



# Universidad Nacional de La Matanza

Departamento de Ingeniería e Investigaciones Tecnológicas

## Trabajo Práctico: Aplicación de IA

Ciclo Lectivo: 2024

Cuatrimestre: 1er Cuatrimestre

Profesores:

Dr. Ierache, Jorge

Dr. Becerra Martín

Ing. Sanz Diego

Integrantes:

Becerra, Diego Ezequiel

Corrales, Mauro Exequiel - DNI: 40137650

Di Nicco, Luis Demetrio - DNI: 43664669

López Ferme, Nahuel Ezequiel - DNI: 43991086

Vivas, Pablo Ezequiel - DNI: 38703964

**Índice:**

**Dominio elegido:.....3**

**Problema identificado:..... 3**

**Casos de Uso:..... 3**

**1. Recomendar Cursada.....3**

**2. Calcular probabilidad de Éxito de una Cursada..... 4**

**Diseño de prototipo:..... 4**

**Esquema de pipeline diseñado:..... 4**

**Modelos Elegidos:..... 27**

**Datasets involucrados:..... 28**

**Test y evaluaciones de modelos:.....33**

**Link a colab realizado:..... 34**

**Aplicación:..... 34**

**Implementación..... 34**

**Presentación final:.....37**

**Hoja de ruta de presentación con los temas a ser abordados.....37**

**Capturas de presentación final realizada..... 38**

**Link a vídeo de presentación realizado..... 42**

**Referencias:.....42**

## **Dominio elegido:**

Nuestro proyecto se centrará en el ámbito educativo, con un enfoque específico en la Universidad Nacional de la Matanza (UNLaM). Esta institución académica, reconocida por su compromiso con la excelencia educativa y la innovación, servirá como el contexto principal para nuestro trabajo. La UNLaM, ubicada en San Justo, provincia de Buenos Aires, Argentina, es un centro educativo de renombre que ofrece una amplia gama de programas de grado y posgrado en diversas disciplinas. En este proyecto se hará foco en la carrera de grado Ingeniería en Informática.

Nuestro objetivo es colaborar con una porción importante de la comunidad universitaria, buscando proporcionar una cursada más acorde a la disponibilidad horaria y a la probabilidad de éxito de la misma del alumnado.

## **Problema identificado:**

El problema que hemos identificado es la dificultad que tienen muchos estudiantes para organizar su cursada al inicio del cuatrimestre. Por inexperiencia o falta de tiempo, muchas veces terminan eligiendo sus materias en forma poco consciente, lo que puede llevar a que no obtengan el éxito esperado.

Nuestro proyecto se centrará en la problemática de agilizar, organizar y optimizar la planificación de materias a cursar en la Universidad Nacional de la Matanza (UNLaM), específicamente de la carrera de Ingeniería en Informática.

Reconocemos la importancia de esta tarea para garantizar que los estudiantes puedan estructurar sus horarios de manera eficiente y maximizar su probabilidad de éxito académico.

El objetivo principal será desarrollar una aplicación con los conocimientos adquiridos en las materias de Inteligencia Artificial e Inteligencia Artificial Aplicada, que permita a los estudiantes de Ingeniería en Informática planificar sus horarios de manera más efectiva, teniendo en cuenta sus tiempos disponibles, si trabaja y la probabilidad de éxito en cada materia. Esto implica la creación de algoritmos que consideren diversos factores, como la disponibilidad horaria, si trabaja o no, la oferta de materia, entre otros.

## **Casos de Uso:**

### **1. Recomendar Cursada**

Descripción: El usuario ingresa la oferta de materias correspondiente al cuatrimestre actual, su historia académica, las materias pendientes de final que posee, su condición laboral y la cantidad de horas que tiene disponible para estudiar. El sistema genera una recomendación de materias en conjunto con el día y horario a cursar, en función a la probabilidad de éxito calculada y los datos proporcionados por el usuario, de forma tal que se ajuste lo más posible a la realidad del alumno para obtener una cursada eficiente.

Tareas identificadas:

- Validar que los parámetros sean correctos.
- Armar el mapa de las materias y sus correlativas de acuerdo al Plan 2023 de la carrera Ingeniería en Informática.
- Procesar los archivos subidos por el usuario “Historia Académica” y “Oferta Académica”.
- Procesar los datos ingresados por el usuario.
- Calcular la situación académica del alumno.
- Aplicar algoritmos genéticos para la generación y selección de cursadas.
- Estimar probabilidad de éxito o fracaso de una cursada.
- Mostrar la recomendación de cursada al usuario.

## 2. Calcular probabilidad de Éxito de una Cursada

Descripción: El usuario ingresa la cursada que le gustaría realizar, su condición laboral y la cantidad de horas que tiene disponible para estudiar. El sistema indica la probabilidad de éxito de la cursada ingresada en función de los datos proporcionados por el usuario.

### Tareas identificadas:

- Validar que los parámetros sean correctos
- Procesar los datos ingresados por el usuario.
- Calcular la situación académica del alumno.
- Estimar probabilidad de éxito o fracaso de la cursada ingresada por el usuario.
- Mostrar la probabilidad de éxito de la Cursada al usuario.

## Diseño de prototipo:

### **Esquema de pipeline diseñado:**

Diseñamos un esquema de pipeline compuesto por 7 partes o tareas principales. A continuación, explicaremos y justificaremos detalladamente qué es lo que se realiza en cada una de ellas.

- Parte 0) Descarga e Importación de Librerías.

Al principio del archivo de google colab, realizamos la descarga e importación de todas las librerías que van a ser usadas durante el desarrollo de todo el prototipo.

- Parte 1) Procesamiento del Plan de Carrera.

En esta primera parte, creamos una estructura para procesar el Plan 2023 de la Carrera de Ingeniería Informática, incluyendo algunos datos adicionales como la dificultad de cada materia y las horas de clase, estudio y práctica que requieren.

Por cada materia completamos la siguiente información:

- Código de Materia (campo clave).
- Nombre de la Materia.
- Dificultad de la Materia.

- Horas semanales de clase (virtuales y/o presenciales).
- Horas semanales que requieren destinadas al estudio.
- Horas semanales que requieren destinadas a la práctica
- A qué rama pertenece.
- A qué año pertenece.

Algunos de los datos que incluimos fueron tomados directamente de la página oficial del Departamento de Ingeniería e Investigación Tecnológicas (DIIT) de la universidad (<https://ingenieria.unlam.edu.ar/index.php?seccion=3&idArticulo=565>), tales como: el código de materia, el nombre de la materia, las horas semanales de clase, a qué rama pertenece y el año al que pertenece.

También, incluimos información adicional sobre cada materia. Los datos relativos a la dificultad y las horas semanales que requieren, tanto para el estudio como para la práctica, fueron calculados arbitrariamente en base a nuestras experiencias personales y tomando en cuenta diversas opiniones de otros alumnos de la facultad. A continuación explicaremos que representan estos 3 valores.:

- *Dificultad*: Es un valor que representa la dificultad de promocionar la materia. Para calcular este valor, tomamos en cuenta la complejidad, cantidad y profundidad de los temas abordados, el conocimiento previo con el que los alumnos llegan en base a las materias correlativas anteriores, la complejidad de los parciales, los métodos de enseñanza y evaluación, y la cantidad de alumnos recursantes que suele haber en cada comisión.

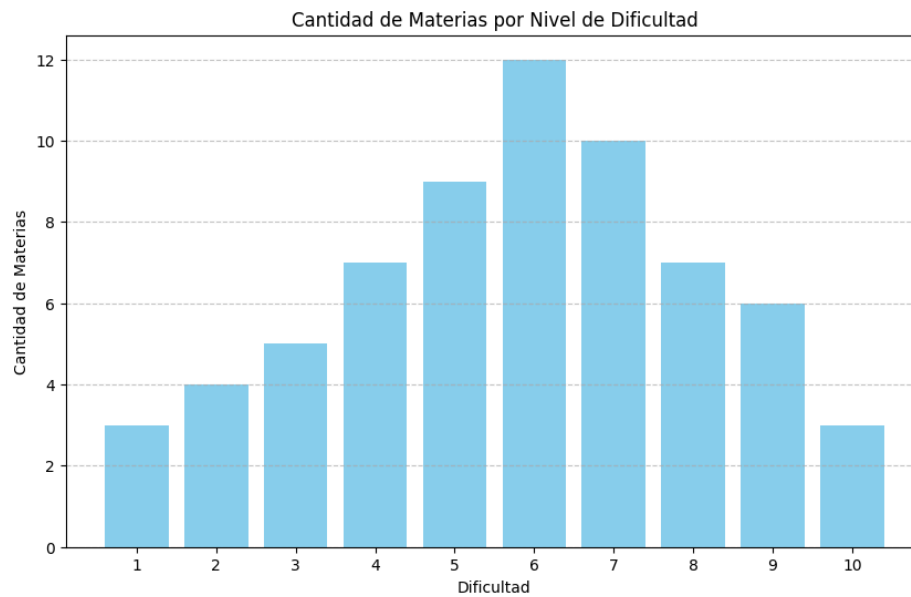
La dificultad es un valor entero positivo que varía entre 1 y 10.

Una dificultad de 1 representa una materia fácil de promocionar mientras que una materia con una dificultad de 10 representa una materia difícil de promocionar que requiere un esfuerzo adicional por parte del alumno.

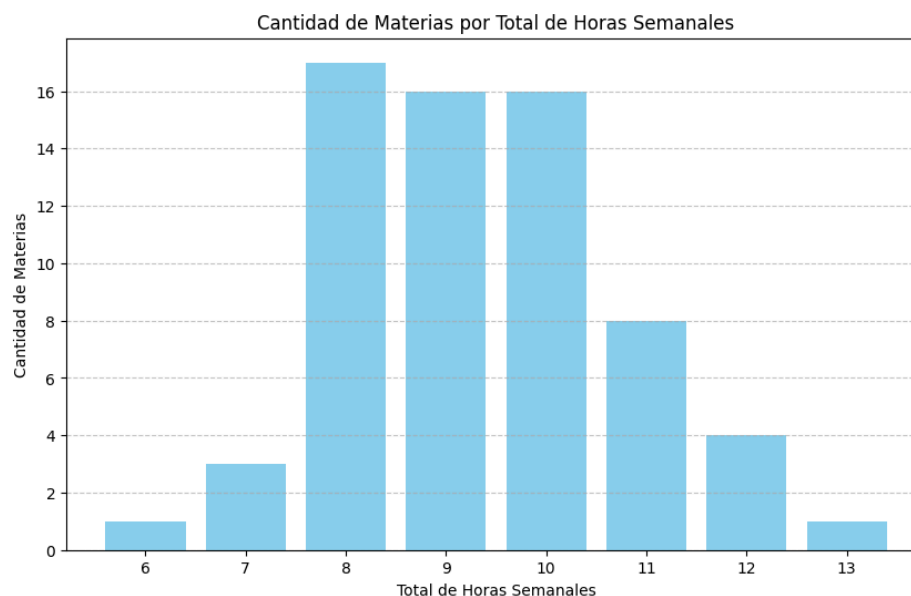
- *Horas semanales que se requieren para el estudio*: Incluye el tiempo por semana que requiere leer y estudiar todo el material bibliográfico de la materia. También tiene en cuenta el tiempo dedicado a la investigación de contenidos en internet.
- *Horas semanales que se requieren para la práctica*: Incluye el tiempo por semana que requiere resolver la guía práctica de ejercicios y/o los trabajos prácticos provistos por los docentes.

Estos 3 valores son estimativos y pueden variar según la opinión de cada persona. Consultamos la opinión de varios estudiantes para tener un panorama más general y que estas métricas no estén tan sesgadas en base a la experiencia personal de los miembros de nuestro grupo.

Como se puede apreciar en el siguiente gráfico, el promedio de dificultad de las materias varía entre cinco y siete. Una materia con una dificultad superior a siete se podría considerar “difícil de promocionar o que requiere un esfuerzo mayor promocionarla”, mientras que una materia con una dificultad menor a cuatro se podría considerar “fácil de promocionar o que requiere significativamente un menor esfuerzo poder promocionarla”.



Como se puede apreciar en este segundo gráfico, la mayoría de las materias requieren entre 8 y 10 horas de estudio semanales. Esto quiere decir, que estimamos que en promedio las materias requieren entre 1 y 1,5 horas destinadas al estudio y práctica por cada hora de clase sincrónica.



Una vez procesado el plan de la carrera con la información de cada materias, creamos un grafo de correlatividades para simplificar las operaciones correspondientes a la correlatividad entre materias.

- Parte 2) Procesamiento de la situación del usuario.

En la segunda parte, se procesa la información del usuario para hacerle una recomendación personalizada.

Los datos que se le solicitarán al usuario serán:

- Cuántas horas semanales va a dedicar a la cursada en el cuatrimestre. ( incluye el tiempo referido a las clases sincrónicas como también al tiempo de estudio.)
- Si trabaja actualmente
- Un archivo pdf con la oferta de materias.
- Un archivo pdf con su historia académica. (\*)
- Las materias que tiene pendiente de final.

(\*) Validamos que el usuario posea historia académica, ya que en el caso de ser ingresante o que no cuente con materias aprobadas en el plan 2023, no puede subir ningún archivo.

