

Universidad de San Carlos de Guatemala
División de Ciencias de la Ingeniería
Ingeniería en Ciencias y Sistemas
Laboratorio de Modelación y Simulación 2
Ing. Pedro Domingo



Proyecto #1

Simulador de horarios

Willians Alberto Orozco López
Carné: 201830221
Fecha: 18/09/2023



Índice

Introducción.....	3
Formulación del problema.....	4
Entidades.....	5
Materias (m).....	5
Salones (s).....	5
Estudiantes (e) y Asignaciones (a).....	6
Carreras (c).....	6
Profesores (p).....	6
Documentación.....	8
Diagrama de bases de datos.....	8
Diagramas de flujo.....	9
Diagrama de casos de uso.....	10
Arquitectura del sistema.....	10
Diagramas de secuencia.....	12
Diagrama de secuencia de la carga de datos.....	12
Diagrama de secuencia de la generación de horarios.....	12
Diagrama de vistas.....	13
Vista creator.....	13
Vista loader.....	13
Vista de horario.....	14
Vista inicial.....	15
Cronograma de actividades.....	16
DDL.....	17
Colocación de objetivos y plan del proyecto global.....	19
Objetivo general.....	19
Objetivos específicos.....	19
Plan del proyecto global.....	20
Conceptualización del modelo.....	21
Recolección y procesamiento de datos.....	21
Ingreso de salones.....	21
Ingreso de cursos.....	22
Ingreso de profesores.....	22
Construcción del modelo.....	23



Prioridades.....	23
Por contratación de profesores.....	23
Por capacidad de salones.....	23
Valores de peso.....	23
Valor por asignación.....	23
Valor por carrera.....	24
Valor por profesor.....	24
Valor por salón.....	24
Porcentaje de reducción por semestre desfasado.....	24
Verificación.....	25
Validación.....	25
Diseño de experimentos.....	26
Producción de corridas de simulación y análisis.....	26
Puesta en marcha del modelo.....	27
Conclusiones.....	29



Introducción

La gestión eficiente de horarios es un desafío crítico en cualquier establecimiento educativo, ya que influye directamente en la calidad de la experiencia académica de los estudiantes y en la productividad de los profesores. En este contexto, la creación de un generador de horarios personalizado se ha convertido en una herramienta esencial para optimizar la asignación de cursos, salones y profesores. Este generador de horarios utiliza valores de peso y prioridades específicas para ofrecer horarios que se adapten a las necesidades únicas de la institución educativa, brindando así una solución eficiente y efectiva para la planificación académica. En este documento, exploraremos en detalle cómo este generador de horarios puede revolucionar la gestión académica en una institución estudiantil al proporcionar horarios optimizados que maximizan la eficiencia y la satisfacción de todos los involucrados en el proceso educativo.



Formulación del problema

En el ámbito educativo y académico, la gestión eficiente de horarios es esencial para el funcionamiento fluido y exitoso de cualquier institución educativa. Un simulador de horarios se erige como una herramienta de gran importancia en este contexto, ya que permite planificar y diseñar horarios óptimos para cursos, profesores y alumnos inscritos. Esta herramienta no solo simplifica la compleja tarea de crear horarios que se ajusten a las necesidades y preferencias de todas las partes involucradas, sino que también aporta una serie de ventajas y beneficios que mejoran significativamente la calidad de la educación y la experiencia de todos los actores implicados.

Uno de los aspectos cruciales que aborda un simulador de horarios es la optimización de recursos. Al considerar múltiples variables, como la disponibilidad de profesores, las preferencias de los alumnos y los requisitos de los cursos, el simulador puede generar horarios que minimicen los solapamientos y maximicen la utilización eficiente de las instalaciones y el tiempo disponible. Esto resulta en una distribución más equitativa de la carga de trabajo para los profesores y en una mejor asignación de espacios físicos, contribuyendo así a un entorno de aprendizaje más productivo y organizado.

En el ámbito de la planificación estratégica, el simulador de horarios facilita la toma de decisiones informadas. Las instituciones educativas pueden probar diferentes configuraciones y escenarios antes de implementar un horario definitivo, lo que les permite evaluar las implicaciones y consecuencias de diferentes opciones. Esto puede ser especialmente útil para anticipar posibles conflictos y encontrar soluciones antes de que afecten negativamente el desarrollo de las clases y la eficiencia general de la institución.



Entidades

En este apartado se detallarán en su totalidad las entidades que conformarán el simulador, identificando sus aspectos prioritarios, puesto que el simulador se basará en un proceso de **pesos o prioridades**.

Materias (m)

Este es elemento fundamental para el simulador, puesto que todo gira en torno a las materias, teniendo que responder preguntas al inicio cómo las siguientes:

- ¿Es la materia impartida sólo por una persona?
- ¿Cuántos alumnos inscritos tiene la materia?
- ¿Necesita hacer uso de un salón específico?
- ¿Corresponde la materia evaluada al semestre en curso?

Cada una de las preguntas nos dará un parámetro importante para el horario final, esto mediante la definición de prioridades respecto a la importancia de las materias. Acá surge una pregunta legítima, *¿Cómo calcular adecuadamente la importancia de una materia?* puesto que para ello pueden haber distintos parámetros que variarán dependiendo de quien esté generando el modelo a seguir.

Teniendo en cuenta lo anterior, se definirá en un futuro apartado los pesos que fueron asignados para cada aspecto evaluado mediante las preguntas anteriores.

Salones (s)

Debido a que los salones únicamente tienen el atributo del número de escritorios que este mismo contiene, solo se tomará en cuenta al momento de asignar como tal un curso. *Se*



debe tomar en cuenta que puede existir un curso que únicamente se puede llevar a cabo en un salón específico. Esta entidad no formará parte del proceso de asignación de pesos.

