillo Infante Mónica, Mgt.**



**UPSE**

**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y  
TELECOMUNICACIONES**

**DECLARACIÓN DE RESPONSABILIDAD**

Yo, **Ricardo Reyes Carlos Steven**

**DECLARO QUE:**

El trabajo de Titulación, Optimización de gestión de horarios académicos para la Universidad Estatal Península de Santa Elena (UPSE), previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 11 días del mes de diciembre del año 2025

**EL AUTOR**

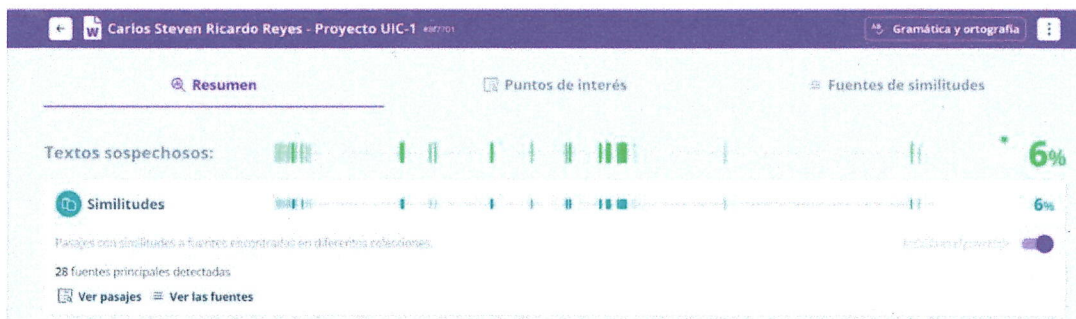
A handwritten signature in blue ink, which appears to read "Ricardo Reyes Carlos Steven", is written over a horizontal line.

**Ricardo Reyes Carlos Steven**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y  
TELECOMUNICACIONES  
CERTIFICACIÓN DE ANTIPLAGIO**

Certifico que después de revisar el documento final del trabajo de titulación denominado Optimización de gestión de horarios académicos para la Universidad Estatal Península de Santa Elena (UPSE), presentado por el estudiante, Ricardo Reyes Carlos Steven, fue enviado al Sistema Anti-plagio, presentando un porcentaje de similitud correspondiente al 6%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



**TUTOR**

**Ing. Jaramillo Infante Mónica, Mgt.**



**UPSE**

**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y  
TELECOMUNICACIONES**

**AUTORIZACIÓN**

**Yo, Ricardo Reyes Carlos Steven**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales del presente trabajo de titulación con fines de difusión pública, además apruebo la reproducción dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor.

La Libertad, a los 11 días del mes de diciembre del año 2025

**EL AUTOR**

---

**Ricardo Reyes Carlos Steven**

## **AGRADECIMIENTO**

A Dios, por darme la fortaleza y salud necesaria para culminar esta etapa.

A mis padres y hermanos, por su comprensión, paciencia y respaldo en cada momento del camino.

Uno muy especial a mi tutora, Ing. Mónica Jaramillo, Ing. Marjorie Coronel e Ing. José Sánchez, por haberme formado y apoyado en todo momento del proceso de titulación.

A los demás docentes y compañeros, por los conocimientos compartidos y por las experiencias vividas durante la formación académica.

A Geoffrey Royce Rojas, por siempre inspirarme y buscar refugio en sus canciones.

A todas las personas que, de una u otra forma, contribuyeron a la realización de este trabajo.

*Ricardo Reyes, Carlos Steven*

## **DEDICATORIA**

A Dios, a mis padres, por su amor incondicional y por enseñarme, con su ejemplo, que el esfuerzo y la constancia siempre valen la pena.

A mi familia, por su apoyo en los momentos difíciles y por ser mi motivación diaria.

*Ricardo Reyes, Carlos Steven*

## ÍNDICE GENERAL

TÍTULO DEL TRABAJO DE TITULACIÓN	I
TRIBUNAL DE SUSTENTACIÓN	II
CERTIFICACIÓN	III
DECLARACIÓN DE RESPONSABILIDAD	IV
DECLARO QUE:	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AUTORIZACIÓN	VI
AGRADECIMIENTO	VII
DEDICATORIA	VIII
ÍNDICE GENERAL	IX
ÍNDICE DE TABLAS	XII
ÍNDICE DE FIGURAS	XIII
RESUMEN	XVI
ABSTRACT	XVII
INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN	3
1.1. Antecedentes	3
1.2. Descripción del Proyecto	5
1.3. Objetivos del Proyecto	6
1.4. Justificación del Proyecto	7
1.5. Alcance del Proyecto	8
1.6. Metodología del Proyecto	9
1.6.1. Metodología de Investigación	9

1.6.2. Beneficiarios del Proyecto	10
1.6.3. Variable	10
1.6.4. Análisis de recolección de datos	11
1.7. Metodología de desarrollo	12
<b>CAPÍTULO 2. PROPUESTA</b>	<b>14</b>
2.1. Marco Contextual	14
2.1.1. Estructura académica y enfoque específico	14
2.1.2. Procesos de gestión de horarios actual y deficiencia	15
2.1.3. Justificación	16
2.2. Marco Conceptual	16
2.2.1. Algoritmo	16
2.2.2. Sistema informático	17
2.2.3. Sistema de información	17
2.2.4. Aplicación web	17
2.2.5. Servidor web	17
2.2.6. Framework web	18
2.2.7. Navegador web	18
2.2.8. Base de datos	18
2.2.9. Entorno de desarrollo	18
2.2.10. Lenguaje Java	19
2.2.11. Programación Orientada a Objetos	19
2.3. Marco Teórico	19
2.3.1. Horarios académicos	20
2.3.2. Problema de generación de horarios académicos (Timetabling Problem)	20

2.3.3. Métodos de optimización aplicados al Timetabling	22
2.3.4. Algoritmos genéticos (AG)	23
2.4. Requerimientos	25
2.4.1. Requerimientos funcionales	25
2.4.2. Requerimientos no funcionales	28
2.5. Componentes de la Propuesta	29
2.5.1. Arquitectura del Sistema	29
2.5.2. Diagramas de casos de uso	31
2.5.3. Modelado de Datos	40
2.5.4. Diseño de interfaces	45
2.6. Pruebas	60
CONCLUSIONES	65
RECOMENDACIONES	65
ANEXOS	71

## ÍNDICE DE TABLAS

Tabla 1. Requerimientos funcionales.	28
Tabla 2. Requerimientos no funcionales.	29
Tabla 3. Especificación de caso de uso de inicio de sesión.	32
Tabla 4. Especificación de caso de uso del uso de distributivo de docentes.	33
Tabla 5. Especificación de casos de uso de restricciones de docentes.	34
Tabla 8. Especificación de casos de uso de administración de paralelos.	35
Tabla 10. Especificación de casos de uso para administración de asignaturas.	36
Tabla 9. Especificación de caso de uso de generación de reportes de horarios.	37
Tabla 6. Especificación de casos de uso de generación de horarios temporales.	39
Tabla 7. Especificación de casos de uso de modificación de horarios temporales.	39
Tabla 11. Prueba de Generación de horario.	61
Tabla 12. Prueba de Almacenamiento y comparación de horarios.	62
Tabla 13. Prueba de Modificación manual y validación en tiempo real.	62
Tabla 14. Prueba de Generación y exportación de reportes.	63
Tabla 15. Prueba de Configuración de entidades horarias.	63
Tabla 16. Prueba de Configuración del distributivo.	64
Tabla 17. Prueba de Configuración de restricciones (docentes).	65

## ÍNDICE DE FIGURAS

Figura 1. Modelo incremental.	13
Figura 2. Ubicación sectorial de la UPSE. Fuente: Google Maps [8].	14
Figura 3. Ejemplo de horario. Fuente: [18].	20
Figura 4. Algoritmo genético. Fuente: [29].	24
Figura 5. Pseudocódigo del Algoritmo Genético Simple. Fuente: [30]	25
Figura 6. Arquitectura del sistema.	29
Figura 7. Caso de uso de inicio de sesión.	31
Figura 8. Caso de uso de distributivo de docentes.	32
Figura 9. Caso de uso de restricciones y preferencias.	33
Figura 10. Caso de uso de administración de paralelos.	34
Figura 11. Caso de uso de administración de asignaturas.	35
Figura 12. Caso de uso de generación de reportes de horarios.	36
Figura 13. Caso de uso de la ejecución del algoritmo para la generación de horarios.	38
Figura 14. Modelo de datos (Parte 1).	40
Figura 15. Modelo de datos (Parte 2).	41
Figura 16. Pantalla de inicio de sesión.	45
Figura 17. Pantalla de administración de asignaturas.	45
Figura 18. Pantalla de filtro para las asignaturas.	46
Figura 19. Pantalla de creación o modificación de registros de asignaturas.	46
Figura 20. Pantalla de administración de paralelos.	47
Figura 21. Pantalla de filtro para los paralelos.	47
Figura 22. Pantalla de creación o modificación de registros de paralelos.	48

Figura 23. Pantalla de administración de docentes.	48
Figura 24. Pantalla de administración de asignaturas y paralelos.	49
Figura 25. Pantalla de creación o modificación de registros de asignaturas y paralelos.	49
Figura 26. Pantalla de administración del distributivo de docentes.	50
Figura 27. Pantalla de filtro para el distributivo de docentes.	50
Figura 28. Pantalla de creación o modificación de registros del distributivo de docentes.	51
Figura 29. Pantalla de administración del bloqueo de docentes (restricciones).	51
Figura 30. Pantalla de creación o modificación de registros de bloqueo de docentes.	52
Figura 31. Pantalla de administración de disponibilidad de docentes.	52
Figura 32. Pantalla de creación o modificación de registros de disponibilidad de docentes.	53
Figura 33. Pantalla de administración de horarios temporales.	53
Figura 34. Pantalla de creación de registros de horarios temporales.	54
Figura 35. Pantalla de selección de múltiples carreras para la generación de horarios.	54
Figura 36. Pantalla de aviso, una vez se esté generando el horario temporal.	55
Figura 37. Pantalla de confirmación para eliminar registros de horarios temporales.	55
Figura 38. Pantalla de detalle de un registro de horario temporal.	55
Figura 39. Pantalla de paralelos correspondiente a una carrera.	56
Figura 40. Pantalla del horario de clases para un paralelo de una carrera.	56
Figura 41. Pantalla de modificación de una clase del horario.	57

Figura 42. Pantalla de lista de espacios disponibles de una clase del horario temporal.	57
Figura 43. Pantalla con mensaje de alerta sobre una restricción en el horario temporal.	58
Figura 44. Pantalla de búsqueda de reporte de docentes por espacios físicos.	58
Figura 45. Pantalla del reporte de horario de clases por espacios físicos generado por el sistema.	59
Figura 46. Pantalla de búsqueda de reporte general de horarios por docentes.	59
Figura 47. Pantalla del reporte de horario de clases por docente generado por el sistema.	60

## RESUMEN

El presente trabajo de titulación desarrolla una aplicación web integral para la generación de horarios académicos en la Universidad Estatal Península de Santa Elena (UPSE), utilizando la metodología del Algoritmo Genético (AG) como una de las tantas soluciones de optimización. Por lo general, la asignación de horarios se vuelve uno de los procesos manuales más complejos en el área académica, ya que implica coordinar múltiples variables como son los docentes, cursos, paralelos, asignaturas, tipos de aulas y demás restricciones institucionales, mismas que generan ciertos conflictos, sobrecarga de trabajo y resultados no óptimos, ni satisfactorios.

El sistema propuesto incorpora la lógica de negocio (backend) desarrollado en Spring Boot, la cual incluye el cálculo evolutivo del horario de clases, validaciones de las restricciones duras y blandas, almacenamiento de posibles soluciones, y la administración y gestión de las demás entidades académicas. El AG trabaja evaluando miles de combinaciones posibles y selecciona aquella solución que minimiza los conflictos una vez habiendo pasado por sus respectivos procesos (aptitud, mutación, cruce y selección).

Los resultados obtenidos muestran que el sistema reduce significativamente el tiempo de respuesta que conlleva cada carrera con la planificación académica, y a su vez, elimina mucho de los conflictos que enfrentaban actualmente cada dirección, ya sea en cruces de horarios, docentes no disponibles, mal uso de recursos institucionales, etc., ofreciendo así, un proceso más legible y transparente hacia la universidad.

**Palabras claves:** Algoritmo genético, horarios, sistema web, restricciones.

## ABSTRACT

This thesis develops a comprehensive web application for generating academic schedules at the Santa Elena Peninsula State University (UPSE), using the Genetic Algorithm (GA) methodology as one of its many optimization solutions. Schedule allocation is generally one of the most complex manual processes in academia, as it involves coordinating multiple variables such as instructors, courses, sections, subjects, classroom types, and other institutional constraints. These constraints often lead to conflicts, workload overload, and suboptimal or unsatisfactory results.

The proposed system incorporates business logic (backend) developed in Spring Boot, which includes the evolutionary calculation of class schedules, validation of hard and soft constraints, storage of possible solutions, and the administration and management of other academic entities. The GA works by evaluating thousands of possible combinations and selects the solution that minimizes conflicts after it has passed through its respective processes (fitness, mutation, crossover, and selection).

The results obtained show that the system significantly reduces the response time that each program entails with academic planning, and in turn, eliminates many of the conflicts that each department currently faced, whether in scheduling conflicts, unavailable teachers, misuse of institutional resources, etc., thus offering a more legible and transparent process to the university.

**Keywords:** Genetic algorithm, schedules, web system, restrictions.

## INTRODUCCIÓN

La planificación de horarios académicos conforma una de las actividades de trabajo con alto grado de errores críticos y una cantidad considerable de tiempo consumido. Este proceso básicamente involucra muchos factores que van desde gestionar paralelos, cursos, docentes, aulas, bloques de horarios, etc., hasta la administración de muchas restricciones institucionales.

Conforme a esto, en las últimas décadas han cobrado mucha relevancia en el mercado las técnicas metaheurísticas, misma que derivan los Algoritmos Genéticos (AG), los cuales permiten encontrar las mejores posibles soluciones a problemas muy complejos y severos. Esta metodología imita prácticamente el proceso de evolución como selección natural, cruce y mutación, evaluando de manera recurrente muchas combinaciones hasta encontrar la las optima o cercanas a la misma.

En este contexto, el presente proyecto propone el desarrollo de un sistema web para optimizar la gestión de horarios académicos, cuya solución se centra en el uso del AG aplicado al modelo de negocio que tiene la universidad en el área estudiantil. El sistema integra un backend robusto desarrollado en Spring Boot con Java, mismo que se encarga de realizar la lógica de negocio con las distintas entidades, dando paso a la implementación y uso de los servicios web, mismos que serán consumidos por el frontend hecho en Angular con Typescript, para que así se puedan visualizar de una manera clara y concisa la fuente de datos del sistema en general.

El desarrollo del proyecto responde a una necesidad real identificada en la institución, misma que desde hace tiempo busca optimizar recursos y mejorar su proceso de planificación académica. Los resultados obtenidos muestran una reducción en el tiempo de elaboración de los horarios de clases, habiendo una mayor coherencia y distribución de las distintas entidades, demostrando así, su efectividad ante el enfoque establecido.

En mención a lo anterior, el proyecto está estructurado de la siguiente forma:

El capítulo I posee la fundamentación de la propuesta y todos los componentes relacionados a la misma, como antecedentes, descripción del proyecto, objetivos, justificación, alcance, metodología de la investigación, beneficios del proyecto, variables, análisis de recolección de datos y metodología de desarrollo.

El capítulo II se conforma por la propuesta, integrando el marco contextual, marco conceptual, marco teórico, requerimientos, componentes de la propuesta, arquitectura del sistema, diagramas de casos de uso, modelado de datos, diseño de interfaces y pruebas. Se finalizan con las conclusiones y recomendaciones del trabajo.

## CAPÍTULO 1. FUNDAMENTACIÓN

### 1.1. Antecedentes

La creación de horarios académicos en una institución de educación es un procedimiento administrativo que consiste en asignar para cada curso un docente, una sala y una programación semanal. Debido a que en todos los períodos académicos las condiciones varían, esta labor debe desarrollarse al inicio de cada período y muchas veces la información de semestres anteriores no puede utilizarse debido al cambio de condiciones [1].

En la actualidad, la planificación de horarios académicos en UPSE recae directamente en los directores de carrera de cada facultad, mismos que deben de invertir una considerable cantidad de tiempo en organizar los horarios de clases de muchas asignaturas en todos sus niveles. Esta tarea puede verse más que tediosa cuando el directivo gestiona más de una carrera, y deben coordinar mucho más el uso de recursos institucionales, la disponibilidad de docentes, cruces de horarios, etc.

El Sistema de Gestión Académica (SGA) de la UPSE, ya cuenta con módulo para la planificación de horarios académicos y asignaciones de cursos, sin embargo si representa un avance significativo en el uso de herramientas comunes como Microsoft Excel u otros sistemas de planificación manual. Sin embargo, este proceso continúa siendo relativamente lento, ya que no cambia en mucho su accionar con las herramientas antes mencionadas. Es decir, la asignación se realiza individualmente por cada elemento que se tenga que considerar, lo que ocasiona retrasos y el tiempo de respuesta suele extenderse más de la cuenta. Esta tarea al principio suele ser fácil, pero conforme se van registrando más asignaturas con otros bloques de horarios, u otros factores, se vuelve muy complejo completarla.

En la actualidad tampoco se cuenta con una manera eficiente de aprovechar los recursos académicos. Esto provoca situaciones comunes, como asignar aulas demasiado grandes para cursos con pocos estudiantes o, al revés, usar salones pequeños para grupos numerosos. En ambos casos se evidencia un uso poco adecuado de la infraestructura disponible. Además, las preferencias o limitaciones

de los docentes, por ejemplo, horarios en los que no pueden dictar clases por motivos laborales, de salud u otros compromisos, no se toman en cuenta de forma automática dentro del proceso de planificación.

Además, existen ciertas restricciones de carácter general, como la programación de reuniones internas o los consejos de facultad. Estos eventos obligan a dejar libres ciertos días y franjas horarias, por lo que los directores de carrera deben ajustar manualmente los horarios que ya estaban definidos. A esto se suma un problema bastante común: asignar por error clases presenciales y virtuales al mismo docente en la misma jornada. Esta situación no es posible de cumplir y requiere una corrección manual para evitar conflictos.

En la Universidad de Chile, se desarrolló un proyecto de Generación de horarios académicos en INACAP, utilizando algoritmos genéticos para el periodo 2014 [2]. Este sistema contribuyó a resolver la problemática de planificación horaria mediante la técnica de los AG, misma que da mucha flexibilidad en los resultados, lo cual permite que la solución se adapte a distintos horarios, universo de salas, etc. No solo aporta una, sino varias soluciones, mostrando de esta manera, una comparativa para verificar cual sería la solución más óptima o cercana a las peticiones dadas. Esto ayudó a reducir el tiempo de respuesta en la creación de horarios, pasando de días o semanas de elaboración, a solo minutos, lo cual representa un ahorro sustancial en recursos de la institución.

En Ecuador, en la Escuela Superior Politécnica de Chimborazo, se trabajó un sistema web para la generación de horarios académicos a través de técnicas metaheurísticas para la Unidad Educativa del Milenio Tarqui en el periodo 2019 [3]. El proyecto resolvió el problema de la poca eficiencia en la creación de horarios para los docentes de la institución, y evaluó la eficiencia del mismo mediante la norma ISO/IEC 9126 a través de la utilización de recursos y comportamientos de tiempo, dando como resultados, que el uso del sistema es más eficiente y rápido en comparación con los procesos manuales que se realizaban antes de la propuesta.

Por su parte, a nivel local, en la Universidad Estatal Península de Santa Elena (UPSE), se desarrolló e implementó un sistema de gestión de horarios de clases

para los docentes del Colegio Península de Santa Elena en el año 2016 [4]. El proyecto también ayudo a mejorar el proceso de elaboración de horarios, puesto que se creaban los mismos en archivos de Microsoft Excel y tomaba mucho tiempo en realizarlos, respectivamente. El uso del algoritmo de doble decisión fue crucial referente en la validación de los cruces de horarios, lo cual permitió optimizar más los recursos del centro educativo en base a cursos, docentes, etc.

Si bien puede observarse a través de la revisión bibliográfica de distintos trabajos sobre la generación automática de horarios académicos que existen algunas propuestas basadas en algoritmos genéticos, heurísticos y otras técnicas de optimización que han resultado eficaces en su aplicación a diversos contextos, ninguna de ellas se ajusta directamente a las particularidades y necesidades de la Universidad Estatal Península de Santa Elena (UPSE). Por tal motivo, en el presente trabajo se plantea la generación de un módulo adicional de complementa al Sistema de Gestión Académica (SGA) vigente en la institución; así, se tratará de un software que permita la generación automatizada y optimización de los horarios académicos.

## **1.2. Descripción del Proyecto**

El presente proyecto consiste en el desarrollo de un módulo web complementario al Sistema de Gestión Académica (SGA), orientado a la generación y optimización de horarios académicos en base a restricciones específicas establecidas por la institución, el personal docente y las políticas administrativas de la carrera.

Entre las características clave del módulo se encuentran:

Gestión de restricciones: permitirá registrar y administrar diferentes tipos de limitaciones que pueden afectar la planificación de horarios, tales como disponibilidad horaria de docentes, actividades institucionales (por ejemplo, reuniones de consejo de facultad), incompatibilidades de modalidad (clases presenciales y virtuales en un mismo día), así como cualquier otra restricción académica o administrativa que incida en la asignación de cursos y espacios. El módulo estará diseñado para soportar un número flexible de restricciones, garantizando que la generación automática de horarios respete todas las condiciones que se definan.

Asignación optimizada de espacios: El algoritmo dará prioridad al uso eficiente de los espacios físicos, comenzando con la asignación de los espacios más pequeños y escalando a las de mayor capacidad en función de la demanda, lo cual permitirá una mejor distribución de los estudiantes y el aprovechamiento de los recursos académicos.

Generación automática de horarios temporales: A partir de las restricciones definidas y la disponibilidad de docentes, espacios y demás cursos, el módulo producirá un horario preliminar que podrá ser revisado, ajustado y validado por el director de Carrera.

Interfaz de administración: El director de carrera podrá administrar la información de docentes, cursos, espacios y las restricciones, así como modificar o reprogramar horarios en casos particulares.

Reportes y exportación: Una vez que se hayan generado los horarios, el sistema permitirá la generación de reportes en formatos accesibles (PDF), con los horarios definitivos para cada carrera.

### **1.3. Objetivos del Proyecto**

#### **Objetivo general**

Desarrollar un módulo web para la generación automática de horarios académicos en la Universidad Estatal Península de Santa Elena (UPSE), aplicando restricciones sobre los recursos institucionales con el propósito de optimizar la planificación y administración académica.

#### **Objetivos específicos**

- Diseñar un modelo de datos relacional que integre los recursos académicos y restricciones, con el fin de estructurar de manera eficiente la información necesaria para optimizar la generación y gestión de horarios académicos en la UPSE.
- Desarrollar un módulo web mediante servicios REST en Java con Spring y un frontend en Angular, orientado a optimizar la gestión y asignación de

horarios académicos, garantizando consistencia en la arquitectura del sistema.

- Evaluar el impacto del módulo web en la reducción del tiempo de planificación, en la disminución de conflictos de horarios y en la mejora de la eficiencia administrativa.
- Generar reportes relevantes en base a la asignación de horarios, uso de aulas y reprogramaciones, que faciliten el seguimiento y la toma de decisiones académicas de la UPSE.

#### **1.4. Justificación del Proyecto**

La planificación de horarios en la universidad constituye una de las tareas administrativas más complejas, ya que implica coordinar aulas, docentes, asignaturas y restricciones institucionales de manera simultánea. Aunque el Sistema de Gestión Académica (SGA) ya dispone de un módulo para registrar horarios y asignar cursos, este proceso continúa siendo manual, materia por materia, lo que demanda una considerable inversión de tiempo por parte de los directores de carrera y aumenta el riesgo de errores humanos.