La recomendación de la cursada se hace tomando en cuenta estos 2 parámetros ingresados por el usuario:

- *¿Trabaja?:* Sirve para conocer las prioridades del usuario. Los alumnos que trabajan suelen priorizar su trabajo por sobre el estudio, o por lo menos, al contar con otras responsabilidades no pueden dedicar el tiempo que les gustaría para avanzar con la carrera. Por lo tanto, para garantizar una buena recomendación, se debe considerar este aspecto. Las recomendaciones para las personas que trabajan cuentan con menor carga horaria y con materias de una dificultad baja o media.

En contraposición, las personas que no trabajan suelen estar más centradas en el estudio, por lo tanto, las recomendaciones para este tipo de alumnos van a maximizar el aprovechamiento de la disponibilidad horaria ingresada por el usuario y pueden incluir algunas materias de complejidad alta.

- *Cantidad de horas semanales que desea dedicar al estudio:* Sirve para conocer la disponibilidad horaria del usuario. La recomendación se diseña de tal forma que se aproveche al máximo esta cantidad de horas semanales.

Gracias a la información estimada en la parte 1) de la cantidad de horas requeridas por cada materia, se compara la cantidad de horas ingresadas por el usuario con la estimación de las horas requeridas de la cursada. Se intenta que en la recomendación final estos dos valores queden lo más parecido posibles para evitar desperdicio o escasez de horas de estudio.

Luego solicitamos al usuario que suba dos archivos:

- El primero es la “Oferta de Materias” correspondiente al cuatrimestre que desea recibir la recomendación. Decidimos hacerlo de esta manera ya que la oferta de materias como tal solo está disponible algunos días previo a la inscripción, por lo tanto, queríamos asegurar que la aplicación funcione independiente del momento de la inscripción.

Este archivo se usará para conocer los días y horarios de las materias ofertadas. Y en base a esos horarios, armar las recomendaciones.

- El segundo archivo es la “Historia Académica” del usuario. Primero, se valida que el usuario cuente con materias aprobadas en el plan de Ingeniería Informática 2023, ya que no tendría sentido que el usuario tenga que subir un archivo vacío. Por otro lado, en caso de que el usuario cuente con una historia

académica, decidimos que lo más práctico es que suba directamente un PDF, en vez de tener que cargar una por una todas sus materias aprobadas.

Este archivo se usará para conocer las materias aprobadas que tiene el usuario y cuales son las que está en condiciones de cursar.

Ambos archivos deben tener un formato y nombre específico.

El archivo correspondiente a la *Oferta de Materias* debe llamarse "OfertaMaterias". Además, se debe seleccionar un tamaño de hoja que permita ver la totalidad de la tabla (tamaño de hoja: "A2", "A1", "A0").

El archivo correspondiente a la *Historia Académica* debe llamarse "HistoriaAcademica". Además, se debe seleccionar un tamaño de hoja que permita ver la totalidad de la tabla (el tamaño de hoja por defecto "A4" es suficiente).

Finalmente, se solicita al usuario que ingrese los códigos de materia que tiene pendiente de final (en caso que tenga algún final pendiente). Decidimos realizarlo de esta forma ya que consideramos que serán pocos códigos de materias y es mucho más sencillo y rápido que descargar el archivo PDF de finales pendientes.

Para simplificar el problema, vamos a considerar que una materia pendiente de final está "aprobada" y la incluiremos con las materias promocionadas.

El recomendador no hace distinción entre las materias que el usuario puede promocionar y las materias que solamente puede cursar (ya que adeuda correlativas por el final pendiente). Consideramos que ponderar de una mejor manera a las materias que puede promocionar por sobre las que solo puede cursar sería agregarle una complejidad innecesaria al proyecto y que escapa del alcance de nuestra aplicación.

Para el desarrollo de las siguientes partes asumiremos que el usuario, en caso de tener finales pendientes, los va a aprobar durante la cursada y por lo tanto, va a estar en condiciones de promocionar todas las materias que le recomendamos.

- Parte 3) Creación del Algoritmo Genético para realizar la recomendación.

En la tercera parte, definimos la estructura del algoritmo genético principal para realizar la recomendación de la cursada. Debido a la naturaleza de nuestro problema, el cual es de búsqueda y optimización, es decir, la búsqueda entre las distintas cursadas posibles con el fin de obtener la cursada más óptima para el usuario, consideramos que una buena forma de resolver este problema sería a través de algoritmos genéticos.

El algoritmo genético definido en esta parte se compone de 6 pasos:

- 1. Representación del Estado del problema / Codificación de los individuos.**

Problema: Determinar la cursada más óptima en base a las características propias de la cursada y los datos del usuario.

Características a representar:

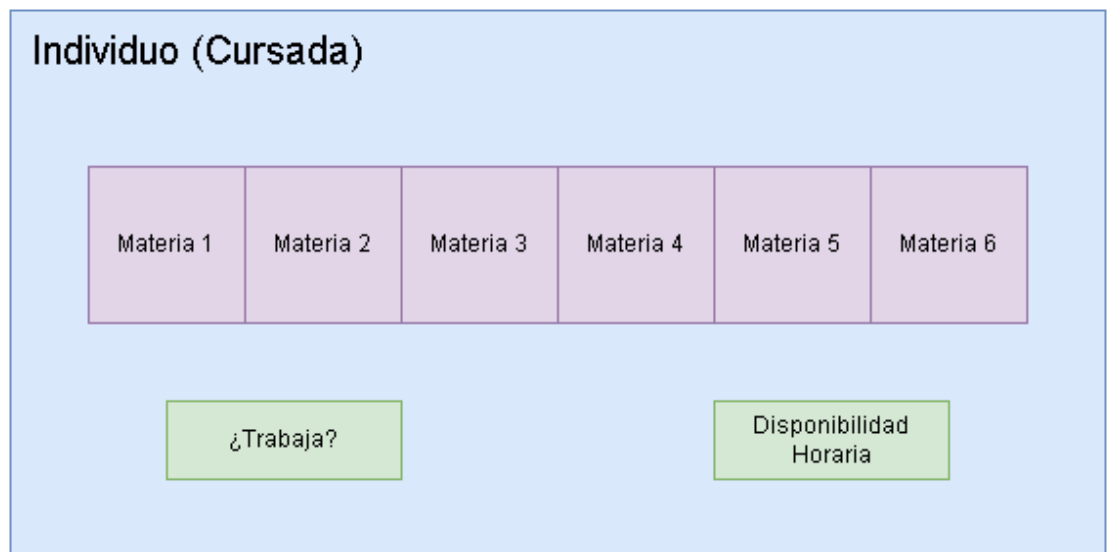
- Dificultad de la Cursada.
- Cantidad de Horas de Clase.



- Cantidad de Horas de Estudio.
- Cantidad de Horas de Práctica.
- Año de cada materia.
- ¿Usuario Trabaja?
- Disponibilidad Horaria del Usuario.

Los individuos serán las cursadas, es decir, estarán conformados por un conjunto de materias. Todas las cursadas incluirán la misma información ingresada por el usuario (si trabaja y las horas que va a dedicar al estudio).

El cromosoma que definimos tiene la siguiente estructura:

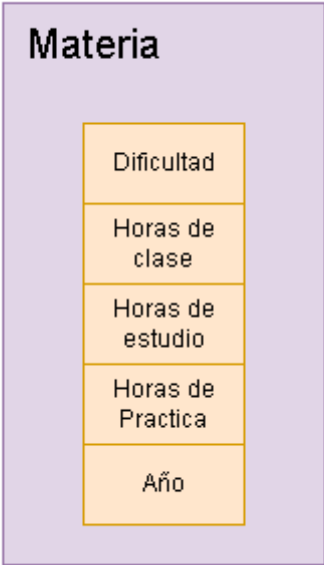


**Aclaración:** Las recomendaciones que generemos tendrán como máximo 6 materias, así como también para el caso de uso "Calcular probabilidad de Éxito de una Cursada" sólo podrán ingresar cursadas de hasta 6 materias (1 materia por día en promedio). Decidimos definir la red neuronal de tal forma que permita como máximo la entrada de 6 materias, ya que el plan de la carrera tiene en cuenta cursadas de hasta 6 materias por cuatrimestre. Además, consideramos que cursadas de 7 o más materias no son realistas y la probabilidad de Éxito es tan pequeña que no merece la pena tenerlas en consideración.

Cada una de las materias posee:

- Dificultad (valor entero entre 1 y 10)
- Horas semanales de clase (valor entero positivo, debido a las características del plan todas las materias son de 4 horas semanales)
- Horas semanales de estudio (valor entero positivo, entre 0 y 168\*)
- Horas semanales de práctica (valor entero positivo, entre 0 y 168\*)
- Año (valor entre 1 a 5. Puede ser T en caso de una materia transversal)

(\*) Aclaración: 168 es la cantidad máxima de horas que hay en una semana (24 \* 7 = 168 horas)

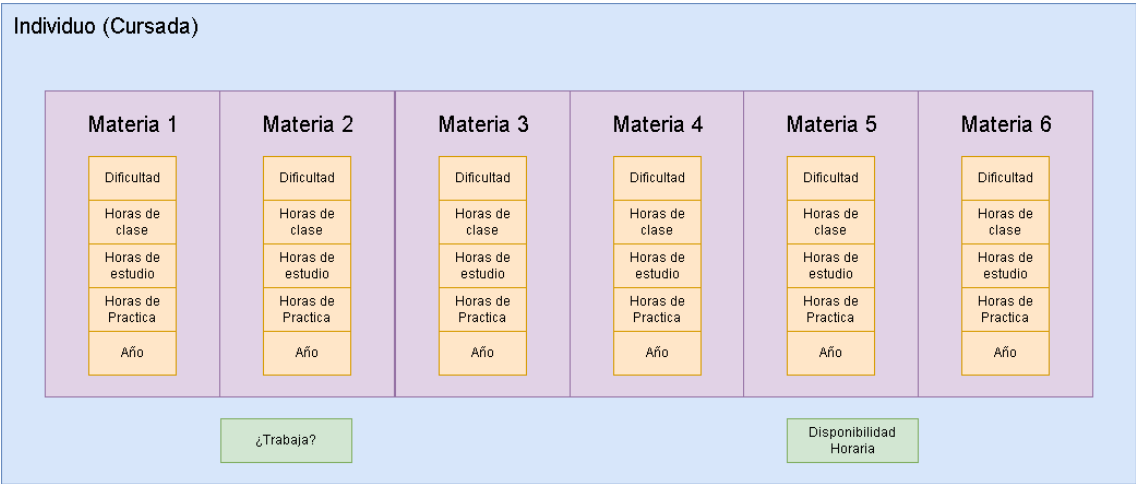


Teniendo en cuenta las características de los valores a representar, decidimos no utilizar una representación binaria ya que hubiese sido agregar una complejidad extra al prototipo el tener que representar todas las características mediante ceros y unos, y además, hay ciertos genes (como la dificultad por ejemplo) que es muy ineficiente representarla con un valor binario.

Luego de investigar los distintos tipos de representaciones decidimos elegir la representación o codificación en árbol. En este tipo de codificación cada cromosoma es árbol con ciertos objetos, lo cual se condice con nuestra forma de ver a la cursada como un conjunto de materias. Cada una de las materias sería un nodo o subárbol del cromosoma principal.

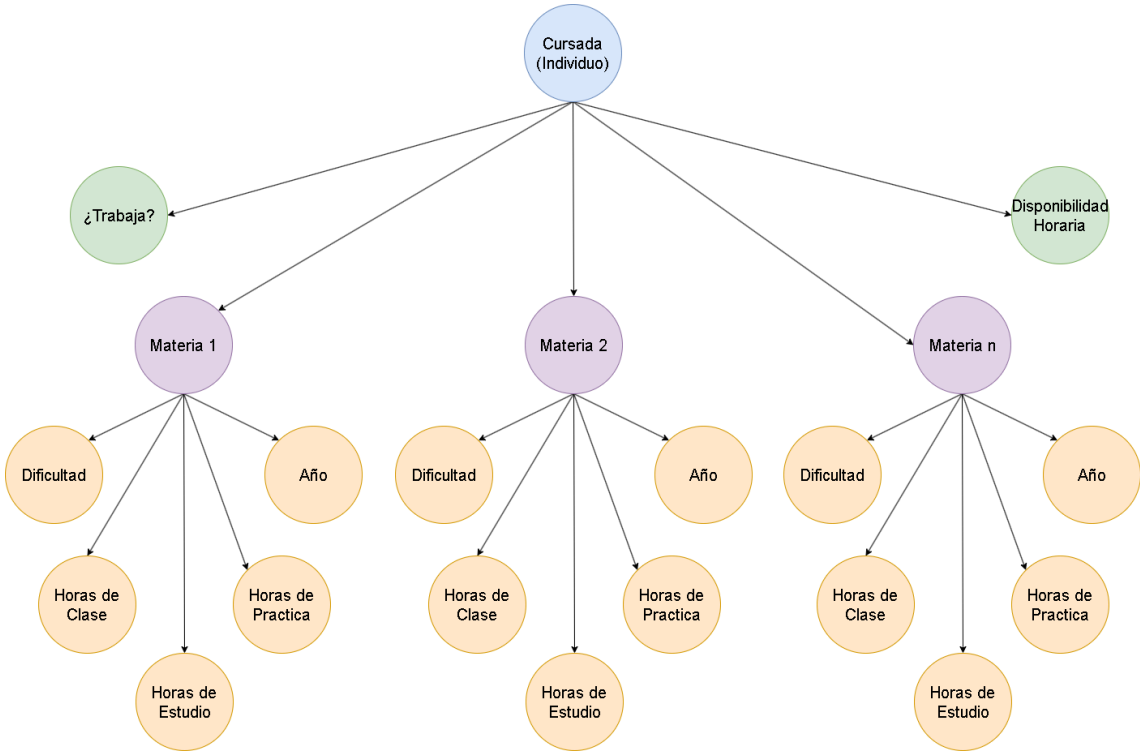
Dentro de las materias, los valores de cada están representados por valor directo. Consideramos que se puede trabajar directamente con esos valores y no se justifica realizar una codificación.