Estudiantes (e) y Asignaciones (a)

Dado que se encuentra definido cuántos estudiantes tiene asignado cada curso, dicho dato se utilizará para calcular el peso del curso, al momento de estar asignándoles una prioridad, y cuándo se le asigne un salón al curso, puesto que de ser posible, deberá de asignársele un salón que cumpla con la capacidad de asignados y si en dado caso suceda lo contrario, mostrar una advertencia. *En este sistema no se tomará en cuenta las personas que se desasignaron un curso,* únicamente se podrá simular aleatoriamente mediante el uso de la estadística presentada en [Deserción: 24.4% de alumnos abandona la carrera en primer año \(universia.net\)](#), que para uso del simulador, se tomará un punto de referencia para el aleatorio de 24%.

Carreras (c)

Como tal *apoyarán a la creación del perfil de un catedrático*, dado que mediante la carrera se definirá a qué áreas puede impartir materias dicho agente de la educación. Además le agrega un grado de complejidad al simulador, puesto que no solo deberá de tomar en cuenta las personas asignadas, sino cuál de las carreras tendrá prioridad sobre las demás.

Para uso de este simulador, se generará un aspecto aleatorio para que cada curso tenga la misma probabilidad de prioridad, aunque se podría basarse en las estadísticas de qué carrera tiene más estudiantes en general.

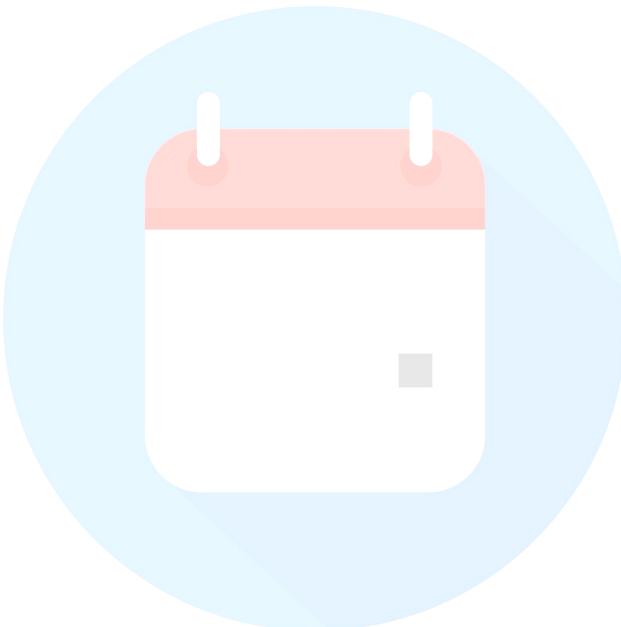
Profesores (p)

Los profesores cuentan con los siguientes aspectos a tomar en cuenta:



- Límite de cursos que puede impartir
- Horario de contratación
- Áreas para las que se encuentra cualificado

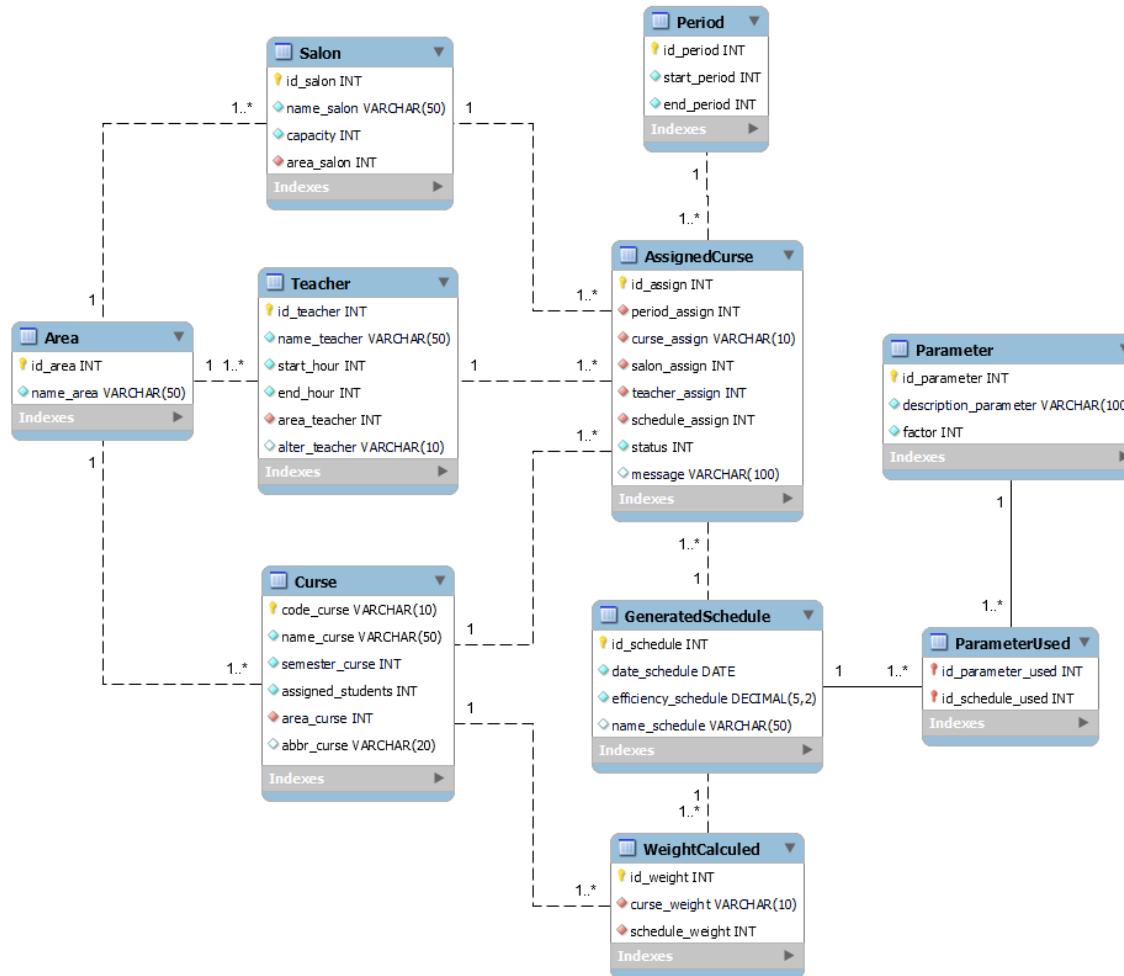
Dichos aspectos serán de uso al momento de **asignar un salón y como tal un catedrático** a la materia, teniendo en cuenta que si no se logra encontrar un profesor que cumpla con los requisitos de la materia, se deberá de **mostrar una advertencia** en el simulador.