En la práctica, esta limitación por lo general se traduce en varios problemas muy concretos. Por ejemplo, se termina usando mal la infraestructura, es decir, se asignan espacios grandes para grupos muy pequeños, o peor aún, espacios muy reducidos para paralelos con muchos estudiantes. También aparecen choques de horarios entre asignaturas y, por ende, se vuelve complicado respetar la disponibilidad real de los docentes y, ante cualquier imprevisto que puede suceder (reuniones internas o académicas, cambios de modalidad de clase, etc.), hay que reprogramar todo manualmente.

Todo esto hace que el trabajo administrativo sea más pesado y menos eficiente, y al final también afecta al grupo de estudiantes porque el tiempo de respuesta en la publicación de horarios se retrasa, hay cambios a última hora y se pierde claridad en la gestión para organizar las clases.

En este escenario, la creación de un módulo web que complemente al SGA y sea capaz de generar y optimizar la gestión de horarios aparece como una propuesta

muy necesaria. Con esta solución, se busca, sobre todo, aliviar un poco el problema de la planificación académica de horarios, el cual representa un proceso tardío y de mucho esfuerzo manual, por lo que su integración hará que se reduzcan los errores y se usen los recursos de la universidad de una mejor manera.

Al mismo tiempo, el proyecto no solo se dedica a resolver la problemática de gestión horaria en la institución, sino que también aporta una propuesta innovadora en el ámbito de gestión educativa, al incorporar herramientas tecnológicas que apuntan a la eficiencia y la capacidad de adaptación.

A futuro, este trabajo deja abierta la puerta para seguir incorporando técnicas mucho más avanzadas para la optimización, o incluso de Inteligencia Artificial (AI), que permitan automatizar aún más la planificación y aumentar con precisión a la optimización académica.

El presente trabajo se alinea al plan de creación de oportunidades, tomando en consideración los siguientes objetivos:

### **Eje social**

**Objetivo 7.** Potenciar las capacidades de la ciudadanía y promover una educación innovadora, inclusiva y de calidad en todos los niveles [5].

### **Eje institucional**

**Objetivo 14.** Fortalecer las capacidades del Estado con énfasis en la administración de justicia y eficiencia en los procesos de regulación y control, con independencia y autonomía [5].

## **1.5. Alcance del Proyecto**

El presente proyecto tendrá como alcance el desarrollo de un módulo complementario al Sistema de Gestión Académica (SGA) de la Universidad Estatal Península de Santa Elena (UPSE), orientado a la generación automática y optimización de horarios académicos de la Carrera de Tecnologías de la Información.

- Registro y gestión de restricciones académicas y administrativas, considerando la disponibilidad de docentes, aulas, laboratorios y actividades institucionales.
- Aprovechar mejor los espacios disponibles, evitando tener espacios vacíos o mal usados, y dándoles prioridad según las necesidades de cada carrera.
- Generar una propuesta de horario temporal que respete las condiciones y restricciones definidas (disponibilidad de docentes, tipos de espacios, jornadas, paralelos, etc.), de modo que se reduzcan al mínimo los cruces de horario y errores en la planificación.
- Ofrecer una pantalla administrativa sencilla y amigable, donde los académicos puedan registrar, modificar o eliminar de manera rápida cuando surjan casos excepcionales o imprevistos.
- Permitir la generación de reportes y la exportación de la información, para que los directores de carrera puedan obtener el horario en formatos accesibles de consultar.

No se contempla en este proyecto la integración con otros módulos del SGA que no estén directamente relacionados con la gestión de horarios, ni la administración de matrículas, calificaciones u otros procesos académicos. El enfoque se limita a la planificación de horarios de la carrera mencionada, sin abarcar a todas las facultades de la institución.

## **1.6. Metodología del Proyecto**

### **1.6.1. Metodología de Investigación**

La investigación experimental se refiere a un tipo de investigación prospectiva. Se presenta mediante la manipulación de una variable experimental no comprobada. En condiciones rigurosamente controladas, con el fin de describir de qué modo o porque causa se produce una situación o acontecimiento particular. En base a esto, el diseño de la investigación será de carácter experimental, ya que implica la implementación controlada de un prototipo del módulo web para optimizar la gestión de horarios académicos. Esto nos permitirá evaluar su funcionamiento e impacto para optimizar parte de la gestión administrativa de la universidad [6].

Además, el trabajo tiene un alcance explicativo, porque no solo se busca mostrar que el módulo web funciona, sino de entender cómo y porque influye en la forma de cómo se planifican los horarios en la universidad. Con este enfoque se pretende más que nada, identificar aquellas muestras que hacen difícil la asignación de horarios académicos (cambios de último momento, falta de visibilidad de la disponibilidad real de docentes y espacios, etc.), y explicar que ocurre cuando ese proceso se automatiza, es decir, como el algoritmo en el módulo web ayuda a organizar mejor los espacios, aprovechar mejor los recursos y respetar la disponibilidad o bloqueo de docentes.

### **1.6.2. Beneficiarios del Proyecto**

El desarrollo de este proyecto beneficiara directamente a los directos de carrera de la Universidad Estatal Península de Santa Elena (UPSE). Ellos contarán con una herramienta que les ayudara a ahorrar tiempo, a reducir errores en las distribuciones, y a organizar mejor los espacios y recursos que la institución tiene a su disposición.

Al mismo tiempo, los docentes también se verán favorecidos, ya que podrán disponer de horarios más ordenados acorde a su disponibilidad y carga de trabajo. Esto disminuye los problemas habituales en la planificación como los que ya se mencionaron al principio.

### **1.6.3. Variable**

Este proyecto se trabajó con cuatro variables principales, pensadas para medir de forma sencilla que tanto ayuda el módulo de generación de horarios:

**Tiempo de generación de horarios:** Se refiere al tiempo que se demora en obtener un horario temporal usando el módulo web, y se compara con el tiempo que se toma hacerlo manualmente, como se viene haciendo actualmente con el SGA.

**Reducción de conflictos de asignación:** Esto hace referencia a la disminución de errores típicos en la planificación manual, como clases que se mezclan, espacios mal asignados o no respetando las demás restricciones locales.

El objetivo es ver si con el módulo web estos problemas, o se eliminan o tienen menos frecuencia.

**Eficiencia en la organización académica:** Mide en qué medida el módulo web ayuda a organizar mejor los horarios de clases, aprovechando de forma más ordenada los espacios, los docentes y las franjas horarias disponibles. No solo se trata de generar el horario, sino de que este mejor distribuido y tengas pocos o cero cambios.

**Satisfacción de los usuarios:** Tiene que ver por como percibe el módulo web los usuarios que lo usan (directores).

#### **1.6.4. Análisis de recolección de datos**

**Entrevista:** Se aplicó esta técnica a través de una entrevista informal con el director de la carrera de Tecnologías de la Información de la Universidad Estatal Península de Santa Elena (UPSE).

Durante esta interacción se expuso la problemática actual relacionada con la planificación y asignación de horarios académicos, identificándose como principales desafíos el elevado tiempo requerido en la elaboración de los horarios, la presencia de conflictos en la asignación de aulas y docentes, así como la ausencia de mecanismos que permitan una reprogramación ágil ante situaciones imprevistas.

**Observación:** Se realizó un análisis del método tradicional utilizado para la planificación de horarios, evidenciando que gran parte del procedimiento se ejecuta de forma manual en el SGA, lo que genera sobrecarga administrativa, duplicidad de esfuerzos y una mayor probabilidad de errores humanos.

Esta observación permitió identificar puntos críticos a mejorar mediante el módulo propuesto.

**Fuentes bibliográficas:** Durante la investigación se consultaron diversas fuentes primarias y secundarias, entre ellas artículos científicos, libros y tesis relacionadas con la generación automática de horarios y la optimización de recursos académicos. Dichas referencias permitieron contextualizar la problemática como parte del denominado Timetabling Problem, analizar técnicas y algoritmos aplicados en proyectos similares, lo que sirvió como una base teórica para fundamentar la propuesta.

## 1.7. Metodología de desarrollo

El modelo incremental o evolutivo se comienza en el desarrollo, satisfaciendo un conjunto de requisitos y partiendo de una base estable. Las siguientes versiones proveen los requisitos que faltan, evolucionando el sistema fase a fase. Esta suele utilizarse en sistemas complejos, donde se admiten versiones del producto. Presentan como ventaja el permitir experimentar con prototipos ejecutables intermedios. Además, la animación grafica utilizada en este tipo de paradigmas sirve para auto explicar su funcionalidad, sin necesidad de mayor documentación, lo que muchas veces sirve para no desecharse el software creado incrementalmente [7].

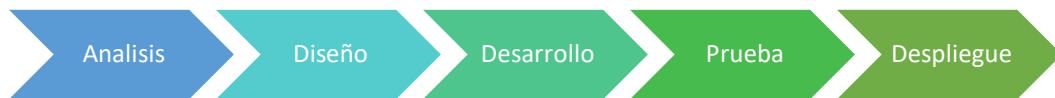
Las fases del proyecto son las siguientes:

- Análisis de requisitos: Obtener y documentar los requisitos funcionales y no funcionales, así también como los procesos principales a gestionar en el módulo web.
- Diseño del sistema: Diseñar una arquitectura modular que permita ir desarrollando el proyecto en base a los frameworks y herramientas tecnológicas.
- Desarrollo y construcción incremental: Construir la lógica de negocio de cada servicio web, permitiendo así una escalabilidad y mantenibilidad en su desarrollo modular, así también como la interfaz gráfica del usuario en base a un desarrollo web escalable, permitiendo consumir los servicios web del backend.
- Pruebas: Realizar los respectivos test de funcionamiento tanto en el backend como en el frontend para asegurar un óptimo funcionamiento del sistema a implementar.
- Despliegue: Realizar el despliegue a producción del módulo web, donde nos permita monitorear la misma para verificar el rendimiento y disponibilidad del software.
- Mantenimiento: Realizar un estudio en base a opiniones de los usuarios acerca del sistema para próximas actualizaciones.

En las definiciones de incrementos consta:

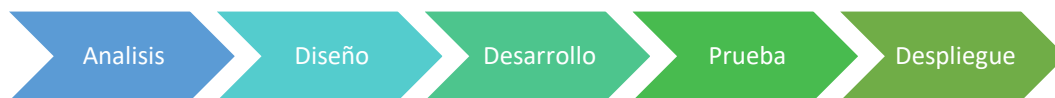
### **I1: Base del Sistema y Algoritmo Esencial**

Este incremento establece la arquitectura (Cliente - Servidor) aplicada a la solución, junto con el modelo de datos básico. El objetivo es implementar el algoritmo genético en la lógica de negocio del proyecto, incluyendo todas las configuraciones y funcionalidades que debe cumplir inicialmente. Se desarrolla la interfaz básica de ejecución y la visualización simple de los resultados.



### **I2: Algoritmo Mejorado y Gestión de Restricciones**

Este incremento se centra en el perfeccionamiento y ajuste del algoritmo genético en el módulo web. Se implementan y prueban todos los operadores (selección, cruce y mutación) y las demás funcionalidades, incluyendo las restricciones duras y blandas. Respecto a la interfaz de usuario, se mejora la visualización del horario en una tabla interactiva y amigable.



### **I3: Reportes, Evaluación y Optimización**

Este último incremento está enfocado en los objetivos de evaluación y reportes. Se realizan ajustes finales al algoritmo genético para maximizar la eficiencia del mismo.

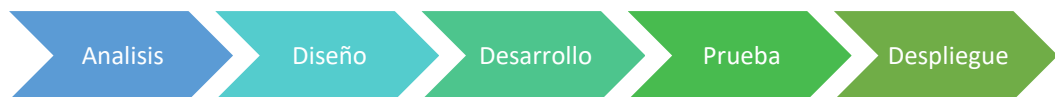


Figura 1. Modelo incremental.

## CAPÍTULO 2. PROPUESTA

### 2.1. Marco Contextual

#### Área de estudio

La Universidad Estatal Península de Santa Elena, conocida por su acrónimo UPSE, es una universidad pública localizada en el cantón La Libertad, provincia de Santa Elena, República del Ecuador. Es el primer centro de enseñanza autónomo que cuenta con la mayor población estudiantil de la zona.

#### Ubicación sectorial



Figura 2. Ubicación sectorial de la UPSE. Fuente: Google Maps [8].

#### Misión

Formar profesionales que aportan al desarrollo sostenible, contribuye a la solución de los problemas de la comunidad y promueve la cultura.

#### Visión

Ser reconocida por su calidad académica, impacto de sus investigaciones y su aporte al desarrollo de la sociedad.

#### 2.1.1. Estructura académica y enfoque específico

La UPSE opera con una estructura académica organizada en 7 facultades principales. Dado que la solución propuesta está diseñada para servir a todas las

carreras de la universidad, el presente proyecto se centra en la aplicación, validación y prueba dentro del contexto de la Facultad de Sistemas y Telecomunicaciones, la cual agrupa carreras como Tecnologías de la Información, Ingeniería de Software, etc., mismas que nos permiten concentrar el análisis en un conjunto de recursos significativos tales como:

**Población académica:** La facultad cuenta con aproximadamente 35 docentes a tiempo completo y parcial, más de 200 estudiantes distribuidos en diferentes cursos y jornadas.

**Recursos críticos:** El proceso de asignación de horarios por lo general es complejo en este ámbito, ya que implica la combinación tanto de aulas básicas como laboratorios especializados.

### **2.1.2. Procesos de gestión de horarios actual y deficiencia**

Actualmente, los procesos de gestión y asignación de horarios académicos en la UPSE se realizan de manera semiautomatizada o manual asistida. Si bien, en el Sistema de Gestión de Académica (SGA) ya existe un apartado que maneja la información de las entidades más relevantes (docentes, asignaturas, aulas, etc.), la optimización y resolución de conflictos no son abordados por alguna herramienta algorítmica.

**Alto consumo de tiempo:** El proceso actual requiere una considerable inversión de tiempo humano por parte de los directores de carrera, quien tienen la tarea de cuadrar la disponibilidad y las restricciones de cada una de sus áreas, lo cual retrasa la planificación por periodo académico.

**Generación de conflictos:** El método empleado actualmente, basado generalmente en heurísticas simples o prueba y error, dificulta constantemente la satisfacción simultánea de todas las restricciones duras o blandas que puedan presentarse, dando como resultados horarios que, aunque se presenten funcionales, no son óptimos y requieren reprogramaciones posteriores.

**Baja eficiencia de recursos:** Ante la falta de una herramienta de optimización se obtiene una ineficiencia del correcto uso de los espacios físicos (aulas y

laboratorios), pudiendo generar casos de subutilización en algunos bloques y sobreutilización en otros.

### **2.1.3. Justificación**

La necesidad de resolver la problemática en la institución, justifica el desarrollo de este proyecto. La idea en sí, es ofrecer una herramienta y servicio que ayude a organizar mejor los horarios de clase y disminuya la carga del proceso manual.

Para lograrlo, se propone la implementación del Algoritmo Genético (AG) en el módulo web. Este tipo de algoritmo trabaja para resolver problemas donde hay demasiadas combinaciones (posibles soluciones) y se necesita encontrar la que este más acorde a las condiciones que queremos resolver, tal y como ocurre en la planificación de horarios académicos. Al explicarlo en este contexto, se busca:

**Reducir el tiempo de planificación:** Automatización de búsqueda de buenas combinaciones de horarios, en lugar de que se esté realizando todo tradicionalmente.

**Disminuir los conflictos:** Evaluando de forma sistemática todas las restricciones académicas (disponibilidad de docentes, uso de recursos, paralelos, etc.), para evitar distribuciones no idóneas.

**Mejora la eficiencia:** A medida que se aprovechen de una mejor manera los recursos y espacios que tiene la universidad, mejor será el trabajo administrativo y, en consecuencia, los estudiantes y docentes tendrán a su disposición un mejor servicio educativo.

## **2.2. Marco Conceptual**

### **2.2.1. Algoritmo**

Un algoritmo es una secuencia ordenada y finita de instrucciones para realizar un cálculo, desarrollar una tarea o resolver un problema específico. Cada paso en un algoritmo está bien definido y puede ser ejecutado para producir un resultado. Los algoritmos son la base de la programación y se utilizan para diseñar programas que ejecuten tareas específicas [9].

### **2.2.2. Sistema informático**

Un sistema informático está formado por un conjunto de elementos necesarios para tratar y procesar la información. Como elemento central de los sistemas informáticos se encuentra el ordenador. Se puede considerar ordenador cualquier dispositivo que tenga la misma estructura, como los móviles actuales o smartphones, las tabletas y cualquier otro dispositivo. Los sistemas informáticos están compuestos por elementos físicos (hardware) y elementos lógicos (software) [10].

### **2.2.3. Sistema de información**

Un sistema de cómputo organiza los datos en una jerarquía que empieza con el bit, el cual representa ya sea un 0 o un 1. Los bits se pueden agrupar para formar un byte para representar un carácter, número o símbolo. Los bytes se pueden agrupar para formar un campo y los campos relacionados se pueden agrupar para formar un registro. Los registros relacionados se pueden conjuntar para formar un archivo y los archivos relacionados se pueden organizar en una base de datos [11].

### **2.2.4. Aplicación web**

Una aplicación web es un software que se ejecuta en el navegador web. Las empresas tienen que intercambiar información y proporcionar servicios de forma remota. Utilizan aplicaciones web para comunicarse con los clientes cuando lo necesiten y de una forma segura. Las funciones más comunes de los sitios web, como los carros de compra, la búsqueda y el filtrado de productos, la mensajería instantánea y los canales de noticias de las redes sociales, tienen el mismo diseño que las aplicaciones web. Le permiten acceder a funcionalidades complejas sin la necesidad de instalar o configurar un software [12].

### **2.2.5. Servidor web**

Un servidor web local es aquel servidor instalado en un equipo determinado con el fin de trabajar offline y online. Es una alternativa especialmente útil si lo que buscamos es un entorno en el que desarrollar un sitio web o una aplicación y que nos permita realizar todo tipo de pruebas sin correr riesgos.

Se trata de un entorno controlado, en el que trabajaremos con los recursos con los que cuenta el propio ordenador, y entre sus ventajas está que no necesitaremos comprar un dominio, ya que el disco duro del equipo se convierte en hosting [13].

#### **2.2.6. Framework web**

Un framework es, en esencia, una estructura de soporte predefinida que proporciona una base sólida sobre la cual los desarrolladores pueden construir aplicaciones. En este sentido, este conjunto de herramientas y bibliotecas permite a los programadores centrarse en la lógica del negocio, al mismo tiempo que les libera de la preocupación por los detalles más bajos de la programación.

Por lo tanto, al utilizar un framework, los desarrolladores pueden ser más eficientes y productivos en su trabajo [14].

#### **2.2.7. Navegador web**

Un navegador web es un software que permite acceder a internet, buscar y ver información, trabajar y acceder a todo tipo de servicios, a través de una conexión con internet. Gracias al uso del Protocolo de Transferencia de Hipertexto, HTTP, esa información, es decir, las imágenes, textos y vídeos, se comparte y se muestra en un formato que pueda verse en cualquier navegador [15].

#### **2.2.8. Base de datos**

Se entiende como la representación a nivel integrado de una colección estructurada de datos que contienen físicamente el diseño lógico de un conjunto de entidades, instancias de las diferentes entidades del sistema de información que se está modelando en una organización y las interrelaciones de las entidades; representación que necesita de una gestión de datos a fin de ser utilizados de una forma compartida por todos los usuarios de una organización en la resolución de sus necesidades de información [16].

#### **2.2.9. Entorno de desarrollo**

Un entorno de desarrollo es un espacio de trabajo que permite a los desarrolladores crear y modificar una aplicación sin impactar la versión final del software. Estos

espacios suelen incluir herramientas que automatizan muchas tareas de programación, como el análisis, el diseño, la escritura de código, las pruebas y el mantenimiento. La elección del entorno de desarrollo adecuado será decisiva en el resultado final, y dependerá de las necesidades de cada programador. Afortunadamente, existen muchos tipos distintos de entornos que veremos a continuación [17].

#### **2.2.10. Lenguaje Java**

Java es una plataforma informática de lenguaje de programación creada por Sun Microsystems en 1995. Ha evolucionado desde sus humildes comienzos hasta impulsar una gran parte del mundo digital actual, ya que es una plataforma fiable en la que se crean muchos servicios y aplicaciones. Los nuevos e innovadores productos y servicios digitales diseñados para el futuro también siguen basándose en Java [18]. Java se distingue de otros lenguajes como Python, C++ y JavaScript en aspectos clave como rendimiento, sintaxis y aplicaciones [19].

#### **2.2.11. Programación Orientada a Objetos**

La programación orientada a objetos (POO) es un modelo de programación que utiliza objetos, ligados mediante mensajes, para la solución de problemas. La idea central es simple: organizar los programas a imagen y semejanza de la organización de los objetos en el mundo real.

Un programa orientado a objetos se compone solamente de objetos. Cada uno de ellos es una entidad que tiene unas propiedades particulares, los atributos, y unas formas de operar sobre ellos, los métodos. Por ejemplo, una ventana de una aplicación Windows es un objeto. El color de fondo, la anchura, la altura, etc. son atributos. Las rutinas, lógicamente transparentes al usuario, que permiten maximizar la ventana y minimizarla son los métodos [19].

### **2.3. Marco Teórico**

En la actualidad, la carga de trabajo en las Instituciones de Educación Superior (IES) ha incrementado; por un lado, la oferta educativa se diversifica de acuerdo a las necesidades económicas/sociales de la comunidad, y, por otro lado, la necesidad

de un mercado laboral más competitivo y complejo ha aumentado el número de estudiantes, y todo lo anterior trae como consecuencia que la administración de los recursos académicos se vuelva compleja [16].

### 2.3.1. Horarios académicos

Según Robles, Ortiz y Gama, los horarios académicos ó Scheduling como comúnmente se conoce; es un proceso que consiste en crear una plantilla en la cual se tengan perfectamente estructuradas las asignaciones, (materias, lugar, día y la hora) que serán impartidas por los docentes sin que existan traslapos entre los profesores o la ubicación de las materias, debido a esto, es considerado un problema de asociación de en variables ya que se tiene que organizar una secuencia de eventos con recursos limitados y restricciones impuestas en un periodo de tiempo especificado [17].

	9 am – 10 am	10 am – 11 am	11 am –12 pm
<b>Monday</b>	MATHS Mr. Green Room 1	ENGLISH Mrs. White Room 2	SCIENCE Mrs. Teacher Room 2
<b>Tuesday</b>			

Figura 3. Ejemplo de horario. Fuente: [18].

Como señala Lance, cada día se divide en franjas horarias; cada franja horaria contiene una lista de las asignaturas impartidas, quién las imparte y dónde. El horario se puede representar de diversas maneras. Cada estudiante tendrá su propio horario según las asignaturas que estudie, al igual que cada profesor y cada aula; todas estas son perspectivas diferentes del mismo horario [18].

### 2.3.2. Problema de generación de horarios académicos (Timetabling Problem)

Bashab y otros autores consideran que, en determinados casos, el problema de la planificación del horario consiste en localizar cualquier horario que cumpla todas las restricciones establecidas, en cuyo caso el problema se formula como un

problema de búsqueda. En otros casos, lo que se necesita es un cronograma que cumpla con todas las restricciones estrictas y, además, minimice o maximice una función objetivo que incorpore las restricciones suaves; en estos casos, el problema se plantea como uno de optimización. El desafío de la optimización es encontrar los mejores valores de acuerdo con una función objetivo establecida, respetando siempre un conjunto de restricciones [19].

En esta misma línea, Fouad, Al-kubaisi y Maha señalan que el problema de los horarios puede describirse como la asignación de cursos a salas y franjas horarias, considerando tanto restricciones duras como restricciones blandas. Este proceso se desarrolla en dos etapas: la primera consiste en obtener una solución factible que satisfaga las restricciones duras, mientras que la segunda busca mejorar dicha solución reduciendo al máximo las violaciones a las restricciones blandas. De esta manera, la solución final no solo es válida, sino que también puede competir en calidad con las propuestas reportadas en la literatura especializada [20].

