

Algoritmo Genético

Calendario Generativo

Reporte

Inteligencia Artificial

Docente: Dr. Carlos Alberto Díaz Hernández.

Nombre del alumno	Matrícula
Adolfo González Juárez	213358
Andrés Juárez Mandujano	213364
Martín De Jesús Rincón Narcia	211160

Índice

1. Introducción	3
2. Análisis del Problema	4
2.1. Problemática	4
3. Resolución de problemas	4
3.1. Captura de progreso académico.	5
3.2. Filtrado de materias por seriación	5
3.2.1. Seriación	6
3.2.2. Depuración de materias	7
3.2.3. Filtrado de materias por disponibilidad horaria	9
3.3. Cuantificación de los datos	10
3.3.1. Umbral de relevancia	11
3.3.2. Franja horaria	14
3.4. Procesamiento de la información	14
3.4.1. Semilla	14
3.4.2. Longitud de semilla	15
3.4.3. Base Aritmética	15
3.4.4. Decodificación de semilla	16
3.4.5. Individuos	17
4. Resultado y salida	17
5. Metodología	17
5.1. Emparejamiento	18
5.2. Mutación	18
5.3. Poda	18
6. Visualización y gráficos	18
6.1. Progreso actual	19
6.2. Grupos disponibles	19
6.3. Resultado	20
6.4. Estadísticas	20
7. Herramientas	21
8. Conclusión	22

Índice de figuras

1. Diagrama del funcionamiento general del algoritmo.	4
2. Captura de pantalla de la sección de selección de materias cursadas, obtenida del maquetado en Figma.	5
3. Mapa curricular de la carrera de Ingeniería en Desarrollo de Software del año 2017.	6
4. Mapa curricular simplificado de ejemplo: Caso 1.	7
5. Mapa curricular simplificado de ejemplo: Caso 2.	8
6. Diagrama de materiales reales de carga.	10
7. Ejemplo ilustrativo de cálculo de relevancia por dependiente.	12
8. Ejemplo ilustrativo de cálculo de relevancia por distancia.	13
9. Representación de los valores de la franja horaria.	14
10. Franja horaria: Ejemplo 1.	14
11. Visualización de longitud de semilla y base aritmética.	15
12. Visualización de decodificación de semilla.	16

13.	Sección de selección de estado actual del usuario.	19
14.	Horario del grupo disponible 4°A en el periodo Enero-Abril de 2024.	19
15.	Resultado para la generación de horario de 1000 generaciones de la selección de la figura 13. .	20
16.	Estadísticas historicas sin colisión de la de la selección de la figura 13, después mil generaciones	21
17.	Estadísticas historicas con colisión de la de la selección de la figura 13, después mil generaciones	21

Lista de ecuaciones

1.	Arreglo de materias sin procesar.	5
2.	Atributos de las asignaturas sin procesar.	6
3.	Condición de seriación para filtro de materias.	8
3.	Lista de horarios disponible por periodo y sus atributos	9
4.	Definición de materiales reales de carga.	10
5.	Definición simplificada de materia apta para procesamiento.	11
5.	Definiciones de datos de entrada.	11
6.	Cálculo de umbral de relevancia por dependientes.	11
7.	Cálculo de umbral de relevancia por distancia.	12
8.	Umbral de relevancia.	13
9.	Condición de individuo sin colisiones.	17
10.	Cálculo del peso total de un individuo.	17

1. Introducción

El presente documento tiene como objetivo presentar un algoritmo genético diseñado para abordar una problemática común en el ámbito estudiantil: la selección óptima de materias para un estudiante en un estado de irregularidad académica. Esta herramienta de software propone una solución que determina la distribución más adecuada de materias teniendo en cuenta los grupos disponibles en el cuatrimestre próximo, así como la secuencia recomendada dentro del plan de estudios y el progreso académico del estudiante. La exposición detallada del análisis, la metodología empleada y el desarrollo del software facilita la comprensión de su funcionamiento y su aplicación práctica en el contexto educativo.

2. Análisis del Problema

Para abordar la resolución de la problemática planteada, se llevó a cabo una exhaustiva identificación de los elementos pertinentes que influyen en el funcionamiento del algoritmo, así como su efectiva interpretación numérica para su implementación en un algoritmo genético.

2.1. Problemática

El núcleo del problema radica en la necesidad de planificar el horario académico del próximo cuatrimestre para un estudiante, basándose en su historial académico, el cual comprende un conjunto de materias que ha cursado o reprobado, teniendo en cuenta su progreso académico actual. Este desafío implica la evaluación de cada materia según su umbral de relevancia 3.3.1 y siempre evitando el solapamientos de horarios entre las disponibles por grupo. El funcionamiento general del algoritmo puede esquematizarse de la siguiente manera:

Funcionamiento

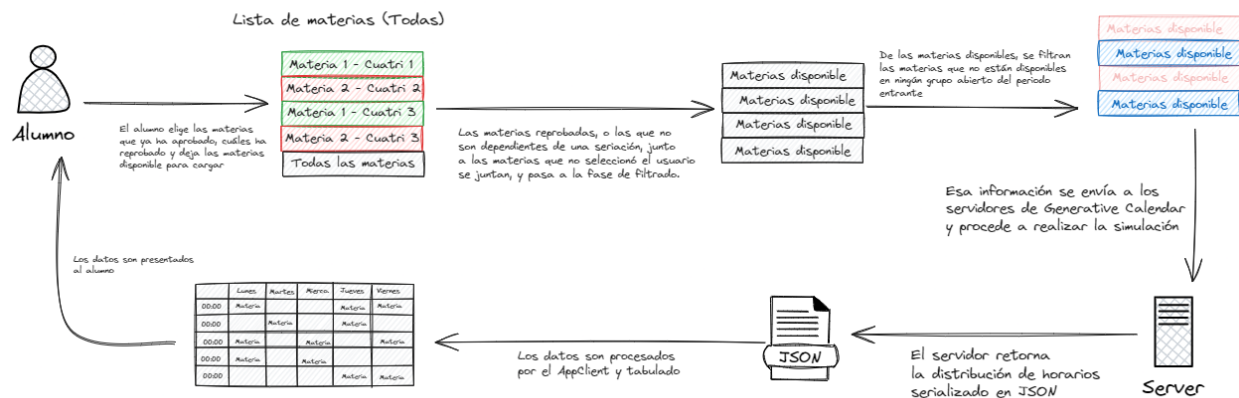


Figura 1: Diagrama del funcionamiento general del algoritmo.

3. Resolución de problemas

En esta sección se describe el enfoque adoptado para abordar el problema y alcanzar los objetivos del programa. La resolución del problema se divide en los siguientes puntos, que se detallarán a través de las diferentes subsecciones del documento.