Documentación

Diagrama de bases de datos





Diagramas de flujo

El diagrama de la izquierda es sobre la *asignación de pesos (prioridades) de los cursos*, mientras que el de la derecha es sobre la *asignación de cursos a salones y períodos*. Para verlo mejor se puede entrar al siguiente enlace draw.io

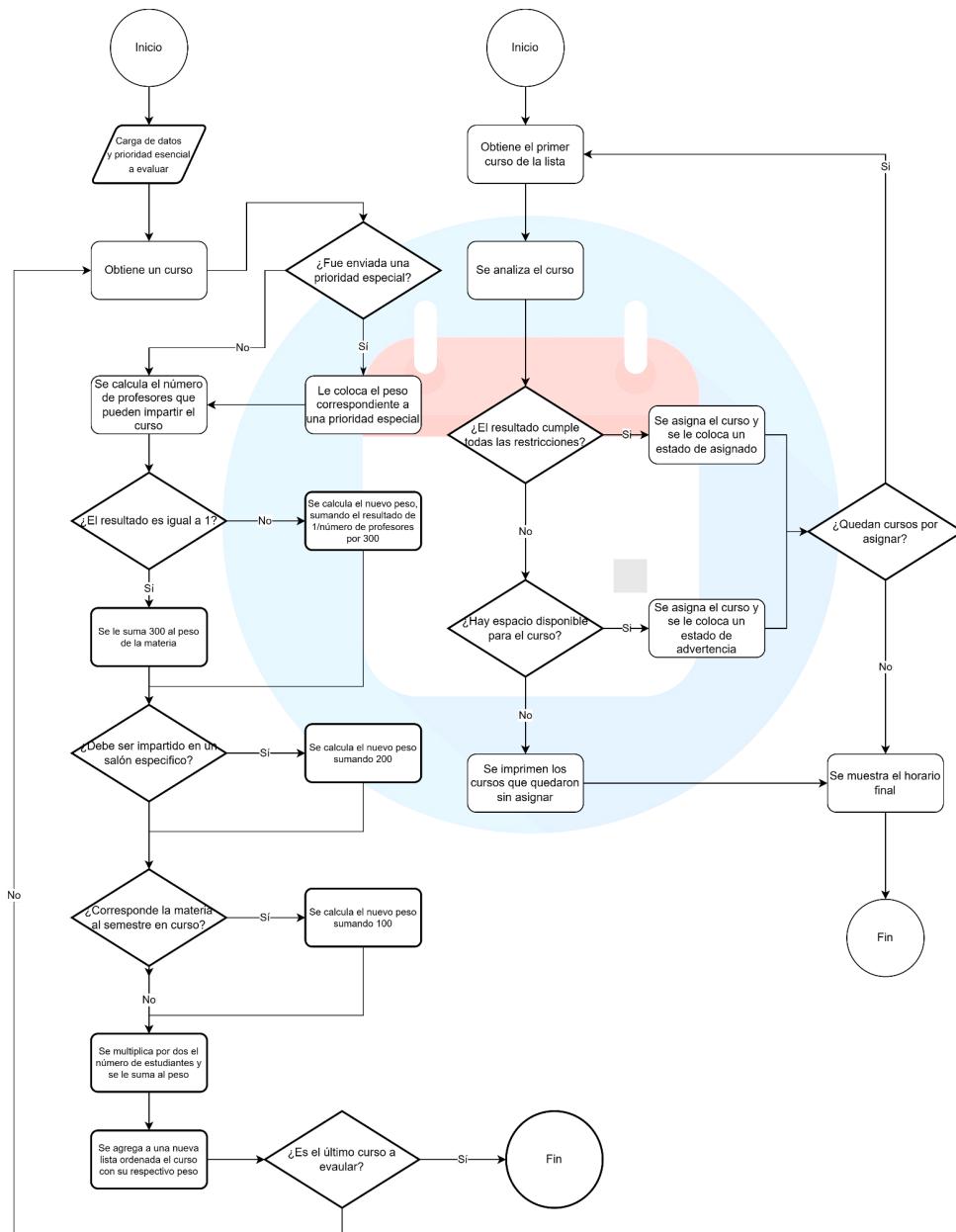
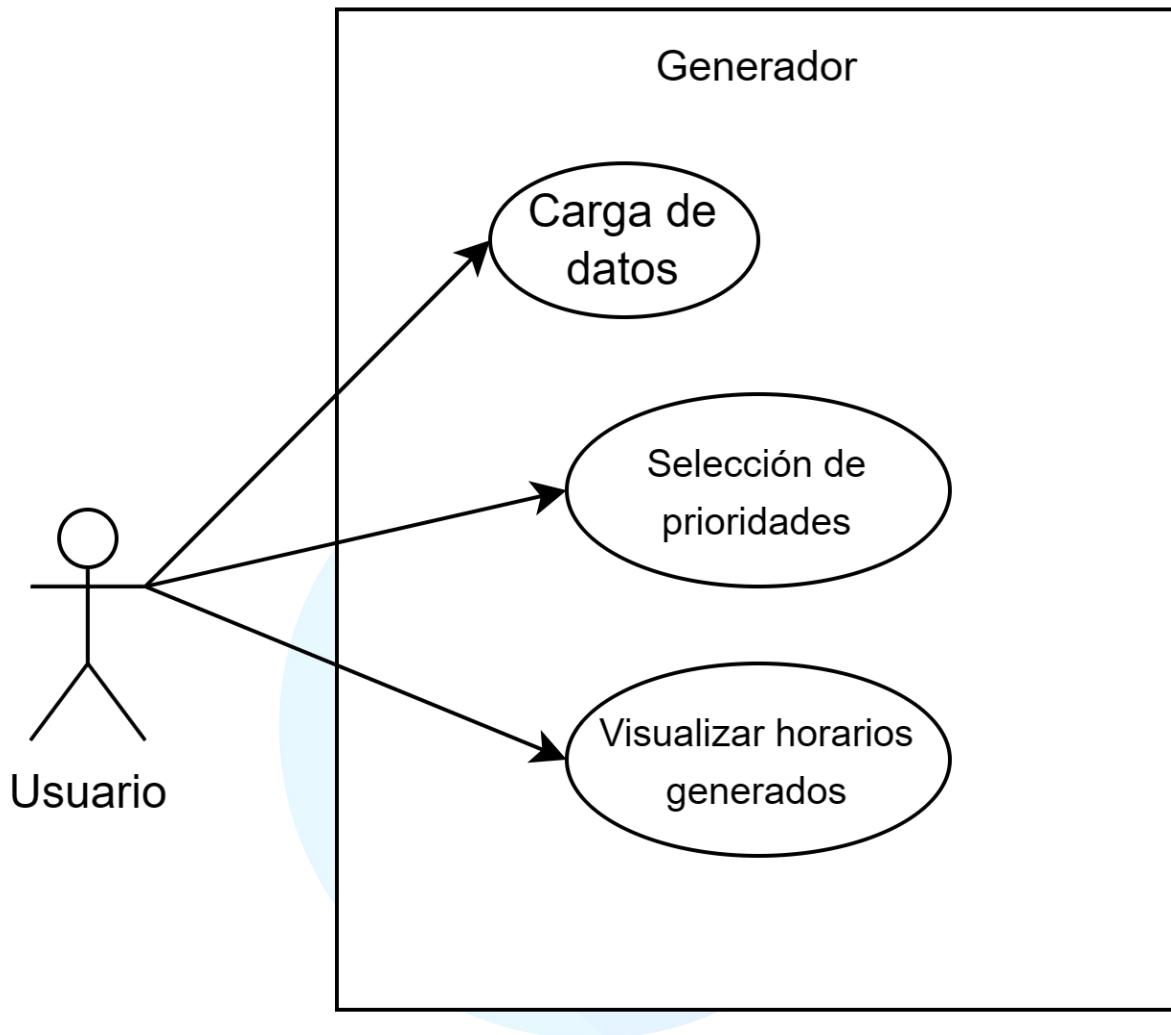