El resultado final es el siguiente:



Trabaja y Horas de Estudio se van a mantener constantes durante todo el proceso del algoritmo genético, ya que son características propias del usuario. Por lo tanto, solo se le realizarán operaciones a las materias.

El árbol del cromosoma quedó con la siguiente estructura:



Correspondencia Genotipo-Fenotipo:

Fenotipo	Genotipo
Dificultad de la Cursada.	$\sum_{i=1}^n$ dificultad Materia i
Cantidad de Horas de Clase.	$\sum_{i=1}^n$ horas de clase Materia i
Cantidad de Horas de Estudio.	$\sum_{i=1}^n$ horas de estudio Materia i
Cantidad de Horas de Práctica.	$\sum_{i=1}^n$ horas de practica Materia i
Año de cada materia	$\sum_{i=1}^n$ Año de cada materia i 1º (5), 2º (4), 3º (3), T(3), 4º(2), 5º(1)

¿Usuario Trabaja?	{SI (1), NO(0)}
Disponibilidad Horaria del Usuario.	Valor entero positivo menor a 168

Entre () figura valorización de cada característica

donde n = La cantidad de materias de materias incluidas en la cursada

## 2. Inicializacion de la Poblacion

En esta etapa decidimos que lo mejor era inicializar a la población aleatoriamente. Utilizamos una función que crea a los individuos (cursadas), agregando una cantidad aleatoria de materias a cada una. Solo se añaden materias que el usuario está en condición de cursar, respetando los días y turnos de la oferta de materias para evitar superposición de horarios.

Como se mencionó en el punto anterior, los genes correspondientes a si el usuario trabaja y la cantidad de horas que va a dedicar al estudio se mantienen constantes en todos los individuos generados.

## 3. Evaluación y Selección de los individuos

### Función de aptitud:

Utilizaremos la función de aptitud para evaluar a los individuos. Consideraremos que cuanto mayor aptitud tenga, mejor es el individuo.

La función de aptitud que diseñamos tiene en cuenta 6 parámetros:

- La Probabilidad de Éxito de la cursada (valor entre 0 a 100)
- La dificultad de la cursada (valor entero entre 1 a 10)
- La cantidad requerida de horas semanales (valor entero entre 4 y 168)
- La cantidad de materias que puede llegar a desbloquear instantáneamente (valor entero positivo)
- La cantidad de materias que puede llegar a desbloquear hasta el final de la carrera (valor entero positivo)
- La cantidad de Materias de los primeros años para el título intermedio

El resultado de la función surge de la siguiente operación:

**Aptitud** = Probabilidad de Éxito - Penalización de Horas - Penalización de Dificultad + Premio Materias Instantáneas Desbloqueadas + Premio Materias Totales Desbloqueadas + Premio materias primeros años

Explicación de cada parámetro:

- La *Probabilidad de Éxito* es un valor proveniente de una red neuronal, la cual evalúa las características de la cursada así como los datos ingresados por el usuario (horas semanales que va a dedicar a la facultad y si trabaja) para predecir la probabilidad de éxito que puede tener el usuario en caso de realizar esa cursada.

*Aclaración:* Se considera “Éxito” en una cursada cuando no se recursó ninguna materia. Llevando todas las materias a final o promocionandolas se considera que fue un cuatrimestre “Exitoso”.

Por otro lado, se considera que un cuatrimestre fue un “Fracaso” cuando se recursa al menos una materia.

Decidimos considerar al “Éxito” o “Fracaso” de esta manera para definir un límite objetivo y poder distinguir entre un cuatrimestre exitoso y uno que no lo es.

- La *Penalización de horas* sirve para cuantificar la diferencia entre horas estimadas por el usuario y horas reales de la cursada. El sentido de esta penalización es evitar que se desperdicie la disponibilidad horaria del usuario.

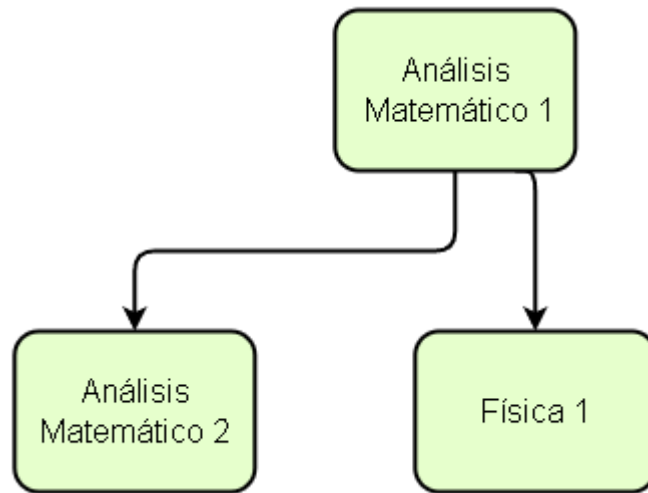
Por ejemplo, si el usuario va a dedicar 50 horas semanales al cuatrimestre. Una cursada de 2 materias posiblemente tenga una probabilidad de éxito cercana al 100%, sin embargo no se está aprovechando de la mejor manera la disponibilidad horaria del usuario. Una cursada de 4 o 5 materias, aunque quizás se reduzca un poco, también tendría una probabilidad de éxito relativamente alta y aprovecharía de una mejor manera esas 50 horas semanales estimadas por el usuario.

- La *penalización de dificultad* sirve para evitar recomendar una cursada muy difícil o una cursada muy fácil. Nuestra intención es que la cursada que se recomiende tenga una dificultad intermedia, para que se adapte de la mejor forma a las capacidades de todos los alumnos. Una cursada muy fácil puede llevar al aburrimiento, mientras que una cursada muy difícil puede llevar a la frustración. Sumado a esto, consideramos que una cursada está bien pensada cuando existe un equilibrio entre materias “fáciles” y materias “difíciles”.
- El *Premio Materias Instantáneas Desbloqueadas* busca priorizar cursadas desbloqueen la mayor cantidad de materias el próximo cuatrimestre.

Por ejemplo, la materia “Responsabilidad Social Universitaria” aunque sea de segundo año, no desbloquea ninguna materia al promocionarla. En cambio, la materia “Estadística Aplicada” pese a ser de cuarto año puede llegar a desbloquear hasta 3 materias. Este premio tiene por finalidad priorizar en este caso “Estadística Aplicada” por sobre “Responsabilidad Social Universitaria”.

Cuanto más materias se desbloqueen, se va a contar con más opciones de inscripción para el próximo cuatrimestre, lo que va a reducir la probabilidad de superposición de horarios o comisiones llenas. Por el contrario, cursar materias que no desbloquean otras, reduce la cantidad de opciones disponibles para la próxima inscripción, lo que puede perjudicar su próxima inscripción.

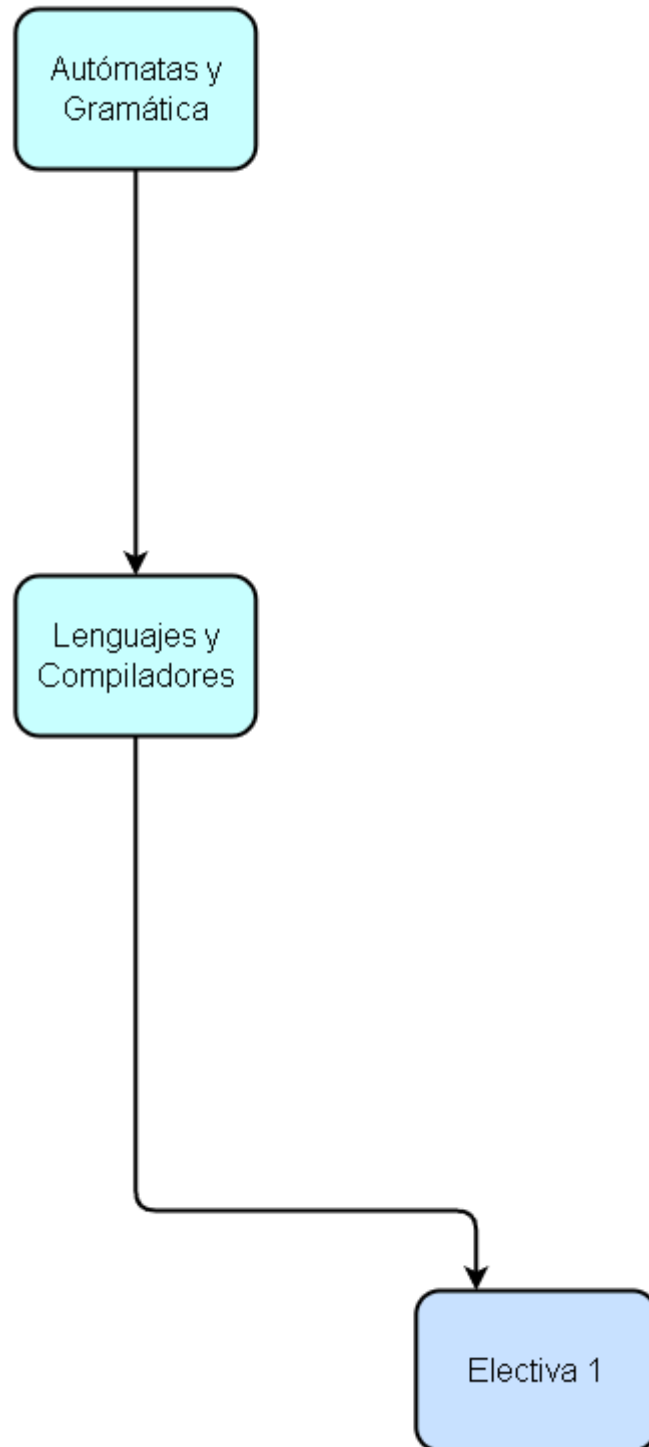
Ejemplo gráfico: La materia “Análisis matemático 1” puede llegar a desbloquear el próximo cuatrimestre 2 materias: “Análisis Matemático 2” y “Física 1”, ya que ambas tienen como correlativa a “Análisis Matemático 1”.



- El *Premio Materias Totales Desbloqueadas* tiene relación con el premio anterior, pero tienen objetivos diferentes. Mientras que el “Premio Materias Instantáneas Desbloqueadas” consideraba únicamente el próximo cuatrimestre, este premio considera toda la cantidad de materias que desbloquea para carrera.

El objetivo es priorizar los “caminos” o “ramas” que cuenten con la mayor cantidad de materias. Esto logra que se prioricen materias más centrales o importantes. De esta forma, se evita que en el futuro el usuario no pueda inscribirse a la cantidad de materias que desea debido a problemas con materias correlativas.

Ejemplo gráfico: La materia “Autómatas y Gramática” desbloquea en total 2 materias hasta el final de la carrera, ya que desbloquea instantáneamente a “Lenguajes y compiladores”, la cual a su vez, desbloquea a la materia “Electiva 2”. Como no hay otra materia que tenga como correlativa a “Electiva 2” se considera que ahí termina la “rama” o el “camino de materias”.



- El *Premio materias primeros años* también mantiene relación con los premios anteriores. El objetivo de este parámetro es priorizar las cursadas de materias de primeros años, para que el alumno pueda obtener el título intermedio lo antes posible. Además de esta manera, el usuario puede cursar materias y adquirir los conocimientos de la forma en la que fue pensado y diseñado el plan de carrera.

La cuenta se realiza en base a la probabilidad de éxito calculada por la red neuronal. Pero como queríamos que también se tengan en cuenta otros

factores que no sean solamente la probabilidad de Éxito, incluimos “premios” y “penalizaciones” a la probabilidad de Éxito calculada. Esta consideración de factores ajenos al Éxito o Fracaso de la cursada permite personalizar de mejor manera la recomendación según las características y necesidades del usuario.

#### Método de selección:

Decidimos utilizar un método de *selección por torneo*, en el cual escogemos de forma aleatoria dos individuos de la población, y el que tiene mayor aptitud se selecciona para que se reproduzca y también para que pase a la siguiente generación. El perdedor es descartado.

De esta forma logramos mantener cierto grado de diversidad en la población y evitamos que converjan tan rápido.

En esta primera selección, se elige al 50% que ganó su torneo para que pase a la siguiente generación. Para completar el 50% restante, se cruzará a los individuos de esta población seleccionada.

#### **4. Cruza de Individuos**

Decidimos utilizar una *cruza multipunto* para conseguir una mayor diversidad entre padres e hijos y así poder explorar todas las combinaciones de cursadas rápidamente.

Elegimos 5 puntos de cruce, uno por cada día luego del lunes, de esta manera, las materias de los padres quedarían intercaladas en base a los días.