De acuerdo con lo anterior, en la literatura se reconocen distintos casos problemáticos utilizados para la validación de algoritmos de horarios. Dichos casos suelen establecer, en primer lugar, un conjunto de restricciones estrictas, tales como:

- En cualquier momento, asignar únicamente un curso a cada estudiante.
- La sala debe cumplir con los requisitos de las características del curso.
- No se puede asignar a un curso un número de estudiantes superior a la capacidad del aula.
- Para cada sala, solo se puede asignar un curso en un mismo horario.
- En segundo lugar, se identifican las restricciones suaves, entre las que se destacan:
  - El último horario del día debe tener un curso programado para el estudiante.
  - Cada estudiante debe contar con al menos dos cursos consecutivos.
  - En un mismo día, el estudiante únicamente debe asistir a un curso.

El objetivo planteado en estas investigaciones consiste en obtener un calendario viable que cumpla con las restricciones duras, minimizando al mismo tiempo las infracciones de las restricciones blandas [20].

### 2.3.3. Métodos de optimización aplicados al Timetabling

Antes de entrar en materia de tipos de algoritmos que se utilizan para la generación de horarios académicos, es necesario aclarar las diferencias entre las heurísticas y metaheurísticas, ya que son conceptos claves para entender cómo se aborda este tipo de problema. De forma sencilla, las heurísticas son reglas o estrategias prácticas que permiten encontrar soluciones factibles de manera rápida, sin garantizar severos cambios, pero ayudando a salir del bucle en problemas complejos. En cambio, las metaheurísticas, van un paso más adelante, ya que toman aquellas soluciones iniciales y las van mejorando (mutación), es decir, que van explorando distintas alternativas y afinando resultados. Las metaheurísticas resultan especialmente útiles para generación de horarios académicos, y, por ende, serán aplicadas al contexto de este proyecto.

#### Heurísticas (HE)

Un método heurístico genera soluciones de un determinado problema mediante un método del cual se sabe, bien por experiencia o por razonamiento teórico, que genera buenas soluciones con una alta probabilidad. Suele clasificarse en tres grupos:

- **Algoritmos de un solo paso:** Son métodos que generan una única solución en cada etapa, tomando decisiones sucesivas que ya no son reconsideradas. Cada vez hay menos alternativas y estas están más condicionadas, de forma que las últimas decisiones pueden ser muy malas.
- **Métodos iterativos:** Son métodos que generan una solución en cada etapa, de manera que las soluciones obtenidas puedan reconsiderarse en etapas anteriores.
- **Métodos de mejora:** Son métodos que, partiendo de una solución determinada, van mejorándola en etapas sucesivas [21].

#### Metaheurísticas

Son metodologías de alto nivel que combinan heurísticas subyacentes, empleando estrategias inteligentes para explorar y explotar el espacio de búsqueda y, al mismo

tiempo, guiando los procedimientos heurísticos con técnicas de aprendizaje. La clasificación de una metaheurística como de solución única o basada en la población depende del número de soluciones examinadas en cada iteración del procedimiento de exploración.

- **Basada en una solución única (MH-S):** Emplea procesos iterativos para mejorar una solución candidata individual mediante operadores de transformación dentro de su entorno, empleando técnicas de búsqueda local.
- **Basada en la población (MH-P):** Comienza estableciendo una colección inicial de soluciones candidatas, manipulándolas iterativamente para mejorar la búsqueda y reemplazando algunas con nuevas soluciones según criterios de calidad para explorarlas y explotarlas eficazmente [22].

#### **2.3.4. Algoritmos genéticos (AG)**

Según Quispe y Gómez, los algoritmos genéticos (AG) representan el modelo evolutivo, de manera que un problema específico actúa como un medio ambiente dentro del cual se desempeña una población formada por individuos que son soluciones candidatas del problema en cuestión. Cada uno de estos individuos (fenotipo) está formado por la información contenida en su genoma (genotipo) y que bien pueden ser las variables independientes, cuyo valor se desea conocer [23].

Similarmente a como se propone para la evolución natural, los AG tienden a encontrar mejores soluciones al problema de manera progresiva, gracias a la "renovación del material genético" en la población por medio de los llamados operadores genéticos. La adaptación de los individuos a su ambiente se determina por la función de aptitud (función fitness), la cual constituye una medida de la calidad de la solución [23].

En base a lo anterior, Leija nos explica la correspondencia existente entre el proceso evolutivo de los organismos y el funcionamiento de los algoritmos genéticos:

- Se parte de una población inicial de individuos (denominados cromosomas), los cuales representan el conjunto potencial de posibles soluciones de un problema.

- Cada uno de estos cromosomas está formado por una cadena o vector de números, y estos números, a su vez, constituyen entidades individuales que son análogas a los genes de los cromosomas biológicos.
- La aptitud de los individuos para producir la mejor solución al problema (adaptabilidad de los individuos al entorno natural) es evaluada mediante una función de bondad o ajuste. Partiendo de esta evaluación es posible escoger los cromosomas (selección) que intercambiarán información entre ellos (cruce) y/o variarán aleatoriamente su información (mutación), dando lugar a una nueva generación de individuos o cromosomas.
- La aptitud o adaptabilidad de los cromosomas es evaluada nuevamente, dando paso a una nueva iteración del algoritmo. Este proceso se repite una y otra vez, logrando que la calidad media de los individuos de cada generación aumente progresivamente con respecto a los de la población inicial de partida [28].

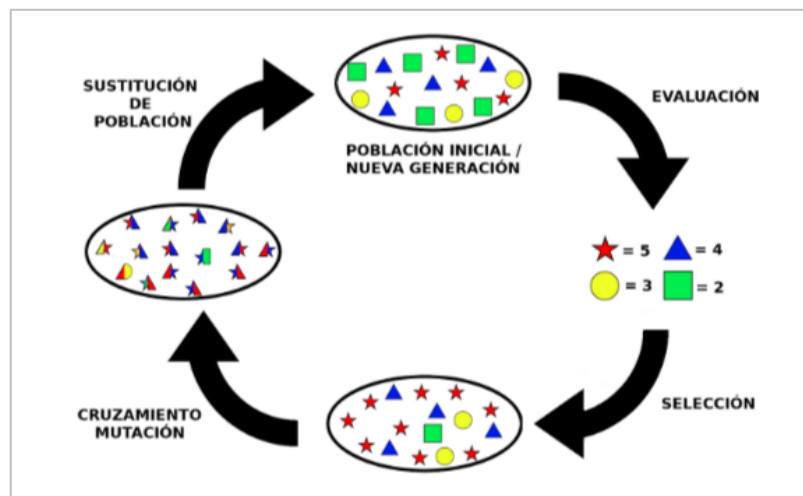


Figura 4. Algoritmo genético. Fuente: [29].

### Esquema del algoritmo genético simple

Adaptando esta solución a la problemática inicial, básicamente se crean los horarios combinando datos de docentes, asignaturas, paralelos, espacios y bloques de horarios. Cada horario se evalúa según las restricciones requeridas (desarrolladas dentro de la lógica de negocio o como parámetros de entrada), como son la disponibilidad de docentes, capacidad de espacios, tipos de espacios, sesiones por

día, jornadas, etc. En cada iteración, las mejores combinaciones de horarios se seleccionan, tienen cruzamiento y mutación, dando paso a la reproducción de nuevas soluciones, repitiendo este paso una y otra vez, hasta obtener un horario valido y optimizado para el actual periodo académico.

```

BEGIN /* Algoritmo Genetico Simple */
  Generar una poblacion inicial.
  Computar la funcion de evaluacion de cada individuo.
  WHILE NOT Terminado DO
    BEGIN /* Producir nueva generacion */
      FOR Tamaño poblacion/2 DO
        BEGIN /*Ciclo Reproductivo */
          Seleccionar dos individuos de la anterior generacion,
          para el cruce (probabilidad de seleccion proporcional
          a la funcion de evaluacion del individuo).
          Cruzar con cierta probabilidad los dos
          individuos obteniendo dos descendientes.
          Mutar los dos descendientes con cierta probabilidad.
          Computar la funcion de evaluacion de los dos
          descendientes mutados.
          Insertar los dos descendientes mutados en la nueva generacion.
        END
      IF la poblacion ha convergido THEN
        Terminado := TRUE
      END
    END
  END
END

```

Figura 5. Pseudocódigo del Algoritmo Genético Simple. Fuente: [30]

## 2.4. Requerimientos

### 2.4.1. Requerimientos funcionales

Código	Descripción	Tipo
RF-GH-01	El sistema debe implementar un algoritmo que genere automáticamente propuestas de horarios en base a la información almacenada.	Procesamiento
RF-GH-02	El sistema debe permitir almacenar múltiples soluciones para poder compararlas.	Procesamiento

RF-GH-03	El sistema debe validar las restricciones antes de aceptar un horario como solución oficial o principal.	Procesamiento
RF-GH-04	El sistema debe permitir el registro de todas las soluciones generadas por el algoritmo en la base de datos.	Persistencia
RF-GH-05	El sistema debe permitir la generación y exportación de reportes de los horarios de clases en formatos legibles e imprimibles (PDF).	Informes
RF-GH-06	El sistema debe validar que el espacio asignado tenga la capacidad suficiente para el paralelo correspondiente.	Validación
RF-GH-07	El sistema debe permitir definir la disponibilidad horaria de cada espacio.	Configuración
RF-GH-08	El sistema debe manejar restricciones fuertes (ej. capacidad de aulas, máximo de horas por docente, etc.) y restricciones suaves (preferencias de jornadas, asignación en tipos de espacios, etc.).	Procesamiento
RF-GH-10	El sistema debe permitir registrar, editar, listar y eliminar docentes y su disponibilidad.	Datos
RF-GH-11	El sistema debe permitir registrar, editar, listar y eliminar las asignaturas de cada carrera.	Datos

RF-GH-12	El sistema debe permitir registrar, editar y eliminar espacios con su respectiva capacidad.	Datos
RF-GH-13	El sistema debe permitir asignar espacios y sus tipos a los paralelos de las asignaturas correspondientes de cada carrera.	Interfaz y datos
RF-GH-14	El sistema debe permitir asignar a los docentes con los paralelos de las carreras mediante la sección del distributivo de clases, misma que tendrá toda la información para su administración.	Interfaz y datos
RF-GH-15	El sistema debe permitir registrar, editar, listar y eliminar bloques horarios con hora de inicio y fin.	Datos
RF-GH-16	El sistema debe permitir generar y visualizar todos los horarios de clases por paralelo de las carreras correspondientes.	Interfaz
RF-GH-17	El sistema debe permitir mover los eventos del horario mediante la interfaz de calendario (drag & drop).	Interfaz
RF-GH-18	El sistema debe actualizar en tiempo real la información del horario al modificar un determinado evento en el calendario de los paralelos correspondientes.	Interfaz
RF-GH-19	El sistema debe mostrar alertas al usuario cuando un movimiento o asignación genere conflicto.	Interfaz

RF-GH-20	El sistema debe validar conflictos por docente, evitando que un docente tenga dos clases simultáneas.	Validación
RF-GH-21	El sistema debe validar estrictamente los determinados conflictos por aula y sus tipos, evitando que una misma se use en dos clases a la vez, dificultando el horario de clases con normalidad.	Validación
RF-GH-22	El sistema debe permitir guardar o revertir cambios en el horario después de una modificación.	Persistencia

Tabla 1. Requerimientos funcionales.

#### 2.4.2. Requerimientos no funcionales

Código	Descripción	Tipo
RNF-GH-01	El sistema debe optimizar recursos al momento de ejecutar el algoritmo, evitando redundancias y malas prácticas.	Rendimiento
RNF-GH-02	El sistema debe generar horarios completos de una o varias carreras (facultad en general) en un tiempo razonable.	Rendimiento
RNF-GH-03	El sistema debe estar preparado para manejar la información de todas las carreras y facultades de la universidad.	Escalabilidad

RNF-GH-04	El sistema debe estar disponible en todo momento durante los procesos de planificación académica.	Disponibilidad
RNF-GH-05	En caso de caída del sistema, el mismo debe garantizar la recuperación de datos.	Recuperación
RNF-GH-06	La interfaz debe ser amigable e intuitiva para los usuarios.	Usabilidad
RNF-GH-07	El código debe estar documentado para facilitar mejoras futuras.	Mantenibilidad
RNF-GH-08	Debe ser accesible desde navegadores modernos.	Portabilidad

Tabla 2. Requerimientos no funcionales.

## 2.5. Componentes de la Propuesta

### 2.5.1. Arquitectura del Sistema

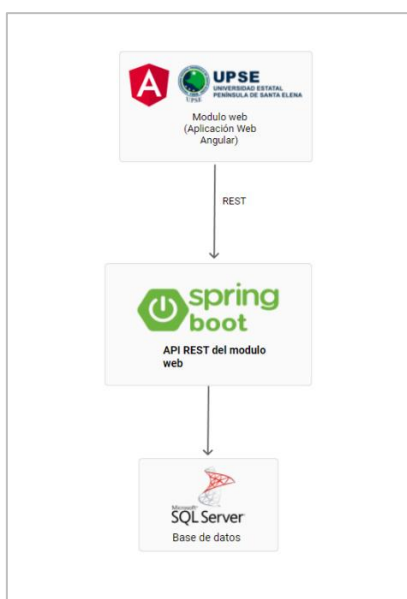


Figura 6. Arquitectura del sistema.

Los estilos arquitectónicos mejoran el "particionamiento" y promueven la reutilización en el diseño mediante soluciones a problemas bien conocidos, así como un lenguaje común independiente de la tecnología. Es posible combinar varios de estos para modelar la solución de un sistema completo [25].

El modelo cliente/servidor (C/S), la computación cliente/servidor, la tecnología cliente/ servidor y la arquitectura cliente/servidor, se refieren a un modelo de diseño que se puede pensar como aplicaciones que se ejecutan en una red de área local (LAN). En términos muy básicos, puede describir que el cliente solicita -y que el servidor ejecuta o de alguna forma realiza las solicitudes de trabajo. Las computadoras en la red se programan para desempeñar eficazmente el trabajo dividiendo las tareas de procesamiento entre clientes y servidores [26].

El proyecto sigue un enfoque clásico "Cliente - Servidor", pero usando tecnologías modernas en cada capa, mismas que se detallaran a continuación:

**Frontend:** Representa el lado del cliente del sistema (Angular). En esta sección de la arquitectura, el usuario puede iniciar la sesión, llamar a los servicios de administración y gestionar la información que requiera.

**Backend:** Es la lógica de negocio del sistema (Spring Boot). En esta sección se reciben todas las solicitudes que venga del lado del cliente, incluyendo las reglas de negocio de la aplicación del Algoritmo Genético (AG) en el sistema, en las cuales constan las siguientes:

- Validación y administración las entidades horarias.
- Inicialización de la población con los datos de entrada (carreras, periodo académico, generaciones y tamaño de población) para generar uno o varios horarios temporales.
- Verificación de restricciones (espacios y tipos, bloques de horarios, etc.).
- Administración de los servicios de API REST y seguridad.
- Conexión con la base de datos, y uso de librerías (Lombok, Mapstruct, JPA, etc.)

**Base de datos:** Es donde se almacena la información utilizada en el sistema (SQL Server).

## 2.5.2. Diagramas de casos de uso

Un diagrama de casos de uso representa las interacciones entre el sistema y los sistemas externos y los usuarios. En otras palabras, describe gráficamente quién utiliza el sistema y la forma en que los usuarios esperan interactuar con el sistema. Estas siguen las especificaciones del lenguaje de modelado unificado (UML) [27].

Los siguientes diagramas muestran la interacción de los actores (director - decano) con las funcionalidades principales del sistema, facilitando la comprensión del comportamiento y administración del sistema en general.

### Inicio de sesión

Este caso de uso describe como los usuarios (director o decano) pueden acceder al módulo web de gestión de horarios académicos. El proceso incluye la validación de credenciales de usuario, verificación del estado, y por consecuente los roles asociados. Si algo falla, el sistema notificara con un mensaje de alerta.

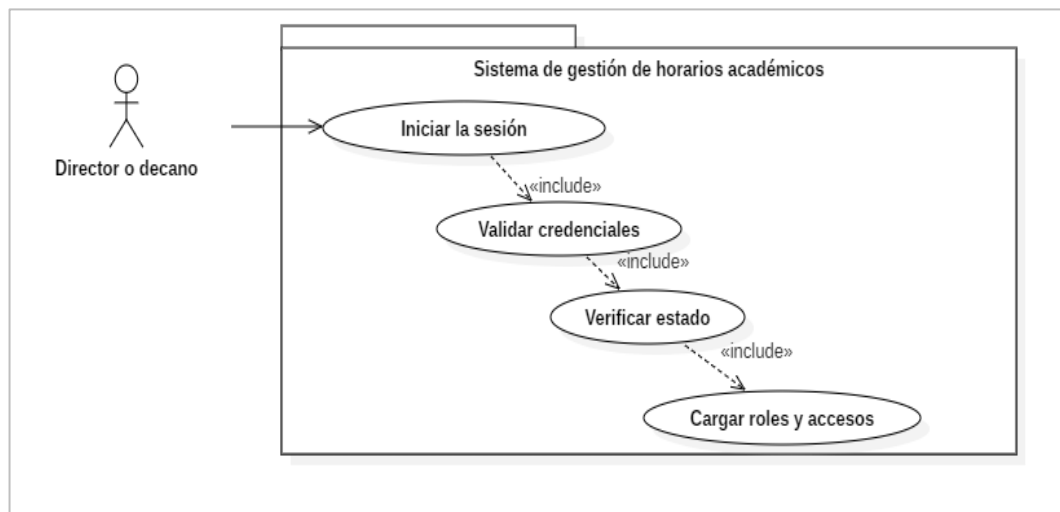


Figura 7. Caso de uso de inicio de sesión.

Caso de uso: Iniciar sesión	
<b>Actores</b>	Director o decano de carrera
<b>Descripción General</b>	Permite el acceso al sistema.
<b>Pre-Condición</b>	Tener privilegios de usuario. Tener abierto el sistema en un navegador web.

<b>Flujo Normal</b>	<ul style="list-style-type: none"> <li>• El director o decano ingresa sus credenciales de usuario al sistema.</li> <li>• El sistema valida la información ingresada y autentica al usuario.</li> </ul>
<b>Post Condición</b>	Ingreso a los módulos del sistema.

Tabla 3. Especificación de caso de uso de inicio de sesión.

### Distributivo

Este caso describe como se construye y mantiene el distributivo de docentes, mismo que sirve para definir y establecer la asignación de cada docente con las asignaturas y paralelos correspondientes a cada carrera de la facultad. El algoritmo utiliza este tipo de datos como parte de las restricciones fuertes al momento de generar y optimizar los horarios académicos. El usuario autenticado puede crear, modificar y eliminar registros de esas asignaciones.

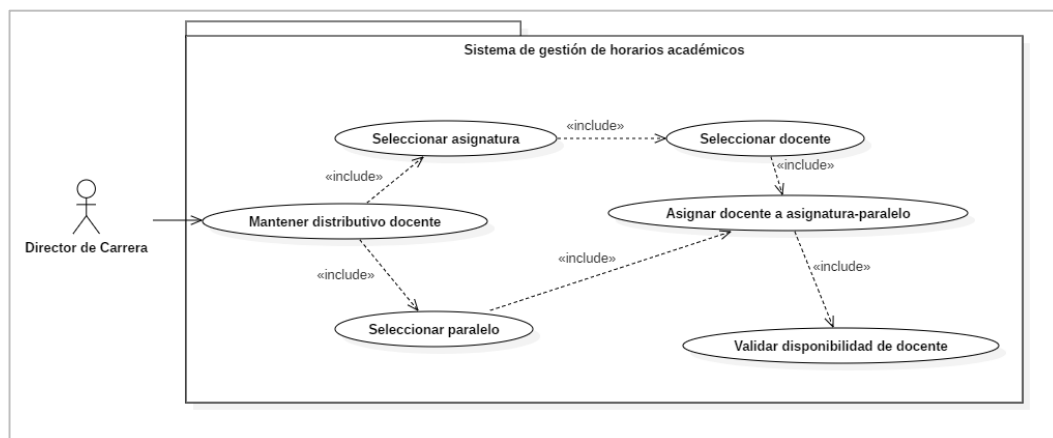


Figura 8. Caso de uso de distributivo de docentes.

Caso de uso: Administración del distributivo	
<b>Actores</b>	Director de Carrera
<b>Descripción General</b>	Permite asignar y mantener a los docentes con las asignaturas y paralelos de las distintas carreras de cada facultad definidas en el sistema.
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• Haber iniciado la sesión.</li> </ul>

	<ul style="list-style-type: none"> <li>• Registros de docentes y asignaturas paralelos.</li> </ul>
<b>Flujo Normal</b>	<ul style="list-style-type: none"> <li>• El usuario autenticado (director) ingresa a la sección de distributivo, en el menú lateral del sistema.</li> <li>• En la sección seleccionada, el usuario podrá crear, modificar o eliminar los registros del distributivo de docentes en el formulario.</li> <li>• Si el proceso es correcto, el sistema muestra un mensaje de éxito.</li> </ul>
<b>Post Condición</b>	La información es procesada y guardada en la base de datos.

Tabla 4. Especificación de caso de uso del uso de distributivo de docentes.

### Restricciones de docentes

Este caso está conformado por las reglas y restricciones en general que el algoritmo toma para condicionar la generación de horarios académicos. El usuario autenticado puede registrar la disponibilidad o situaciones puntuales de cada docente, así también como la definición de uso de distintos tipos de espacios físicos en asignaturas, configurar franjas horarias, definir horas libres y establecer jornadas con los respectivos paralelos.

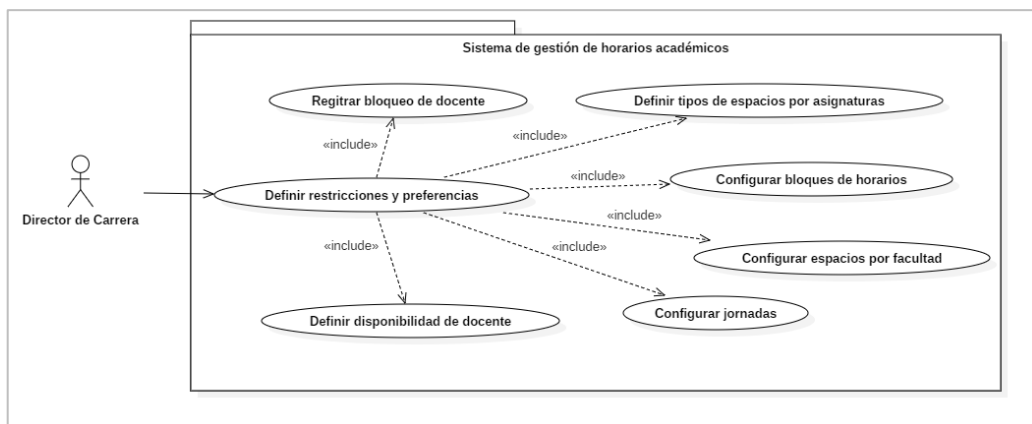


Figura 9. Caso de uso de restricciones y preferencias.

Caso de uso: Administración de restricciones de docentes	
<b>Actores</b>	Director de carrera
<b>Descripción General</b>	Permite definir las restricciones o bloqueos en el sistema.
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• Haber iniciado la sesión.</li> <li>• Tener registro de docentes.</li> </ul>
<b>Flujo Normal</b>	<ul style="list-style-type: none"> <li>• El usuario autenticado (director) ingresa a la sección de restricciones, bloqueo de docentes en el sistema.</li> <li>• En la sección seleccionada, el usuario podrá crear, modificar o eliminar los registros de bloqueo en el formulario, así también como la de establecer reglas en otras secciones.</li> <li>• Si el proceso de registro es correcto, el sistema muestra un mensaje de éxito.</li> </ul>
<b>Post Condición</b>	La información es procesada y guardada en la base de datos.

Tabla 5. Especificación de casos de uso de restricciones de docentes.