Diagrama de casos de uso



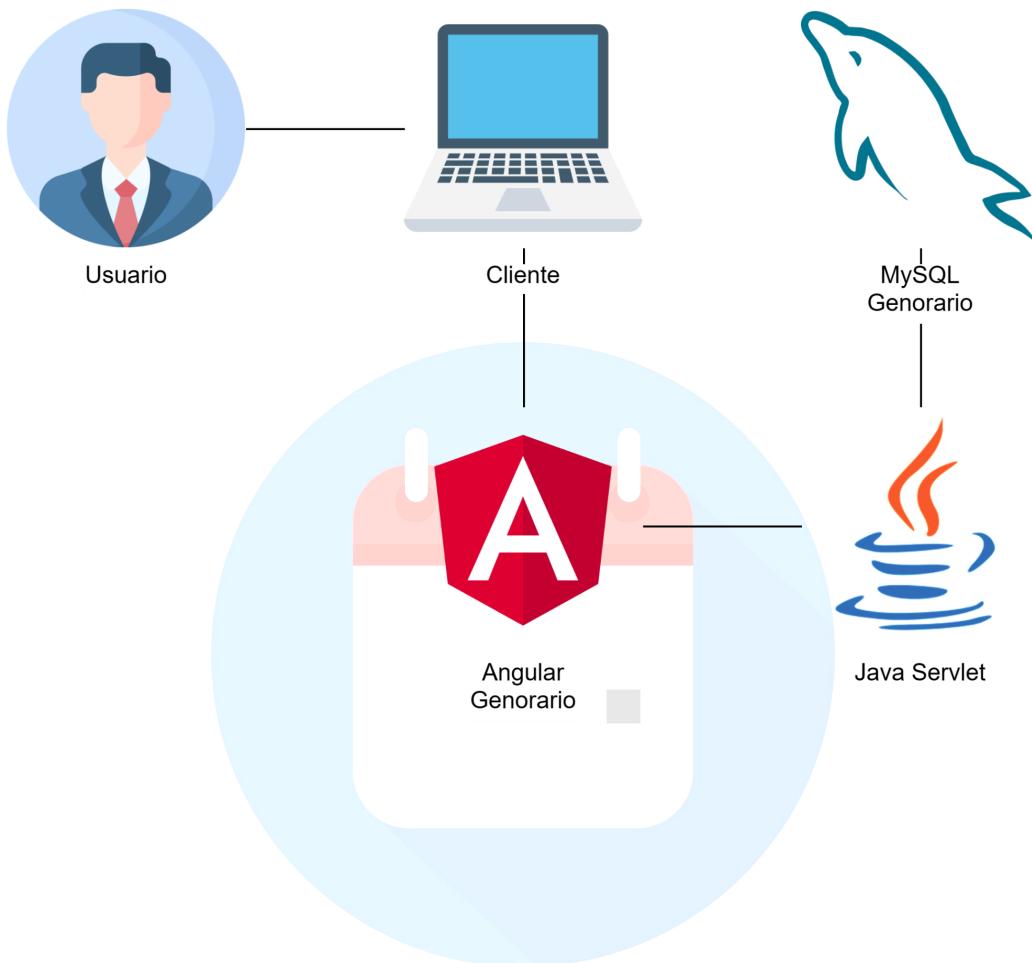
Arquitectura del sistema

El sistema consiste de los siguientes aspectos esenciales:

- Angular 16: como frontend por defecto, puesto que es una salida bastante sencilla y asequible para utilizar.
- Java Servlet: como backend por defecto, puesto que el uso de objetos facilita el análisis de los cursos y la asignación de los pesos.