Funciona de la siguiente manera:

- Cruza Progenitor A x Progenitor B = Hijo A

Donde Hijo A tiene el horario del Progenitor A los días Lunes, Miércoles y Viernes y el horario del Progenitor B los días Martes, Jueves y Sábado.

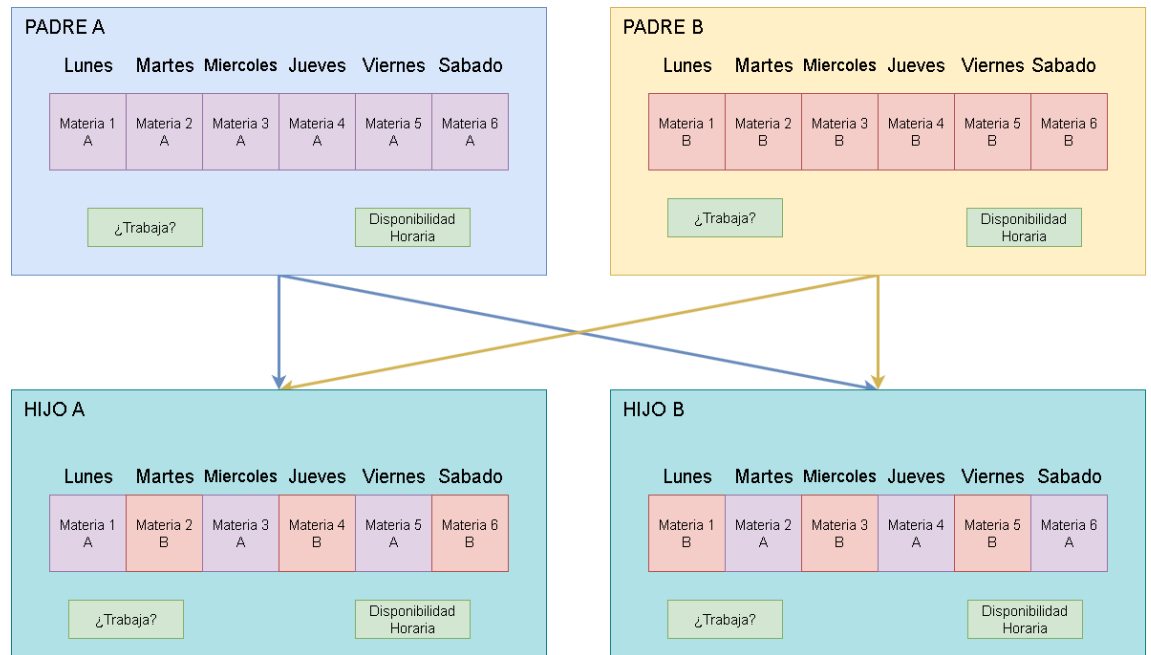
- Cruza Progenitor A x Progenitor B = Hijo B

Donde Hijo B tiene el horario del Progenitor B los días Lunes, Miércoles y Viernes y el horario del Progenitor A los días Martes, Jueves y Sábado.

*Importante:* En caso que haya alguna superposición de materias, es decir, el hijo quedó con la misma materia en dos horarios distintos, se elimina aleatoriamente uno de los dos horarios de la materia.

Explicación gráfica de como queda una cruce:





## 5. Mutación

Cómo elegimos una codificación en árbol, en nuestro caso las mutaciones se producen sobre ramas enteras (materias).

La mutación se le realizará a la población conformada por los padres e hijos del paso anterior.

Decidimos generar distintos tipos de mutaciones para aumentar la diversidad de la población.

Los tipos de mutaciones posibles son los siguientes:

- No realizar ninguna mutación (Probabilidad: 70%)
- Agregar una materia al individuo (Probabilidad: 10%)
- Intercambiar horarios entre materias del mismo individuo (Probabilidad: 10%)
- Eliminar una materia al individuo (Probabilidad: 10%)

En total hay un 30% de probabilidades que el individuo sufra alguna mutación y un 70% de probabilidades de que no se le realice ningún cambio.

Establecimos la probabilidad de mutación en 30%, la cual es relativamente alta, ya que debido a las limitaciones de la oferta de materias por más que se intente realizar alguna mutación, muchas veces no es posible. En los casos donde no es posible realizar dicha mutación, se deja al individuo sin mutar y se pasa al siguiente de la población.

Utilizamos la *técnica de mutación simple*, es decir, siempre se tiene la misma probabilidad de realizar la mutación a los individuos. Si bien esto no es lo más óptimo, consideramos que al tener pocas iteraciones el algoritmo genético no es necesario usar técnicas adaptativas.

## 6. Reemplazo, Iteración y Terminación

La técnica de reemplazo que decidimos utilizar es “Gap Generacional”. Evaluamos a la población resultante luego de realizar la mutación y realizamos una selección por torneo para quedarnos con el 50% de la población.

Este 50% seleccionado se combina con el 50% seleccionado en el paso 3 para pasar a la siguiente generación.

Al finalizar evaluamos la aptitud general de la población y continuamos iterando a partir del paso 3.

Definimos dos condiciones de corte: la primera es un límite máximo de 20 generaciones (iteraciones) y la segunda es un límite de 5 generaciones sin encontrar un mejor individuo que supere la mejor de aptitud histórica. Luego de 20 generaciones el algoritmo termina, a menos que durante el proceso se detecte un estancamiento de la aptitud por 5 generaciones. En ese caso, el algoritmo se detiene ya que consideramos que se alcanzó un “óptimo”.

Una vez obtenida la población final, se realiza una selección elitista donde elegimos al cromosoma (cursada) de mayor aptitud. Esa es la cursada que se le recomendará al usuario.

- **Parte 4) Aumentación de Datos.**

En la cuarta parte, crearemos un dataset artificial para simular un historial de cursadas de alumnos, con el fin de poder entrenar la red neuronal que predecirá la probabilidad de éxito de una cursada. Utilizaremos un segundo algoritmo genético para generar y seleccionar los individuos más “realistas”.

Comenzamos ingresando el historial de nuestras cursadas reales. Las usamos como punto de partida para inicializar la población y también como referencia cuando planteamos la función de aptitud. Para balancear el dataset inicial, consultamos las cursadas de otros estudiantes de la facultad para tener mayor variedad de cursadas.

Una vez finalizado este paso, obtuvimos un dataset inicial de 64 cursadas reales.

Luego, definimos un segundo algoritmo genético para generar los datos de entrenamiento de la red.

A continuación explicaremos brevemente los pasos que seguimos para definir este algoritmo genético:

- 1. Representación del Estado del problema / Codificación de los individuos.**

Problema: Generar historiales de cursadas lo más realistas posibles.

Características a representar:

- Cantidad estimada de horas requeridas por las materias de la cursada.
- Cantidad de horas semanales de estudio por parte del usuario.
- Trabaja
- Resultado del Cuatrimestre (Éxito/Fracaso)

Los individuos en este caso serán historiales de cursadas, los cuales incluirán las materias cursadas, la información del alumno (si trabaja y horas de estudio).

Sin embargo, existe una diferencia respecto a los cromosomas definidos en el algoritmo genético anterior. En este caso, se agrega un nuevo gen al cromosoma, que representa el resultado obtenido en la cursada (Éxito/Fracaso).

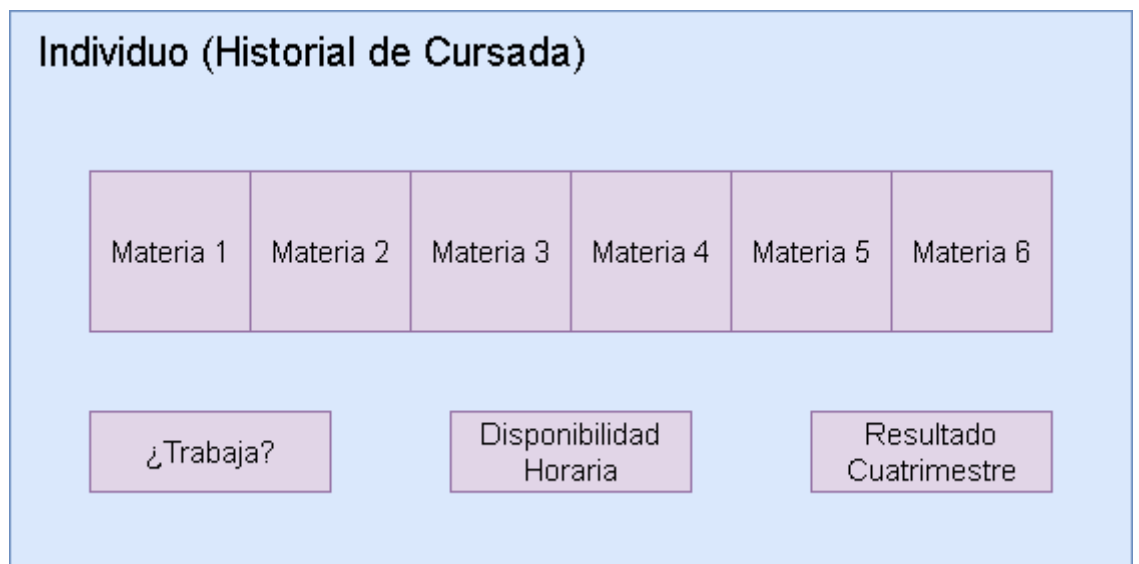
El cromosoma que definimos tiene la siguiente estructura:

*Aclaración:* Al igual que en el algoritmo genético anterior, cómo definimos la red neuronal para que reciba como entradas hasta 6 materias, los historiales de cursadas tendrán como máximo 6 materias.

Al igual que en la parte 3, cada materia posee:

- Dificultad (valor entre 1 y 10)
- Horas semanales de clase (valor entero positivo, debido a las características del plan todas las materias son de 4 horas semanales)
- Horas semanales de estudio (valor entero positivo, entre 0 y 168\*)
- Horas semanales de práctica (valor entero positivo, entre 0 y 168\*)
- Año (valor entre 1 a 5. Puede ser T en caso de una materia transversal)

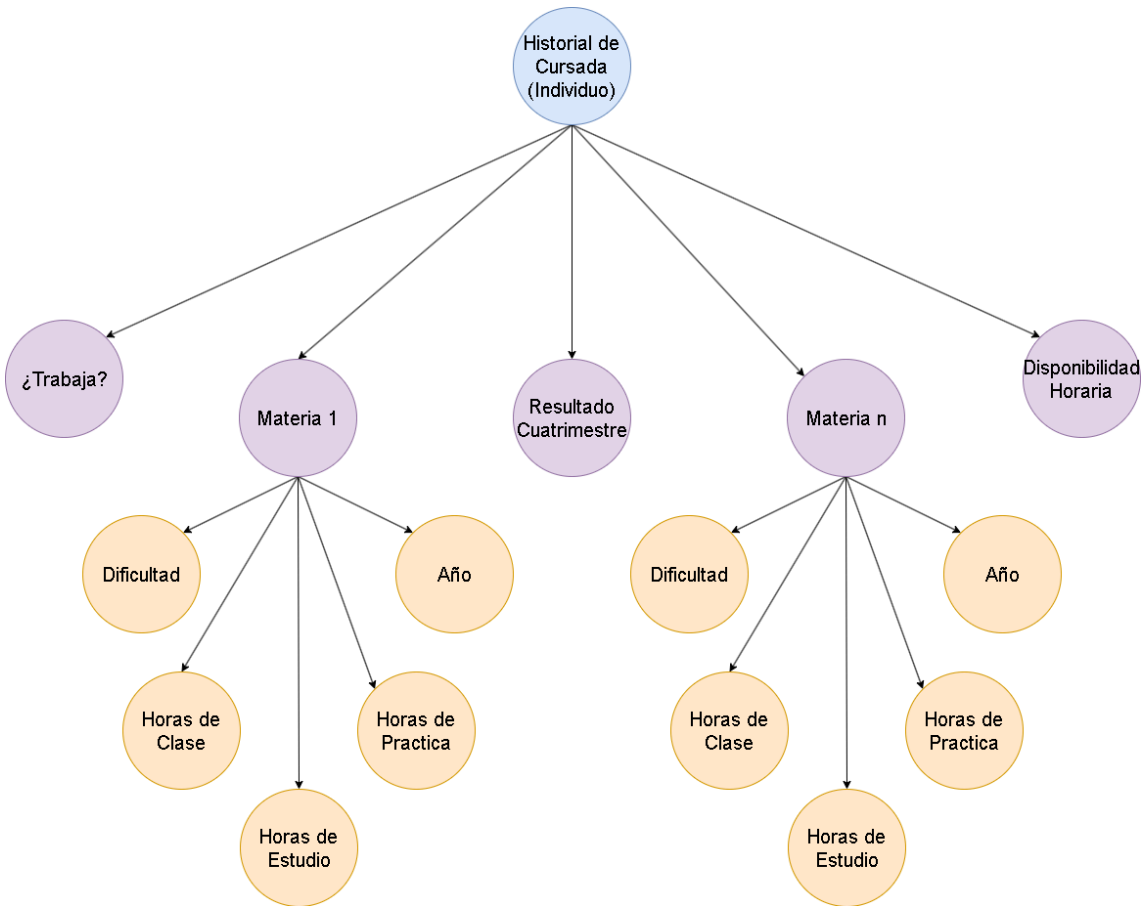
El cromosoma que definimos tiene la siguiente estructura:



Decidimos utilizar una codificación similar al algoritmo genético anterior, ya que el problema a representar es bastante similar. Nuevamente, utilizaremos una codificación en árbol para representar a los individuos de la población.