1. Captura de progreso académico. 3.1
2. Filtrado de materias por seriación 3.2
3. Filtrado de materias por disponibilidad horaria. 3.2.3
4. Cuantificación de los datos. 3.3
5. Procesamiento de la información. 3.4

3.1. Captura de progreso académico.

La primera etapa del proceso del algoritmo se centra en la captura del progreso académico actual del alumno, es decir, las materias cursadas o reprobadas por el alumno. En esta fase, el alumno seleccionará manualmente las materias que ha aprobado y las que ha reprobado, dejando libres aquellas que aún no ha cursado. Esta acción proporciona al sistema un contexto claro sobre el estado y el progreso del alumno. Además, la implementación de la seriación de materias, facilitará el completado y filtrado de las materias

3.2. Selecciona tu estado actual

CUATRIMESTRE 1	MATERIA 1 MATERIA 2 MATERIA 3 MATERIA 4 MATERIA 5 MATERIA 6 MATERIA 7	CUATRIMESTRE 2	MATERIA 1 MATERIA 2 MATERIA 3 MATERIA 4 MATERIA 5 MATERIA 6 MATERIA 7	CUATRIMESTRE 3	MATERIA 1 MATERIA 2 MATERIA 3 MATERIA 4 MATERIA 5 MATERIA 6 MATERIA 7	CUATRIMESTRE 4	MATERIA 1 MATERIA 2 MATERIA 3 MATERIA 4 MATERIA 5 MATERIA 6 MATERIA 7
CUATRIMESTRE 5	MATERIA 1 MATERIA 2 MATERIA 3 MATERIA 4 MATERIA 5 MATERIA 6 MATERIA 7	CUATRIMESTRE 6	MATERIA 1 MATERIA 2 MATERIA 3 MATERIA 4 MATERIA 5 MATERIA 6 MATERIA 7	CUATRIMESTRE 7	MATERIA 1 MATERIA 2 MATERIA 3 MATERIA 4 MATERIA 5 MATERIA 6 MATERIA 7	CUATRIMESTRE 8	MATERIA 1 MATERIA 2 MATERIA 3 MATERIA 4 MATERIA 5 MATERIA 6 MATERIA 7
CUATRIMESTRE 9	MATERIA 1 MATERIA 2 MATERIA 3 MATERIA 4 MATERIA 5 MATERIA 6 MATERIA 7						

Figura 2: Captura de pantalla de la sección de selección de materias cursadas, obtenida del maquetado en Figma.

3.2. Filtrado de materias por seriación

El proceso de filtrado de materias tiene como objetivo eliminar aquellas materias que, debido a restricciones de seriación, no deben considerarse como materias aptas para carga. Estas materias, denominadas “materias aptas para carga”, son aquellas que cumplen con sus requerimiento de materias previas, y como consecuencia, pueden ser consideradas aptas para el procesamiento del algoritmo. Es importante destacar que el Algoritmo Genético (AG) procesa únicamente conjuntos de materias de forma individual, sin tener un contexto completo de la seriación o adeudos de materias del alumno, por lo que este paso es importante para generar un horario válido y un funcionamiento correcto.

Una vez que el usuario ha seleccionado las materias aprobadas, estas se guardan en un arreglo de materias M , donde se almacenan todas las materias iniciales N etiquetadas según la selección del alumno. De esta forma, el arreglo de materias M se define como:

$$M : \{N_1, N_2, \dots, N_i\} \quad (1)$$

Donde N_i es la materia alojada en la posición i del arreglo M , y se podría definir a N_i de la siguiente manera:

$$N_i : (id_i, name_i, status_i, [dependencias]_i, [dependientes]_i, cuatrimestre_i, peso_i) \quad (2)$$

Donde cada atributo de los objetos materia obedece las siguientes definiciones:

- id: es el identificador único de la materia conformado por caracteres alfabéticos de longitud 4.
- name: es el nombre común de la materia en cuestión.
- status: es el estado que ha elegido el alumno, que puede ser: aprobado, reprobado, o sin cursar.
- [dependencias]: es un arreglo de longitud n que contiene todos los identificadores de las materias que se deben cursar antes de poder cursar la materia actual.
- [dependientes]: es un arreglo de todas las materias que dependen de la materia actual, es decir, aquellas materias que se podrán cursar una vez que la materia actual haya sido aprobada.
- cuatrimestre: corresponde a un identificador numérico del 1 al 9, que indica el cuatrimestre regular en el que debería ser cursada dicha materia.
- peso: corresponde a un número entero mayor o igual a 0, que representa la cantidad de materias que dependen de la asignatura actual, ver sección 3.3.1.

3.2.1. Seriación

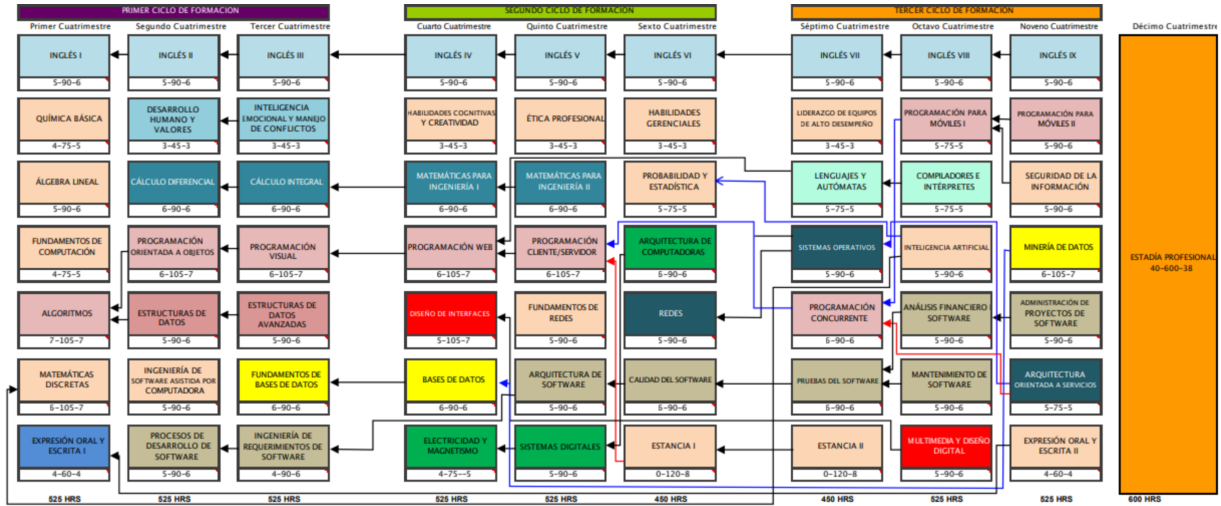


Figura 3: Mapa curricular de la carrera de Ingeniería en Desarrollo de Software del año 2017.

Es importante destacar que las materias que se maneja internamente en el programa, tienen una lista de materias dependientes y una de materias de dependencias. Esta característica se debe a la seriación, la cual dicta qué materias se podrán cursar en función del progreso del alumno.

La seriación, como es comúnmente conocida, se ajusta al mapa curricular de la carrera para la cual se desarrolló el algoritmo. En el caso actual, dicho mapa curricular corresponde a la carrera de **Ingeniería en Desarrollo de Software (IDS)**.

El algoritmo se basó en el mapa curricular de IDS del año 2017. Como se puede observar en la figura 3, existen materias que solo se podrán cursar una vez que todas las materias requeridas sean aprobadas.

3.2.2. Depuración de materias

Es fundamental tener en cuenta que existen materias que no pueden ser cursadas hasta que se aprueben una o más asignaturas de su línea de seriación. Por lo tanto, se debe eliminar de la lista de materias todas aquellas asignaturas que, según el caso de cada alumno, no sean aptas para ser cargadas debido a motivos de reprobación o porque su esquema de seriación está incompleto.

