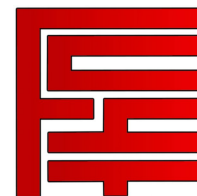


UNIVERSIDAD MAYOR DE SAN SIMÓN
FACULTAD DE CIENCIAS Y TECNOLOGIA
INGENIERÍA INFORMÁTICA



**DESARROLLAR UN ASSET PARA FACILITAR LA CREACIÓN
PROCEDURAL DE MAPAS TIPO MOSAICO 2D Y 3D A
DESARROLLADORES DE VIDEOJUEGOS.**

Proyecto de Grado Presentado para optar al Diploma Académico de Licenciatura en Ingeniería
Informática

Presentado por: Ríos Cardozo Nicolás Luis
Tutor: Lic. -

COCHABAMBA - BOLIVIA
II, 2025

DEDICATORIA

Dedicado a

AGRADECIMIENTOS

Agradezco a ...

FICHARESUMEN

mi resumen

Índice general

DEDICATORIA	I
AGRADECIMIENTOS	III
FICHARESUMEN	V
1 INTRODUCCIÓN	1
1.1 ANTECEDENTES	1
1.2 DESCRIPCIÓN DEL PROBLEMA	1
1.2.1 Definición del Problema	1
1.3 OBJETIVOS	2
1.3.1 Objetivo general	2
1.3.2 Objetivos específicos	2
1.4 JUSTIFICACIÓN	3
1.5 LÍMITES Y ALCANCES	3
1.6 METODOLOGÍA DE DESARROLLO	3
2 MARCO TEÓRICO	5
2.1 MAPAS ESTILO MOSAICO O TILEMAPS	5
2.2 WAVE FUNCTION COLLAPSE	5
2.2.1 Algoritmos genéticos paralelos y distribuidos (PGA , DGA)	6
2.2.2 Algoritmos genéticos híbridos (HGA)	6
2.2.3 Algoritmos genéticos adaptativos (AGA)	6
2.2.4 Algoritmos genéticos rápido desordenado (FmGA)	6
2.2.5 Algoritmos genéticos muestreo independiente (ISGA)	6
2.3 APLICACIONES ACTUALES EN BASE A ALGORITMOS GENÉTICOS	6
2.4 MOTORES DE VIDEOJUEGOS	6
2.5 ESTADO DEL ARTE	7
2.6 TECNOLOGÍAS	7
2.6.1 Lenguaje de programación	7
2.6.2 Spring	7
2.6.3 Jenetics	7
2.6.4 Base de datos	8
2.6.5 Angular	8
2.7 PROCESO DE DESARROLLO	8
2.7.1 Kanban	8
2.7.2 Por qué no otros procesos	8
3 MARCO DE APLICACIÓN	9
3.1 SUBTITULO	9
3.2 SUBTITULO	9

3.3	DISEÑO	9
3.3.1	subsubtitulo	9
3.4	IMPLEMENTACIÓN	9
3.4.1	Herramientas de software	9
4	DESARROLLO DEL PROYECTO	11
4.1	DISEÑO DE ARQUITECTURA	11
4.2	DISEÑO DE ALGORITMO GENÉTICO	12
4.3	DISEÑO DE BASE DE DATOS	12
4.4	DISEÑO DE INTERFAZ DE USUARIO	12
5	CONCLUSIONES Y RECOMENDACIONES	15
5.1	SUBTÍTULO	15
5.1.1	Subsubtitulo	15
	BIBLIOGRAFÍA	17
	ANEXOS	17

Índice de figuras

1.1	Árbol de problemas, centrado el la generación de mapas para videojuegos.	2
2.1	experimento de la doble rendija.	6
4.1	Diagrama de componentes	11
4.2	Diagrama entidad relación	12
4.3	Menu de selección de materias, grupos	13
4.4	Menu de selección de prioridad de problema	13
4.5	Tabla de visualización de horario	14

Índice de cuadros

1.1	Cronograma de actividades	3
-----	-------------------------------------	---

Capítulo I

INTRODUCCIÓN

El presente proyecto consiste en resolver la necesidad de herramientas mas accesibles en la generación procedural de mapas en el desarrollo de videojuegos

1.1 ANTECEDENTES

El desarrollo de videojuegos es un área en crecimiento mas accesible de ingresar gracias a la facilidad que generan el uso de motores de juegos, y herramientas para estos generados por sus respectivas comunidades.