## Paralelos

Este caso describe como se construye y mantiene la información de los paralelos, y las definiciones jornadas asociadas, las cuales forman parte de las restricciones.

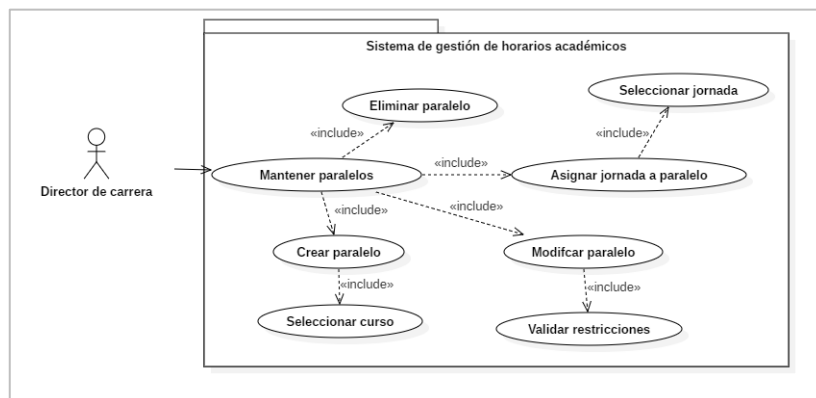


Figura 10. Caso de uso de administración de paralelos.

Caso de uso: Administración de paralelos	
<b>Actores</b>	Director de carrera
<b>Descripción General</b>	Permite administrar la información de los paralelos asociados a cada curso de una determinada carrera,
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• Haber iniciado la sesión.</li> <li>• Tener registro de docentes y jornadas.</li> </ul>
<b>Flujo Normal</b>	<ul style="list-style-type: none"> <li>• El usuario autenticado (director) ingresa a la sección de paralelos en el menú lateral del sistema.</li> <li>• En la sección seleccionada, el usuario podrá crear o modificar los registros de paralelos en el formulario.</li> <li>• Si el proceso de registro es correcto, el sistema muestra un mensaje de éxito.</li> </ul>
<b>Post Condición</b>	La información es procesada y guardada en la base de datos. El valor de jornada es determinante en este registro ya que se considera una restricción fuerte.

Tabla 6. Especificación de casos de uso de administración de paralelos.

## Asignaturas

Este caso describe como se administran las asignaturas de todas las carreras. El mismo define también un tipo de restricción dirigida a los espacios físicos.

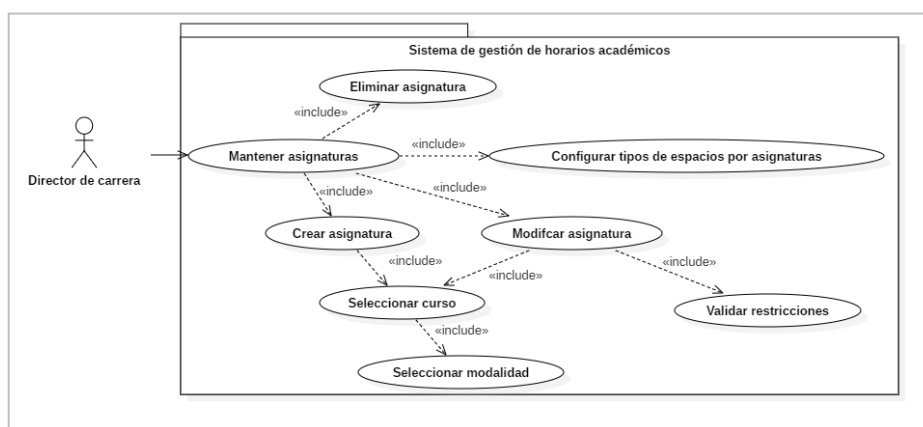


Figura 11. Caso de uso de administración de asignaturas.

Caso de uso: Administración de asignaturas	
<b>Actores</b>	Director de carrera
<b>Descripción General</b>	Permite administrar la información de las asignaturas de una determinada carrera.
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• Haber iniciado la sesión.</li> <li>• Tener registro de asignaturas, cursos, etc.</li> </ul>
<b>Flujo Normal</b>	<ul style="list-style-type: none"> <li>• El usuario autenticado (director) ingresa a la sección de asignaturas en el menú lateral.</li> <li>• En la sección seleccionada, el usuario podrá crear o modificar los registros de asignaturas (nombre, curso, etc.) en el formulario.</li> <li>• Si el proceso de registro es correcto, el sistema muestra un mensaje de éxito.</li> </ul>
<b>Post Condición</b>	La información es procesada y guardada en la base de datos.

Tabla 7. Especificación de casos de uso para administración de asignaturas.

## Reportes

Este caso describe como se gestionan los reportes generales de cada generación de horarios académicos de cada carrera o facultad, respectivamente.

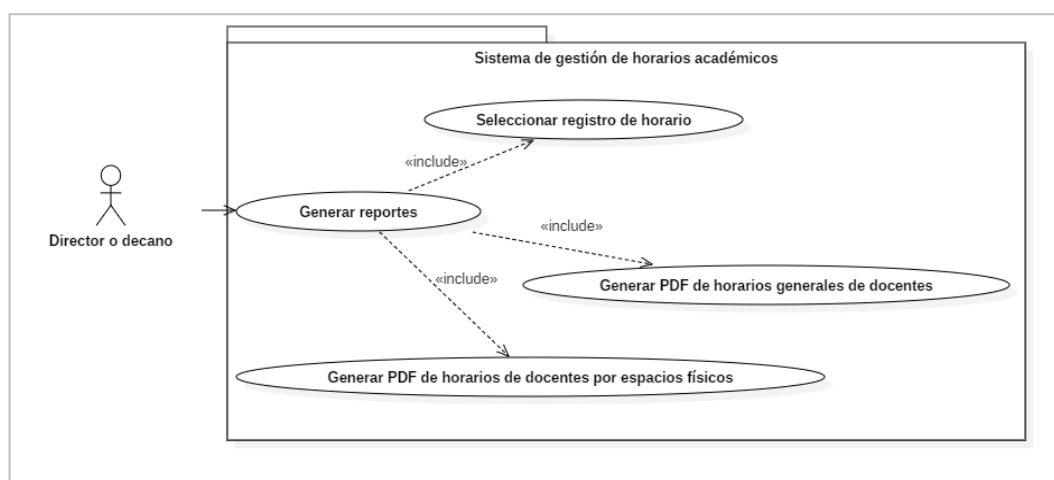


Figura 12. Caso de uso de generación de reportes de horarios.

<b>Caso de uso: Generar Reporte de Horarios</b>	
<b>Actores</b>	Director o decano de carrera
<b>Descripción General</b>	Permite generar un reporte consolidado de los horarios académicos por curso, paralelo, aula o docente.
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• Haber iniciado la sesión.</li> <li>• Contar con horarios registrados en el sistema.</li> </ul>
<b>Flujo Normal</b>	<ul style="list-style-type: none"> <li>• El usuario autenticado (director o decano) ingresa a la sección de reportes en horario temporal, en el menú lateral.</li> <li>• Para generar los reportes, el usuario debe seleccionar que tipo de reportes va a generar (Horario de docentes en general o por tipos de espacios físicos), además de seleccionar el registro el horario generado.</li> <li>• En la sección seleccionada, el usuario podrá generar los reportes en formato .PDF</li> <li>• Si el proceso de generación es correcto, el sistema muestra un mensaje de éxito.</li> </ul>
<b>Post Condición</b>	El reporte queda disponible para consulta o descarga, al igual que la información queda a total disponibilidad

Tabla 8. Especificación de caso de uso de generación de reportes de horarios.

### **Horarios temporales**

Este caso de uso describe el proceso mediante el cual se construye automáticamente los horarios académicos en base a la solución establecida (Algoritmo genético). El usuario autenticado selecciona el periodo académico, la carrera o facultad en general, seguido de los parámetros de entrada (número de población y generaciones), que son determinantes para encontrar a las posibles soluciones.

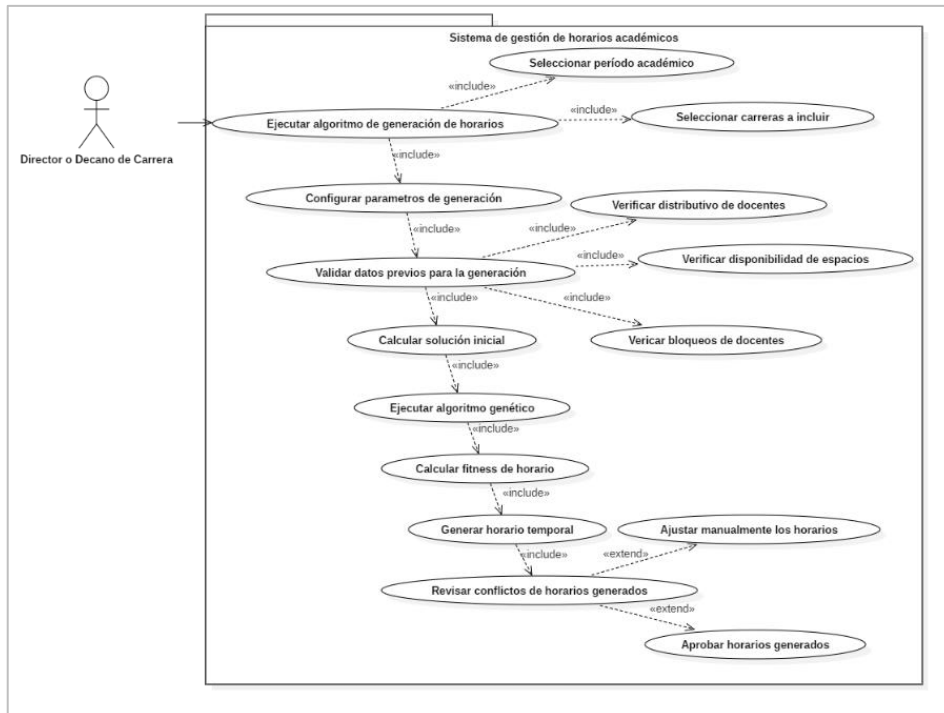


Figura 13. Caso de uso de la ejecución del algoritmo para la generación de horarios.

Caso de uso: Generación de horarios temporales	
<b>Actores</b>	Director o decano de facultad
<b>Descripción General</b>	Permite definir los parámetros necesarios para la generación de horarios académicos de una carrera y facultad.
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• Haber iniciado la sesión.</li> <li>• Tener registro de distributivo, carreras, restricciones de docentes, bloques de horarios, etc., en general.</li> </ul>
<b>Flujo Normal</b>	<ul style="list-style-type: none"> <li>• El usuario autenticado (director) ingresa a la sección de horarios temporales en el sistema.</li> <li>• En la sección seleccionada, el usuario podrá crear los registros de horarios temporales (carreras, periodo académico, generaciones, etc.) en el formulario.</li> </ul>

	<ul style="list-style-type: none"> <li>• Si el proceso de registro es correcto, el sistema muestra un mensaje de éxito.</li> </ul>
<b>Post Condición</b>	La información es procesada y guardada en la base de datos.

Tabla 9. Especificación de casos de uso de generación de horarios temporales.

<b>Caso de uso: Modificación de horarios temporales</b>	
<b>Actores</b>	Director o decano de facultad
<b>Descripción General</b>	Permite la modificación de registros generados por el algoritmo, en cuanto a los horarios temporales.
<b>Pre-Condición</b>	<ul style="list-style-type: none"> <li>• Haber iniciado la sesión.</li> <li>• Tener registros de horarios temporales.</li> </ul>
<b>Flujo Normal</b>	<ul style="list-style-type: none"> <li>• El usuario autenticado (director) ingresa a la sección de horarios temporales y busca un registro en el sistema.</li> <li>• En la sección seleccionada, el usuario podrá modificar por completo una sesión de clase de un paralelo en particular, mediante la acción de arrastrar y soltar los eventos de clase en el calendario. El sistema muestra los espacios y horas disponibles de acuerdo al evento seleccionado y modificado.</li> <li>• El sistema valida la disponibilidad de docentes, espacios, bloqueos, etc.</li> <li>• Si el proceso de modificación es correcto, el sistema muestra un mensaje de éxito.</li> </ul>
<b>Post Condición</b>	La información es procesada y guardada en la base de datos.

Tabla 10. Especificación de casos de uso de modificación de horarios temporales.



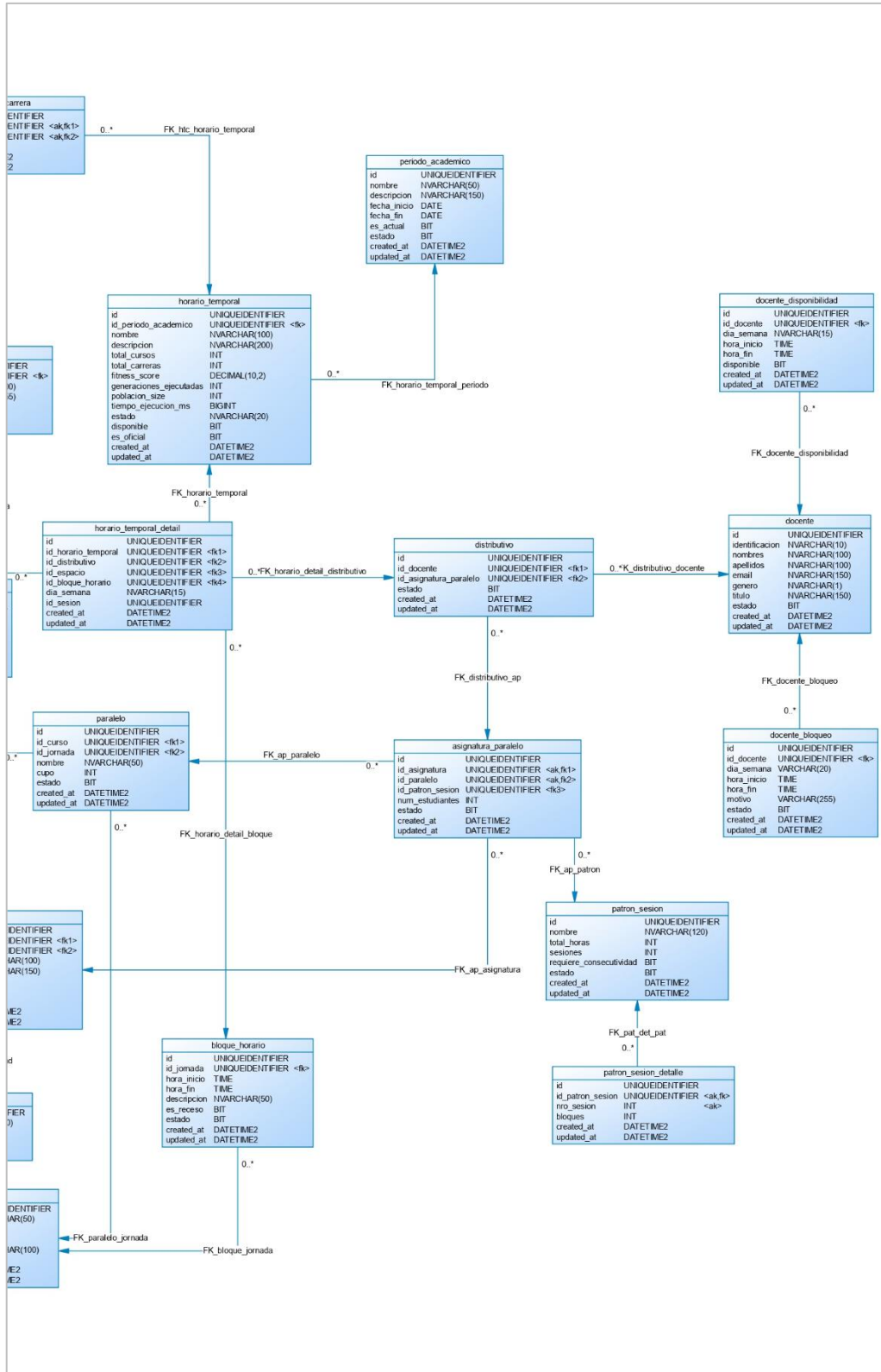


Figura 15. Modelo de datos (Parte 2).

Un modelo de datos se define como un conjunto de símbolos, conceptos y reglas que nos permiten representar los datos que se van a almacenar en una base de datos. El resultado de la aplicación de un modelo de datos, es decir, la plasmación de la parte de la realidad para la cual deseamos crear la base de datos (UD) mediante el empleo de un determinado modelo de datos, da lugar a lo que se denomina un esquema. Existen varios tipos de modelos de datos aplicables en distintos momentos a lo largo del proceso de creación de una base de datos (modelos conceptuales, modelos lógicos y modelos físicos) dando lugar a diferentes tipos de esquemas (esquemas conceptuales, esquemas lógicos y esquemas físicos, respectivamente) [25]. Las entidades del modelo de datos utilizados en el desarrollo del proyecto se describen a continuación:

## **ESTRUCTURA ACADÉMICA**

**Facultad:** Son las grandes divisiones de la universidad. Sirven para agrupar varias carreras bajo una misma unidad. Por ejemplo, “Facultad de Sistemas y Telecomunicaciones”.

**Carrera:** Son programas de estudios en concreto, y que pertenecen a una facultad en particular, y a su vez, se relaciona con cursos, asignaturas, y paralelos. Por ejemplo, “Tecnologías de la Información”.

**Curso:** Representa un nivel dentro una carrera, y permite ubicar en que plan de estudios se encuentra con cada distribución de asignaturas y paralelos. Por ejemplo: “Curso 1”.

**Asignatura:** Son las materias que se imparten en la universidad, mismas que son claves en la distribución con los docentes. Por ejemplo, “Programación I”.

**Paralelo:** Representa al grupo de estudiantes de una asignatura en un curso y jornada respectivamente. El paralelo determina un cupo máximo, el cual es quien termina ocupando un espacio en un horario determinado. Por ejemplo: “1”.

**Patrón sesión:** Son las sesiones establecidas para cada asignación del distributivo. Por ejemplo: “Matemáticas I - Horas por semana (4) - Sesiones (2)”.

**Patrón sesión detalle:** Establece el número de bloques de horarios por cada sesión.

## PERSONAS Y DISPONIBILIDAD

**Docente:** Representa a una persona, a la que se le asignan las asignaturas en el distributivo. Por ejemplo: “Jaime Orozco”.

**Docente Disponibilidad:** Registro que indica los días y horas hábiles para cada docente. Por ejemplo: “Disponible de 10:00AM a 14:00PM”.

**Docente bloqueo:** Son todas las excepciones dentro de la disponibilidad. Este registro valida que no se asigne un horario en un tiempo determinado. Por ejemplo: “Jaime Orozco - Permiso por clases de doctorado”.

## MODALIDAD Y ESPACIOS FÍSICOS

**Modalidad:** Representa si una asignatura es de carácter presencial, virtual, híbrida, etc. Este registro es importante ya que nos ayuda a determinar si necesita o no un espacio en el horario. Por ejemplo: “Legislación de Proyectos - Virtual”.

**Tipo Espacio:** Clasifica los tipos de escenarios donde se imparten las clases. Este registro representa el tipo que necesita cada asignatura. Por ejemplo: “Laboratorio”.

**Sector - Edificación - Espacio:** En conjunto representan un espacio físico al cual pertenece cada facultad de la universidad. Por ejemplo: “B - B1 - 101”.

**Facultad Edificación:** Representa a las relaciones de cada facultad con una edificación determinada. Por ejemplo: “Sistemas y Telecomunicaciones - B”.

**Asignatura Tipo Espacio:** Indica que tipo de espacio requiere cada asignatura. Por ejemplo: “Programación II - Laboratorio”.

## BLOQUES DE TIEMPO Y JORNADA

**Jornada:** Representa los turnos del día laborable. Este registro se relaciona con cada paralelo y bloque de horario para saber que turno se dicta cada clase. Por ejemplo: “Matutina - 07:30AM a 13:00:PM”

**Bloque horario:** Son los intervalos de tiempo específicos que tiene una jornada. El AG usa este registro para ubicar las sesiones en las clases. Por ejemplo: “07:30AM a 08:00AM”.

## **DISTRIBUTIVO**

**Asignatura Paralelo:** Representa la asignación de una asignatura con un paralelo (grupo de estudiantes). Por ejemplo: “Métodos numéricos - Curso 4/1”.

**Distributivo:** Representa una lista de clases para cada docente a impartir en la semana laboral, pero sin ningún día, espacio asignado u hora establecida, respectivamente. Por ejemplo: “Jaime Orozco - Aplicaciones móviles - Curso 6/1”.

## **HORARIOS GENERADOS**

**Horario temporal:** Representa el registro de un horario de clases de una o más carreras asociadas creado por el algoritmo. Por ejemplo: “Horario temporal borrador - Tecnologías de la Información - Software”.

**Horario Temporal Carrera:** Relaciona un horario temporal con una o varias carreras.

**Horario Temporal Detalle:** Representa al detalle de cada generación automática del horario temporal, el cual contiene la información del espacio, registro del distributivo, bloques de horarios, y la sesión, respectivamente.

## **USUARIOS Y ROLES**

**Rol:** Representa los diferentes tipos de usuarios en el sistema. Por ejemplo: “Administrador, Director, Docente, etc.”.

**Usuario:** Representa a las personas que se conectan al sistema. Por ejemplo: “carlos@upse.edu.ec”.

**Usuario Rol:** Representa a la relación de muchos usuarios con muchos tipos de usuarios. Por ejemplo: “José Sánchez - Director”.

**Carrera Director:** Representa la vinculación de un usuario con una carrera en particular de la universidad. Por ejemplo: “José Sánchez - Carrera Software”.

**Facultad Decano:** Representa la relación de cada facultad universitaria con el decano (usuario) de la misma. Por ejemplo: “Sistemas y Telecomunicaciones - Washington Torres”.

## 2.5.4. Diseño de interfaces

### Autenticación de usuarios

Para acceder al módulo web, los usuarios (directores, decanos) deben de ingresar con sus respectivas credenciales de usuario.

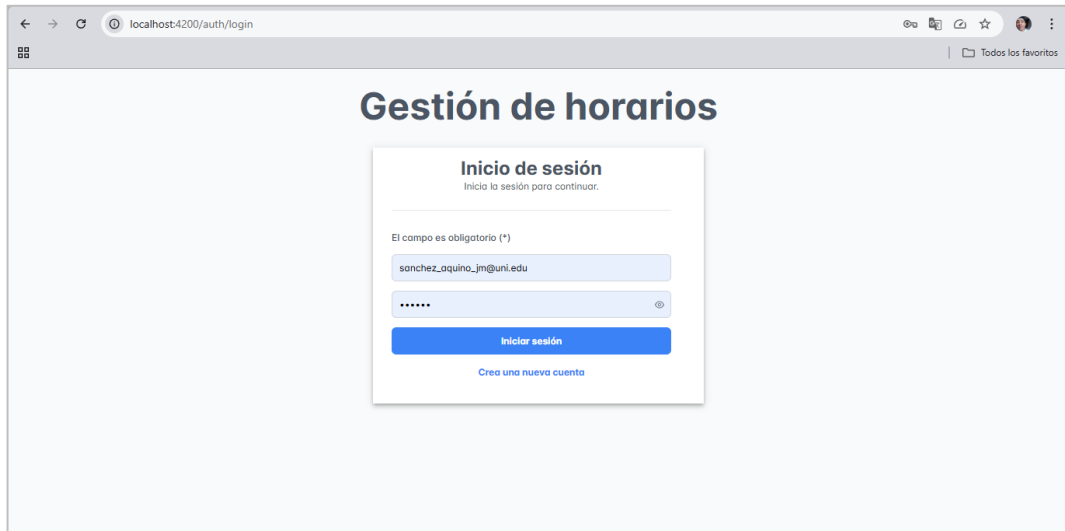


Figura 16. Pantalla de inicio de sesión.

### Administración de asignaturas

En esta sección, el usuario (director autenticado), podrá administrar la información de todas las asignaturas de las carreras que le correspondan.