Para ilustrar este concepto, consideremos un ejemplo y simplifiquemos el mapa curricular como se muestra en la figura 4:

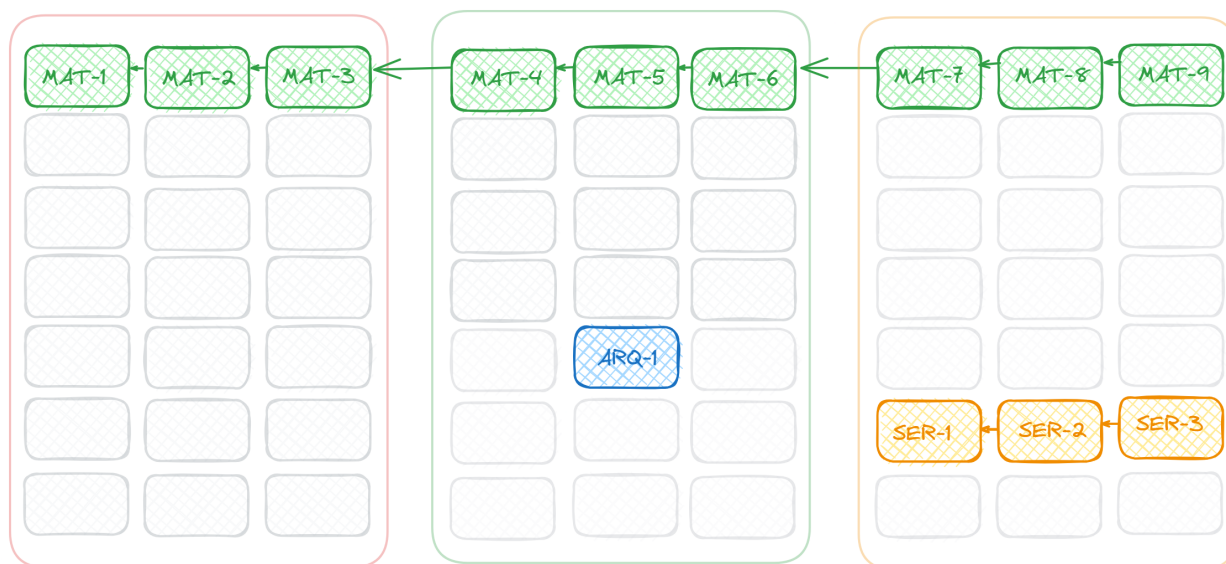


Figura 4: Mapa curricular simplificado de ejemplo: Caso 1.

Teniendo un mapa curricular como el de la figura 4, cada materia tiene una lista de asignaturas que dependen de ella. Por ejemplo:

- MAT-7 tiene una lista de materias dependientes donde se hallan MAT-8 y MAT-9.
- MAT-6 tiene una lista de materias dependientes donde se hallan MAT-7, MAT-8 y MAT-9.

Cuando una materia, por ejemplo ARQ-1, no tiene ninguna asignatura en su conjunto de materias dependientes, se le considera una materia débilmente seriada. Esto significa que ARQ-1 puede ser cursada sin tener ninguna otra materia como requisito previo. En otras palabras, no existen restricciones de seriación que limiten la capacidad de cursar ARQ-1 en función de otras asignaturas.

Ahora, si una materia es marcada como “reprobada”, todas las materias pertenecientes a la lista de materias dependientes de esa materia reprobada se marcarán como “deshabilitadas”. Este proceso se aplica recursivamente a todas las materias dependientes de las materias deshabilitadas. Es decir, si una materia directamente dependiente de la materia reprobada se deshabilita, entonces todas las materias que dependen de esa materia también se deshabilitarán. La materia originalmente marcada como reprobada, sin embargo, será apta para volver a cargarse en futuros intentos académicos.

De manera similar, si una materia es seleccionada por el usuario y se encuentra dentro de al menos un conjunto de materias marcadas como reprobadas o no cursadas, esta materia también será marcada como “deshabilitada”. Este proceso garantiza que no se incluyan materias que tengan dependencias no cumplidas o que hayan sido reprobadas previamente, garantizando así la coherencia del horario académico del estudiante.

Esta delimitación se puede representar mediante una serie de condiciones algebraicas que describen el proceso de manera precisa.

Se define M como una materia existente en el plan académico de la carrera, R como una materia marcada como reprobada o no cursada, P como el conjunto de materias con posibilidad de carga, y N como el conjunto de materias sin posibilidad de carga.

Entonces, estas condiciones pueden expresarse de la siguiente manera:

- Si M no pertenece al conjunto de dependientes de R y no está en el conjunto N , entonces M pertenece al conjunto P .
- Si M pertenece al conjunto de dependientes de R , entonces M se coloca en el conjunto N y se elimina de P .

Esto se puede expresar en términos algebraicos de la siguiente manera:

$$\begin{aligned} (M \notin R[\text{dependientes}]) \wedge (M \notin N) &\implies M \in P \\ M \in R[\text{dependientes}] &\implies (M \in N) \wedge (M \notin P) \end{aligned} \quad (3)$$

Esta expresión algebraica define de manera clara y precisa las condiciones bajo las cuales una materia puede ser considerada para su carga en función del estado de otras materias y sus dependencias.

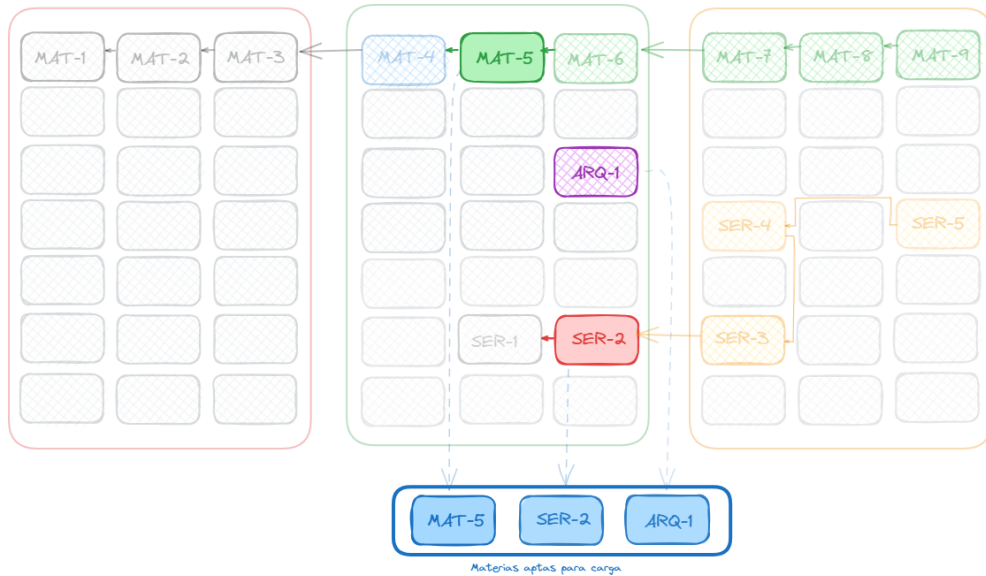


Figura 5: Mapa curricular simplificado de ejemplo: Caso 2.

Para ilustrar este proceso, supongamos que el usuario ha aprobado las asignaturas MAT-4, pero ha reprobado SER-2 y no ha cursado ARQ-1. En este caso, las únicas materias disponibles para ser cargadas

serán MAT-5, ya que es la continuación de MAT-4, SER-2, que es la materia que podrá volver a cursar en un nuevo intento académico, como se muestra en el mapa curricular simplificado de ejemplo en la figura 5, y ARQ-1, que fue marcado como no cargada, y podrá ser cursada en el curso entrante como primera oportunidad.

Estas materias son recopiladas dentro de un nuevo conjunto, que en el diagrama es ilustrado como “Materias aptas para carga”, por otro lado, la materia SER-5, aunque no sea directamente dependiente de SER-2, está dentro de la línea de seriación gracias de SER-4 que depende de SER-3 que a su vez depende de SER-2, lo que significa que todas esas materias listadas deberán ser deshabilitadas, de esta manera solo las asignaturas que tienen su lista dependientes aprobadas, pueden pasar a la siguiente etapa.