- MySQL: como base de datos, para almacenar los cursos, profesores, salones y demás entidades que sean necesarias, así como los horarios generados y los valores de las prioridades.





Diagramas de secuencia

Diagrama de secuencia de la carga de datos

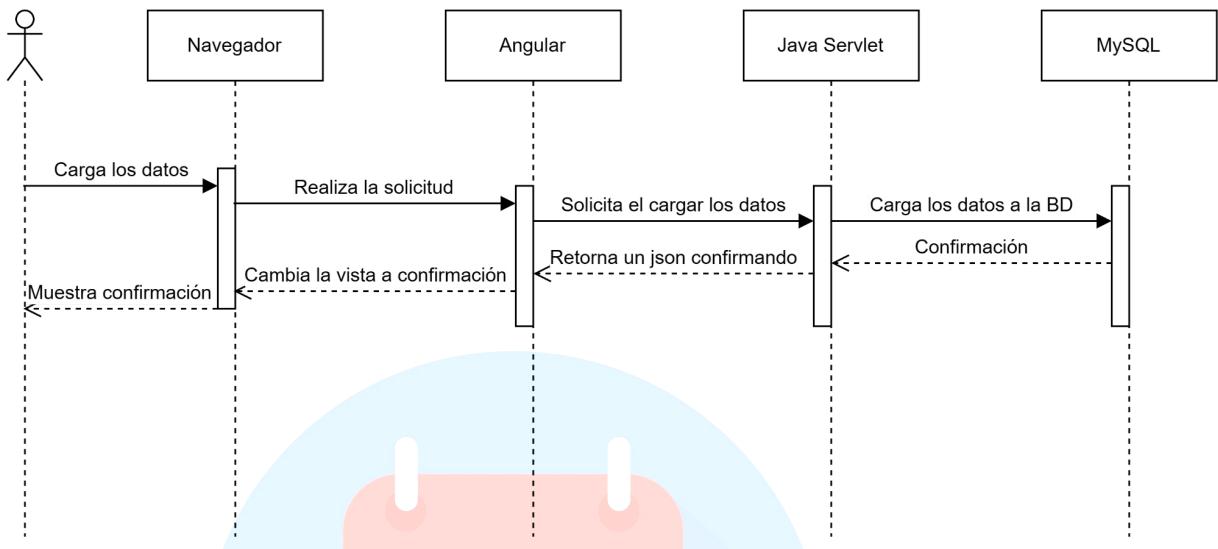


Diagrama de secuencia de la generación de horarios

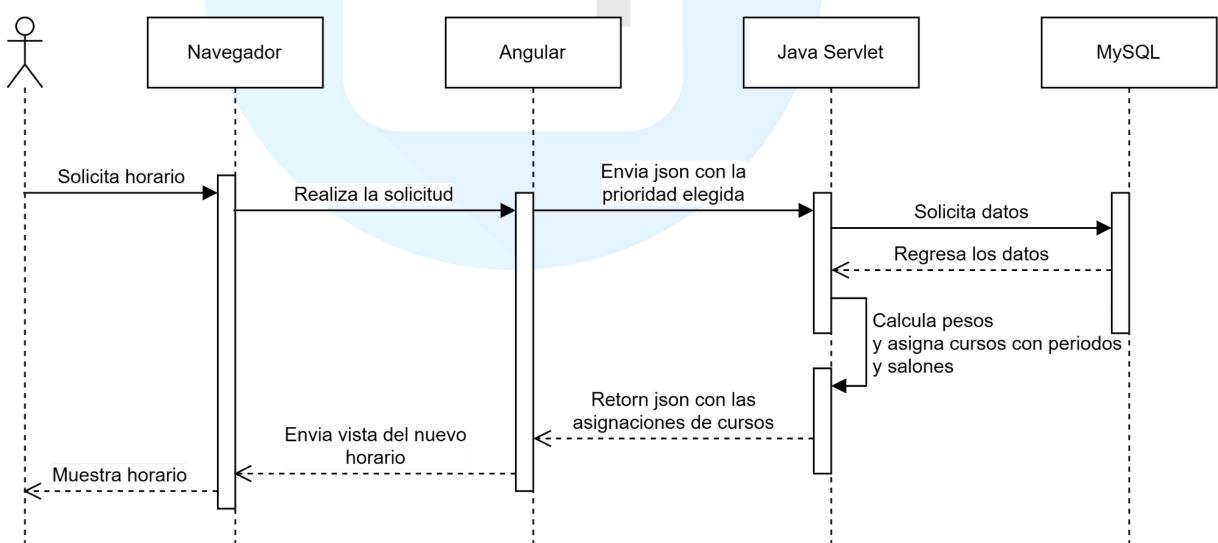




Diagrama de vistas

Vista creator

Esta vista nos permitirá crear un horario mediante el envío de las prioridades que queramos, así como nos permitirá modificar los valores con los que se calcula el peso de cada curso.

The screenshot shows a web browser window with the URL <https://localhost:4200/creator>. At the top, there are navigation icons for back, forward, and refresh, along with a file icon labeled "Page 1". On the right side of the header are three circular progress indicators. Below the header, there are four tabs: "Genorario", "Carga", "Generador", and "Horario". The "Generador" tab is active. Underneath the tabs, there are two main configuration sections. The left section, titled "Configuracion", contains five dropdown menus with arrows: "Contratacion", "Capacidad", "Asignacion", "Carrera", and "Profesor". The right section, titled "Prioridades", also has a "Configuracion" title. It includes three input fields: "Nombre del horario", "Prioridad" (with a dropdown arrow), and "Elige curso prioritario" (with a search icon). A large blue button at the bottom of this section is labeled "Generar".