Como cada historial de cursada (individuo) corresponde a distintas personas, los genes “Trabaja” y “Horas de Estudio”, así como también el “Resultado del Cuatrimestre”, van a variar según el individuo. A diferencia del algoritmo genético anterior donde solo se le realizaban operaciones a las materias, en este caso se le realizaron operaciones a todos los genes por igual.

El árbol del cromosoma quedó con la siguiente estructura:



Correspondencia Genotipo-Fenotipo:

Fenotipo	Genotipo
Cantidad estimadas de Horas Necesarias	$\sum_{i=1}^n$ horas de clase Materia i + horas de estudio Materia i + horas de practica Materia i
Disponibilidad Horaria del Usuario.	Valor entero positivo menor a 168
¿Usuario Trabaja?	{SI (1), NO(0)}
Resultado del Cuatrimestre.	{Éxito (1), Fracaso(0)}

Entre () figura valorización de cada característica.

donde n = La cantidad de materias de materias incluidas en el historial de cursada.

2. Inicializacion de la Poblacion

Basándose en el historial de cursadas reales, se crea la población inicial de 200 individuos.

El procedimiento es el siguiente:

1. Se selecciona una cursada real
2. Si el resultado fue "Exitoso", se le saca una materia y se le disminuye la cantidad de horas de estudio del usuario en función a la materia eliminada.
3. Si el resultado fue "Fracaso", se le agrega una materia y se le aumenta la cantidad de horas de estudio del usuario en función a la materia agregada.
4. Cada materia de la cursada tiene una probabilidad del 75% de intercambiarse por otra materia del mismo año.
5. Hay una probabilidad del 50% de intercambiar el valor de "¿Usuario Trabaja?" al valor opuesto.

Luego de aplicar todas estas modificaciones queda creado un nuevo historial de cursada que se agrega a la población inicial del algoritmo genético.

Al finalizar este proceso, queda generada la población inicial la cual queda conformada por los datos reales de cursadas pasadas y por los nuevos historiales de cursadas creados artificialmente.

### 3. Evaluación y Selección de los individuos

#### Función de aptitud:

Utilizaremos la siguiente función para calcular la aptitud de los individuos.

Decidimos considerar estos tres aspectos o parámetros para determinar la aptitud de los individuos:

1. Parámetro 1: Evalúa si el resultado del cuatrimestre es realista
2. Parámetro 2: Evalúa si las horas de estudio son realistas en función de las horas necesarias y su condición laboral
3. Parámetro 3: Evalúa la cantidad de materias

Estos 3 parámetros se encargan de medir el "realismo" de la cursada, en el caso de que el individuo tenga un parámetro realista se aumenta la aptitud. Por el contrario, en caso de que algún parámetro no sea realista, se reduce la aptitud del individuo en función de qué tan alejado de la realidad esté.

La función de aptitud la definimos de la siguiente manera:

**Aptitud** = Evaluación del resultado obtenido en el cuatrimestre + Evaluación de las horas de estudio del usuario + Evaluación de la cantidad de materias

En caso de que las evaluaciones determinen que el parámetro es realista, tendrá un valor positivo. En cambio, si luego de la evaluación se determina que el parámetro no es realista, tendrá un valor negativo el cual su magnitud dependerá de lo alejado que esté de los límites esperados.

Consideramos que para este caso, lo mejor era utilizar una función simple que tenga en cuenta únicamente los aspectos más importantes de la cursada. Por eso mismo, unificamos en un solo parámetro las horas requeridas por cada

materia, así como también dejamos afuera del análisis la dificultad de cada materia.

Explicación de cada parámetro:

- *Evaluación del resultado obtenido en el cuatrimestre:* Valida el resultado obtenido en el cuatrimestre. Compara las horas de estudio del usuario con las horas estimadas necesarias de la cursada.

De esta manera se penaliza si se tuvo éxito dedicando muchas menos horas de las necesarias o si se fracasó dedicando muchas más horas de las necesarias.

- *Evaluación de las horas de estudio del usuario:* Esta evaluación está dividida según 4 distintos escenarios:
  - Si el alumno trabajaba y tuvo éxito, se valida que las horas dedicadas al estudio por parte del usuario estén entre un valor un poco menor a las horas requeridas por las materias cursadas y un límite máximo de horas semanales realista para una persona que trabaja.
  - Si el alumno trabajaba y fracasó, se valida que las horas dedicadas al estudio por parte del usuario estén entre un límite inferior que representa las horas de clase sincrónica de la cursada (es decir, únicamente asistió a clases) y un valor un poco mayor a las horas requeridas por las materias cursadas.
  - Si el alumno no trabajaba y tuvo éxito, se valida que las horas dedicadas al estudio por parte del usuario estén entre un valor un poco menor a las horas requeridas por las materias cursadas y un límite máximo de horas semanales realista para una persona que no trabaja.
  - Si el alumno no trabajaba y fracasó, se valida que las horas dedicadas al estudio por parte del usuario estén entre un límite inferior que representa las horas de clase sincrónica de la cursada (es decir, únicamente asistió a clases) y un valor un poco mayor a las horas requeridas por las materias cursadas.

En esta evaluación realizamos un pequeño ajuste. Consideramos que para tener el mismo éxito en una cursada, una persona que trabaja necesita esforzarse más que una persona que no trabaja. Por lo tanto, las horas requeridas por las materias cursadas tienen un aumento del 5% en caso de que la persona trabaje.

- *Evaluación de la cantidad de materias:* Valida que la cantidad de materias de la cursada esté en un rango de 3 a 6 materias. Consideramos que cursadas con 2 o menos materias no son muy frecuentes y no aportan mucha información para el entrenamiento de la red neuronal. Por otro lado, cursadas de 7 o más materias quedan fuera del alcance de nuestra aplicación, ya que solo nos dedicaremos a recomendar y evaluar cursadas de hasta 6 materias.

Método de selección:

Del mismo modo que en el algoritmo genético anterior, decidimos utilizar el método de selección por torneo para aumentar la diversidad de cada generación. El funcionamiento sigue siendo el mismo, se seleccionan dos individuos aleatoriamente, el de mayor aptitud pasa a la siguiente generación y es utilizado para la cruce, mientras que el perdedor es descartado.

**4. Cruza de Individuos**

Decidimos utilizar una *cruza simple*. En este caso decidimos elegir el método clásico de la cruce para simplificar la complejidad del problema.

Funciona de la siguiente manera:

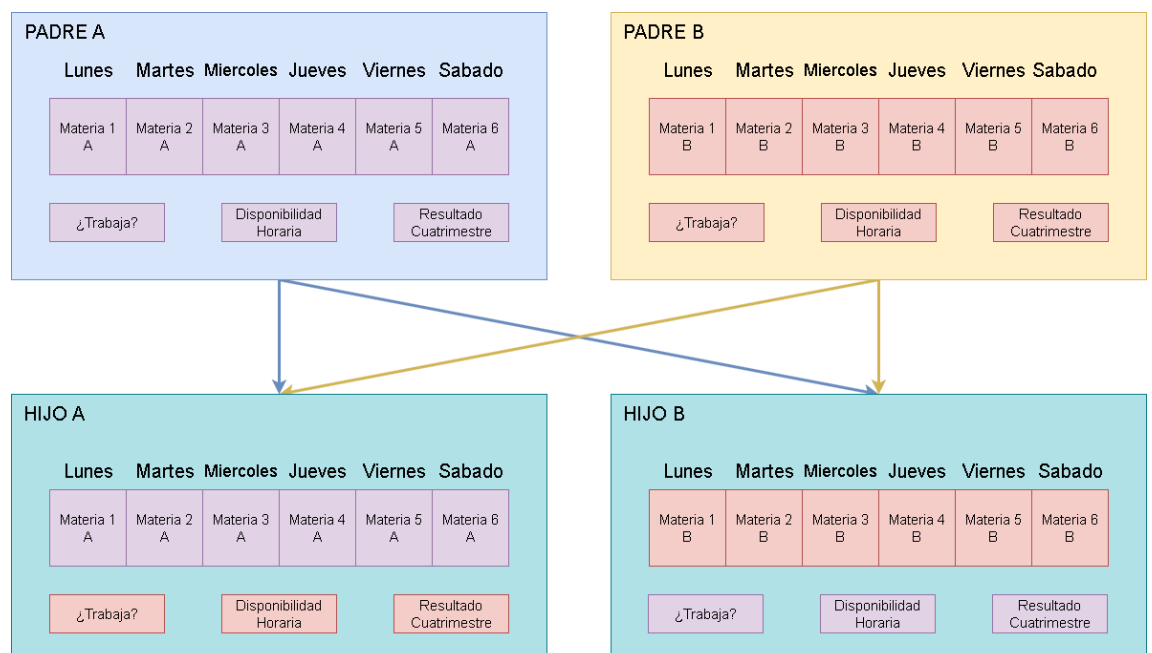
- Cruza Progenitor A x Progenitor B = Hijo A

Donde Hijo A tiene las materias del Padre A, y la información del alumno y resultado del cuatrimestre del padre B.

- Cruza Progenitor A x Progenitor B = Hijo B

Donde Hijo B tiene las materias del Padre B, y la información del alumno y resultado del cuatrimestre del padre A.

Explicación gráfica de la cruce:

**5. Mutación**

Al haber elegido de nuevo una codificación en árbol, las mutaciones se producirán sobre los nodos, sin embargo, en este caso si se modificarán los valores de disponibilidad horaria del usuario, si trabaja y el resultado del cuatrimestre.

Los distintos tipo de mutaciones que decidimos utilizar fueron los siguientes:

Decidimos generar distintos tipos de mutaciones para aumentar la diversidad de la población.

Los tipos de mutaciones posibles son los siguientes:

- Invertir el resultado del cuatrimestre (Probabilidad 10%)
- Invertir el valor del gen "Trabaja" (Probabilidad: 10%)
- Aumentar un 20% las horas de estudio del alumno (Probabilidad: 10%)
- Disminuir un 20% las horas de estudio del alumno (Probabilidad: 10%)
- Intercambiar una materia de la cursada por otra del mismo año (Probabilidad: 10%)

Cada una de estas mutaciones tienen un 10% de probabilidades de ocurrir. Nuestro objetivo es aumentar la diversidad de la población lo mayor posible, de forma tal que cuando se usen estos datos para entrenar la red neuronal, se tengan en cuenta los distintos escenarios y alternativas que pueden ocurrir en una cursada.

También volvimos a utilizar la técnica de mutación simple, ya que consideramos que no hacía falta implementar alguna técnica de mutación adaptativa. No vemos necesario aumentar la probabilidad de mutación a lo largo de las generaciones, ya que queremos aumentar la diversidad desde el principio, así como tampoco vemos necesario reducir la probabilidad de mutación a lo largo de las generaciones ya que no buscamos acercarnos a un valor óptimo, sino que la población final quede en promedio con un buen valor de aptitud.

## 6. Reemplazo, Iteración y Terminación

Nuevamente, la técnica de reemplazo que decidimos utilizar es "*Gap Generacional*". Evaluamos a la población resultante luego de realizar la mutación y realizamos una selección por torneo para quedarnos con el 50% de la población.

Este 50% seleccionado se combina con el 50% seleccionado en el paso 3 para pasar a la siguiente generación.

Al comienzo de cada iteración evaluamos la aptitud general de la población para verificar cómo están evolucionando los individuos y continuamos iterando a partir del paso 3.

Definimos dos condiciones de corte: la primera es un límite máximo de 20 generaciones (iteraciones) y la segunda es un límite de 5 generaciones sin mejora de aptitud poblacional. Luego de 20 generaciones el algoritmo termina, a menos que durante el proceso se detecte un estancamiento de la aptitud global por 5 generaciones. En ese caso, el algoritmo se detiene ya que consideramos que se alcanzó un "óptimo".

Una vez obtenida la población final, se realiza una *selección elitista* donde elegimos a los 50 cromosomas (cursadas) de mayor aptitud. Esos individuos serán los que se incluyan en el dataset de entrenamiento de la red neuronal.

- Parte 5) Creación de la Red Neuronal.



En la quinta parte, se crea la red Neuronal que se encargará de estimar la probabilidad de éxito o fracaso de una cursada, la cual es necesaria para calcular la aptitud de un individuo en el algoritmo genético que realizará la recomendación final al usuario.

La probabilidad de éxito se estima a partir de las características de una cursada (como su dificultad y cantidad de horas de clase y estudio), así como los datos ingresados por el usuario (horas semanales que va a dedicar al estudio y si trabaja).

Dicha Red Neuronal está implementada en este prototipo por medio de las librerías TensorFlow y Keras.

La creación de este modelo consta de 4 partes:

A) Generación de los Datasets:

Se generan 3 datasets a partir de el dataset artificial de 100.000 tuplas (obtenido en la Sección de Aumentación de Datos):

- train\_ds: Dataset de entrenamiento que consta de 80.000 tuplas
- val\_ds: Dataset de validación que consta de 10.000 tuplas
- test\_ds: Dataset de prueba que consta de 10.000 tuplas

El tamaño del lote de los 3 datasets es de 256

Además los datos son preprocesados con funciones de TensorFlow para que puedan ser fácilmente utilizados por el modelo.