3.2.3. Filtrado de materias por disponibilidad horaria

Una vez que hemos definido las materias del alumno, seleccionando solo aquellas que se ajustan a su estado académico actual, es necesario realizar un filtrado adicional basado en la disponibilidad de las materias para el próximo curso. Este paso es fundamental para garantizar la eficacia del algoritmo. En cada cuatrimestre, se establecen grupos de acuerdo con el número de alumnos y la disponibilidad de maestros en la carrera. Para los propósitos prácticos de nuestro algoritmo, nos basamos en los grupos abiertos durante el periodo de Enero-Abril de 2024. Dado que los números de grupo varían según el curso, esto influye directamente en la disponibilidad de las materias para su carga. Por lo tanto, debemos realizar un filtro adicional en nuestra lista de materias aptas para cargar, eliminando aquellas que no estén disponibles en el curso entrante, específicamente, durante el periodo de Enero-Abril de 2024.

Esta información se comunica a los alumnos a través de una lista que contiene los calendarios de todos los grupos disponibles. Cada horario S contiene las materias M ubicadas en su franja horaria correspondiente para cada grupo. Para facilitar la gestión de datos, nuestro equipo ha convertido este archivo a un formato de serialización de datos, lo que permite un acceso más accesible para el equipo de desarrollo sin agregar complejidad a la lectura de los datos. Estos horarios se presentan como un conjunto C , que engloba todos los horarios S disponibles para el próximo periodo.

Cada horario S pertenece a un grupo S_{grupo} , que según la cantidad de alumnos, pueden ser enumerados alfabéticamente de la A a la Z y a un cuatrimestre $S_{\text{cuatrimestre}}$ en específico. Los horarios contienen una lista de asignaturas M que corresponden a las materias pertenecientes a ese cuatrimestre según el mapa curricular de la carrera (ver Figura 3). Cada materia perteneciente al horario S contiene atributos como M_{id} , que es un identificador único para identificar una asignatura entre todas las disponibles en el periodo; M_{nombre} , que corresponde al nombre común de la materia; $M_{\text{canonical}}$, que corresponde al identificador único de la materia sin procesar a la que hace referencia, definida en la sección 1; M_{maestro} , que hace referencia al nombre del docente que imparte la materia para el grupo y cuatrimestre en específico; y $M_{\text{[horas]}}$, a un arreglo de horas que representa a qué franja horaria de la semana ocupa dicha asignatura. Cada hora está representada siguiendo la definición de la sección 3.3.2.

En este sentido, podemos sintetizar cada parte de esta lista de la siguiente manera:

$$\begin{aligned} C &: ([S]) \\ S_i &: (\text{cuatrimestre}_i, \text{grupo}_i, [M]_i) \\ M_{ik} &: (\text{id}_{ik}, \text{nombre}_{ik}, \text{canonical}_{ik}, \text{maestro}_{ik}, [\text{horas}]_{ik}) \end{aligned}$$



Figura 6: Diagrama de materiales reales de carga.

Ahora, para obtener las materias reales para carga, debemos de tomar aquellas asignaturas disponibles de carga, que cumplen con la regla definida en la ecuación 3, y conservar únicamente aquellas materias que también estén presente dentro de las asignaturas de los horarios que están en la lista de grupos disponibles para el periodo próximo, como se representa en la figura 6.

La definición de la existencia de una materia M en el conjunto de materias reales de carga R se expresa como:

$$M \in R \iff M \in S, \text{ para algún } S \in G \quad (4)$$

Donde:

M es una materia.

R es el conjunto de materiales reales de carga.

S es una asignatura.

G es el conjunto de horarios disponibles en la lista de horarios disponibles L .

3.3. Cuantificación de los datos

Otra parte importante del proceso previo al procesamiento de las materias por el algoritmo genético, es la cuantificación de los datos, es decir, añadir valores numéricos a las materias con el fin de realizar operaciones algebraicas y definir de una manera más concreta si una materia es apta para el alumno.

Después de realizar las delimitaciones de las materias, como se detalló en las secciones 3.2.2 y 3.2.3, nos queda una lista de materias que son aptas para ser procesadas por el algoritmo genético. Sin embargo, es necesario agregar información adicional y eliminar información innecesaria para mejorar la eficiencia del programa y garantizar que los datos proporcionados al algoritmo genético sean óptimos para generar un horario ideal.

En este sentido, se eliminarán ciertas propiedades de las materias disponibles de carga, como “dependientes”, “dependencias”, “status”, que fueron utilizadas únicamente en la sección 3.2.2 para eliminar las materias con seriaciones incompletas.

Por otro lado, se introducen nuevas propiedades de la materia. Sin embargo, para los fines explicativos de este documento, nos centraremos únicamente en dos atributos principales: el Umbral de Relevancia (μ), también conocido internamente como peso, y un horarios H que describen que franja horaria de la semana

se lleva acabo la clase de la materia en cuestión. La definición de estas horas se especifican en la sección 3.3.2.

$$M : (\mu, [H]) \quad (5)$$

Estas materias pertenecen a un conjunto de materiales reales de carga D que se adjunta a otro conjunto de entrada E que se enviará al servidor del algoritmo genético para su procesamiento. Este conjunto E posee propiedades adicionales que hacen referencia a información adicional para el funcionamiento, como cuatrimestre actual del alumno y configuraciones técnicas para el algoritmo gnético.

Descrito explícitamente podríamos decir

$$\begin{aligned} D &: \{M_1, M_2, \dots, M_i\} \\ E &: (D, C) \end{aligned}$$

Este conjunto de datos son los que serán procesado por el algoritmo genético para generar el horario más óptimo para el alumno, abordemos cada uno de los campos a continuación.

3.3.1. Umbral de relevancia

El umbral de relevancia, también conocido como peso, es un número entero mayor a 0 asociado a cada materia. Este número indica la importancia de la materia y es utilizado por la función que determina el horario final del alumno.

El cálculo del umbral se rige por las siguientes reglas:

- Por cada materia dependiente, se añade 1 punto al umbral.
- Si la materia se encuentra en un cuatrimestre menor al cuatrimestre actual del alumno, se suma 1 punto al umbral por cada cuatrimestre de diferencia entre el último cuatrimestre actual y el cuatrimestre de la materia.

Para un mejor entendimiento, dividiremos el cálculo de este umbral en dos partes: calcular el peso de las dependientes y el cálculo de peso por distancia de cuatrimestre.

Para calcular el umbral de relevancia por dependientes, primero definimos el conjunto de materias dependientes como D , donde $|D|$ representa el número de materias dependientes de una materia. Entonces, el umbral de relevancia μ_r por dependientes se calcula como:

$$\mu_r = |D| + \sum_{d \in D} \mu_r \quad (6)$$

Donde:

μ_r es el umbral de relevancia por dependientes.

D es el conjunto de materias dependientes de una materia.

$|D|$ es el número de materias dependientes de una materia.

μ_r es el umbral de relevancia de cada materia dependiente.

Para la primera parte, que es el cálculo de peso por dependientes, el conjunto de materias se recorre de forma inversa, es decir, de mayor cuatrimestre a menor cuatrimestre. Por cada materia en dependientes, se suma 1 punto a su peso; por cada materia dependiente, se suma el peso de esa materia. Esto se representa visualmente en la Figura 7.

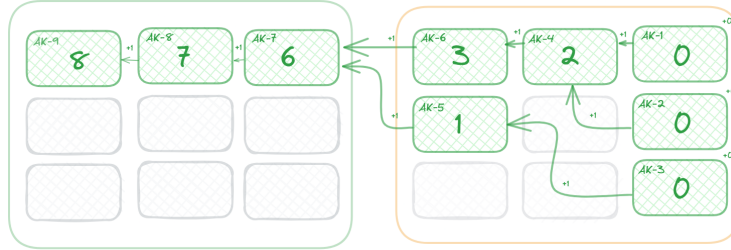


Figura 7: Ejemplo ilustrativo de cálculo de relevancia por dependiente.