A pesar de todos los beneficios de usar un motor de videojuegos como base para el desarrollo, hacer videojuegos es actualmente una tarea que demanda de muchos aspectos en los que trabajar por lo que se puede terminar tomando mucho tiempo en terminar de implementar todos los aspectos necesarios que lo involucra

Entre las necesidades mas comunes para un desarrollador de videojuegos es la creación de mapas que puede ser una tarea tardía, por lo que muchos proyecto pequeños optan por la generación automática de estos por diferentes métodos

cita ejemplo (?)

1.2 DESCRIPCIÓN DEL PROBLEMA

Un videojuego es un proyecto pesado que incluye muchas áreas a desarrollar entre ellos diseñar el mapa de los niveles así que para aligerar la carga los desarrolladores pueden usar Assets.

Un Asset es como una biblioteca que puede incluir scripts de código, imágenes, modelos 3D/2D, etc disponibles a usar para un nuevo proyecto, estos suelen estar publicados para proyectos usando algún motor de videojuegos específico para ser compatible.

1.2.1 Definición del Problema

Dificultad para la generación de mapas en en videojuegos.

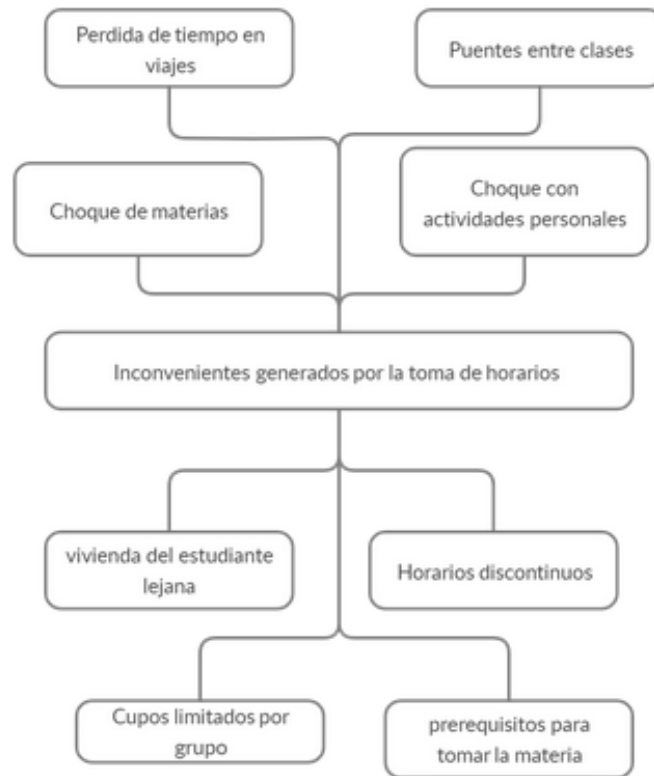


Figura 1.1: Árbol de problemas, centrado en la generación de mapas para videojuegos.

1.3 OBJETIVOS

A continuación se presentan el objetivo general y los objetivos específicos en este proyecto de grado.

1.3.1 Objetivo general

Desarrollar un Asset para facilitar la creación procedural de mapas tipo mosaico 2D y 3D a desarrolladores de videojuegos

1.3.2 Objetivos específicos

1. Investigar la técnica de wave function collapse para la generación de mapas
2. Investigar la técnica de seeding/semillas para añadir control y reproducibilidad de los mapas resultados generados
3. Investigar un motor de videojuegos para implementar un Asset compatible con esa tecnología
4. Implementar la funcionalidad de generación de mapas compatible con las herramientas del motor de videojuegos
5. Definir casos de uso para las pruebas de la funcionalidad

6. Hacer el Asset publico para cualquier desarrollador de videojuegos

1.4 JUSTIFICACIÓN

Como se menciona en puntos anteriores hacer videojuegos es un tarea pesada, entre las necesidades mas comunes para un desarrollador de videojuegos es la creación de mapas, para aligerar tal carga se busca publicar un Asset que ayude a desarrolladores a facilitar esa área del desarrollo de videojuegos.