Vista loader

Esta es la vista destinada a la carga de datos, ya sea mediante el uso de archivos de carga del tipo JSON y en la estructura definida en [Recolección y procesamiento de datos](#), la cuál también hace presencia en un apartado puesto que se muestran unos ejemplos para que al usuario le resulte más fácil el comprender cómo deberá de ir estructurado el archivo que este mismo decida subir, puesto que si en dado caso contiene errores se le deberá de informar al usuario.



Page 2

https://localhost:4200/loader

Genorario Carga Generador Horario

Carga de datos

Insertar mediante JSON

Choose File No file chosen

Materia Profesional Salón

Nombre de la materia

Área

INSERTAR

Ejemplo de como se tiene que hacer los JSON acá

Vista de horario

Acá en esta vista se pueden visualizar como quedan asignados los cursos en sus respectivos períodos y salones, mostrando la cantidad de alumnos que tiene asignados, el código del curso, el nombre del curso, y el estado de la asignación, pudiendo tener advertencia ya sea porque se traslapa con otro curso del mismo semestre o porque no se logró colocar cómo se debía.



Page 3

https://localhost:4200/schedule

Genorario Carga Generador Horario

Horario generado mediante prioridades

Periodo/Salon	Aula 1
12:00 - 13:40	IPC1
8:20 - 9:10	IPC2

Vista inicial

Solo introduce datos de la aplicación en un bonito carrusel.

Page 4

https://localhost:4200/

Genorario Carga Generador Horario

< First slide >

First slide label

Nulla vitae elit libero, a pharetra augue mollis interdum.

— — —



Cronograma de actividades

#	3/8/23	5/8/23	7/8/23	9/8/23	11/8/23	13/8/23	15/8/23	17/8/23	19/8/23	21/8/23	23/8/23	25/8/23	27/8/23	31/8/23	4/9/23	8/9/23
1	Toma de requerimientos															
2		Análisis del sistema														
3			Definición de tareas													
4				Creación de base de datos												
5					Creación del modelo a utilizar											
6						Creación de las vistas del proyecto										
7													Presentación de vistas			
8														Mejoras de las vistas		
9														Conexión vistas y backend		
10						Optimización de la BD									Pruebas de conexión	
11							Creación de endpoints									
12								Definición de modelo								
13									Prueba de características							
14										Optimización de errores						
15															Pruebas de integración	
16																Presentación
17																



DDL

```
CREATE DATABASE GENORARIO;
USE GENORARIO;
CREATE TABLE salons(
    id INT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    capacity INT NOT NULL
);
CREATE TABLE area(
    id VARCHAR(50) PRIMARY KEY
);
CREATE TABLE priority(
    id VARCHAR(50) PRIMARY KEY,
    description VARCHAR(100),
    factor INT
);
CREATE TABLE simulation(
    id INT auto_increment PRIMARY KEY,
    date_simulation datetime,
    name_simulation VARCHAR(50)
);
CREATE TABLE curse(
    code_curse VARCHAR(10) PRIMARY KEY,
    name_curse VARCHAR(50) NOT NULL,
    abbr_curse VARCHAR(20) NULL,
    semester_curse INT NOT NULL,
    assign_curse INT NOT NULL,
    area_curse VARCHAR(50) NOT NULL,
    FOREIGN KEY (area_curse) REFERENCES area(id)
);
CREATE TABLE teacher(
    id_teacher INT AUTO_INCREMENT PRIMARY KEY,
    name_teacher VARCHAR(50) NOT NULL,
    alter_teacher VARCHAR(20) NOT NULL,
    start_hour INT NOT NULL,
    end_hour INT NOT NULL,
```



```
area_teacher VARCHAR(50) NOT NULL,
FOREIGN KEY (area_teacher) REFERENCES area (id)
);
CREATE TABLE weight_calculated(
    id INT auto_increment PRIMARY KEY,
    code_curse VARCHAR(10) NOT NULL,
    FOREIGN KEY (code_curse) REFERENCES curse(code_curse)
);
CREATE TABLE period(
    id INT auto_increment PRIMARY KEY,
    start_period INT NOT NULL,
    end_period INT NOT NULL,
    duration INT DEFAULT 50
);
CREATE TABLE assigned_curse(
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_period INT NOT NULL,
    code_curse VARCHAR(10) NOT NULL,
    id_salon INT NOT NULL,
    id_teacher INT NOT NULL,
    id_simulation INT NOT NULL,
    status_assigned INT NOT NULL,
    message_assigned VARCHAR(100),
    FOREIGN KEY (id_period) REFERENCES period(id),
    FOREIGN KEY (code_curse) REFERENCES curse(code_curse),
    FOREIGN KEY (id_salon) REFERENCES salon(id),
    FOREIGN KEY (id_teacher) REFERENCES teacher(id_teacher),
    FOREIGN KEY (id_simulation) REFERENCES simulation(id)
);
CREATE TABLE priority_used(
    id_priority VARCHAR(50) NOT NULL,
    id_simulation INT NOT NULL,
    PRIMARY KEY (id_priority, id_simulation),
    FOREIGN KEY (id_priority) REFERENCES priority(id),
    FOREIGN KEY (id_simulation) REFERENCES simulation(id)
);
```



Colocación de objetivos y plan del proyecto global

Para la colocación de objetivos y la ideación del plan del proyecto, es necesario seguir una estructura clara y detallada qué logre explicar brevemente a lo que se quiere llegar con el simulador de horarios.

Objetivo general

Desarrollar un sistema de generación de horarios automatizado basado en asignación de prioridades, que optimice la distribución eficiente de recursos y tiempo en entornos con múltiples restricciones, para mejorar la planificación y organización de actividades.