En la ilustración 7, se muestra el proceso de cálculo de las materias. El peso de las materias de AK-1 al AK-3 es 0, ya que al ser la última materia del plan curricular, no tiene materias que dependan de ella.

Podemos ver que la materia AK-4 tiene un peso de 2. Esto resulta de tener dos materias que dependen de ella: AK-1 y AK-2, sumando 1 punto por cada una. Además, se suma el valor de dichas dependientes, que son 0. Al final, el peso de la materia mostrada será de 2.

Tomemos un ejemplo más complejo, como el de la materia AK-7, que tiene dos materias dependientes: AK-6 y AK-5. Ambas suman 1 punto al peso de AK-7, teniendo un peso inicial de 2. Ahora debemos sumar el peso de cada dependiente a la materia AK-7, que para AK-6 sería 3 y para AK-5 sería 1. Teniendo estos valores, tendríamos que el valor total del peso de AK-7 es 6, ya que $3 + 1 + 1 + 1 = 6$.

De esta forma, nos aseguramos de que las materias con la mayor cantidad de dependientes tengan prioridad al generar el horario, en comparación con las materias con pocas dependientes.

Es importante realizar este cálculo antes que todo, de esta forma evitamos tomar en cuenta pesos que no corresponden exclusivamente a la cantidad de materias dependientes de cada asignatura.

El umbral de relevancia por distancia se calcula como la diferencia entre el cuatrimestre de la materia (M) y el cuatrimestre del alumno (C), siempre y cuando M sea mayor que C . De lo contrario, no se suma nada y el umbral es igual a 0.

$$\mu_d = \begin{cases} M - C, & \text{si } M > C \\ 0, & \text{si } M \leq C \end{cases} \quad (7)$$

Donde:

M : Cuatrimestre de la materia.

C : Cuatrimestre del alumno.

μ_d : Umbral de relevancia por distancia.

Para el cálculo de relevancia por distancia, usamos una operación un poco más simple. Sabiendo el cuatrimestre del alumno, a cada materia de cuatrimestre menor al cuatrimestre del alumno, se le va a sumar un punto por cada materia de distancia que tenga al cuatrimestre actual del usuario.

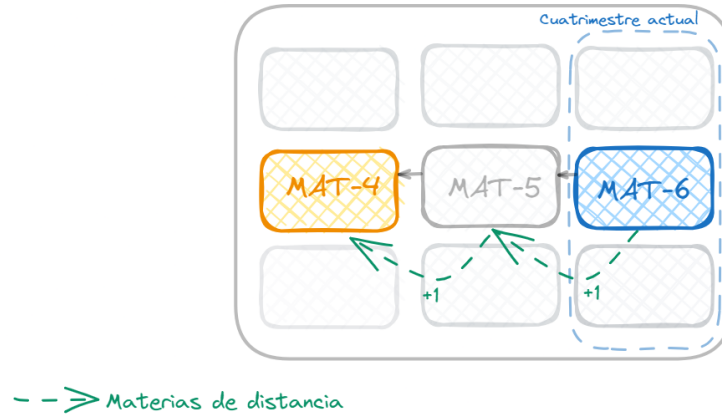


Figura 8: Ejemplo ilustrativo de cálculo de relevancia por distancia.

Por ejemplo, consideremos el caso del umbral de la materia MAT-4 de la Figura 8 para un alumno en sexto semestre, mientras que el cuatrimestre donde se halla la materia MAT-4 está a 2 cuatrimestres de distancia. Por cada cuatrimestre de diferencia entre el cuatrimestre actual del alumno y el cuatrimestre de la materia, se suma un punto adicional al umbral de relevancia de la materia. Por lo tanto, se añadirán 2 puntos al umbral de MAT-4.

Adicionalmente, después de todos los cálculos realizados, se sumará 1 punto adicional a cada umbral para evitar tener materias con 0 puntos y mejorar la eficiencia del cálculo de nuestro algoritmo genético.

Finalmente, nuestro umbral de relevancia es el resultado de la suma de estos dos umbrales más 1, como se muestra a continuación.

$$\mu = \mu_r + \mu_d + 1 \quad (8)$$

3.3.2. Franja horaria

La franja horaria es una forma numérica de definir que horas de la semana se imparte una clase, esto se puede definir a través de números enteros entre el 1 y el 45, cada número representa una hora entre las 8:00hrs. a las 17:00hrs. de algún día entre lunes a viernes como se muestra en la ilustración 9.

	Lunes	Martes	Miércoles	Jueves	Viernes
8:00-9:00	1	10	19	28	37
9:00-10:00	2	11	20	29	38
10:00-11:00	3	12	21	30	39
11:00-12:00	4	13	22	31	40
12:00-13:00	5	14	23	32	41
13:00-14:00	6	15	24	33	42
14:00-15:00	7	16	25	34	43
15:00-16:00	8	17	26	35	44
16:00-17:00	9	18	27	36	45

Figura 9: Representación de los valores de la franja horaria.

De esta forma, una materia que tenga un franja horaria con los siguientes datos [2, 3, 10, 26, 27, 38, 39, 40] tendría una distribución visual en el horario como se muestra en la figura 10

	Lunes	Martes	Miércoles	Jueves	Viernes
8:00-9:00					
9:00-10:00					
10:00-11:00					
11:00-12:00					
12:00-13:00					
13:00-14:00					
14:00-15:00					
15:00-16:00					

Figura 10: Franja horaria: Ejemplo 1.

3.4. Procesamiento de la información

Ahora, hablaremos sobre cómo nuestro algoritmo procesó a las materias reales para carga (ver definición 4), para esto es indispensable definir 3 nuevos conceptos que nos ayudó a tener un registro de aquellos horarios que cumplen con todos los criterios para poder ser cursado, y evaluar la mejor alternativa, estos conceptos son: semilla, longitud de semilla y base aritmética, que se explica a profundidad en las secciones 3.4.1, 3.4.2 y 3.4.3 respectivamente.

3.4.1. Semilla

La semilla corresponde a un identificador único para cada nuevo horario o individuo generado, consta de una longitud de caracteres definido en la longitud de semilla de la sección 3.4.2, y sirve para poder replicar

un individuo en cualquier parte del tiempo y bajo el mismo contexto de materias. Esto permite replicar de forma seguro e inequívoca a aquellos horarios pertinentes para visualización.

3.4.2. Longitud de semilla

La longitud de semilla corresponde a la cantidad de caracteres que tendrá una semilla, esto es importante para hacer que las semillas generadas cubran por completo todas las posibles combinaciones de horarios y evitando dejar generaciones imposibles de replicar o semillas tan largas que abarcan generaciones fuera del rango de posibilidades.

Este número se genera tomando en cuenta la cantidad de materias reales de carga (ver visualización 6), es decir, si en una instancia del algoritmo genético hay 7 materias reales de carga, la longitud de los caracteres de la semilla será de 7. El valor de los caracteres varían según lo definido en la sección 3.4.3.

3.4.3. Base Aritmética

La base aritmética representa el rango de incremento de cada carácter en la semilla, es decir, una base aritmética de 2, también conocido como binario, los valores en el cual el carácter puede incrementar son dos posibilidades: 0 y 1.

Este valor se define tomando el valor máximo de grupos abiertos para una materias reales disponibles existente en la lista de materiales reales de carga que se envía a través de los datos de entrada definido en la ecuación 3.3.

Para ilustrar estos conceptos de una forma más clara, tomemos la ilustración 11 de ejemplo.