1.5 LÍMITES Y ALCANCES

El presente trabajo de grado se enfoca en los siguientes aspectos:

-
-
-
-

1.6 METODOLOGÍA DE DESARROLLO

Al momento de avanzar en el procedimiento se definió que el proceso a seguir necesitaría ser un proceso ágil para poder avanzar evitando interrupciones y acomodarse a necesidades surgientes en el desarrollo. Entre los procesos ágiles conocidos se eligió kanban por el echo de poder amoldarse al trabajo en solitario requerido y ser flexible con flujo de trabajo para evitar posibles atascos. Las tareas principales de investigación que se definió serían las siguientes:

Cuadro 1.1: Cronograma de actividades

Nro. Objetivo Específico	Actividades	Recursos Necesarios	Resultados a obtener
1	Investigación de la implementación de las técnicas wave function collapse y seeding		Diseño de algoritmos a usar
2	Investigación del motor a usar y herramientas para generar mapas	lenguaje de programación compatible con el motor	Diseño de uso en el motor y herramientas a usar de este
3	Diseñar y desarrollar biblioteca de generación de mapas	lenguaje de programación compatible con el motor	biblioteca de generación de mapas
4	Desarrollar pruebas para la biblioteca creada	Motor y biblioteca de unit test	conjunto de test de unidad
5	publicar Asset implementado	Motor, biblioteca de generación de mapas	Asset

Gran parte de los puntos no requieren la totalidad de la investigación de puntos anteriores así que se irán generando tareas a cumplir según lo de lo que ya no se tenga bloqueos para su desarrollo.

Capítulo II

MARCO TEÓRICO

En este capítulo se explican los conceptos de los mapas estilo mosaico y las técnicas a usar para poder generarlos procedural-mente, también se profundizara en la tecnología de los motores de videojuegos a usar ¹.

2.1 MAPAS ESTILO MOSAICO O TILEMAPS

Los mapas estilo mosaico o también llamados tilemaps son una técnica común en el desarrollo de videojuegos especialmente en juegos 2D, que consiste en construir el mapa del mundo o nivel de juego a base de pequeñas imágenes con forma usualmente cuadrada a los que se les llaman mosaicos o tiles, los beneficios de usar esto es que no necesitar grandes imágenes que pueden pesar mucho en cambio se construye usando pequeñas imágenes que se pueden repetir varias veces en diferentes partes del mapa.

Otro beneficio de esto es que se pueden poner en matrices de 2 y 3 dimensiones lo que hace sencillo poder definir las posiciones de los tiles usando algoritmos.

En el caso de mapas 3D se usan modelos 3D o también llamados 3D mesh, en vez de imágenes obteniendo los mismos beneficios.

2.2 WAVE FUNCTION COLLAPSE

El colapso de la función de onda o wave function collapse es un algoritmo basado en un concepto de física cuántica con el mismo nombre. Cuando se mide un sistema cuando en superposición de estados su función de onda se colapsa a un solo estado, por lo que se considera el estado es indeterminado hasta que se haga una medición, experimentos conocidos de este fenómeno son el gato de Schrödinger o el experimento de la doble rendija.

¹métodos adaptativos: métodos basados en la estimación del error

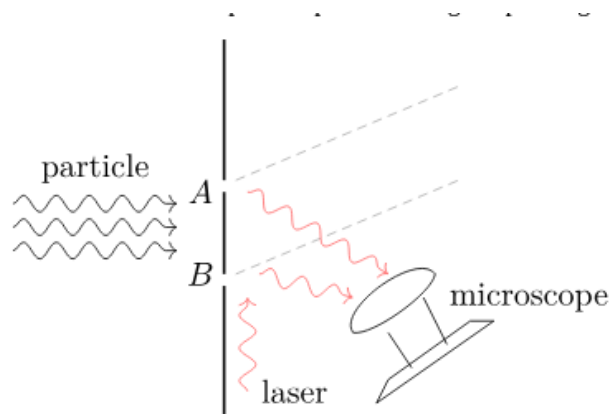


Figura 2.1: experimento de la doble rendija.