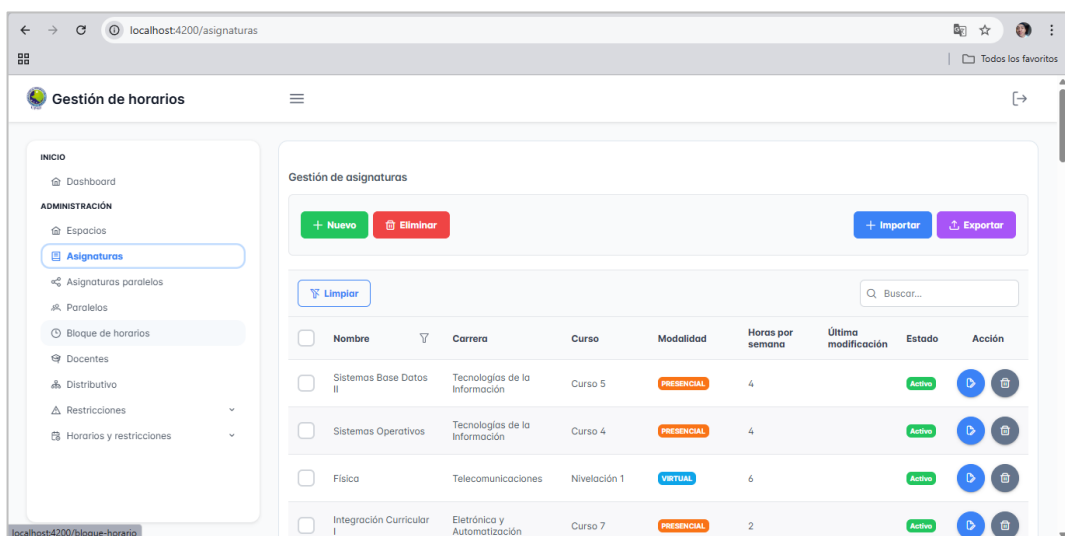


Figura 17. Pantalla de administración de asignaturas.

Para la búsqueda de múltiples registros, la gestión de asignaturas cuenta con un campo de texto (filtro) para poder acceder de manera directa a la información que sea requerida por el usuario.

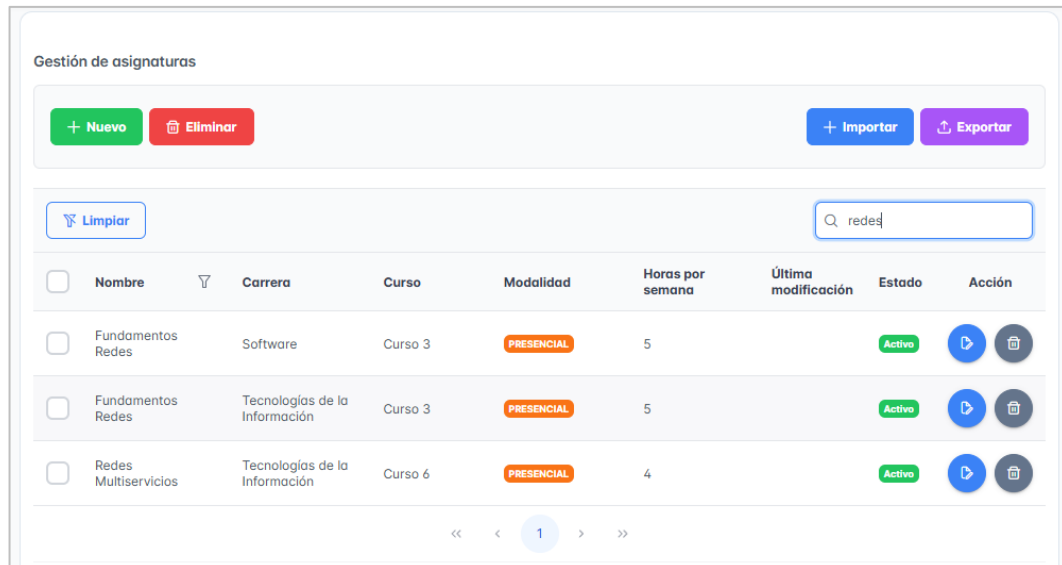


Figura 18. Pantalla de filtro para las asignaturas.

Para administrar la información de cada asignatura, respecto a una carrera independiente, el usuario podrá modificar los campos de entrada que sean necesarios (nombre, descripción, modalidad, curso, tipo de espacio, horas por semana, etc.), según la administración en general.

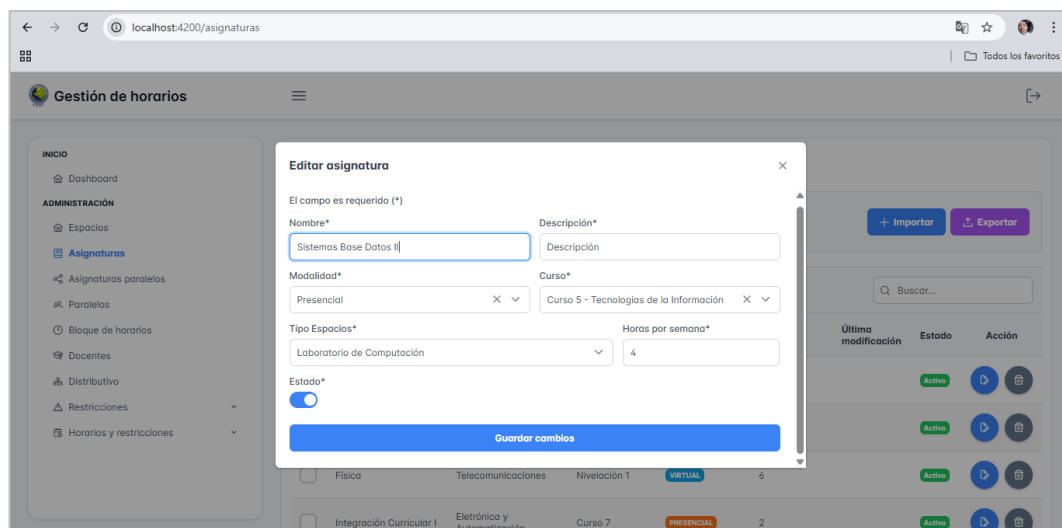


Figura 19. Pantalla de creación o modificación de registros de asignaturas.

## Administración de paralelos

En esta sección, el usuario (director autenticado) podrá administrar la información de todos los paralelos asociados a sus respectivos cursos, de las carreras que le correspondan.

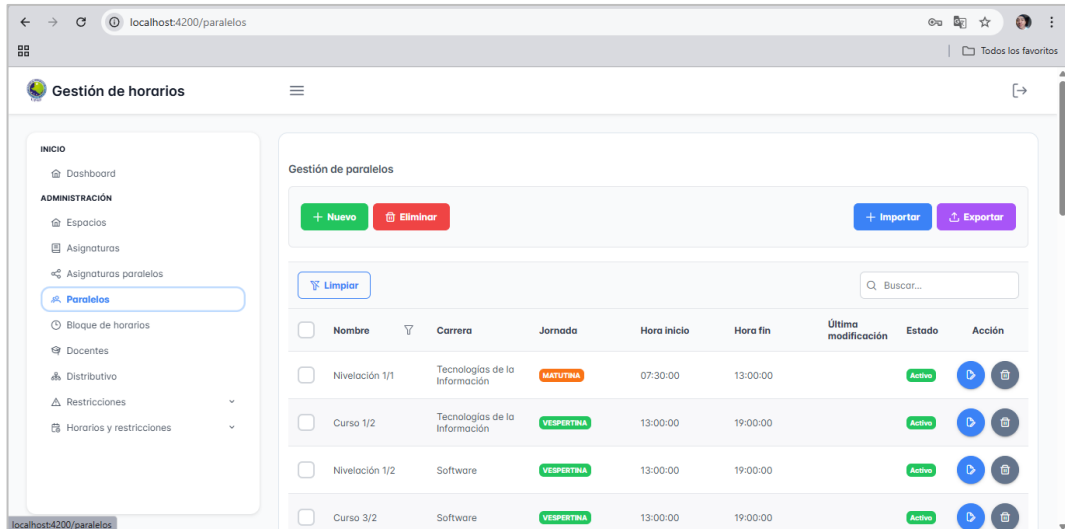


Figura 20. Pantalla de administración de paralelos.

Para la búsqueda de múltiples registros, la gestión de paralelos cuenta con un campo de texto (filtro) para poder acceder de manera directa a la información que sea requerida por el usuario.

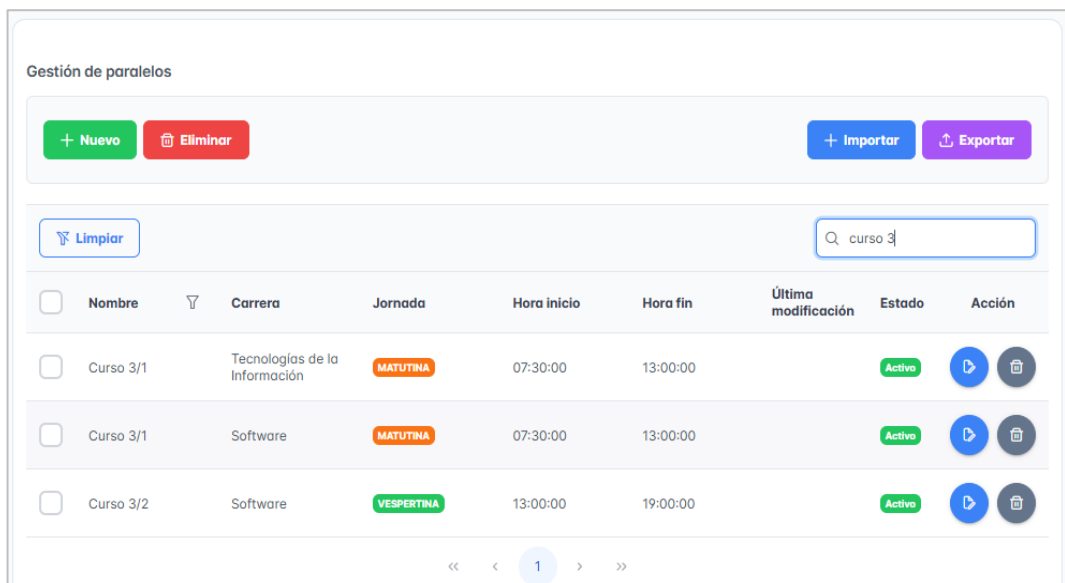


Figura 21. Pantalla de filtro para los paralelos.

Para administrar la información de cada paralelo, respecto a un curso de una carrera independiente, el usuario podrá modificar los campos de entrada que sean necesarios (nombre, cupo de estudiantes, jornada, curso, etc.), según la administración en general.

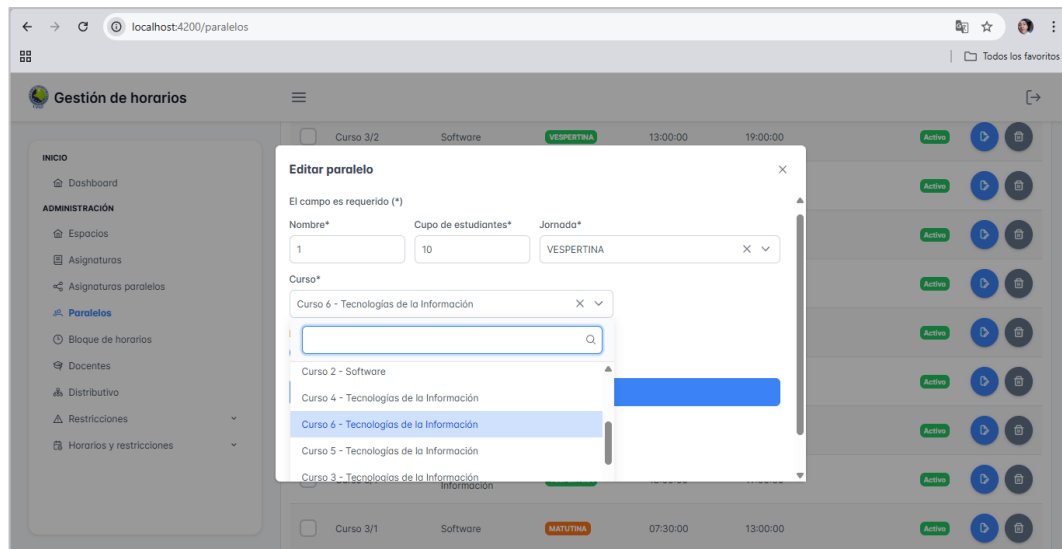


Figura 22. Pantalla de creación o modificación de registros de paralelos.

## Administración de docentes

En esta sección, el usuario (director autenticado) podrá administrar la información de todos los docentes.

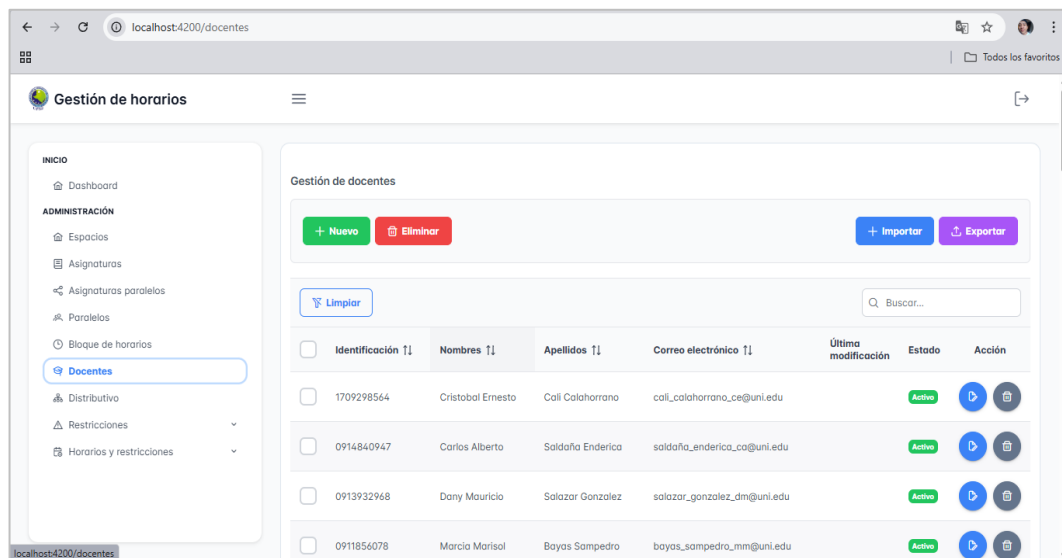


Figura 23. Pantalla de administración de docentes.

## Administración de asignaturas y paralelos

En esta sección, el usuario (director autenticado) podrá administrar la información de asignaturas y paralelos.

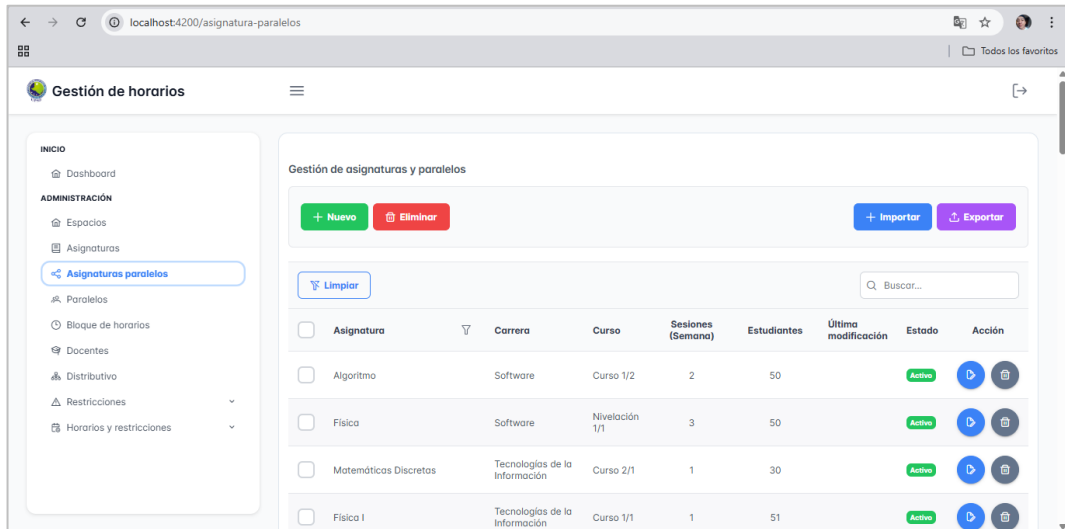


Figura 24. Pantalla de administración de asignaturas y paralelos.

Para administrar la información de cada asignación de asignaturas y paralelos, el usuario podrá modificar los campos de entrada que sean necesarios (asignaturas, paralelo, patrón sesión, etc.), según lo requiera cada situación en particular. La información registrada sirve para validar y configurar el distributivo de docentes en la siguiente captura.

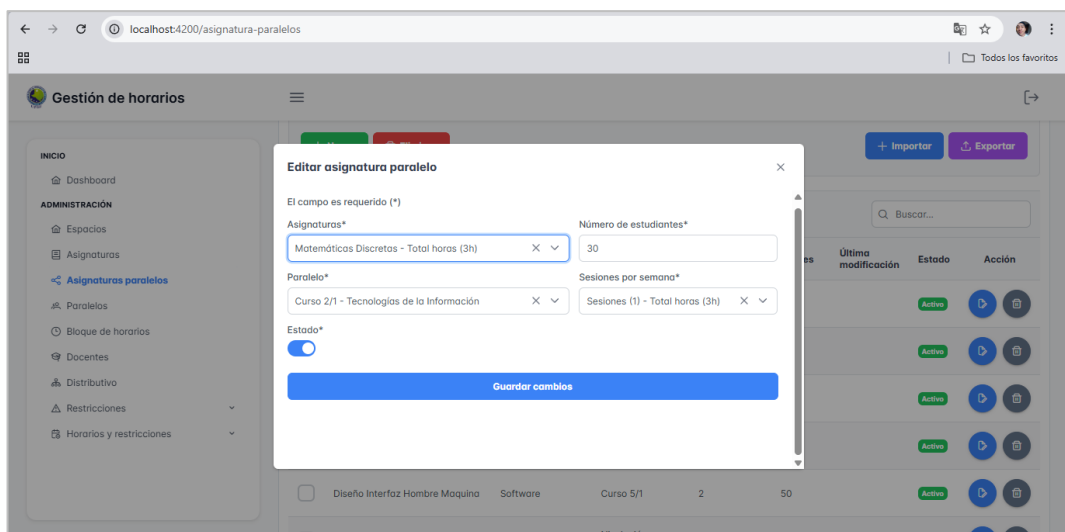


Figura 25. Pantalla de creación o modificación de registros de asignaturas y paralelos.

## Administración del distributivo de docentes

En esta sección, el usuario (director autenticado) podrá administrar la información del distributivo de docentes.

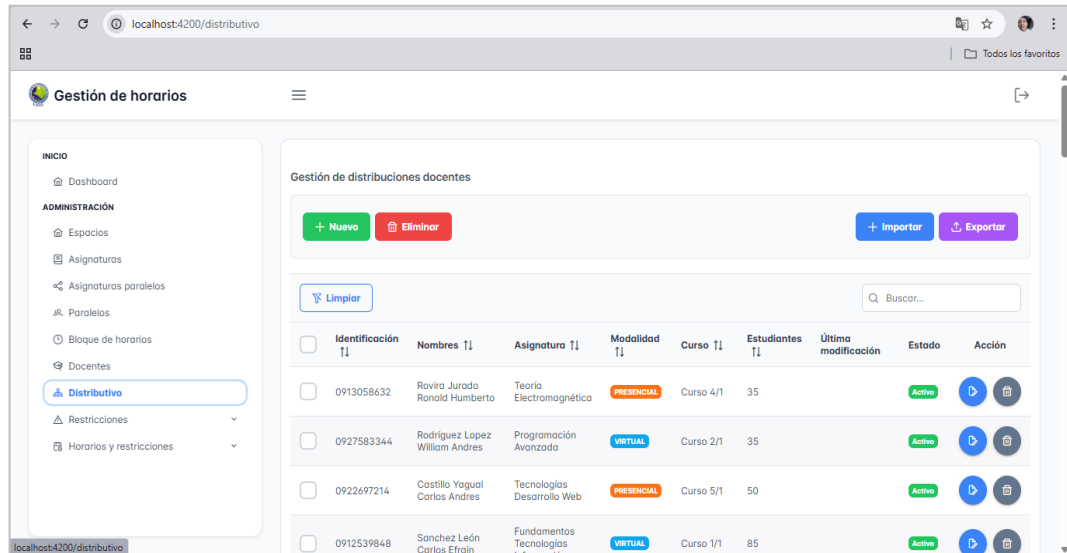


Figura 26. Pantalla de administración del distributivo de docentes.

Para la búsqueda de múltiples registros, la gestión del distributivo de docentes cuenta con un campo de texto (filtro) para poder acceder de manera directa a la información que sea requerida por el usuario.

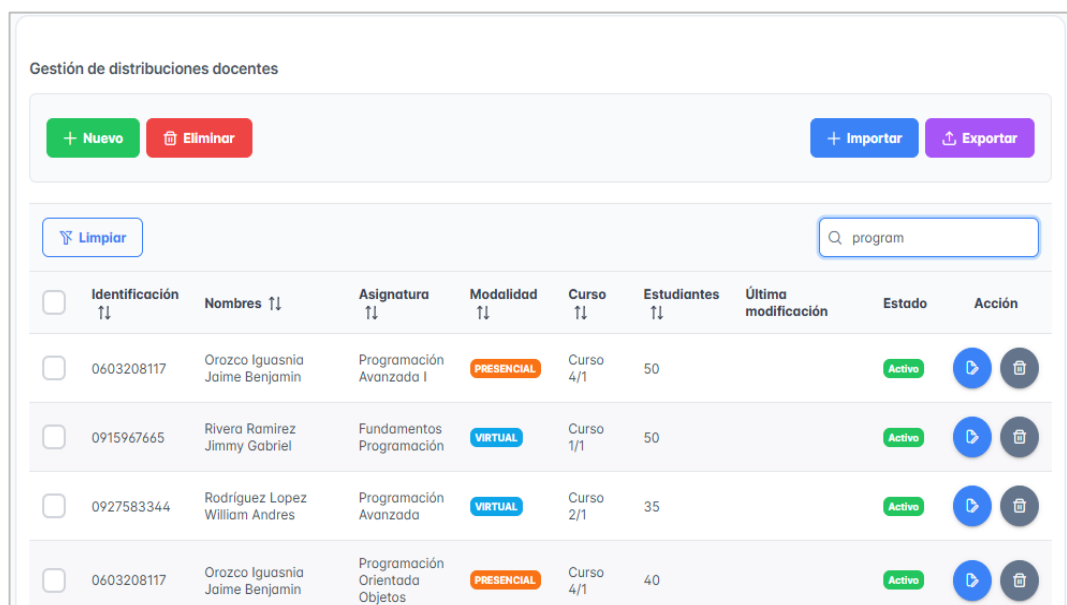


Figura 27. Pantalla de filtro para el distributivo de docentes.

Para administrar la información de cada distributivo, respecto a una carrera independiente, el usuario podrá modificar los campos de entrada que sean necesarios (docente, asignación de asignaturas y paralelos, etc.), según la administración en general.