B) Definición del Modelo:

La red neuronal se estructura de la siguiente forma:

- 32 Capas de Entrada que representan cada parámetro del DataSet (6 códigos de materias, 6 dificultades correspondientes a cada materia, 6 horas de clase, 6 horas de estudio y 6 horas de práctica correspondiente a cada materia, si trabaja y la cantidad de horas semanales de estudio).
- 32 Capas de Normalización de datos, una para capa de entrada, a fin de estabilizar el entrenamiento de la red y mejorar la convergencia del algoritmo.
- 1 Capa de Concatenación que se encarga de concatenar todas las features normalizadas en un único tensor, para luego ser pasado a través de capas posteriores del modelo.
- 1 Capa Densa de 32 neuronas que recibe el tensor de la capa de Concatenación y realiza una función de activación ReLu en cada una de sus neuronas.
- 1 Capa DropOut con una tasa de abandono del 50%, es decir que la mitad de los valores provenientes de la capa Densa se establecerán en cero utilizando un patrón de selección aleatorio. Por lo cual la salida de esta capa tendrá 16 valores en 0 y 16 que conservarán su valor original de la capa anterior. Esto se realiza principalmente para prevenir el overfitting, generando que el modelo sea más robusto y menos propenso a memorizar el ruido en los datos de entrenamiento.

- 1 Capa Densa de salida con una sola neurona y sin ninguna función de activación. La salida de esta última capa es la que se utiliza para realizar las predicciones.

C) Entrenamiento del Modelo:

Se entrena al modelo con 50 Epochs de 313 iteraciones cada una, debido a que el dataset de entrenamiento posee 80000 tuplas y lotes de 256. Además se utiliza el dataset de validación para ir analizando el progreso del modelo en cada Epochs, visualizando la evolución de las métricas “Accuracy” y “Loss”.

En caso de que durante el entrenamiento la red empiece a empeorar su accuracy, utilizamos el callback de “EarlyStopping” para detener el entrenamiento y restablecer los mejores pesos.

D) Evaluación del Modelo:

Utilizando el DataSet de Prueba, se determina el Accuracy y el Loss que posee el Modelo.

Luego de entrenar el modelo obtuvimos los siguientes resultados:

- Accuracy 0.9003999829292297
  - Loss 0.20118089020252228
- Parte 6) Puesta en marcha del algoritmo genético para realizar la recomendación.

En la sexta parte, se pone en marcha el algoritmo genético que obtendrá la cursada con mayor aptitud para ser recomendada al usuario. Dicho algoritmo trabajará con poblaciones de 200 individuos, y obtenida la población final se procederá a realizar una selección elitista para quedarse con el individuo más apto. En caso de que pasen 5 generaciones sin una mejora en la aptitud se procede a detener el algoritmo genético. Además establecimos un límite máximo de 20 generaciones, es decir que el algoritmo realizará como máximo 20 iteraciones.

Una vez obtenida la cursada ideal para el usuario, se procederá a mostrar la misma en forma de un calendario semanal, para que pueda visualizar de forma más amigable la recomendación de cursada.

Resultado del proceso de recomendación:

Cursada Recomendada						
	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Mañana					Análisis de Sistemas	
Tarde						
Noche	Autómatas y Gramáticas	Virtualización de Hardware	Ciencia de Datos	Gestión de las Organizaciones		

**Aclaración:** El proceso del algoritmo genético es normal que tarde algunos minutos en generar la recomendación. Para optimizar el tiempo, generamos una estructura que “cachea” la aptitud de las cursadas, lo que agiliza el cálculo de la aptitud. Sin embargo, la primera recomendación es normal que lleve un poco más de tiempo, ya que esa “caché” se encuentra vacía.

Sin embargo, para evitar que el proceso de recomendación tarde demasiado en la aplicación, decidimos reducir la cantidad de individuos y generaciones del algoritmo genético principal. De esta manera, si bien se logra llegar a una recomendación lógica y relativamente buena, el algoritmo genético no llega a explorar todas las posibles cursadas, por lo tanto no está garantizado un “óptimo”. Debido a esto puede suceder que si se ejecuta varias veces el mismo proceso de recomendación para el mismo usuario, se obtengan resultados ligeramente distintos.

A fin de realizar la demostración en clase en pocos minutos, decidimos priorizar la velocidad por sobre la búsqueda del resultado más “óptimo”. En caso de que el usuario desee realmente realizar una búsqueda más detallada sobre todas las posibles combinaciones de su cursada, puede ingresar al colab y aumentar manualmente los parámetros de cantidad de individuos y cantidad de generaciones.

### Modelos Elegidos:

Para el cálculo de la probabilidad de éxito implementamos un clasificador de datos estructurados basado en una red neuronal de TensorFlow. Se eligió este modelo por la facilidad de representar los datos del alumno y su cursada generados anteriormente en un vector numérico que puede usarse directamente como entrada de la red.

### Formato de las entradas

- Por cada materia del plan de cursada (6 máximo, rellena con 0 si hay menos de 6): (Código,Dificultad,Horas de clase,Horas de Estudio, Horas de Práctica)
- Datos del alumno: (¿El alumno Trabaja? (0/1),Horas de estudio por semana)

La variable objetivo de la red es el Éxito Global, 1 para éxito, 0 para fracaso. Representa si el alumno promocionó todas las materias.

### Capas de la red

#### Preprocesamiento:

- Input: Las entradas definidas anteriormente.
- Normalization: Normalizar las entradas para facilitar el entrenamiento y mejorar los resultados.
- Concatenate: Unir todos los inputs y combinar características en un tensor que sirve de entrada de la siguiente capa.

#### Entrenamiento:

- Dense: Capa completamente conectada que utiliza la función de activación ReLU
- Dropout: Eliminar conexiones de forma aleatoria para prevenir overfitting.
- Dense: Utilizada como salida, para proyectar el tensor al resultado final (Exitó\_Global)

### Análisis del Resultado para el caso de Uso “Calcular probabilidad de Éxito de cursada”:

- Resultado entre 0 y 25%: Es muy probable que el alumno al cursar esa combinación de materias tenga que recurrir al menos una de ellas. El alumno debería repensar todo el armado de su cursada.
- Resultado entre 26% y 50%: Es probable que el alumno al cursar esa combinación de materias deba recurrir al menos una de ellas. El alumno debería hacer algún cambio a su cursada, como por ejemplo, eliminar alguna materia de su cursada, reemplazar materias exigentes por otras más sencillas o aumentar el tiempo que va a dedicar a la facultad.
- Resultado entre 51% y 75%: Es probable que el alumno al cursar esa combinación de materias pueda aprobar o promocionar todas ellas, sin embargo, requerirá mayor compromiso y dedicación al estudio para que esto suceda. El alumno puede dejar la cursada tal como está y comprometerse a esforzarse durante todo el cuatrimestre o realizar alguna leve modificación en las materias, o en su disponibilidad horaria, para aumentar esta probabilidad.
- Resultado 76% y 100%: Es muy probable que el alumno al cursar esa combinación de materias pueda aprobar o promocionar todas ellas. La cursada es acorde a las características del alumno y no requiere ninguna modificación.

### **Datasets involucrados:**

Se crea un dataset artificial para simular un historial de cursadas de alumnos, con el fin de poder entrenar la red neuronal que predecirá la probabilidad de éxito de una cursada. Se utiliza un segundo algoritmo genético para generar y seleccionar los individuos más “realistas”.

Se comienza ingresando el historial de cursadas de los integrantes del grupo, las cuales se utilizan como punto de partida para inicializar la población y también como referencia para plantear la función de aptitud.

Luego, se define el segundo algoritmo genético para generar los datos de entrenamiento de la Red Neuronal.

Una vez finalizada toda esta sección, se obtiene un dataset final de 100.000 tuplas para entrenar la Red Neuronal.

Dicho dataset es dividido en 3 datasets:

- train\_ds: Dataset de entrenamiento que consta de 80.000 tuplas
- val\_ds: Dataset de validación que consta de 10.000 tuplas
- test\_ds: Dataset de prueba que consta de 10.000 tuplas

El tamaño del lote de los 3 datasets es de 256.

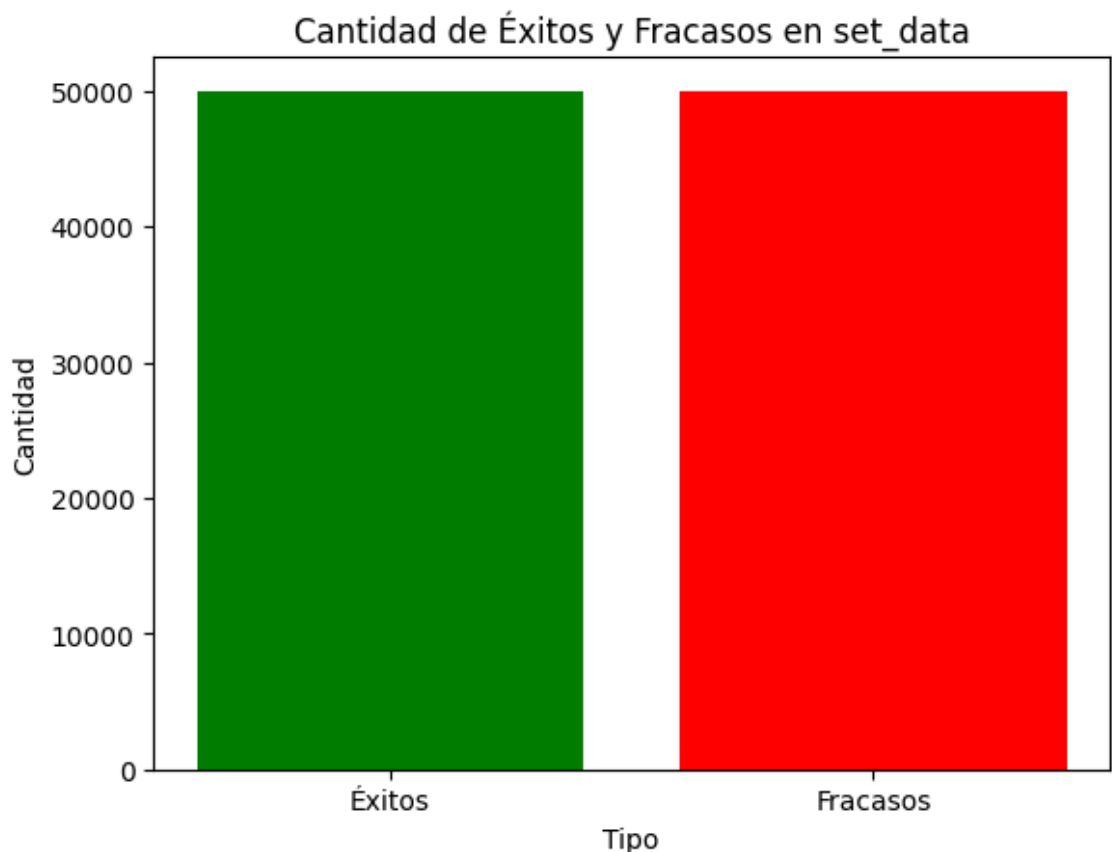
#### Análisis del dataset generado:

Para este análisis ejecutamos el algoritmo genético para obtener un dataset de 100.000 cursadas y de esta manera tener un valor representativo de cómo es la composición de los datos generados. Los resultados fueron los siguientes:

- Proporción de Éxitos y Fracazos: Este es el único valor que ajustamos manualmente. La única condición que agregamos al proceso de creación del dataset fue que se genere un 50% de Cursadas que fueron Éxito y un 50% de Cursadas que fueron Fracaso.

Consideramos que necesitábamos equilibrar la cantidad de Éxitos y Fracazos para que el modelo no se entrene con una tendencia hacia uno u otro.

Como resultado de generar 100.000 cursadas artificiales, obtuvimos 50.000 que obtuvieron “Éxito” y 50.000 que obtuvieron “Fracaso”.