Objetivos específicos

1. Identificar y recopilar los requisitos específicos de los usuarios y las restricciones del entorno para la generación de horarios.
2. Diseñar un algoritmo de asignación que tome en cuenta las prioridades de las materias, considerando factores como la importancia de la materia, la disponibilidad de escritorios en los salones y el horario de contratación de los docentes.
3. Crear una interfaz de usuario intuitiva que permita a los usuarios ingresar información relevante, establecer prioridades y visualizar los horarios propuestos, facilitando la interacción con el sistema.
4. Integrar la capacidad de manejar diferentes tipos de prioridades, como la capacidad de los salones, las materias disponibles o los horarios de contratación.
5. Refinar el algoritmo y la implementación para mejorar la eficiencia en la generación de horarios, reduciendo el tiempo necesario para producir soluciones y aumentando la calidad de las mismas.



6. Preparar documentación detallada y recursos de capacitación para los usuarios finales, explicando cómo utilizar el sistema de generación de horarios, interpretar los resultados y ajustar las prioridades según sea necesario.

Plan del proyecto global

El plan del proyecto consiste en el desglose de las tareas y actividades necesarias para cumplir con los objetivos establecidos. Sabiendo lo anterior, podríamos dividir dicho plan en las siguientes etapas:

1. Análisis y diseño

- a. Meditar los aspectos fundamentales a los que se sujetará un horario de clases.
- b. Definir los requisitos funcionales y no funcionales del generador de horarios.
- c. Diseñar la interfaz de usuario y la estructura del algoritmo de asignación de periodos y salones.

2. Desarrollo

- a. Se crea la interfaz de usuario utilizando las mejores prácticas de diseño y usabilidad.
- b. Implementar el algoritmo designado para la asignación de pesos para cada materia ingresada.
- c. Integrar los componentes del generador de horarios y realizar pruebas de funcionalidad.

3. Testing y correcciones:

- a. Realizar pruebas exhaustivas del generador para identificar posibles errores o fallos.
- b. Corregir y optimizar el código en base a los resultados de las pruebas.

4. Documentación y explicaciones:



- a. Crear material acerca de la aplicación para que los usuarios puedan comprender y hacer uso de sus funciones.
- b. Documentar todos los aspectos técnicos de la aplicación para poder garantizar un buen mantenimiento en un futuro.

5. Evaluación y mejora continua:

- a. Recolectar comentarios y sugerencias de los usuarios para mejorar el generador.
- b. Realizar actualizaciones periódicas para mantener el generador actualizado y funcional.

Conceptualización del modelo

El modelo parte del concepto de peso y asignación, el cuál asigna un valor de prioridad a los elementos de una colección, para luego extraer dichos elementos y asignarlos según los procedimientos definidos previamente.

Recolección y procesamiento de datos

Para la recolección de datos se definirá una estructura base para el ingreso de los mismos.

Ingreso de salones

```
[  
  {  
    "type": "fileSalons",  
    "data": [  
      {  
        "id": 1,  
        "nombre": "Aula 1",  
        "capacidad": 30,  
        "area": 2  
      }  
    ]  
  }  
]
```



Ingreso de cursos

```
[  
  {  
    "type": "fileCourses",  
    "data": [  
      {  
        "curso": "Social Humanistica 2",  
        "codigo": "029",  
        "abreviatura": "Humanistica 2",  
        "area": 1,  
        "semestre": 2,  
        "asignados": 50,  
        "catedraticos": [  
          {  
            "id": "32"  
          }  
        ]  
      }  
    ]  
  }  
]
```

Ingreso de profesores

```
[  
  {  
    "type": "fileTeachers",  
    "data": [  
      {  
        "id": 1,  
        "nombre": "Jose Moises Granados",  
        "alter": "Moises",  
        "entrada": 1400,  
        "salida": 2000,  
        "area": 2  
      }  
    ]  
  }  
]
```



Construcción del modelo

La construcción del modelo se puede definir en dos prioridades y cinco valores de peso que ayudarán a la asignación de los cursos. Las dos prioridades, se ejecutarán cuando se esté asignando como tal el curso en un periodo y salon, siendo que si se hace elección de una, dicha prioridad se ejecutará primero, mientras que los valores de peso, ayudarán a calcular los pesos que ordenarán la cola de los cursos, ordenando de forma descendente, del curso con más peso, al de menor valor.

Prioridades

Por contratación de profesores

Dicha opción asigna un periodo que le quede cómodo al futuro profesor que impartirá el curso. Luego se ejecutará la otra prioridad.

Por capacidad de salones

Dicha opción asigna un periodo que le quede cómodo al futuro salón dónde se impartirá el curso. Luego se ejecutará la otra prioridad.

Valores de peso

Valor por asignación

Suma al valor total del peso, la multiplicación del valor definido por el número de estudiantes asignados que tenga el curso.



Valor por carrera

Suma, al valor total del peso, la multiplicación de 100 por el nivel asignado a la carrera.

Valor por profesor

Basado en la ecuación

$$P_{profesores} = F_{profesores} * \frac{1}{T_{profesores}}$$

donde $P_{profesores}$ es el peso que se sumará al total del peso, $F_{profesores}$ el factor definido previamente y $T_{profesores}$ el número total de profesores que pueden dar el curso. Dicho total se obtiene ya sea de la asignación de profesores a cursos, o según los profesores del área que sea el curso.

Valor por salón

Basado en la ecuación

$$P_{salones} = F_{salones} * \frac{1}{T_{salones}}$$

donde $P_{salones}$ es el peso que se sumará al total del peso, $F_{salones}$ el factor definido previamente y $T_{salones}$ el número total de salones en donde se puede impartir el curso. Dicho total se obtiene ya sea de la asignación de salones a cursos, o según los salones que tiene el área del curso.

Porcentaje de reducción por semestre desfasado

Dicho porcentaje se resta del total del peso.



Verificación

Para la verificación se recopilaron algunos horarios reales, creados de diferentes semestres y años estudiantiles.