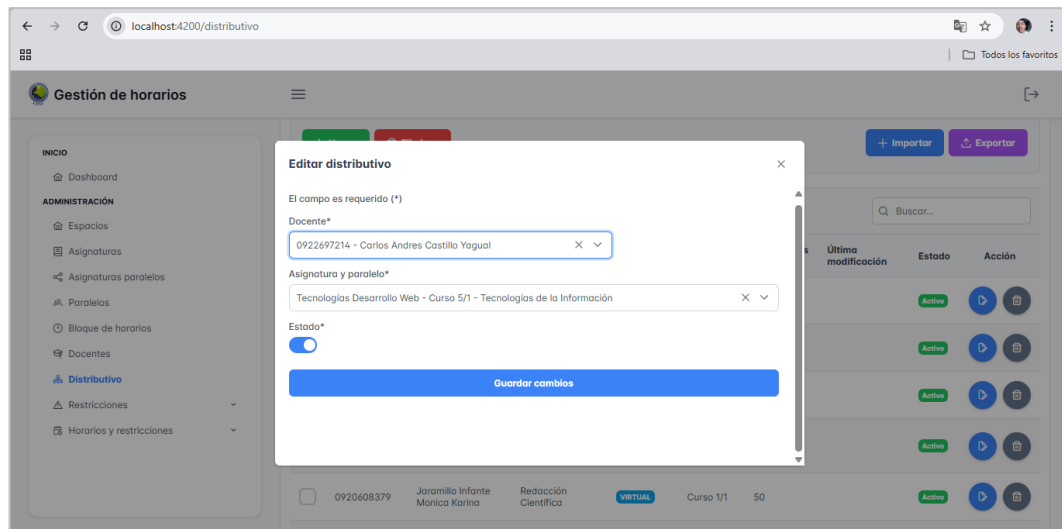


Figura 28. Pantalla de creación o modificación de registros del distributivo de docentes.

### Administración del bloqueo de docentes (restricciones)

En esta sección, el usuario (director autenticado) podrá administrar la información del bloqueo de docentes.

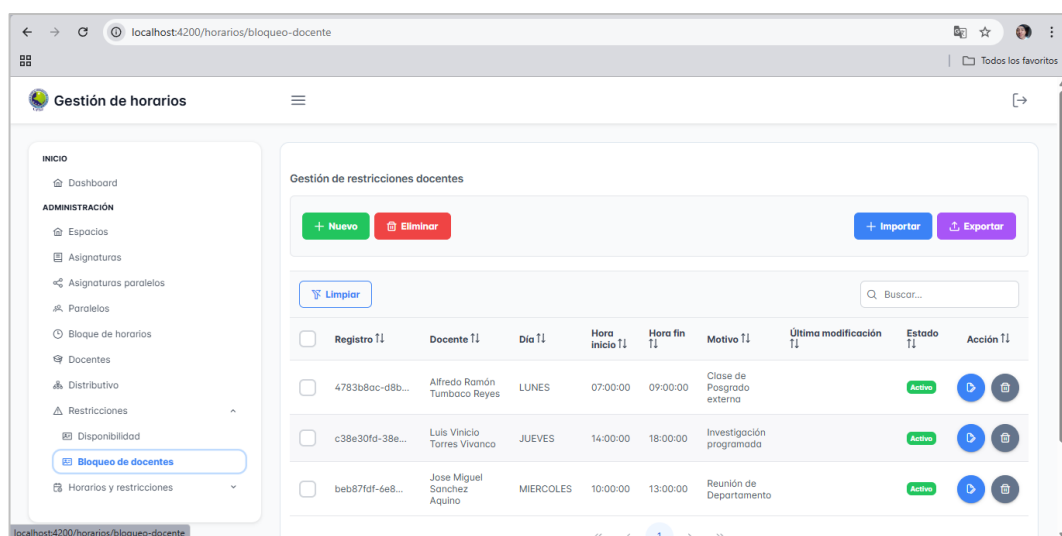


Figura 29. Pantalla de administración del bloqueo de docentes (restricciones).

Para administrar la información de cada bloqueo de docente, el usuario podrá modificar los campos de entrada que sean necesarios (día laborable, docente, motivo, hora de inicio y fin, etc.), según lo requiera cada situación de un docente en particular.

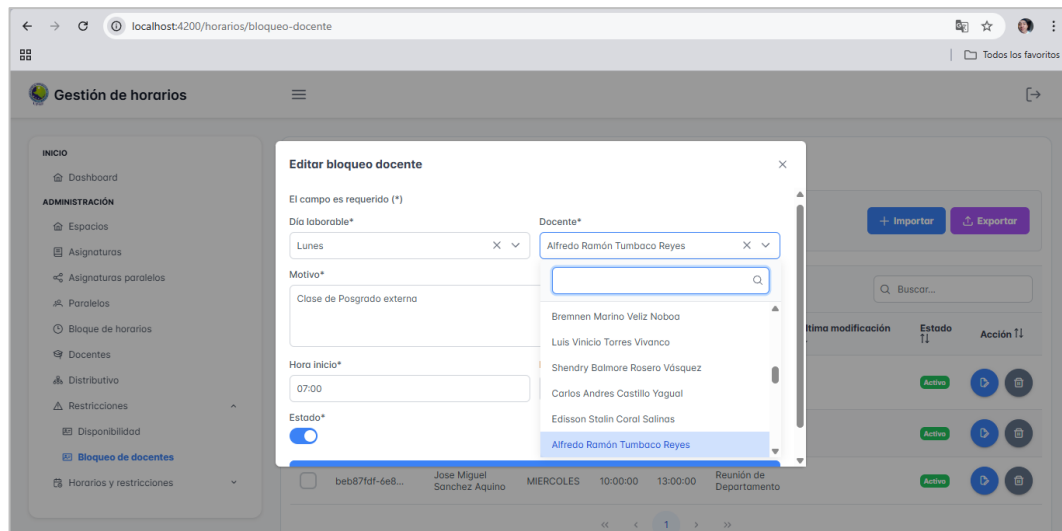


Figura 30. Pantalla de creación o modificación de registros de bloqueo de docentes.

## Administración de disponibilidad de docentes

En esta sección, el usuario (director autenticado) podrá administrar la información de la disponibilidad de docentes.

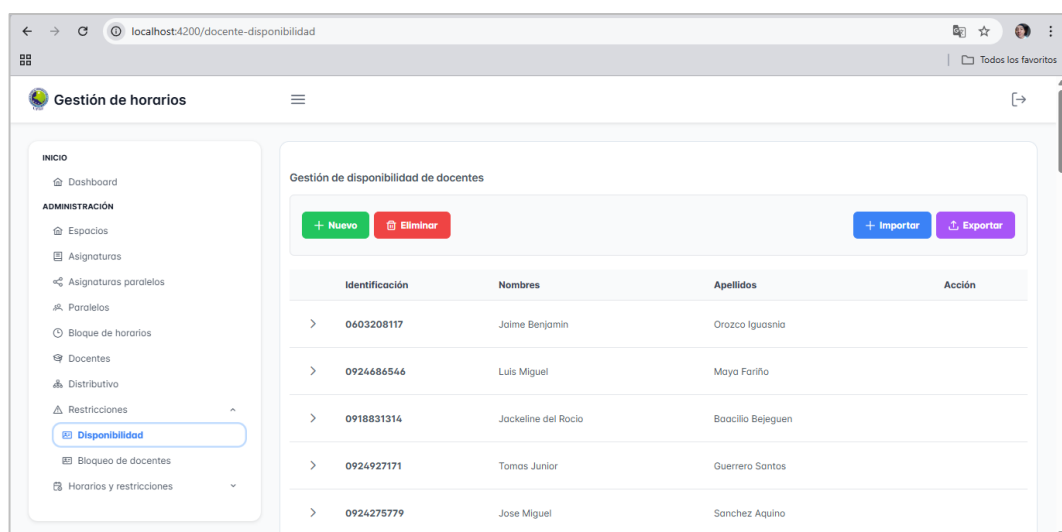


Figura 31. Pantalla de administración de disponibilidad de docentes.

Para administrar la información de la disponibilidad de cada docente, el usuario podrá modificar los campos de entrada que sean necesarios (docente, día laborable, hora de inicio y fin, etc.), según la administración en general.

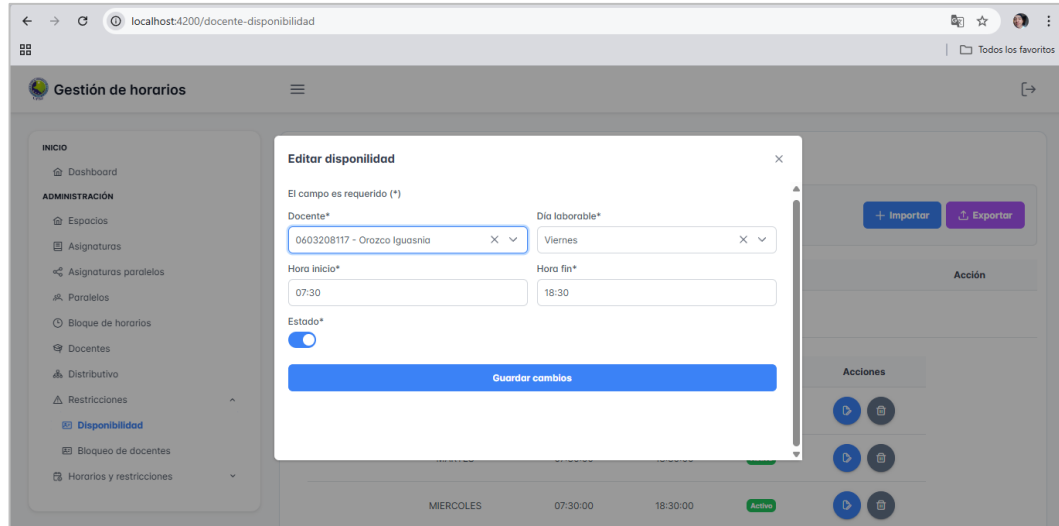


Figura 32. Pantalla de creación o modificación de registros de disponibilidad de docentes.

### Administración de horarios temporales

En esta sección, el usuario (decano autenticado) podrá administrar la información de todos los horarios temporales, ya sea de una o varias carreras, o toda la facultad, según sea el requerimiento.

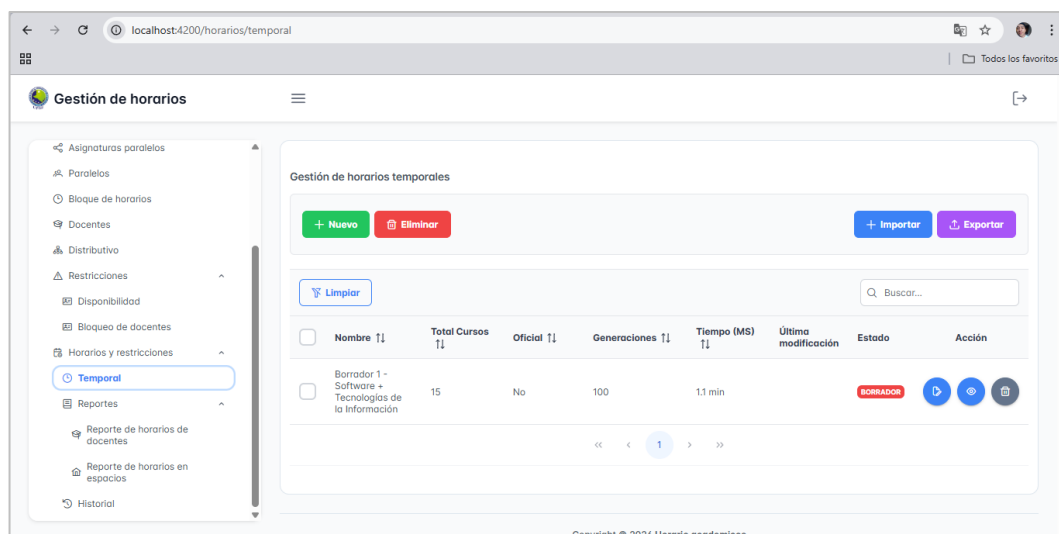


Figura 33. Pantalla de administración de horarios temporales.

Para la creación de un nuevo registro de horario temporal, el usuario podrá modificar los campos de entrada que sean necesarios (carreras, periodo académico, generaciones y población), según lo requiera cada situación en particular, ya sea tanto para generar con una o varias carreras, como la facultad en general, respectivamente.

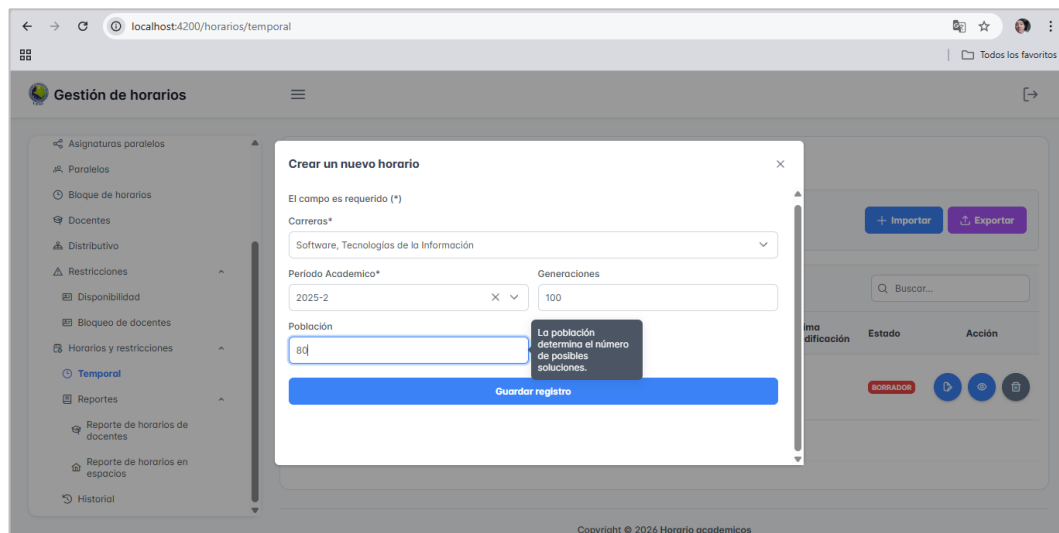


Figura 34. Pantalla de creación de registros de horarios temporales.

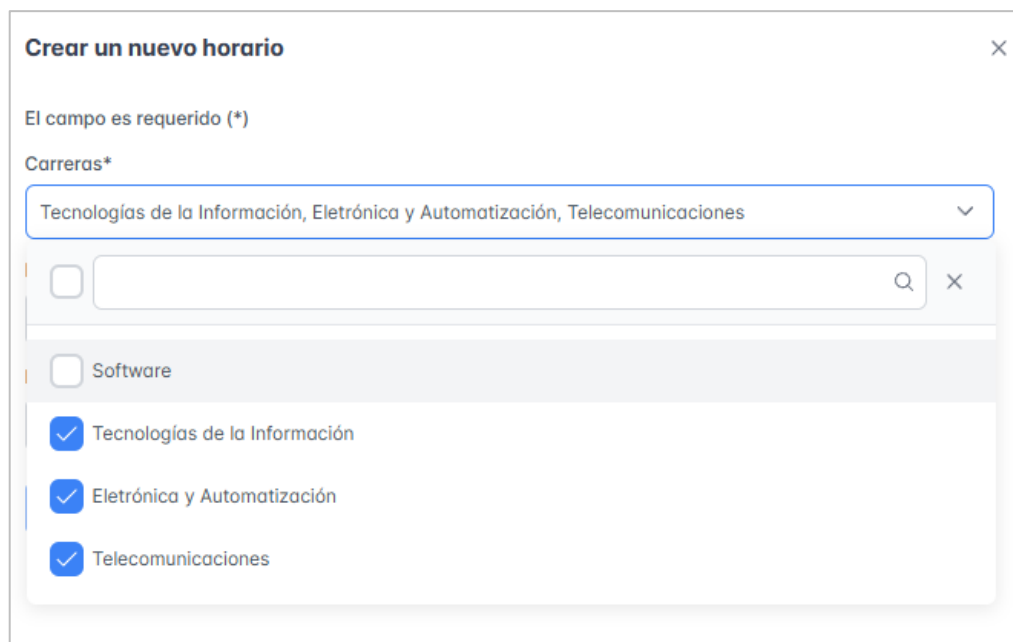


Figura 35. Pantalla de selección de múltiples carreras para la generación de horarios.

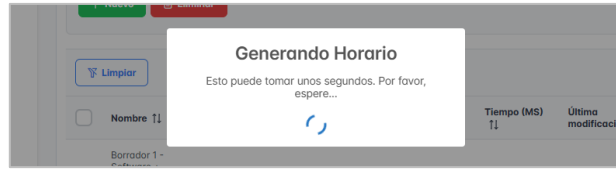


Figura 36. Pantalla de aviso, una vez se esté generando el horario temporal.

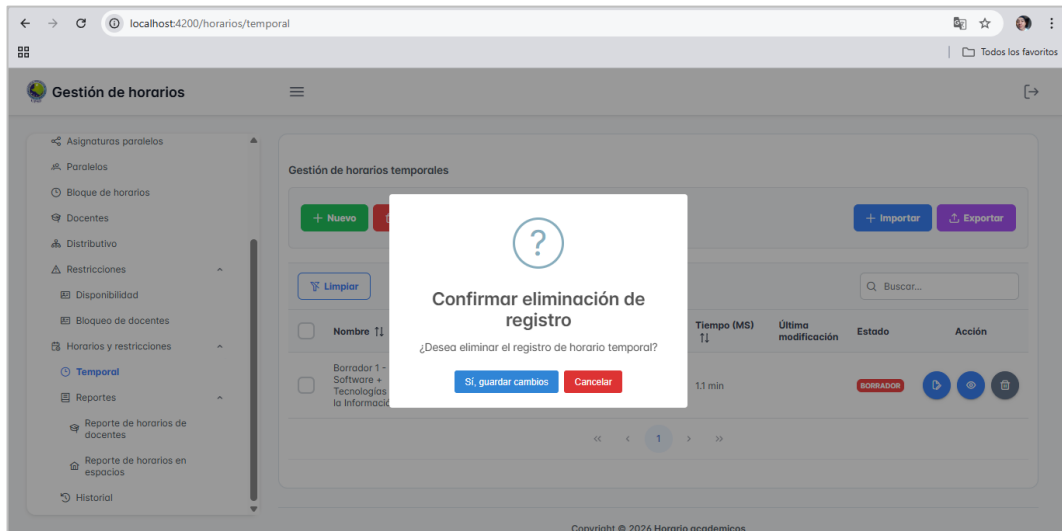


Figura 37. Pantalla de confirmación para eliminar registros de horarios temporales.

Una vez se haya creado el registro del horario temporal, en las opciones, se puede verificar las carreras seleccionadas con los paralelos correspondientes.

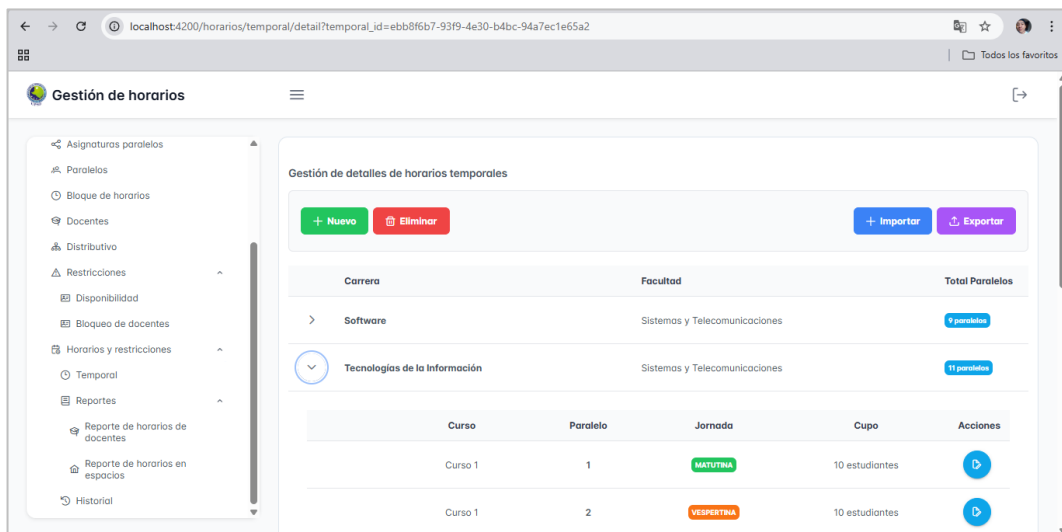
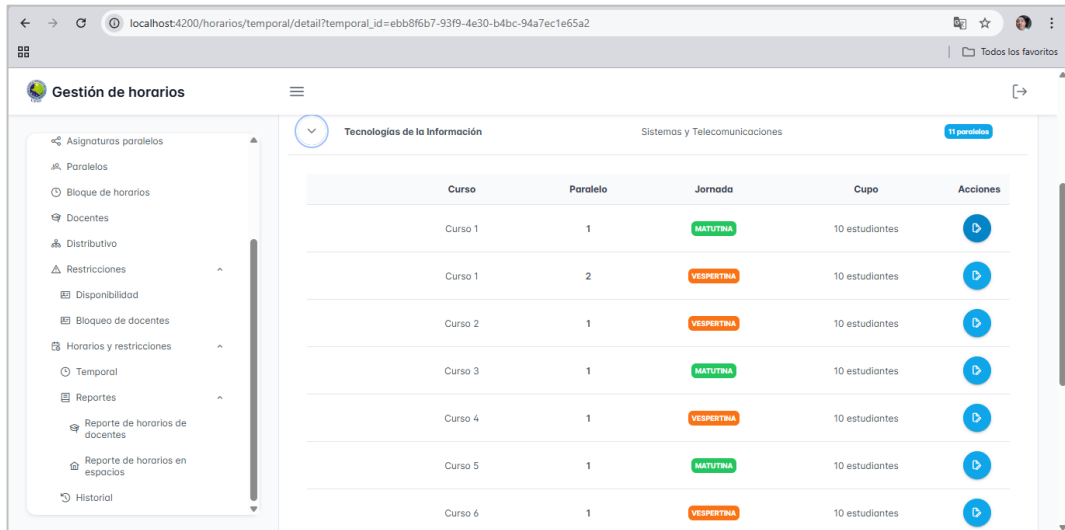


Figura 38. Pantalla de detalle de un registro de horario temporal.

## Gestión de detalles de horarios temporales

El algoritmo ha generado horarios académicos para cada uno de los paralelos correspondientes a cada carrera, validando tanto las restricciones de cada asignatura como también la disponibilidad de los docentes participantes, respetando todos los espacios físicos y la capacidad para alojar a cada uno de ellos.

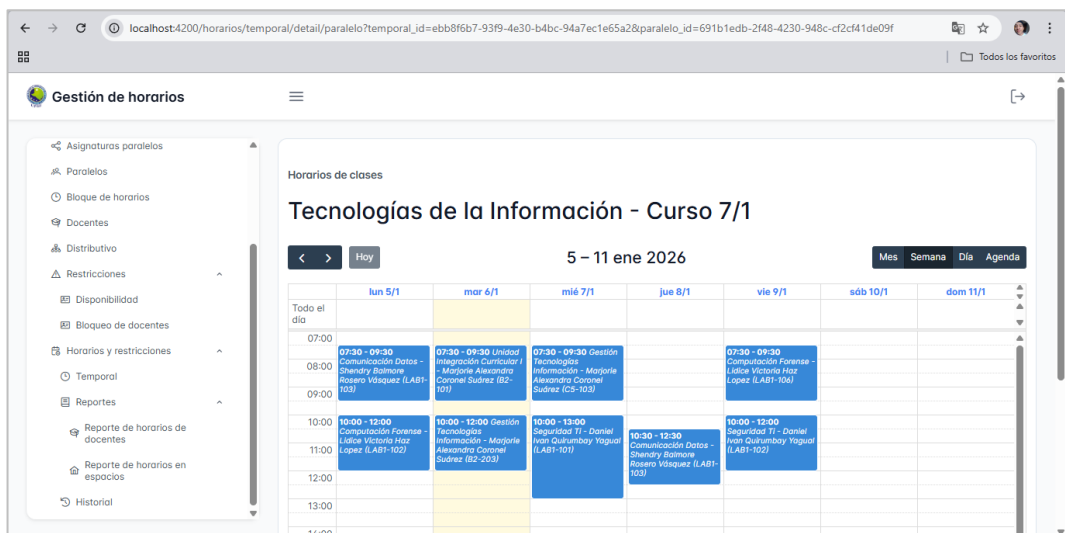


The screenshot shows a web application interface for managing temporary schedules. The main content area displays a table with the following data:

Curso	Paralelo	Jornada	Cupo	Acciones
Curso 1	1	MATUTINA	10 estudiantes	[Icono de acción]
Curso 1	2	VESPERTINA	10 estudiantes	[Icono de acción]
Curso 2	1	VESPERTINA	10 estudiantes	[Icono de acción]
Curso 3	1	MATUTINA	10 estudiantes	[Icono de acción]
Curso 4	1	VESPERTINA	10 estudiantes	[Icono de acción]
Curso 5	1	MATUTINA	10 estudiantes	[Icono de acción]
Curso 6	1	VESPERTINA	10 estudiantes	[Icono de acción]

Figura 39. Pantalla de paralelos correspondiente a una carrera.