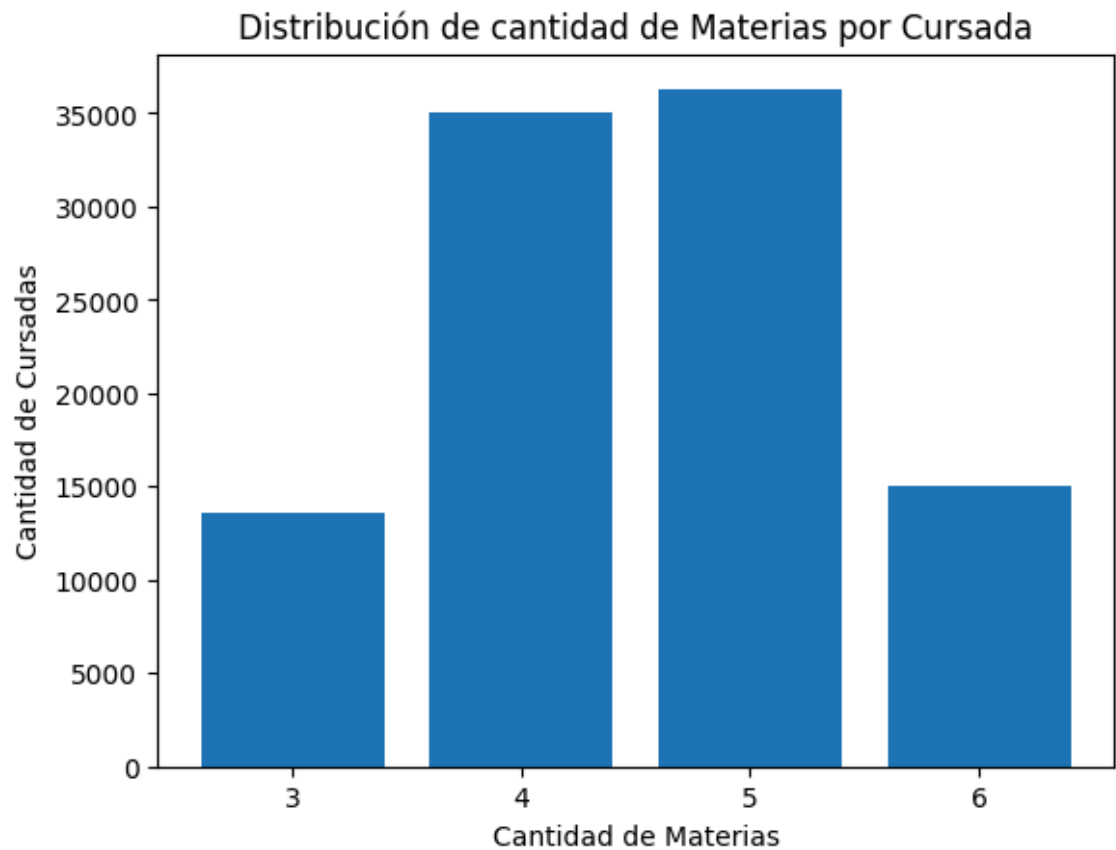


- Proporción de cantidad de materias:

Se puede apreciar que hay una mayor tendencia a generar cursadas de 4 o 5 materias, mientras que las cursadas de 3 o 6 materias son menos frecuentes.

Consideramos que estos resultados son bastante realistas, ya que la mayoría de nuestras cursadas fueron dentro del rango de 4 o 5 materias.

Debido a que en la función de aptitud se realizaba una penalización a los individuos que tenían menos de 3 materias o más de 6, no se incluyeron ese tipo de cursadas en el dataset de entrenamiento.



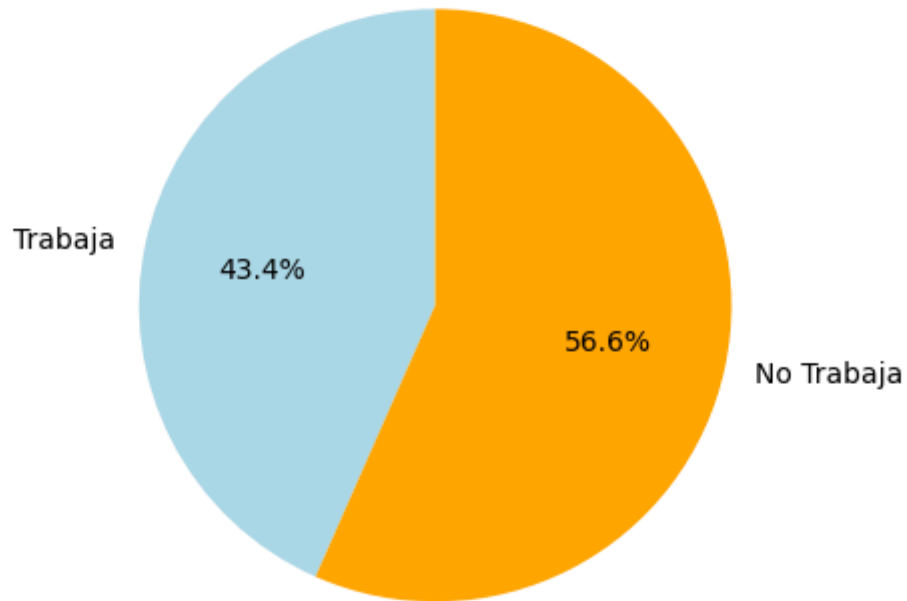
- Proporción de alumnos que trabajaban:

Como se puede apreciar en el siguiente gráfico, hay una ligera mayor proporción de alumnos que no trabajan.

Consideramos que estos resultados son realistas, ya que en promedio los alumnos de ingeniería suelen comenzar a trabajar a partir de 2º o 3º año. Muchos de ellos, una vez que ingresan al mundo laboral, abandonan la carrera para dedicarse a tiempo completo a sus trabajos. Por lo tanto, hay menos estudiantes que deciden combinar el estudio con el trabajo simultáneamente.

Además, hay una mayor cantidad de alumnos en los primeros años, en los cuales es menos probable que trabajen, a comparación de los últimos años de la carrera donde la mayoría de los estudiantes ya consiguieron su primer trabajo.

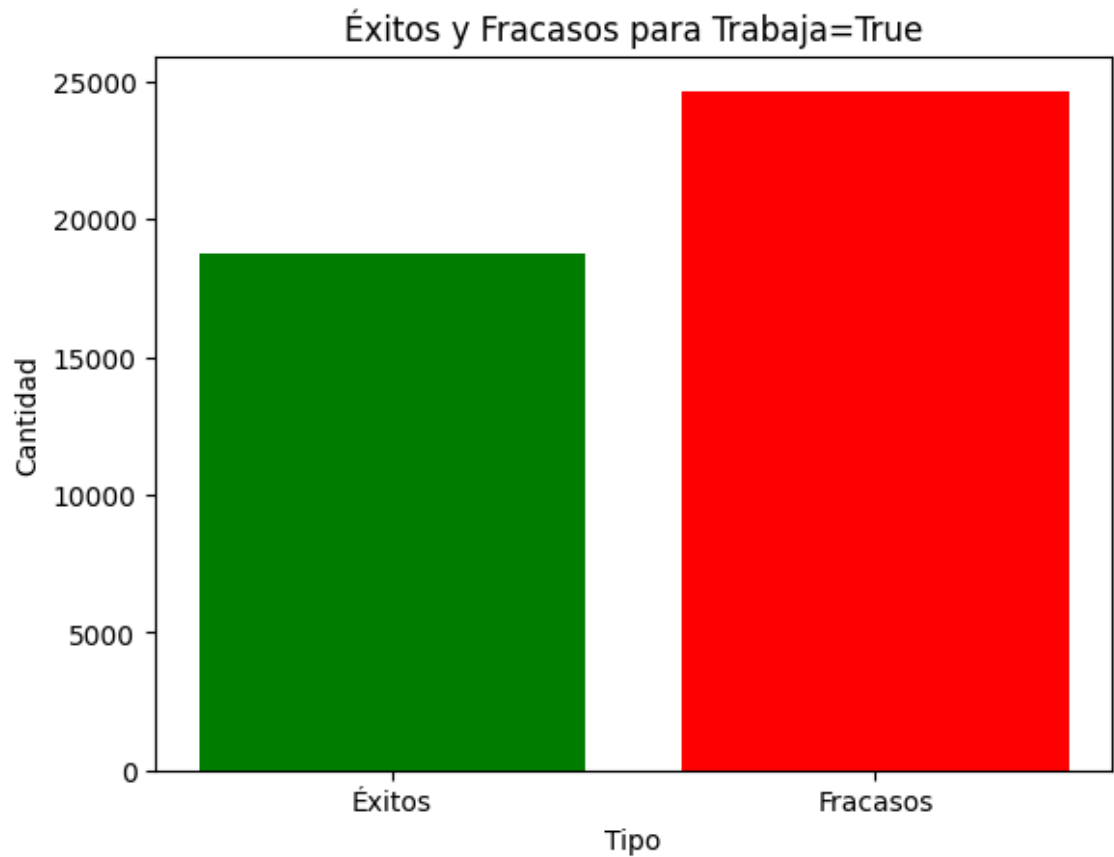
### Porcentaje de cursadas que el alumno trabajaba/no trabajaba



- Resultados de las personas que trabajaban:

Como se puede apreciar en el siguiente gráfico, las personas que trabajan tienen una ligera tendencia a recursar o abandonar al menos una de las materias que se inscribieron en el cuatrimestre.

Consideramos que este resultado es realista, es frecuente que los estudiantes que trabajan no lleguen a estudiar para un parcial, tengan menos tiempo disponible o no puedan llevar al día las materias, entre muchas otras razones, ya que tienen que priorizar las responsabilidades que tienen en su trabajo.

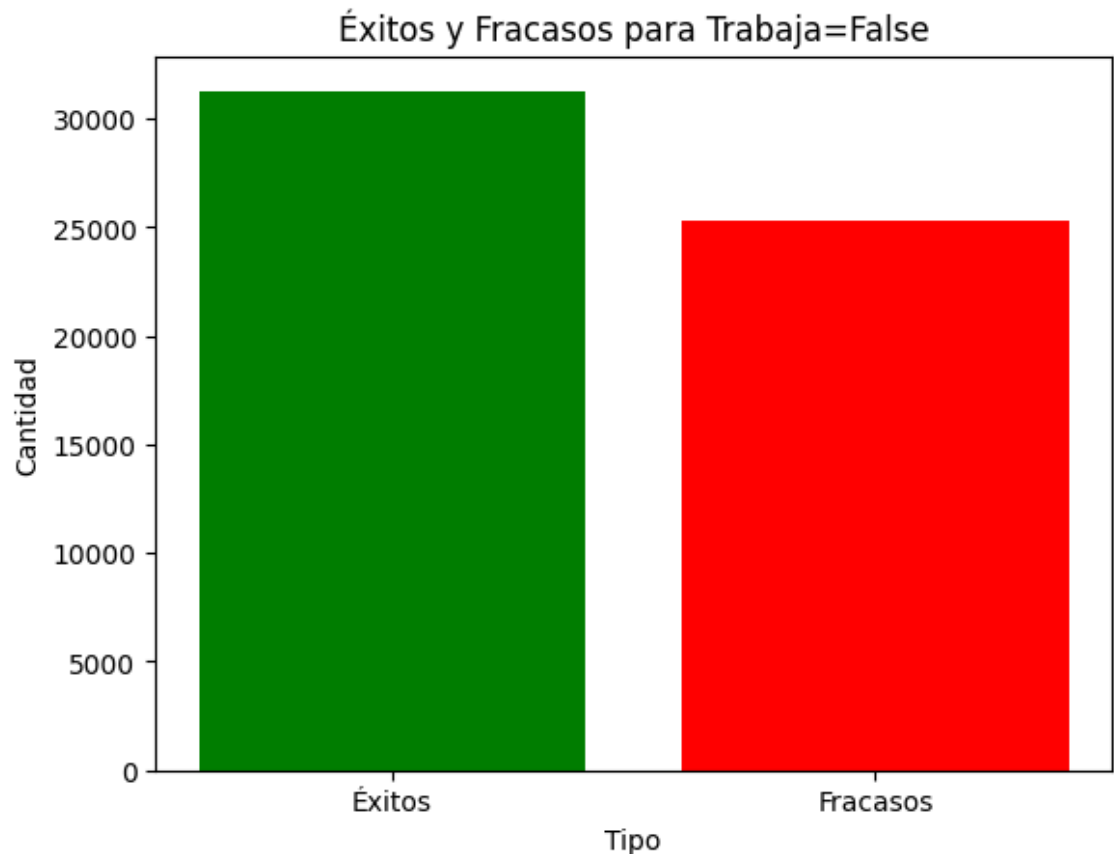


- Resultados de Personas que no trabajaban:

A diferencia de las personas que trabajan, se puede apreciar que en este caso hay una ligera tendencia a tener “Éxito” en las cursadas (es decir, no se recusa ninguna materia).

Al no tener que atender responsabilidades laborales, los estudiantes cuentan con mayor disponibilidad horaria para dedicarle a la cursada. Además, pueden poner toda su atención en el seguimiento y estudio de las materias que se inscribieron.





### Test y evaluaciones de modelos:

#### Evaluación de la red neuronal:

Teniendo en cuenta únicamente el código de materia, la disponibilidad horaria del usuario y si el usuario trabaja:

- Accuracy: 0.8650000095367432
- Loss: 0.2895848751068115

Tomando en cuenta también la dificultad de las materias

- Accuracy: 0.8815000057220459
- Loss: 0.24686376750469208

Se puede ver que incluir esta información subió el accuracy y bajó el loss, por lo tanto, se mejoró la precisión del modelo

Tomando también en cuenta las horas de clase, de estudio y de practica tambien sube el accuracy:

- Accuracy 0.901199996471405
- Loss 0.20377621054649353

De esta manera, se volvió a mejorar el modelo al tener también en cuenta las horas requeridas por cada materia.

Decidimos que estos valores son satisfactorios, por lo tanto vamos a usar el último modelo evaluado.

#### Evaluación de la velocidad del proceso de recomendación:


Características del algoritmo genético: 200 individuos, 10 generaciones como máximo,

Para que todos los algoritmos realicen la misma cantidad de iteraciones, vamos a descartar la condición extra de corte en caso del estancamiento de la población.