Validación

La validación del modelo implica asegurarse de que el simulador calcule los pesos de los horarios de forma correcta, esto mediante el encontrar los valores que generen la mejor eficiencia del horario, esto mediante lo siguiente:

1. **Colocación de datos de prueba reales:** Se recopila cierta cantidad de información para la prueba del correcto funcionamiento del generador del horario, se ajustan los valores antes definidos hasta obtener una eficiencia idónea.
2. **Comparación de los horarios generados con los reales:** Se comparan los horarios generados, con horarios de años anteriores y mediante la experiencia al momento de analizar un horario.
3. **Pruebas automatizadas:** Se desarrollan pruebas automatizadas que verifiquen la precisión del generador en diferentes escenarios.
4. **Documentación y transparencia:** Se asegurará el documentar adecuadamente cómo funciona el generador y cómo se realiza la asignación de cursos a un salón y periodo.

Para la validación del horario generado se dispone de un valor de eficiencia que permitirá visualizar si un horario es mejor que otro.



Diseño de experimentos

En el diseño de experimentos no se puede realizar tanta variación más que el estar ajustando los valores de peso definidos previamente, dichos valores influyen directamente en la eficiencia del horario generado, teniendo distintos estados de la asignación:

- OK: el curso se asignó correctamente a un salón y periodo, colocando un profesor adecuado al área y a los requerimientos del curso.
- SALONDIF: el curso se asignó a un salón que le pertenece a otra área que no es el área común ni el área del curso.
- TEACHEROUT: el curso se le asignó a un profesor que ya tiene un curso en el mismo horario.
- OVERLAP: el curso se traslapa con otro curso de la misma área y del mismo semestres.
- CRASH: el curso choca con otro curso en el mismo salón.
- OVERQUORUM: el curso se asignó a un salón que no tiene la capacidad para tantos estudiantes como los asignados al mismo.

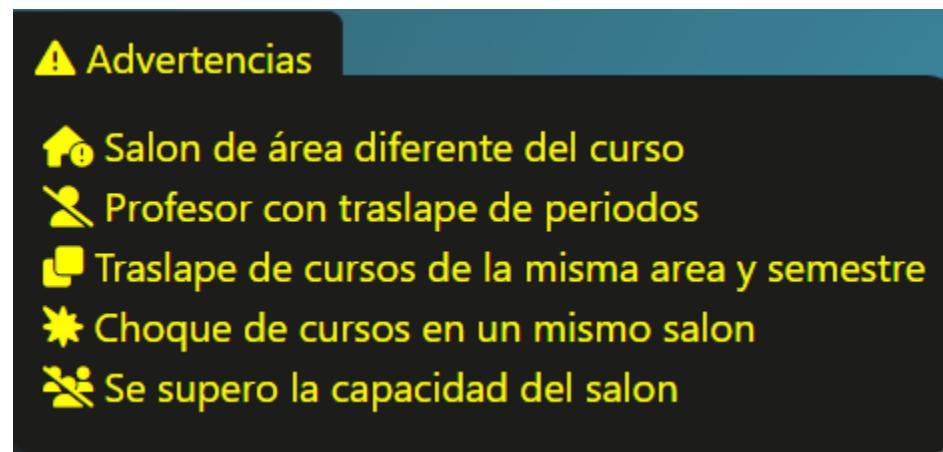
Producción de corridas de simulación y análisis

Para la producción de corridas se establece un número de pruebas para el generador, teniendo en cuenta la definición de los factores previamente establecidos, y se analiza qué valores dan un resultado mejor.

Para lo anterior se dispuso de la creación de una interfaz donde se pueda localizar las deficiencias en las asignaciones de los cursos, ya sea qué se asignen recursos a salones que no



tienen la capacidad para tantos estudiantes asignados. Dichas asignaciones se muestran de la siguiente manera en la interfaz definida:



Puesta en marcha del modelo

Se adjuntan capturas de la puesta en marcha del modelo

Horario Generado Eficiencia 84.78							
Periodo / Salón	Aula 1	Aula 2	Aula 3	Aula 4	Aula 5	Aula 6	Aula 7
13:40 - 14:30	2840 Sistemas Aplicados 2 10						
14:30 - 15:20	2810 Compi 2 6	029 Humanistica 2 2	690 Economica 1 8	779 Conta 2 6	292 Inter 3 4	2819 SOPE2 8	670 Electric 6
15:20 - 16:10	2808 TS1 6	2820 Redes 2 8	2838 Inteligencia 2 10	912 Planea 10			
16:10 - 17:00	2836 Software Avanzado 10	170 Básica 2 2	2666 Orientación 1				



GENORARIO Carga Generador Horario

Advertencias Horario Generado Eficiencia 78.89 Ⓜ

Periodo / Salón	Aula 1	Aula 2	Aula 3	Aula 4	Aula 5	Aula 6	Aula 7
13:40 - 14:30							
14:30 - 15:20	2810 Compi 2 6	029 Humanistica 2 2					
15:20 - 16:10	2836 Software Avanzado 10	690 Economica 1 8	292 Inter 3 4	2670 Gestion 6			
16:10 - 17:00	2838 Inteligencia 2 10	170 Básica 2 2	085 Lógica 4	949 Estadistica 1 4	670 Electrica 1 6	904 Aplicada 2 6	
	2800	2811	2812	779	685	2841	905





Conclusiones

- La capacidad de personalizar un horario según las preferencias y prioridades individuales es esencial para maximizar la eficiencia. Al permitir a los usuarios asignar pesos y establecer prioridades a los cursos, el generador de horarios puede adaptarse de manera óptima a las necesidades específicas de cada establecimiento. Esto resulta en horarios que son más eficientes y efectivos, ya que se ajustan a los objetivos.
- La eficiencia en la gestión del tiempo es fundamental en la vida moderna. Un generador de horarios que tenga en cuenta factores como el número de estudiantes asignados al curso, la cantidad de profesores que pueden impartir un curso o la cantidad de salones dónde se puede impartir un curso.
- La creación de un generador de horarios basado en valores de peso y prioridades es un ejemplo de cómo la tecnología puede tener un impacto positivo en la calidad de vida del estudiante. Al proporcionar una herramienta que simplifica la planificación y organización de los períodos, se les ayuda a tomar decisiones más informadas.