2.2.1 Algoritmos genéticos paralelos y distribuidos (PGA , DGA)

2.2.2 Algoritmos genéticos híbridos (HGA)

2.2.3 Algoritmos genéticos adaptativos (AGA)

2.2.4 Algoritmos genéticos rápido desordenado (FmGA)

2.2.5 Algoritmos genéticos muestreo independiente (ISGA)

2.3 APLICACIONES ACTUALES EN BASE A ALGORITMOS GENÉTICOS

El uso de algoritmos genéticos ha estado subiendo exponencialmente en los últimos años. Los algoritmos genéticos son usados en su mayoría para los problemas de optimización y la resolución de problemas NP difíciles. Una de las áreas donde se puede ver a los algoritmos genéticos ser aplicado es el aprendizaje de máquinas, Velez-Langs Oswaldo y Santos Carlos (2014) usaron algoritmos genéticos para su implementación de un sistema recomendador, que demostró su correctitud en la recomendación de películas al usuario. Lars Bungum y Björn Gambäck (2010) demostraron que la programación evolutiva puede ser efectivamente usada en las diferentes etapas del procesamiento del lenguaje natural.

2.4 MOTORES DE VIDEOJUEGOS

Usar motores de videojuegos como base de un proyecto es actualmente lo mas frecuente en el desarrollo de videojuegos ya que bastante costoso implementar desde 0 todas las funciones que aportan estos.

Por lo que los Assets necesitan ser compatibles con el motor de videojuegos que se elija usar entre ellos las opciones mas conocidas y completas serian Unity y Unreal

Unity y Unreal son motores de videojuegos los cuales han estado dominando la industria por un buen tiempo por lo cual tienen un buen repertorio de Assets publicados dando varias opciones a nuevos desarrolladores, aparte de estos hay varios otros motores entre los cuales uno adquirió popularidad recientemente Godot.

Godot es un motor de videojuegos gratuito de código abierto creado originalmente en Argentina por Ariel Manzur y Juan Linietsky como un proyecto cerrado el cual pasaría a lanzarse como código abierto el 14 de enero de 2014 con licencia MIT Aunque adquirió popularidad rápidamente el aporte de la comunidad es bastante pequeño comparado con otros motores que llevan mas tiempo encabezando el mercado.

Una gran ventaja de usar Godot por encima de motores como Unity y Unreal es que es completamente gratuito por lo que desarrolladores no necesitan pagar licencias para publicar sus juegos, también que al ser código abierto no esta sujeto a las políticas de ninguna compañía por lo que se puede tener mas libertad creativa.

Otra ventaja de Godot es que al poder usar el mismo lenguaje de programación C# usado en Unity y Unreal estos proyectos serian mas fáciles de exportarse a este nuevo motor de ser necesario y viceversa.

2.5 ESTADO DEL ARTE

2.6 TECNOLOGÍAS

2.6.1 Lenguaje de programación

Para lograr el cometido se eligió el lenguaje de programación Java por múltiples motivos que veremos a continuación.

Java es un lenguaje popular que fue publicado en 1995 por Sun Microsystems, entre las ventajas que tiene es que es un lenguaje moderadamente fácil de entender y mantiene un alto rendimiento. Al ser uno de los lenguajes más populares que existen java cuenta con una comunidad grande y extensa documentación disponible para usar, tambien le da el beneficio de tener una basta cantidad de frameworks y diferentes bibliotecas hechas para este lenguaje. de las cuales veremos a continuación las que se eligió para llevar a cabo este proyecto.

2.6.2 Spring

Spring es un framework para Java posiblemente el más popular en la actualidad que ofrece varias herramientas al desarrollador para poder simplificar bastante el trabajo en la creación de aplicaciones web. Spring es bastante modular permitiendo solo importar las librerías relevantes para el trabajo.

En este proyecto se busca poder usar Spring para poder dar un API capaz de ser usado para poder dar los resultados a los estudiantes de los horarios que desean optimizar y también pueda dar un sistema con un mínimo aceptable de seguridad a los administradores para popular la información necesaria de los horarios.

2.6.3 Jenetics

Jenetics es una biblioteca para java que implementa varios de los conceptos de algoritmos evolutivos y utiliza la funcionalidad de Stream de java para la iteración de generaciones, lo que permite concentrarse en la resolución del problema de este proyecto que es la optimización de los horarios elegidos por estudiantes, que principalmente radica en la definición de la función fitness.

2.6.4 Base de datos

Para el almacenamiento de información necesaria en este proyecto se eligió utilizar mysql por su sencillez popularidad y estabilidad.

2.6.5 Angular

En cuanto la interfaz de usuario se utilizó Angular al ser un framework bastante completo y robusto, conteniendo todo lo necesario para las necesidades de este proyecto.