En esta sección, el usuario (director o decano autenticado) podrá administrar la información de todos los detalles de los horarios temporales.



The screenshot shows a class schedule for 'Tecnologías de la Información - Curso 7/1' for the week of January 5-11, 2026. The schedule is displayed in a grid format with the following data:

Todo el día	lun 5/1	mar 6/1	mié 7/1	jue 8/1	vie 9/1	sáb 10/1	dom 11/1
07:00							
08:00	07:30 - 09:30 Comunicación Datos - Shendry Balmora Rosero Vázquez (LAB1-103)	07:30 - 09:30 Unidad Integración Curricular I - Marjorie Alejandra Coronel Sudrez (B2-101)	07:30 - 09:30 Gestión Tecnológica - Marjorie Alejandra Coronel Sudrez (B2-103)		07:30 - 09:30 Computación Forense - Lidice Victoria Hoz Lopez (LAB1-106)		
09:00							
10:00	10:00 - 12:00 Computación Forense - Lidice Victoria Hoz Lopez (LAB1-102)	10:00 - 12:00 Gestión Tecnológica - Marjorie Alejandra Coronel Sudrez (B2-203)	10:00 - 13:00 Seguridad TI - Daniel Iván Quiroz Viquez (LAB1-101)	10:30 - 13:00 Comunicación Datos - Shendry Balmora Rosero Vázquez (LAB1-103)	10:00 - 12:00 Seguridad TI - Daniel Iván Quiroz Viquez (LAB1-102)		
11:00							
12:00							
13:00							
14:00							

Figura 40. Pantalla del horario de clases para un paralelo de una carrera.

Si el usuario no está conforme con los resultados obtenidos en la generación automática, manualmente se puede editar esos registros con la acción de “arrastrar y soltar eventos” en el calendario, y una vez se valide que el evento (clase actual), no tenga alguna restricción, ya sea con la disponibilidad de docentes, espacios libres o respetando las jornadas, saldrá la siguiente pantalla, en la cual el usuario podrá seleccionar o el mismo espacio u otro diferente.

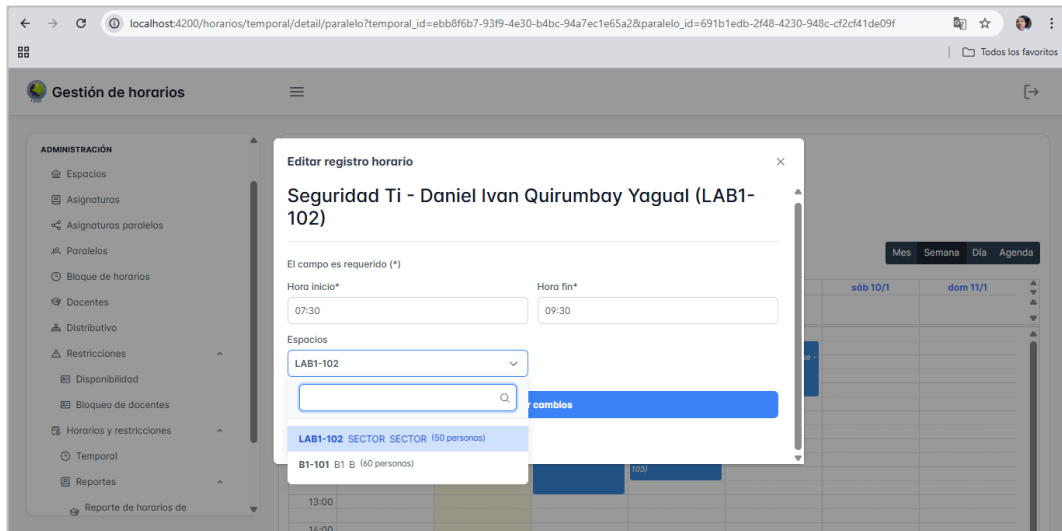


Figura 41. Pantalla de modificación de una clase del horario.

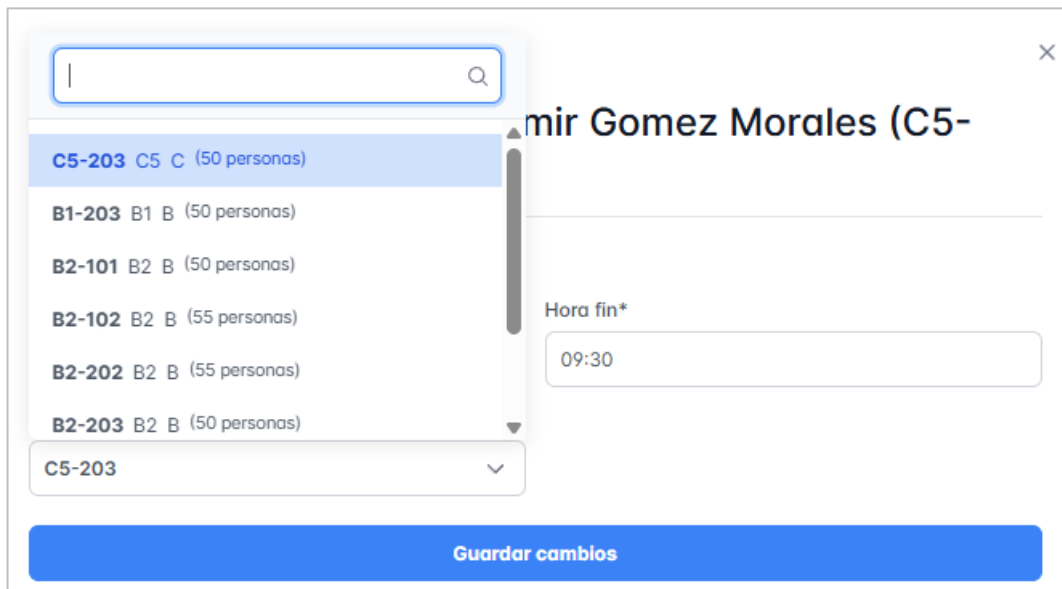


Figura 42. Pantalla de lista de espacios disponibles de una clase del horario temporal.

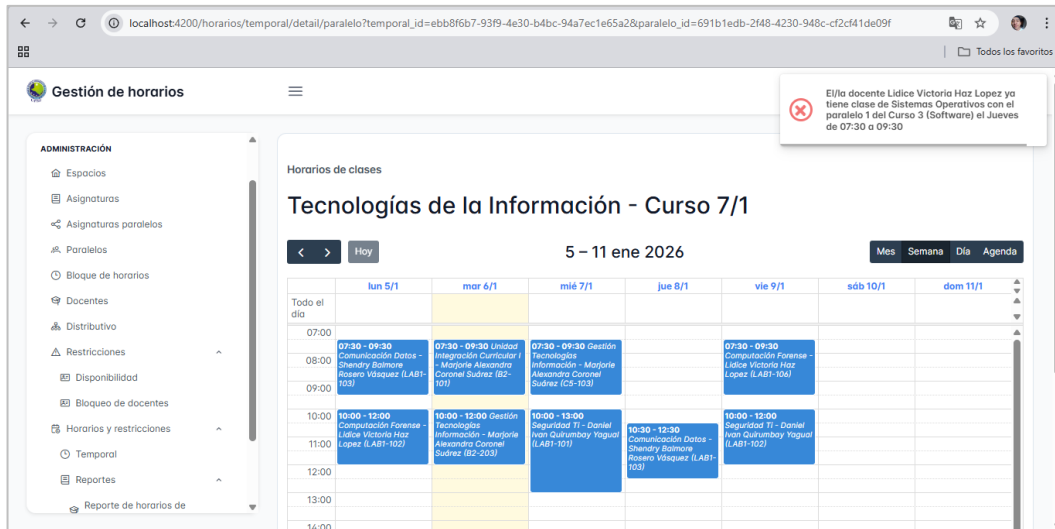


Figura 43. Pantalla con mensaje de alerta sobre una restricción en el horario temporal.

### Reporte general de docentes por espacios físicos

En esta sección, el usuario (decano o director autenticado) podrá generar los reportes generales de docentes por espacios físicos en un formato legible (.PDF), para las respectivas verificaciones.

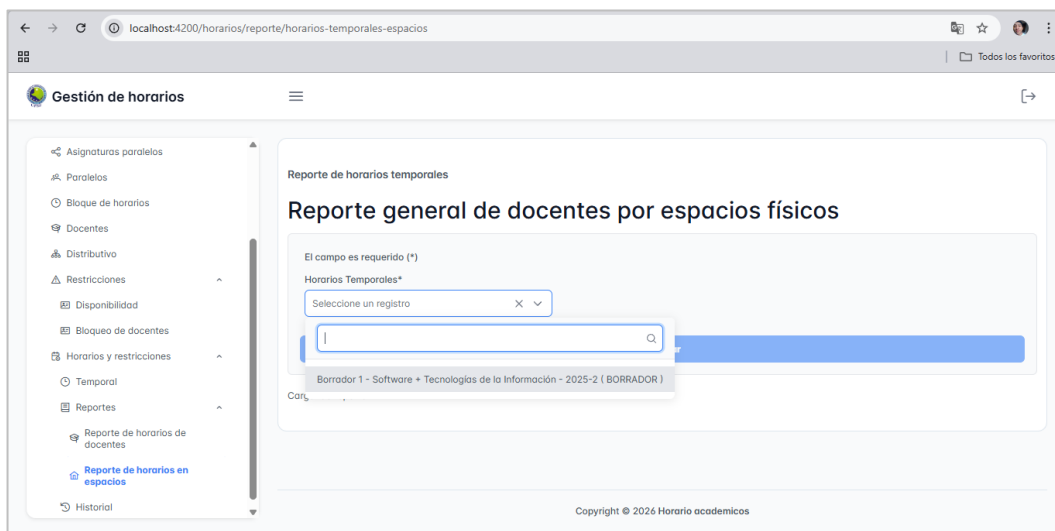


Figura 44. Pantalla de búsqueda de reporte de docentes por espacios físicos.

Una vez, ya generado los horarios temporales, indiferentemente la carrera o facultad que hayan seleccionado, se emitirá un reporte en formato .PDF, en el cual consta

toda la información detallada de un horario de clase (docente, asignatura, espacio, hora inicio y fin, estudiantes, etc.).

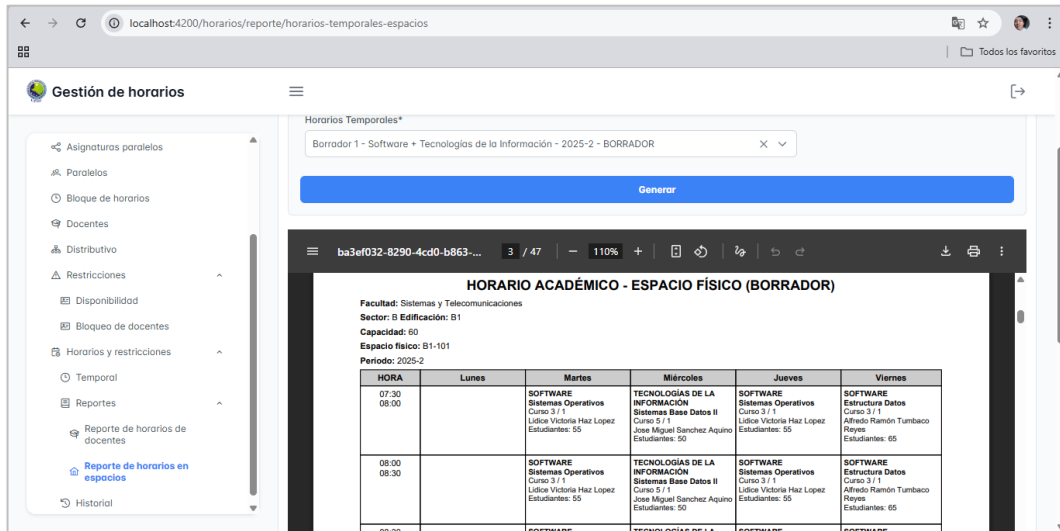


Figura 45. Pantalla del reporte de horario de clases por espacios físicos generado por el sistema.

## Reporte general de horarios por docentes

En esta sección, el usuario (decano o director autenticado) podrá generar los reportes generales de los horarios por docentes en un formato legible (.PDF), para las respectivas verificaciones.

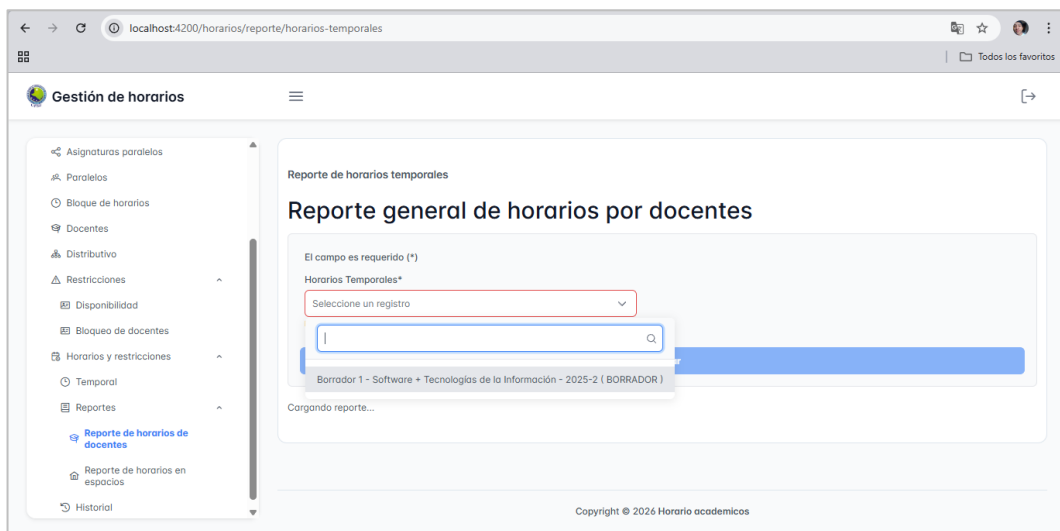


Figura 46. Pantalla de búsqueda de reporte general de horarios por docentes.

Una vez, ya generado los horarios temporales, indiferentemente la carrera o facultad que hayan seleccionado, se emitirá un reporte en formato .PDF, en el cual consta toda la información detallada de un horario de clase del docente (carrera, asignatura, espacio, hora inicio y fin, etc.).

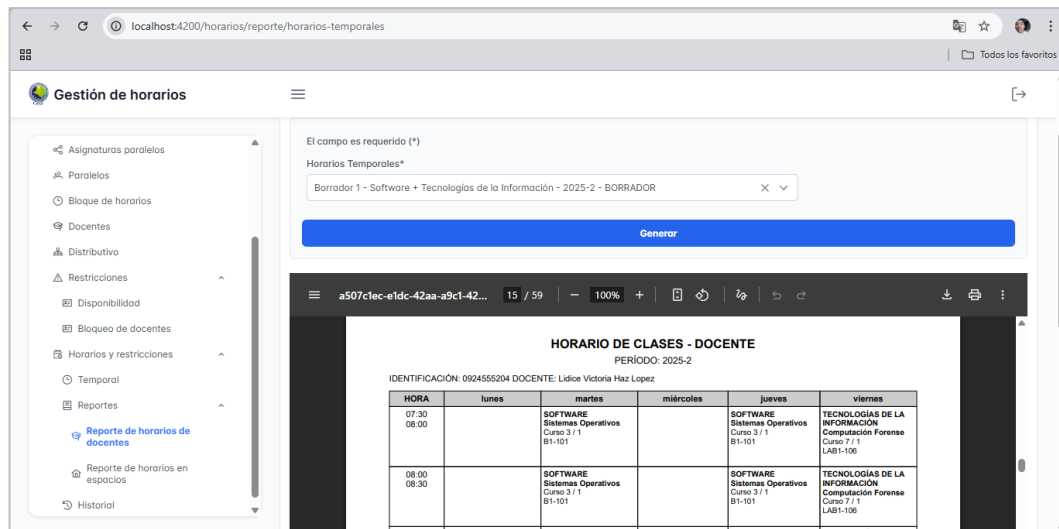


Figura 47. Pantalla del reporte de horario de clases por docente generado por el sistema.

## 2.6. Pruebas

Prueba N°1: Generación de horario	
<b>Propósito</b>	El sistema debe ejecutar el Algoritmo Genético (AG) para generar un horario que satisfaga las restricciones duras y minimice las restricciones blandas.
<b>Roles</b>	Director de carrera o decano de facultad
<b>Proceso</b>	Exitoso y Optimo
<b>Procedimiento</b>	
<ul style="list-style-type: none"> <li>• El director o decano previamente debe configurar las restricciones tanto duras como blandas.</li> <li>• El usuario debe tener acceso a la información de todas las asignaturas, docentes, espacios y recursos necesarios para la generación en ese periodo académico</li> </ul>	

<ul style="list-style-type: none"> <li>• El sistema muestra una barra de progreso mientras se hace la ejecución del algoritmo para buscar soluciones.</li> <li>• El sistema presenta la propuesta del horario en una interfaz gráfica amigable.</li> </ul>
<b>Resultados requeridos</b>
<ul style="list-style-type: none"> <li>• El sistema genera una solución al problema de horarios académicos en un tiempo razonable.</li> <li>• El resultado garantiza que haya cero violaciones a las distintas restricciones.</li> <li>• El valor fitness de la solución debe ser el más bajo posible, lo cual indica una alta optimización.</li> </ul>

Tabla 11. Prueba de Generación de horario.

<b>Prueba N°2: Almacenamiento y comparación de horarios</b>	
<b>Propósito</b>	El sistema debe almacenar varias propuestas de horarios generados por el Algoritmo Genético. Esto permite al director comparar las respuestas antes de seleccionar la versión que este más acorde a las necesidades de cada carrera.
<b>Roles</b>	Director de carrera de decano de facultad
<b>Proceso</b>	Exitoso y comparativo
<b>Procedimiento</b>	
<ul style="list-style-type: none"> <li>• El director o decano ejecuta la generación de horarios las veces que crea necesario.</li> <li>• El usuario va a la sección de horarios temporales en el sistema y navega entre las distintas soluciones de horarios académicos que puede tener cada carrera con sus diferentes cursos, paralelos docentes y horas laborables consecutivas.</li> <li>• La interfaz gráfica del sistema permite visualizar de manera clara y concisa los horarios por paralelos.</li> </ul>	

<ul style="list-style-type: none"> <li>• En caso de varias carreras, el sistema debe validar tanto la disponibilidad en espacios como los docentes, permitiendo así, una comparativa real.</li> </ul>
<b>Resultados requeridos</b>
<ul style="list-style-type: none"> <li>• El sistema permite almacenar múltiples soluciones generadas por el Algoritmo Genético.</li> <li>• El sistema muestra las diferentes soluciones en el calendario, mediante la navegación por cursos y paralelos.</li> </ul>

Tabla 12. Prueba de Almacenamiento y comparación de horarios.

<b>Prueba N°3: Modificación manual y validación en tiempo real</b>	
<b>Propósito</b>	Validar que el director de carrera pueda hacer ajustes manuales en el horario generado temporalmente, y que el sistema detecte o alerte automáticamente si existen conflictos con los cambios propuestos.
<b>Roles</b>	Director de carrera o decano de facultad
<b>Proceso</b>	Alerta de conflicto
<b>Procedimiento</b>	
<ul style="list-style-type: none"> <li>• El director de carrera debe ir al apartado de horarios temporales e ir un horario de clases generado de un paralelo en particular.</li> <li>• El director debe arrastrar el evento (clase) a un día y hora distinto al que estaba originalmente.</li> <li>• Si existe algún conflicto el sistema alertara de inmediato que no es posible una modificación.</li> </ul>	
<b>Resultados requeridos</b>	
<ul style="list-style-type: none"> <li>• El sistema no permite una nueva reasignación.</li> <li>• El sistema emite una alerta inmediata con los detalles del problema.</li> <li>• El sistema permite guardar los cambios en caso de que no existan conflicto alguno con las demás reasignaciones.</li> </ul>	

Tabla 13. Prueba de Modificación manual y validación en tiempo real.

<b>Prueba N°4: Generación y exportación de reportes</b>	
<b>Propósito</b>	Generar reportes relevantes para la toma de decisiones y exportar el horario final a un formato imprimible.
<b>Roles</b>	Director de carrera
<b>Proceso</b>	Exitoso y exportable
<b>Procedimiento</b>	
El director de carrera navega en la sección de reportes, establece los parámetros de búsqueda o exporta lo que crea necesario para su trabajo.	
<b>Resultados requeridos</b>	
<ul style="list-style-type: none"> <li>• El sistema genera y descarga los reportes en formato PDF, de manera que se vea de manera legible los resultados ya sea del uso de aulas, como la de los mismos horarios de clases generados por el algoritmo.</li> <li>• Los datos de los informes coinciden con la información almacenada en el sistema.</li> </ul>	

Tabla 14. Prueba de Generación y exportación de reportes.

<b>Prueba N°5: Configuración de entidades horarias</b>	
<b>Propósito</b>	Registrar y configurar las entidades en el sistema.
<b>Roles</b>	Director de carrera
<b>Proceso</b>	Exitoso y correcto
<b>Procedimiento</b>	
<ul style="list-style-type: none"> <li>• El director de carrera debe ir al apartado de administración en el sistema, donde podrá gestionar las diferentes entidades utilizadas por el algoritmo (aulas, asignaturas, docentes, bloque de horarios, etc.).</li> <li>• El sistema valida la información que se requiera registrar o modificar antes de enviar la petición al servicio web.</li> </ul>	
<b>Resultados requeridos</b>	
El sistema almacena la información en la base de datos, lo cual permite al director tener actualizados los registros de las entidades empleadas.	

Tabla 15. Prueba de Configuración de entidades horarias.

<b>Prueba N°6: Configuración del distributivo</b>	
<b>Propósito</b>	Validar el registro de asignaciones de docentes, cursos y asignaturas en el distributivo, asegurando la integridad de los datos de entrada para el AG.
<b>Roles</b>	Director de carrera
<b>Proceso</b>	Exitoso y completo
<b>Procedimiento</b>	
<ul style="list-style-type: none"> <li>• El director de carrera debe ir al apartado de administración - distributivo, donde se podrá visualizar el listado de asignaciones de los docentes y asignaturas con sus distintos cursos, además de otros atributos generales.</li> <li>• El sistema valida la información que se requiera registrar o modificar antes de enviar la petición al servicio web.</li> </ul>	
<b>Resultados requeridos</b>	
El sistema almacena la información en la base de datos, lo cual permite al director tener actualizados los registros del mismo distributivo.	

Tabla 16. Prueba de Configuración del distributivo.

<b>Prueba N°7: Configuración de restricciones (docentes)</b>	
<b>Propósito</b>	Validar el registro y modificación de restricciones (docentes) a utilizarse para generar el horario académico en el AG.
<b>Roles</b>	Director de carrera
<b>Proceso</b>	Exitoso y completo
<b>Procedimiento</b>	
<ul style="list-style-type: none"> <li>• El director de carrera debe ir al apartado de horarios y restricciones, donde se podrá visualizar el listado de registro de restricciones (docentes).</li> <li>• El director debe ingresar la información del docente, motivos, y horas específicas.</li> <li>• El sistema valida la información que se requiera registrar o modificar antes de enviar la petición al servicio web.</li> </ul>	

<b>Resultados requeridos</b>
El sistema almacena la información en la base de datos, lo cual permite al AG tener actualizadas las restricciones (docentes) antes de generar el horario académico.