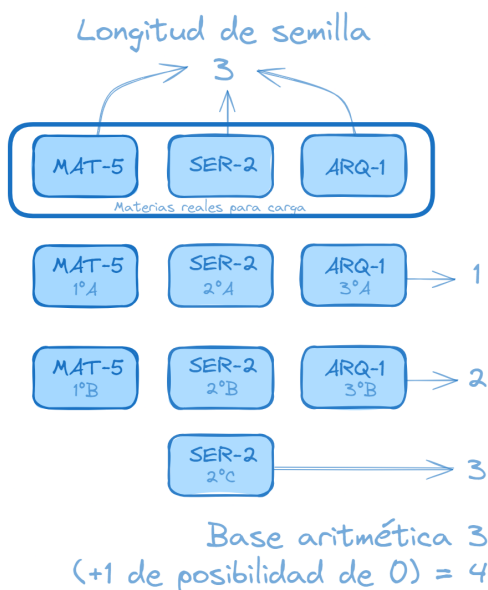


Figura 11: Visualización de longitud de semilla y base aritmética.

En la ilustración anterior 11 podemos ver como para ese caso, la base aritmética es 4, y la longitud de semilla es 3.

La longitud, como se define en la sección 3.4.2, es el valor de las materias reales para carga de esa instancia del algoritmo genético, que en este caso son: MAT-5, SER-2 y ARQ-1, dando como resultado una longitud de semilla de 3.

Para la base aritmética, se considera la mayor cantidad de grupos disponibles (ver 3.2.3) para alguna materia existente en la lista de materias reales de carga en la instancia del algoritmo genético actual. Para el ejemplo anterior, esto es 3, debido a que SER-2 tiene tres grupos disponibles para cursar dicha materia, que son: 2°A, 2°B y 2°C. A esta longitud, debemos incrementar una base más, para que pueda interpretarse el 0 como un dígito válido para la semilla y tener una buena interpretación de la semilla como se define en la sección 3.4.4.

3.4.4. Decodificación de semilla

La decodificación de semilla es la parte del algoritmo en la que dada una semilla, y ya definido su longitud y base aritmética, procedemos a decodificar a qué materias hace referencia dicha semilla, para ilustrar este paso tomemos la figura 12.

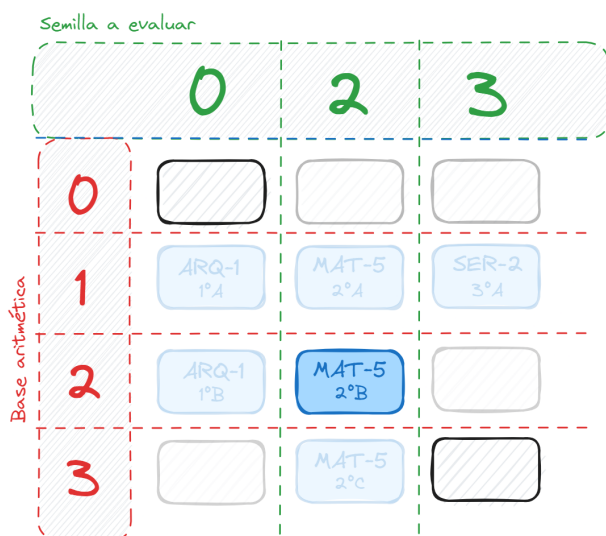


Figura 12: Visualización de decodificación de semilla.

En el ejemplo anterior, la semilla tiene una longitud de 3, y una base aritmética de 4, esto lo calculamos previamente como lo explicamos en las secciones 3.4.2 y 3.4.3, lo que nos indica que las semillas posibles oscilan entre "000" y "333".

El siguiente paso es ordenar las materias por su id canonical (ver sección 3.2.3) de forma alfabética, este id, al ser único entre todas las materias, y al ordenarlo alfabéticamente nos permite tener una mayor consistencia al momento de codificar las semillas bajo el mismo contexto de materias.

Posteriormente, asignamos cada carácter de la semilla a una materia, y el valor corresponde a la alternativa o grupo de la materia en cuestión, estas materias seleccionadas se almacenan dentro de una lista llamada **lista de materias activas**, que es usada para saber si el individuo está con colisión o no (ver sección 3.4.5).

Si el valor del carácter es 0, la materia no se añade a la lista de materias activas, como es el caso de la materia ARQ-1 del ejemplo de la figura 12, por otro lado si el valor excede a la cantidad de posibilidades reales de la materia, esta materia tampoco es añadida a la lista de materias activas, como es el caso de la materia SER-2 del ejemplo de la figura 12.

3.4.5. Individuos

Dentro del algoritmo genético implementado en la aplicación, conocemos como individuo a aquel objeto capaz de interpretar una semilla (ver sección 3.4.1), dado un contexto específico, es decir, una lista de materias reales de carga.

Con dicha interpretación de la semilla, puede calcular sus pesos totales y definirse como un individuo con colisiones o sin colisiones según los siguientes parámetros.

Un individuo es considerado **sin colisión** si, según la decodificación de la semilla (ver sección 3.4.4), ninguna hora de la franja horaria de cualquier materia perteneciente a la lista de materias activas se encuentra dentro de otra franja horaria de otra materia de la misma lista de materias activas. Matemáticamente, esto se expresa como:

$$\text{sin_colisiones}(i, j) = \begin{cases} \text{verdadero,} & \text{si no hay colisión entre las franjas horarias de las materias } i \text{ y } j \\ \text{falso,} & \text{en caso contrario} \end{cases} \quad (9)$$

donde $\text{no_hay_colisión}(i, j)$ es una función que retorna verdadero si no hay colisión entre las franjas horarias de las materias i y j , y falso en caso contrario.

En caso contrario, se considera al individuo como **con colisiones**, y no es usado como posible horario apto para el alumno.

Independientemente del estado del individuo, se calcula un **peso total** para cada individuo calculado, que es la suma de todos los pesos de todas las materias existentes en la lista de materias activas. Matemáticamente, esto se expresa como:

$$\text{peso_total} = \sum_{i \in \text{materias_activas}} \text{peso}(i) \quad (10)$$

donde $\text{peso}(i)$ es el peso de la materia i .

4. Resultado y salida

Tras finalizar todas las generaciones del algoritmo genético y evaluar a cada individuo generado en cada iteración, regresamos al usuario al usuario el horario del individuo sin colisión con mayor peso (o umbral de relevancia, ver sección 3.3.1). Para esto, decodificamos la semilla (ver sección 3.4.5) del individuo con el umbral más alto y se lo enviamos al usuario para que la aplicación web renderice el horario correspondiente (ver figura 15).

5. Metodología

En esta sección se presenta en detalle la metodología empleada para el desarrollo del algoritmo genético utilizado en la aplicación.

5.1. Emparejamiento

El proceso de emparejamiento es crucial en los algoritmos genéticos, ya que determina cómo se combinan los individuos para generar descendencia. En nuestro enfoque, utilizamos dos estrategias distintas para el emparejamiento: A2 y C2.

La estrategia A2 consiste en evaluar, para cada individuo de la población, si se cruzará con otro individuo basándose en un umbral de probabilidad de cruce (P_c). Si la evaluación es positiva, se selecciona aleatoriamente uno o más individuos como pareja para el cruce.

Por otro lado, la estrategia C2 implica el uso de múltiples puntos de cruce para el intercambio de información genética entre los individuos emparejados. Para cada pareja a cruzar, se elige aleatoriamente la cantidad de puntos de cruce, y luego para cada punto de cruce se selecciona aleatoriamente la posición en la secuencia genética (semilla) donde se realizará el intercambio.

La combinación de estas estrategias nos permite explorar de manera eficiente el espacio de búsqueda de soluciones al permitir una variedad de intercambios genéticos entre los individuos emparejados.