2.7 PROCESO DE DESARROLLO

Como se mencionó en el anterior capítulo este proyecto estará desarrollado usando el proceso ágil conocido como Kanban. en los siguientes puntos se profundizará más respecto al proceso y el motivo de su elección

2.7.1 Kanban

2.7.2 Por qué no otros procesos

Capítulo III

MARCO DE APLICACIÓN

3.1 SUBTITULO

3.2 SUBTITULO

3.3 DISEÑO

3.3.1 subsubtitulo

3.4 IMPLEMENTACIÓN

3.4.1 Herramientas de software

Capítulo IV

DESARROLLO DEL PROYECTO

4.1 DISEÑO DE ARQUITECTURA

Se opta por una arquitectura MVP(modelo vista presentador) para mantener desacoplada la interfaz de usuario y tener un API de uso genérico que puede ser reutilizado por otros proyectos. Spring basa su arquitectura en MVC(Modelo vista controlador) y es la arquitectura que seguiría el componente de backend en este proyecto pero, al integrar también una interfaz de usuario ajena a Spring boot el controlador desde el punto de vista general del proyecto pasaría a usarse como un intermediario funcionando como un presentador que se encarga de la lógica y enviar o recibir información de la vista a petición.

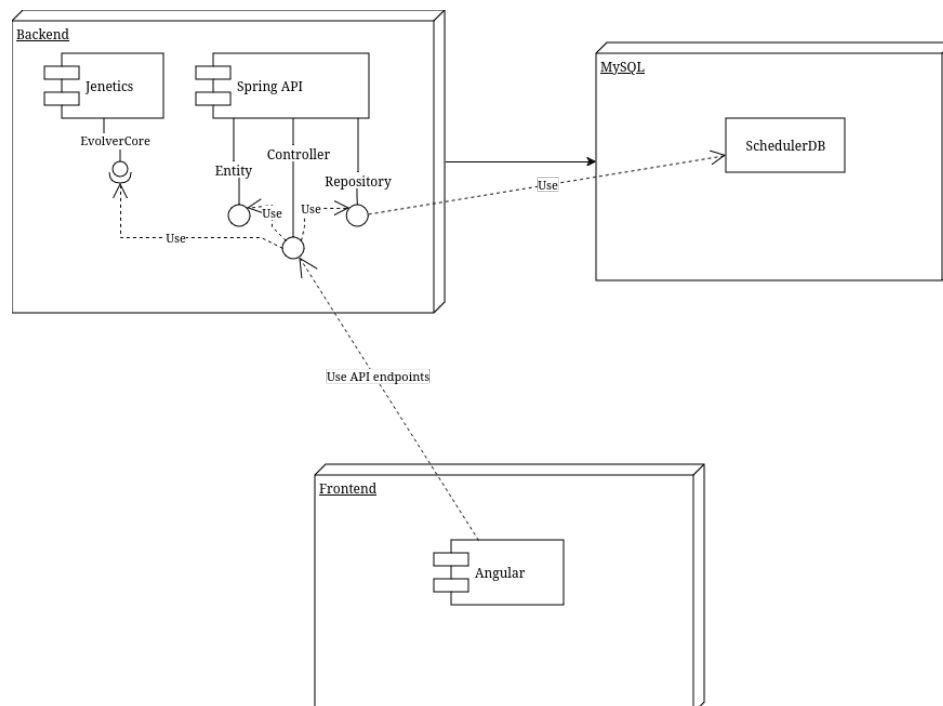


Figura 4.1: Diagrama de componentes

4.2 DISEÑO DE ALGORITMO GENÉTICO

Tras el análisis hecho a las variaciones mencionadas en el capítulo se decidió por usar la siguiente estructura [TODO]. algoritmo genético simple.[TODO] analizar matemáticamente las probabilidades y cantidades máximas de combinaciones, permutaciones, relevancia en complejidad del problema, como son 6 max de materias tomadas posiblemente se utilice el algoritmo genético base pero cada parte también tiene formas de implementar revisar cuales estan disponible en Jenetics..[TODO].. dado ya el algoritmo a seguir, para el desarrollo de esto se utilizó como apoyo Jenetics una librería para el lenguaje de programación Java que cuenta con bases para la implementación de algoritmos genéticos que ayudan a agilizar la implementación de la aplicación