Tabla 17. Prueba de Configuración de restricciones (docentes).

## **CONCLUSIONES**

- El uso de los Algoritmos Genéticos (AG) permitió generar soluciones bastante optimas en tiempos de respuestas muy reducidos, inclusive si la cantidad de la población inicial es muy alta. En este escenario real aplicando estas metodologías metaheurísticas son prueba y afirmación que las mismas pueden resolver el problema de la planificación académica de horarios con muchas restricciones.
- La solución permite optimizar el trabajo semiautomatizado o manual asistido de los directores de carrera de una manera notable.
- El desarrollo de la propuesta en Spring Boot, facilito muchas de las gestiones que se hacen en la lógica de negocio, siendo esta parte fundamental de una arquitectura escalable para futuras mejoras y optimizaciones, sin comprometer la estabilidad del sistema.
- El sistema ofrece una solución integral, manteniendo un historial con múltiples versiones que nos puedan ayudar a analizar las distintas generaciones de horarios y ver cuál es la más viable.

## **RECOMENDACIONES**

- Implementar pruebas automatizadas para fortalecer la calidad del sistema, ya sean pruebas unitarias, integración o de rendimiento, lo cual garantizara futuras mejoras y evitar errores en producción.
- La auditoría es indispensable hoy en día para analizar posibles conflictos o problemas recurrentes a lo largo de la vida del software, por lo cual, a futuro sería útil contar con sistemas que permitan registrar estos imprevistos, y así tratar de mitigarlos y llegar una solución más sostenible.

- Si bien, la metodología adaptada a este proyecto se basó en los AG, sería más útil poder mejorar el mismo en la lógica de negocio del sistema, ya sea detectando automáticamente el tamaño de la población, mejoras en los procesos de selección, para así acelerar a soluciones mucho más óptimas sin requerir tantos ajustes.

## BIBLIOGRAFÍA

- [1] J. Ahumada, «GENERACIÓN DE HORARIOS ACADÉMICOS EN INACAP UTILIZANDO ALGORITMOS GENÉTICOS,» 2014.
- [2] J. Ahumada, «Repositorio Universidad de Chile,» 2014. [En línea]. Available:  
[https://repositorio.uchile.cl/bitstream/handle/2250/131197/Generaci%  
b3n-de-horarios-acad%  
gen%  
a9ticos.pdf?sequence=1&isAllowed=y](https://repositorio.uchile.cl/bitstream/handle/2250/131197/Generaci%c3%b3n-de-horarios-acad%c3%a9micos-en-INACAP-utilizando-algoritmos-gen%c3%a9ticos.pdf?sequence=1&isAllowed=y). [Último acceso: 06 09 2025].
- [3] J. Gómez, «Repositorio ESPOCH,» 2016. [En línea]. Available:  
[https://dspace.esPOCH.edu.ec:8080/server/api/core/bitstreams/074cd4fe-  
de35-41f7-8541-f86f8e050e80/content](https://dspace.esPOCH.edu.ec:8080/server/api/core/bitstreams/074cd4fe-de35-41f7-8541-f86f8e050e80/content). [Último acceso: 09 09 2025].
- [4] A. Barzola, «Implementación de un Sistema de Gestión de Horarios de Clases,» La libertad, 2016.
- [5] «Secretaría Nacional de Planificación,» 2021. [En línea]. Available:  
[https://www.planificacion.gob.ec/wp-content/uploads/2021/09/Plan-de-  
Creacio%  
81n-de-Oportunidades-2021-2025-Aprobado.pdf](https://www.planificacion.gob.ec/wp-content/uploads/2021/09/Plan-de-Creacio%CC%81n-de-Oportunidades-2021-2025-Aprobado.pdf).
- [6] M. Rodríguez, Metodología de la Investigación, México: Universidad Juárez Autónoma de Tabasco, 2005.
- [7] J. Barranco de Areba, Metodología del análisis estructurado de sistemas, España: Universidad Pontificia Comillas (Publicaciones), 2001.
- [8] «Google Maps,» Google, 09 09 2025. [En línea]. Available:  
<https://maps.app.goo.gl/5SpX6eWWBEyRGokQ6>. [Último acceso: 09 09 2025].

- [9] «Universidad Europea,» 02 09 2024. [En línea]. Available: <https://universidadeuropea.com/blog/que-es-algoritmo/>. [Último acceso: 01 11 2025].
- [10] M. d. P. Alegre, Sistemas Informaticos, Paraninfo, SA, 2023.
- [11] K. C. Laudon, «economicas.unsa.edu.ar,» 2012. [En línea]. Available: [https://www.economicas.unsa.edu.ar/sigeco/archivos/gti\\_material/Capitulo%20BD%20Laudon%20y%20Laudon%202013.pdf](https://www.economicas.unsa.edu.ar/sigeco/archivos/gti_material/Capitulo%20BD%20Laudon%20y%20Laudon%202013.pdf).
- [12] Amazon, «amazon.com,» [En línea]. Available: <https://aws.amazon.com/es/what-is/web-application/>.
- [13] U. Revista, «uni.net,» [En línea]. Available: <https://www.unir.net/revista/ingenieria/servidor-web-local/>.
- [14] itsqmet, «itsqmet.edu.ec,» [En línea]. Available: <https://itsqmet.edu.ec/framework/>.
- [15] Telefonica, «Telefonica,» [En línea]. Available: <https://www.telefonica.com/es/sala-comunicacion/blog/historia-evolucion-navegadores-web/>. [Último acceso: 12 08 2025].
- [16] . Capacho y . Nieto Bernal, Diseño de bases de datos, Colombia: Universidad del Norte, 2017.
- [17] «universidadeuropea,» 22 08 2024. [En línea]. Available: <https://universidadeuropea.com/blog/entorno-desarrollo/>. [Último acceso: 11 11 2025].
- [18] «java.com,» 2025. [En línea]. Available: [https://www.java.com/es/download/help/whatis\\_java.html](https://www.java.com/es/download/help/whatis_java.html). [Último acceso: 11 11 2025].

- [19] «Universidad Europea,» 26 02 2025. [En línea]. Available: <https://universidadeuropea.com/blog/programacion-java/>. [Último acceso: 12 12 2025].
- [20] J. Ceballos Sierra, Programación Orientada a Objetos con C++. 4ta edición, Grupo Editorial RA-MA, 2007.
- [21] S. Viñas, N. Rodríguez, E. Corona y A. Jiménez, «Software para la generación automática de horarios académicos,» Instituto Tecnológico Superior de Ciudad Hidalgo, Ciudad Hidalgo, 2018.
- [22] . O. Robles Montero, . Ortiz Quiroga y . Gama Campillo, «Sistema Integral de Asignación de Carga Académica con la Utilización de Algoritmos Evolutivos para Resolución del Problema de Horarios en la UPVM. Alemania,» GRIN Verlag, México, 2016.
- [23] L. D. Chambers, Practical Handbook of Genetic Algorithms: Complex Coding Systems, Volume III, Reino Unido: CRC Press, 2019.
- [24] A. O. I. I. A. T. H. K. A. Abeer Bashab, M. Fadhil, F. A. Ghaleb y A. Abdelzahir, «Optimization Techniques in University Timetabling Problem: Constraints, Methodologies, Benchmarks, and Open Issues,» Tech Science Press, 2022.
- [25] F. H. Awad, A. Al-kubaisi y M. Mahmood, «Large-scale timetabling problems with adaptive tabu search,» Freund Publishing House, Reino Unido, 2022.
- [26] . Fernández y . Sallán, Programación lineal, Catalunya: Universitat Politècnica de Catalunya, 2016.
- [27] K. S. E. S., S. S. N., G. S. L., K. G., S. N. R. y A. S., A Survey of Solution Methodologies for Exam Timetabling Problems, Malaysia: IEEE, 2024.

- [28] E. Quispe y G. Julio, Desequilibrio de tensiones en motores de inducción: Modelado, impacto en el desempeño energético, determinación de la eficiencia., Colombia: PROGRAMA EDITORIAL UNIVER, 2020.
- [29] L. Leija, Métodos de procesamiento avanzado e inteligencia artificial en sistemas sensores y biosensores., Alemania: Reverté, 2021.
- [30] «Inesdi,» 20 11 2015. [En línea]. Available: <https://www.inesdi.com/algoritmos-geneticos-cp/>. [Último acceso: 15 12 2025].
- [31] «sc.ehu.es,» 2024. [En línea]. Available: <http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/temageneticos.pdf>. [Último acceso: 11 12 2025].
- [32] . García Rubio y . Piattini Velthuis, Desarrollo Global de Software, España: RA-MA Editorial, 2014.
- [33] . Kendall, . Kendall y . Núñez Ramos, Análisis y diseño de sistemas, México: Pearson Educación, 2005.
- [34] . Fernández Alarcón, Desarrollo de Sistemas de Información una Metodología Basada en el Modelado, España: UPC, S.L., Edicions, 2010.
- [35] J. Piñeiro, Diseño de bases de datos relacionales, España: Ediciones Paraninfo, S.A., 2024.

# ANEXOS

## Anexo 1. Vista del horario académico por espacios físicos (Laboratorios).

### HORARIO ACADÉMICO - ESPACIO FÍSICO (BORRADOR)

Facultad: Sistemas y Telecomunicaciones

Sector: SECTOR Edificación: SECTOR

Capacidad: 50

Espacio físico: LAB1-101

Período: 2025-2

HORA	Lunes	Martes	Miércoles	Jueves	Viernes
07:30 08:00	<b>TECNOLOGÍAS DE LA INFORMACIÓN</b> <b>Comunicación Datos</b> Curso 7 / 1 Shendry Balmore Rosero Vásquez Estudiantes: 50	<b>TECNOLOGÍAS DE LA INFORMACIÓN</b> <b>Administración</b> <b>Sistemas Operativos</b> Curso 5 / 1 Ivan Alberto Coronel Suárez Estudiantes: 50		<b>TECNOLOGÍAS DE LA INFORMACIÓN</b> <b>Computación Forense</b> Curso 7 / 1 Lidice Victoria Haz Lopez Estudiantes: 50	
08:00 08:30	<b>TECNOLOGÍAS DE LA INFORMACIÓN</b> <b>Comunicación Datos</b> Curso 7 / 1 Shendry Balmore Rosero Vásquez Estudiantes: 50	<b>TECNOLOGÍAS DE LA INFORMACIÓN</b> <b>Administración</b> <b>Sistemas Operativos</b> Curso 5 / 1 Ivan Alberto Coronel Suárez Estudiantes: 50		<b>TECNOLOGÍAS DE LA INFORMACIÓN</b> <b>Computación Forense</b> Curso 7 / 1 Lidice Victoria Haz Lopez Estudiantes: 50	
08:30 09:00	<b>TECNOLOGÍAS DE LA INFORMACIÓN</b> <b>Comunicación Datos</b> Curso 7 / 1 Shendry Balmore Rosero Vásquez Estudiantes: 50	<b>TECNOLOGÍAS DE LA INFORMACIÓN</b> <b>Administración</b> <b>Sistemas Operativos</b> Curso 5 / 1 Ivan Alberto Coronel Suárez Estudiantes: 50		<b>TECNOLOGÍAS DE LA INFORMACIÓN</b> <b>Computación Forense</b> Curso 7 / 1 Lidice Victoria Haz Lopez Estudiantes: 50	
09:00 09:30	<b>TECNOLOGÍAS DE LA INFORMACIÓN</b> <b>Comunicación Datos</b> Curso 7 / 1 Shendry Balmore Rosero Vásquez Estudiantes: 50	<b>TECNOLOGÍAS DE LA INFORMACIÓN</b> <b>Administración</b> <b>Sistemas Operativos</b> Curso 5 / 1 Ivan Alberto Coronel Suárez Estudiantes: 50		<b>TECNOLOGÍAS DE LA INFORMACIÓN</b> <b>Computación Forense</b> Curso 7 / 1 Lidice Victoria Haz Lopez Estudiantes: 50	
09:30 10:00					
10:00 10:30				<b>SOFTWARE</b> <b>Algoritmo</b> Curso 1 / 1 Alfredo Ramón Tumbaco Reyes Estudiantes: 50	
10:30 11:00	<b>SOFTWARE</b> <b>Programación Funcional</b> Curso 5 / 1 Luis Vinicio Torres Vivanco Estudiantes: 40			<b>SOFTWARE</b> <b>Algoritmo</b> Curso 1 / 1 Alfredo Ramón Tumbaco Reyes Estudiantes: 50	
11:00 11:30	<b>SOFTWARE</b> <b>Programación Funcional</b> Curso 5 / 1 Luis Vinicio Torres Vivanco Estudiantes: 40		<b>TELECOMUNICACIONES</b> <b>Líneas Transmisión</b> Curso 5 / 1 Carlos Efraim Andrade Caiche Estudiantes: 25	<b>SOFTWARE</b> <b>Algoritmo</b> Curso 1 / 1 Alfredo Ramón Tumbaco Reyes Estudiantes: 50	
11:30 12:00	<b>SOFTWARE</b> <b>Programación Funcional</b> Curso 5 / 1 Luis Vinicio Torres Vivanco Estudiantes: 40		<b>TELECOMUNICACIONES</b> <b>Líneas Transmisión</b> Curso 5 / 1 Carlos Efraim Andrade Caiche Estudiantes: 25	<b>SOFTWARE</b> <b>Algoritmo</b> Curso 1 / 1 Alfredo Ramón Tumbaco Reyes Estudiantes: 50	
12:00 12:30	<b>SOFTWARE</b> <b>Programación Funcional</b> Curso 5 / 1 Luis Vinicio Torres Vivanco Estudiantes: 40		<b>TELECOMUNICACIONES</b> <b>Líneas Transmisión</b> Curso 5 / 1 Carlos Efraim Andrade Caiche Estudiantes: 25		
12:30 13:00			<b>TELECOMUNICACIONES</b> <b>Líneas Transmisión</b> Curso 5 / 1 Carlos Efraim Andrade Caiche		

## Anexo 2. Vista del horario académico por espacios físicos (Aulas).

### HORARIO ACADÉMICO - ESPACIO FÍSICO (BORRADOR)

Facultad: Sistemas y Telecomunicaciones

Sector: B Edificación: B1

Capacidad: 60

Espacio físico: B1-101

Período: 2025-2

HORA	Lunes	Martes	Miércoles	Jueves	Viernes
07:30 08:00		<b>SOFTWARE Estructura Datos</b> Curso 3 / 1 Alfredo Ramón Tumbaco Reyes Estudiantes: 65		<b>SOFTWARE Fundamentos Redes</b> Curso 3 / 1 Monica Karina Jaramillo Infante Estudiantes: 70	<b>TECNOLOGÍAS DE LA INFORMACIÓN Seguridad TI</b> Curso 7 / 1 Daniel Ivan Quirumbay Yagual Estudiantes: 50
08:00 08:30		<b>SOFTWARE Estructura Datos</b> Curso 3 / 1 Alfredo Ramón Tumbaco Reyes Estudiantes: 65		<b>SOFTWARE Fundamentos Redes</b> Curso 3 / 1 Monica Karina Jaramillo Infante Estudiantes: 70	<b>TECNOLOGÍAS DE LA INFORMACIÓN Seguridad TI</b> Curso 7 / 1 Daniel Ivan Quirumbay Yagual Estudiantes: 50
08:30 09:00		<b>SOFTWARE Estructura Datos</b> Curso 3 / 1 Alfredo Ramón Tumbaco Reyes Estudiantes: 65		<b>SOFTWARE Fundamentos Redes</b> Curso 3 / 1 Monica Karina Jaramillo Infante Estudiantes: 70	<b>TECNOLOGÍAS DE LA INFORMACIÓN Seguridad TI</b> Curso 7 / 1 Daniel Ivan Quirumbay Yagual Estudiantes: 50
09:00 09:30		<b>SOFTWARE Estructura Datos</b> Curso 3 / 1 Alfredo Ramón Tumbaco Reyes Estudiantes: 65		<b>SOFTWARE Fundamentos Redes</b> Curso 3 / 1 Monica Karina Jaramillo Infante Estudiantes: 70	<b>TECNOLOGÍAS DE LA INFORMACIÓN Seguridad TI</b> Curso 7 / 1 Daniel Ivan Quirumbay Yagual Estudiantes: 50
09:30 10:00					
10:00 10:30		<b>TECNOLOGÍAS DE LA INFORMACIÓN Programación Avanzada I</b> Curso 3 / 1 Luis Vinicio Torres Vivanco Estudiantes: 40	<b>TECNOLOGÍAS DE LA INFORMACIÓN Tecnologías Desarrollo Web</b> Curso 5 / 1 Carlos Andres Castillo Yagual Estudiantes: 50	<b>SOFTWARE Fundamentos Redes</b> Curso 3 / 1 Monica Karina Jaramillo Infante Estudiantes: 70	
10:30 11:00	<b>TECNOLOGÍAS DE LA INFORMACIÓN Sistemas Base Datos II</b> Curso 5 / 1 Jose Miguel Sanchez Aquino Estudiantes: 50	<b>TECNOLOGÍAS DE LA INFORMACIÓN Programación Avanzada I</b> Curso 3 / 1 Luis Vinicio Torres Vivanco Estudiantes: 40	<b>TECNOLOGÍAS DE LA INFORMACIÓN Tecnologías Desarrollo Web</b> Curso 5 / 1 Carlos Andres Castillo Yagual Estudiantes: 50	<b>SOFTWARE Fundamentos Redes</b> Curso 3 / 1 Monica Karina Jaramillo Infante Estudiantes: 70	<b>SOFTWARE Estructura Datos</b> Curso 3 / 1 Alfredo Ramón Tumbaco Reyes Estudiantes: 65
11:00 11:30	<b>TECNOLOGÍAS DE LA INFORMACIÓN Sistemas Base Datos II</b> Curso 5 / 1 Jose Miguel Sanchez Aquino Estudiantes: 50	<b>TECNOLOGÍAS DE LA INFORMACIÓN Programación Avanzada I</b> Curso 3 / 1 Luis Vinicio Torres Vivanco Estudiantes: 40	<b>TECNOLOGÍAS DE LA INFORMACIÓN Tecnologías Desarrollo Web</b> Curso 5 / 1 Carlos Andres Castillo Yagual Estudiantes: 50	<b>SOFTWARE Fundamentos Redes</b> Curso 3 / 1 Monica Karina Jaramillo Infante Estudiantes: 70	<b>SOFTWARE Estructura Datos</b> Curso 3 / 1 Alfredo Ramón Tumbaco Reyes Estudiantes: 65
11:30 12:00	<b>TECNOLOGÍAS DE LA INFORMACIÓN Sistemas Base Datos II</b> Curso 5 / 1 Jose Miguel Sanchez Aquino Estudiantes: 50	<b>TECNOLOGÍAS DE LA INFORMACIÓN Programación Avanzada I</b> Curso 3 / 1 Luis Vinicio Torres Vivanco Estudiantes: 40	<b>TECNOLOGÍAS DE LA INFORMACIÓN Tecnologías Desarrollo Web</b> Curso 5 / 1 Carlos Andres Castillo Yagual Estudiantes: 50	<b>SOFTWARE Fundamentos Redes</b> Curso 3 / 1 Monica Karina Jaramillo Infante Estudiantes: 70	<b>SOFTWARE Estructura Datos</b> Curso 3 / 1 Alfredo Ramón Tumbaco Reyes Estudiantes: 65
12:00 12:30	<b>TECNOLOGÍAS DE LA INFORMACIÓN Sistemas Base Datos II</b> Curso 5 / 1 Jose Miguel Sanchez Aquino Estudiantes: 50	<b>TECNOLOGÍAS DE LA INFORMACIÓN Programación Avanzada I</b> Curso 3 / 1 Luis Vinicio Torres Vivanco Estudiantes: 40		<b>SOFTWARE Fundamentos Redes</b> Curso 3 / 1 Monica Karina Jaramillo Infante Estudiantes: 70	<b>SOFTWARE Estructura Datos</b> Curso 3 / 1 Alfredo Ramón Tumbaco Reyes Estudiantes: 65

### Anexo 3. Vista del horario académico de docentes.

#### HORARIO DE CLASES - DOCENTE

PERÍODO: 2025-2

IDENTIFICACIÓN: 0927583344 DOCENTE: William Andres Rodríguez Lopez

HORA	lunes	martes	miércoles	jueves	viernes
07:30 08:00	TELECOMUNICACIONES Software Simulación Curso 3 / 1 VIRTUAL		TELECOMUNICACIONES Sistemas Digitales Curso 5 / 1 C5-101		TELECOMUNICACIONES Sistemas Digitales Curso 5 / 1 C5-205
08:00 08:30	TELECOMUNICACIONES Software Simulación Curso 3 / 1 VIRTUAL		TELECOMUNICACIONES Sistemas Digitales Curso 5 / 1 C5-101		TELECOMUNICACIONES Sistemas Digitales Curso 5 / 1 C5-205
08:30 09:00	TELECOMUNICACIONES Software Simulación Curso 3 / 1 VIRTUAL		TELECOMUNICACIONES Sistemas Digitales Curso 5 / 1 C5-101		TELECOMUNICACIONES Sistemas Digitales Curso 5 / 1 C5-205
09:00 09:30	TELECOMUNICACIONES Software Simulación Curso 3 / 1 VIRTUAL		TELECOMUNICACIONES Sistemas Digitales Curso 5 / 1 C5-101		TELECOMUNICACIONES Sistemas Digitales Curso 5 / 1 C5-205
09:30 10:00					
10:00 10:30		ELETRÓNICA Y AUTOMATIZACIÓN Software Simulación Curso 3 / 1 C5-205			
10:30 11:00		ELETRÓNICA Y AUTOMATIZACIÓN Software Simulación Curso 3 / 1 C5-205			
11:00 11:30		ELETRÓNICA Y AUTOMATIZACIÓN Software Simulación Curso 3 / 1 C5-205			
11:30 12:00		ELETRÓNICA Y AUTOMATIZACIÓN Software Simulación Curso 3 / 1 C5-205			
12:00 12:30		ELETRÓNICA Y AUTOMATIZACIÓN Software Simulación Curso 3 / 1 C5-205			
12:30 13:00		ELETRÓNICA Y AUTOMATIZACIÓN Software Simulación Curso 3 / 1 C5-205			
13:00 13:30					
13:30 14:00					
14:00 14:30		TELECOMUNICACIONES Programación Avanzada Curso 2 / 1 VIRTUAL			
14:30 15:00		TELECOMUNICACIONES Programación Avanzada			

#### **Anexo 4. Encuesta de evaluación.**

**TEMA: OPTIMIZACIÓN DE GESTIÓN DE HORARIOS ACADÉMICOS PARA LA UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA (UPSE).**

**Objetivo:** Conocer su percepción sobre el impacto que tiene este módulo en su trabajo de planificación de horarios, en comparación con la forma tradicional en que usted lo realiza.

#### **ENCUESTA DE EVALUACIÓN**

- 1. ¿El proceso de elaboración de horarios me resultaba tedioso y demandaba mucho tiempo?**
  - Definitivamente si
  - Probablemente si
  - Indeciso
  - Probablemente no
  - Definitivamente no
  
- 2. ¿Con el módulo web, la detección de conflictos de horario es más rápida y clara?**
  - Totalmente de acuerdo
  - De acuerdo
  - Ni de acuerdo
  - En desacuerdo
  - Muy en desacuerdo
  
- 3. ¿Con el módulo web, me siento más confiado de que el horario final tiene menos errores?**
  - Totalmente de acuerdo
  - De acuerdo
  - Ni de acuerdo
  - En desacuerdo

Muy en desacuerdo

**4. ¿El módulo facilita realizar cambios y regenerar el horario sin rehacer todo desde cero?**

Totalmente de acuerdo

De acuerdo

Ni de acuerdo

En desacuerdo

Muy en desacuerdo

**5. ¿En general, el uso del módulo web ha mejorado mi trabajo de planificación académica?**

Totalmente de acuerdo

De acuerdo

Ni de acuerdo

En desacuerdo

Muy en desacuerdo