5.2. Mutación

La mutación es un operador esencial en los algoritmos genéticos, ya que introduce diversidad en la población y permite explorar nuevas soluciones. En nuestro enfoque, adoptamos la estrategia M2 para la mutación. Esta estrategia implica el intercambio de posición de caracteres de la semilla de un individuo (ver sección 3.4.1). En otras palabras, si un individuo sufre una mutación, se elige aleatoriamente una posición en su semilla y se intercambia el carácter en esa posición con otro carácter seleccionado aleatoriamente.

La estrategia M2 se seleccionó debido a su capacidad para modificar las soluciones existentes de manera no determinista, lo que puede ser beneficioso para explorar el espacio de búsqueda de manera más amplia.

5.3. Poda

La poda de la población es un paso importante para controlar el tamaño y la diversidad de la población en cada iteración del algoritmo genético, así como mejorar el rendimiento del algoritmo. En nuestro enfoque, implementamos la estrategia P2 para la poda. Esta estrategia implica una eliminación aleatoria de individuos de la población, asegurando que al menos el mejor individuo de la población se conserve en cada iteración.

La elección de esta estrategia se basa en su capacidad para mantener una población diversa y promover la convergencia hacia soluciones óptimas al conservar los individuos más prometedores.

6. Visualización y gráficos

En esta sección, a forma de galería de imágenes, mostramos algunas de las interfaces implementadas para el aplicativo y algunas de las técnicas usada para hacer una experiencia de usuario agradable.

6.1. Progreso actual

Esta sección es donde el usuario selecciona que materias ha aprobado, reprobado y se ve visualmente qué materias son inhabilitadas por no cumplir su seriación, y cuales son las que pondrán ser cargadas.

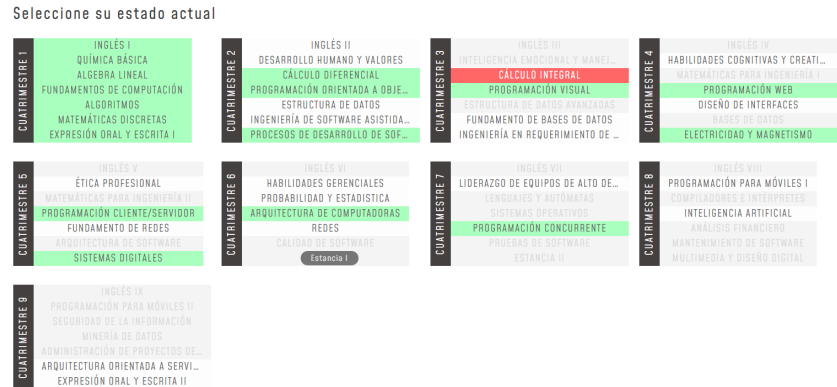


Figura 13: Sección de selección de estado actual del usuario.

A pesar de ser muy similar a la figura 2, la figura 13 es la vista que está implementada en la versión final del programa.

6.2. Grupos disponibles

Como vimos en la sección 3.2.3, la disponibilidad de una materia depende de los grupos que se abren en un periodo determinado, para mostrar estos grupos disponibles, nuestra aplicación tiene un apartado donde podemos ver aquellos grupos abiertos para el periodo que usa el algoritmo genético.

4°A	Lunes	Martes	Miércoles	Jueves	Viernes
8:00 - 9:00	Matemáticas para Ing... Marcelo Álvarez Hernández		Diseño de Interfaces José Pablo Vázquez Cruz	Habilidades Cognitiva... Guillermo Salazar Arias	Matemáticas para Ing... Marcelo Álvarez Hernández
9:00 - 10:00	Matemáticas para Ing... Marcelo Álvarez Hernández	Electricidad y Magneti... Horacio Iran Solís Cisneros	Diseño de Interfaces José Pablo Vázquez Cruz	Habilidades Cognitiva... Guillermo Salazar Arias	Matemáticas para Ing... Marcelo Álvarez Hernández
10:00 - 11:00	Electricidad y Magneti... Horacio Iran Solís Cisneros	Electricidad y Magneti... Horacio Iran Solís Cisneros			Habilidades Cognitiva... Guillermo Salazar Arias
11:00 - 12:00		Matemáticas para Ing... Marcelo Álvarez Hernández		Programación Web Carlos Alberto Díaz Hernández	
12:00 - 13:00		Matemáticas para Ing... Marcelo Álvarez Hernández		Programación Web Carlos Alberto Díaz Hernández	Programación Web Carlos Alberto Díaz Hernández
13:00 - 14:00		Diseño de Interfaces José Pablo Vázquez Cruz	Programación Web Carlos Alberto Díaz Hernández	Programación Web Carlos Alberto Díaz Hernández	Programación Web Carlos Alberto Díaz Hernández
14:00 - 15:00	Bases de Datos José Alonso Macías Montoya	Diseño de Interfaces José Pablo Vázquez Cruz	Bases de Datos José Alonso Macías Montoya	Bases de Datos José Alonso Macías Montoya	Electricidad y Magneti... Horacio Iran Solís Cisneros
15:00 - 16:00	Bases de Datos José Alonso Macías Montoya	Diseño de Interfaces José Pablo Vázquez Cruz	Bases de Datos José Alonso Macías Montoya	Bases de Datos José Alonso Macías Montoya	Inglés IV Luis María Ventura
16:00 - 17:00		Inglés IV Luis María Ventura	Inglés IV Luis María Ventura	Inglés IV Luis María Ventura	Inglés IV Luis María Ventura

Figura 14: Horario del grupo disponible 4°A en el periodo Enero-Abril de 2024.

Hemos asignado colores para cada materia perteneciente a cada horario para identificar de una manera más sencilla la distribución de una materia.

6.3. Resultado

Tras finalizar el algoritmo genético, y obtener los resultados, la aplicación web se encarga de mostrar el horario correspondiente a la mejor opción y meta dato como detalles de la semilla.

A diferencia de los calendarios de la figura 14, en este horario de resultado se muestra el grado y grupo de la materia en cada celda horaria.

¡Listo! Este es el mejor horario para ti.

	Lunes	Martes	Miércoles	Jueves	Viernes
8:00 - 9:00		2°A Ingeniería de Softwar... Efraín Alejandro Macías Ca...	4°A Diseño de Interfaces José Pablo Vázquez Cruz	4°A Habilidades Cognitiva... Guillermo Salazar Arias	2°A Estructura de Datos Sirgei García Ballinas
9:00 - 10:00		2°A Ingeniería de Softwar... Efraín Alejandro Macías Ca...	4°A Diseño de Interfaces José Pablo Vázquez Cruz	4°A Habilidades Cognitiva... Guillermo Salazar Arias	2°A Estructura de Datos Sirgei García Ballinas
10:00 - 11:00				2°A Desarrollo Humano y V... Guillermo Salazar Arias	4°A Habilidades Cognitiva... Guillermo Salazar Arias
11:00 - 12:00					
12:00 - 13:00			2°A Estructura de Datos Sirgei García Ballinas		
13:00 - 14:00	2°A Estructura de Datos Sirgei García Ballinas	4°A Diseño de Interfaces José Pablo Vázquez Cruz	2°A Estructura de Datos Sirgei García Ballinas	2°A Ingeniería de Softwar... Efraín Alejandro Macías Ca...	2°A Desarrollo Humano y V... Guillermo Salazar Arias
14:00 - 15:00		4°A Diseño de Interfaces José Pablo Vázquez Cruz		2°A Ingeniería de Softwar... Efraín Alejandro Macías Ca...	2°A Desarrollo Humano y V... Guillermo Salazar Arias
15:00 - 16:00		4°A Diseño de Interfaces José Pablo Vázquez Cruz		2°A Ingeniería de Softwar... Efraín Alejandro Macías Ca...	2°A Inglés II Luis María Ventura
16:00 - 17:00		2°A Inglés II Luis María Ventura	2°A Inglés II Luis María Ventura	2°A Inglés II Luis María Ventura	2°A Inglés II Luis María Ventura

Detalles de tu población.
Alternativa 12,490,174 de 48,828,124

Valor: 48

Semilla: 11144141144

Base Aritmética(semilla): 5

Longitud de semilla: 11

Figura 15: Resultado para la generación de horario de 1000 generaciones de la selección de la figura 13.