4.3 DISEÑO DE BASE DE DATOS

Para implementar un sistema capaz de recomendar horarios a los estudiantes, primero se necesita la información con la que trabajar, en vez de tediosamente re ingresar todos los datos para cada uso es más conveniente poder almacenar la información de los horarios en una base de datos a la cual el sistema tenga acceso para poder revisar en cada request de algún usuario para poder generar un horario óptimo. Dado lo anterior dicho se trabajó en un diseño de la información a guardar respecto a los horarios que incluye tanto los tiempos de ingreso de las materias y de finalización, como también información básica de los docentes que darán tales materias y los grupos disponibles que estarán en la gestión.

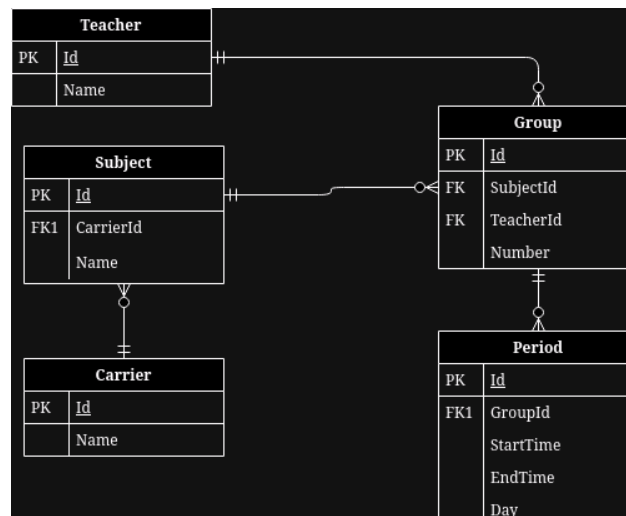


Figura 4.2: Diagrama entidad relación

En la anterior imagen se puede visualizar la relación entre los datos necesario para definición de horarios

4.4 DISEÑO DE INTERFAZ DE USUARIO

En la interfaz de usuario se decidió optar por una vista simple que sea capaz de mostrar el horario resultado de las preferencias del estudiante y un menu para seleccion dichas preferencia.

Un menu desplegable para seleccionar las materias en las que este interesado el alumno o mas

específicamente grupos si hay uno en concreto que quiera tomar por encima de los otros ofertados por la institución.

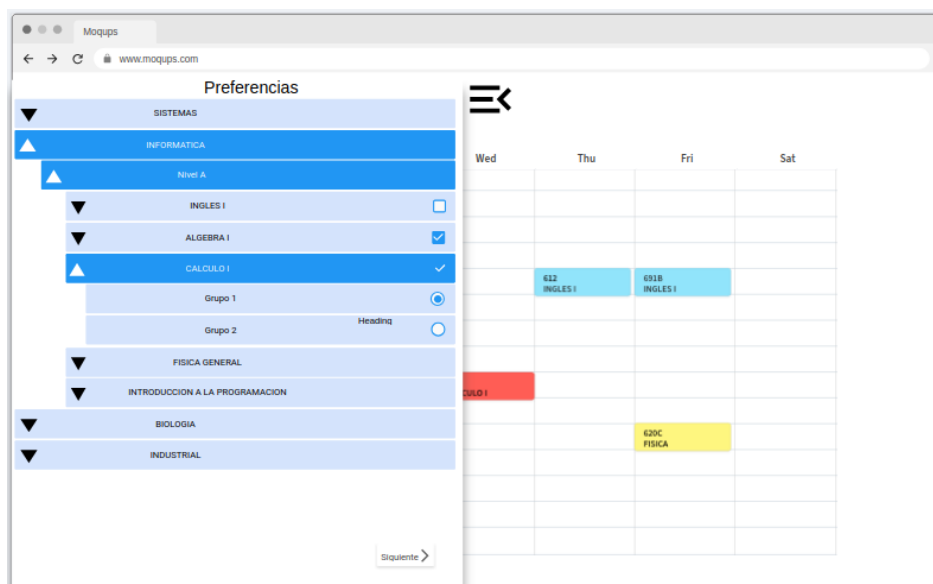


Figura 4.3: Menu de selección de materias, grupos

Una segunda sección del menu para indicar que prioridad tiene el estudiante respecto a evitar choques entre sus materias tomadas en la gestion o los puentes de periodos libres entre estas para evitar perder mucho tiempo.

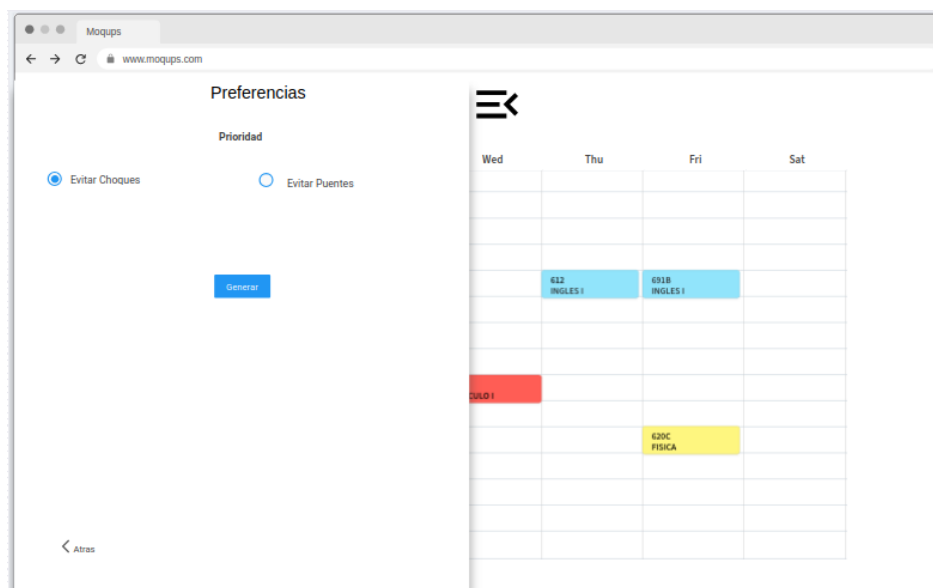


Figura 4.4: Menu de selección de prioridad de problema

Una vez seleccionado las preferencias del estudiante puede mandar a generar un horarios tomando en cuenta dichas preferencia y vizualizar el resultado en una tabla como se muestra a en la imagen a continuación.

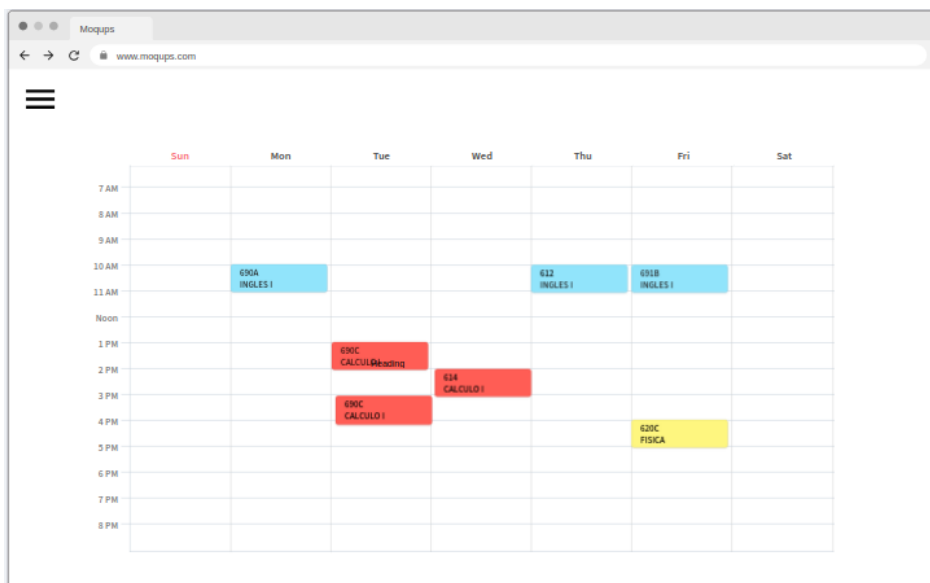


Figura 4.5: Tabla de visualización de horario

Capítulo V

CONCLUSIONES Y RECOMENDACIONES

5.1 SUBTITULO

Lorem ipsum dolor sit amet

5.1.1 Subsubtitulo

Lorem ipsum dolor sit amet

ANEXOS