6.4. Estadísticas

La parte estadística muestra la generación histórica a través de la n cantidad de generaciones definida por el usuario en opciones avanzadas.

Estas gráficas se muestran en la sección de resultado, justo de bajo del calendario óptimo (ver figura 15), y muestra de forma visual, el mejor caso, que corresponde al individuo de mayor peso de dicha generación; el peor caso, que corresponde al individuo de menor peso de dicha generación; y promedio, que corresponde al peso promedio de todos los individuos calculados en dicha generación (incluyendo al mejor y peor caso).

Estas gráficas son generadas para los individuos sin colisiones y para los individuos históricos (incluyendo los individuos con colisiones) (ver sección 3.4.5).

Para la gráfica de los individuos sin colisiones (figura 16), se preservan al mejor y al peor histórico, por lo que el mejor caso de la generación actual no es tomado en cuenta, al menos que supere al mejor caso

histórico.

Para la gráfica de los individuos con colisiones (figura 17), se gráficán únicamente los mejores y peores casos de la generación actual, esto por temas de rendimiento.

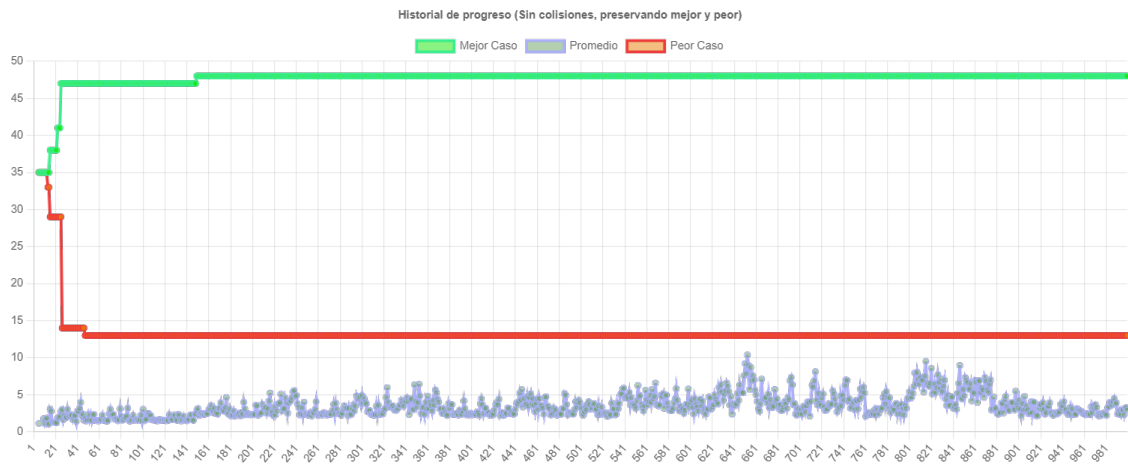


Figura 16: Estadísticas históricas sin colisión de la de la selección de la figura 13, después mil generaciones

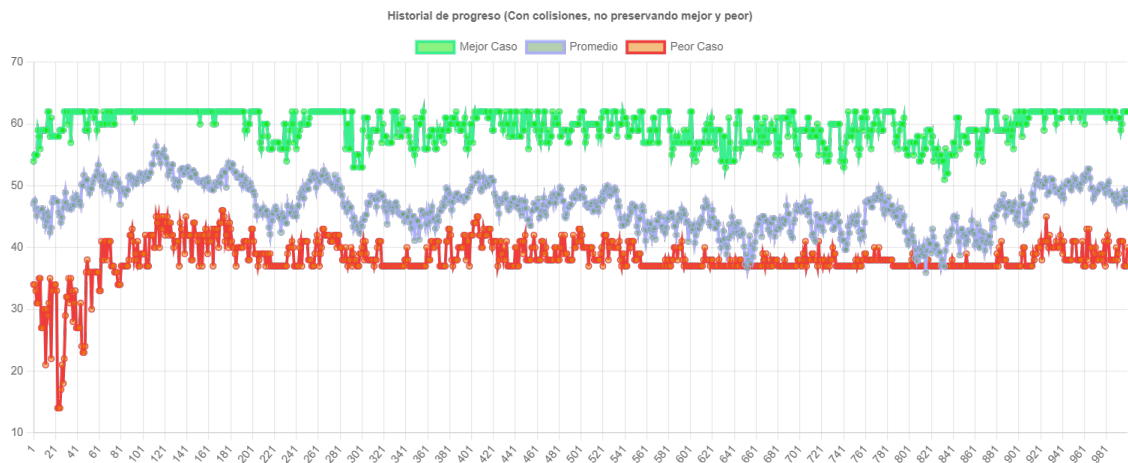


Figura 17: Estadísticas históricas con colisión de la de la selección de la figura 13, después mil generaciones

7. Herramientas

En esta sección se detallan las herramientas utilizadas en el desarrollo de la aplicación, abarcando tanto el frontend como el backend, así como las tecnologías empleadas para el despliegue del servicio.

Para la interfaz de usuario de la aplicación, se empleó React JS, un framework de JavaScript ampliamente utilizado para la construcción de interfaces de usuario interactivas y eficientes. Además, se utilizó Chart.js para la visualización de gráficos web, lo que permitió representar de manera clara y concisa la información generada por el algoritmo genético.

En el lado del servidor, se utilizó Flask, un framework de Python para el desarrollo de aplicaciones web, junto con Python para la implementación del algoritmo genético. Para exponer el servicio del algoritmo

genético, se optó por utilizar Waitress, un servidor web WSGI de Python que proporciona un equilibrio entre simplicidad y rendimiento, siendo adecuado para entornos de producción.

Además, para mejorar el rendimiento y tiempo de carga de la aplicación, se optó por implementar ciertas funcionalidades de manera nativa en Python, aprovechando su potencial en términos de procesamiento de datos y eficiencia.

Finalmente, para orquestar el lanzamiento del servicio en Waitress y el build de React servido con la imagen de Nginx, se empleó Docker Compose, una herramienta que permite definir y ejecutar aplicaciones Docker multi-contenedor de forma sencilla y eficiente. Esto facilitó la ejecución y la gestión de la aplicación en diferentes entornos de desarrollo y producción.

8. Conclusión

El algoritmo genético diseñado y presentado en este documento ofrece una solución innovadora y eficaz para abordar la problemática de la selección óptima de materias en un contexto de irregularidad académica. Al considerar tanto los grupos disponibles en el cuatrimestre próximo como la secuencia recomendada dentro del plan de estudios y el progreso académico del estudiante, esta herramienta de software proporciona una distribución más adecuada de las materias, facilitando así la toma de decisiones informadas por parte de los estudiantes.

La exposición detallada del análisis realizado, la metodología empleada y el desarrollo del software en este documento no solo brinda una comprensión clara del funcionamiento del algoritmo genético, sino que también destaca su aplicación práctica y su relevancia en el contexto educativo actual. Se espera que esta herramienta proporcione un apoyo valioso a los estudiantes y administradores educativos al optimizar el proceso de selección de materias y mejorar la eficiencia académica en general.