- Sin la estructura “cache”, volviendo a calcular la aptitud en cada etapa de selección: 4 minutos 10 segundos.
- Con la estructura “caché” vacía, calculando una única vez la aptitud de la cursada: 2 minutos y 16 segundos.
- Con la estructura “cache” llenada con varias aptitudes de las cursadas posibles: 1 minuto y 54 segundos.

#### **Link a colab realizado:**

Link al prototipo realizado en google colab:

 TP IA Aplicada Grupo 3.ipynb


#### **Aplicación:**

##### **Implementación.**

Opción 1: Link al repositorio de hugging face para ejecutar la aplicación:

[https://huggingface.co/spaces/MauroC97/TP\\_IA](https://huggingface.co/spaces/MauroC97/TP_IA)

Opción 2: Link a un archivo de google colab para correr la aplicación desde ahí mismo o en la página web de gradio:

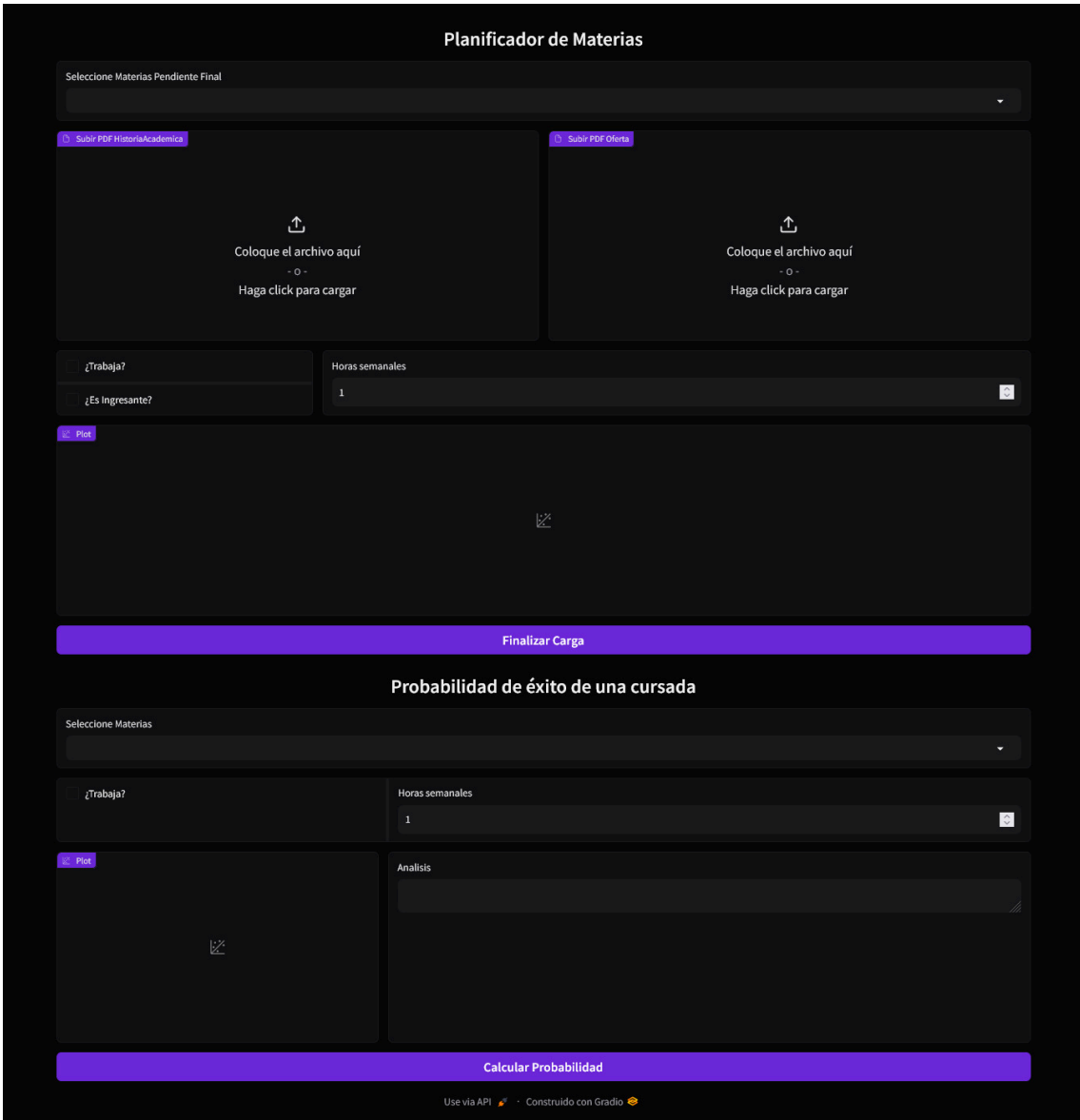
 Aplicación.ipynb

#### Capturas de pantalla del sistema:

El primer ingreso de datos corresponde al caso de uso “Recomendar Cursada”, luego de ingresar los datos y archivos correspondiente, y presionar el botón de finalizar carga, se genera la recomendación de la cursada.

La segunda parte de ingreso de datos corresponde al caso de uso “Calcular probabilidad de Éxito de una Cursada”, luego de ingresar los datos correspondiente, y presionar calcular probabilidad, se genera el cálculo y el análisis de la probabilidad de éxito.

Captura de la aplicación sin carga de Datos:



Ejemplo de la aplicación en funcionamiento:

Planificador de Materias

Seleccione Materias Pendiente Final

Física II

Subir PDF HistoriaAcademica

HistoriaAcademica.pdf

89.0 KB

Subir PDF Oferta

Oferta1C2024.pdf

1.2 MB

¿Trabaja?

¿Es Ingresante?

Horas semanales

60

PLOT

Cursada Recomendada

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Mañana					Análisis de Sistemas	
Tarde						
Noche	Responsabilidad Social Universitaria	Gestión de Proyectos	Ciencia de Datos		Redes de Computadoras	

Finalizar Carga

Probabilidad de éxito de una cursada

Seleccione Materias

Virtualización de Hardware

Inteligencia Artificial Aplicada

Seguridad Aplicada y Forense

Matemática Aplicada

¿Trabaja?

Horas semanales

40

PLOT

73% de Éxito

Analisis

Analisis del Resultado: Es probable que puedas aprobar o promocionar todas las materias, sin embargo, requerirá cierto grado de compromiso y dedicación de tu parte. Podés dejar la cursada tal como está y esforzarte durante todo el cuatrimestre o realizar alguna leve modificación en las materias, o en tu disponibilidad horaria, para aumentar esta probabilidad.

Calcular Probabilidad

Use via API

Construido con Gradio

Captura de la recomendación de Cursada generada:

Cursada Recomendada						
	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Mañana					Análisis de Sistemas	
Tarde						
Noche	Responsabilidad Social Universitaria	Gestión de Proyectos	Ciencia de Datos		Redes de Computadoras	

Ejemplo del Cálculo de la Probabilidad de Éxito de una cursada:



Presentación final:

Hoja de ruta de presentación con los temas a ser abordados

Temas a ser abordados:

- Introducción
- Dominio Elegido
- Problema Identificado
- Solución Propuesta
- Casos de Uso
  - Recomendar Cursada
  - Calcular Probabilidad de éxito de una Cursada
- Recomendar Cursada - Tareas del Sistema
- Calcular Probabilidad de éxito de una Cursada - Tareas del Sistema

- Esquema de Pipeline
  - Descarga e Importación de Librerías
  - Procesamiento del Plan de Carrera
  - Procesamiento de la situación del usuario
  - Creación del Algoritmo Genético
  - Aumentación de Datos
  - Creación de la Red Neuronal
  - Recomendación
- Funcionamiento del Modelo en la Aplicación

### Capturas de presentación final realizada

A continuación incluimos algunas de las diapositivas que preparamos para ilustrar el video de la presentación.



**Dominio Elegido:**

El proyecto se centra en el ámbito educativo, específicamente en la Universidad Nacional de la Matanza (UNLaM).

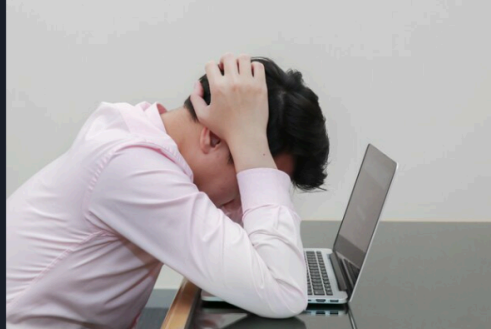
El objetivo es colaborar con una porción importante de la comunidad universitaria, particularmente con los estudiantes de la carrera de grado Ingeniería en Informática, buscando proporcionar una cursada más acorde a la disponibilidad horaria y a la probabilidad de éxito del alumnado.

  
Universidad Nacional  
de La Matanza

  
Departamento de Ingeniería e  
Investigaciones Tecnológicas

## Problema Identificado:

El problema que hemos identificado es la dificultad que tienen muchos estudiantes para organizar su cursada al inicio del cuatrimestre. Por inexperiencia o falta de tiempo, muchas veces terminan eligiendo sus materias en forma poco consciente, lo que puede llevar a que no obtengan el éxito esperado.



## Solución Propuesta:

Por lo cual hemos desarrollado una aplicación con los conocimientos adquiridos en las materias de Inteligencia Artificial e Inteligencia Artificial Aplicada, que permitirá a los estudiantes de Ingeniería en Informática planificar sus horarios de manera más efectiva, teniendo en cuenta sus tiempos disponibles, trabajo y la probabilidad de éxito en cada materia.



## Casos de Uso:

### 1) Recomendar Cursada:

El usuario ingresa la oferta de materias correspondiente al cuatrimestre actual, su historia académica, las materias pendientes de final que posee, su condición laboral y la cantidad de horas que tiene disponible para estudiar. El sistema genera una recomendación de materias en conjunto con el día y horario a cursar, en función a la probabilidad de éxito calculada y los datos proporcionados por el usuario.

### 2) Calcular Probabilidad de Éxito de una Cursada:

El usuario ingresa la cursada que le gustaría realizar, su condición laboral y la cantidad de horas que tiene disponible para estudiar. El sistema indica la probabilidad de éxito de la cursada ingresada en función de los datos proporcionados por el usuario.

## Esquema de Pipeline







## Función de Aptitud:

La función de Aptitud tiene en cuenta 6 parámetros:

- 1) La Probabilidad de Éxito de la cursada (valor entre 0 a 100)
- 2) La dificultad de la cursada (valor entero entre 1 a 10)
- 3) La cantidad requerida de horas semanales (valor entero entre 4 y 168)
- 4) La cantidad de materias que puede llegar a desbloquear instantáneamente (valor entero positivo)
- 5) La cantidad de materias que puede llegar a desbloquear hasta el final de la carrera (valor entero positivo)
- 6) La cantidad de Materias de los primeros años para el título intermedio

El resultado de la función surge de la siguiente operación:

$$\text{Aptitud} = \text{Probabilidad de Éxito} - \text{Penalización de Horas} - \text{Penalización de Dificultad} + \text{Premio Materias Instantáneas Desbloqueadas} + \text{Premio Materias Totales Desbloqueadas} + \text{Premio materias primeros años}$$



## 5) Aumentación de Datos

En esta sección se crea un dataset artificial para simular un historial de cursadas de alumnos, con el fin de poder entrenar la red neuronal que predecirá la probabilidad de éxito de una cursada. Se utiliza un segundo algoritmo genético para generar y seleccionar los individuos más "realistas".

Se comienza ingresando el historial de cursadas de los integrantes del grupo, las cuales se utilizan como punto de partida para inicializar la población y también como referencia para plantear la función de aptitud.

Luego, se define el segundo algoritmo genético para generar los datos de entrenamiento de la Red Neuronal.

Una vez finalizada toda esta sección, se obtiene un dataset final de 100.000 tuplas para entrenar la Red Neuronal.

## B) Definición del Modelo:

La red neuronal se estructura de la siguiente forma:

- **32 Capas de Entrada**, que representan cada parámetro del DataSet (6 códigos de materias, 6 dificultades correspondientes a cada materia, 6 horas de clase, 6 horas de estudio y 6 horas de práctica correspondiente a cada materia, si trabaja y la cantidad de horas semanales de estudio).
- **32 Capas de Normalización de datos**, una para capa de entrada, a fin de estabilizar el entrenamiento de la red y mejorar la convergencia del algoritmo.
- **1 Capa de Concatenación**, que se encarga de concatenar todas las features normalizadas en un único tensor, para luego ser pasado a través de capas posteriores del modelo.
- **1 Capa Densa de 32 neuronas**, que recibe el tensor de la capa de Concatenación y realiza una función de activación ReLu en cada una de sus neuronas.
- **1 Capa Dropout**, con una tasa de abandono del 50%, es decir que la mitad de los valores provenientes de la capa Densa se establecerán en cero utilizando un patrón de selección aleatorio. Por lo cual la salida de esta capa tendrá 16 valores en 0 y 16 que conservarán su valor original de la capa anterior. Esto se realiza principalmente para prevenir el overfitting, generando que el modelo sea más robusto y menos propenso a memorizar el ruido en los datos de entrenamiento.
- **1 Capa Densa de salida con una sola neurona** y sin ninguna función de activación. La salida de esta última capa es la que se utiliza para realizar las predicciones.

## 7) Recomendación

En esta sección final se pone en marcha el algoritmo genético que obtendrá la cursada con mayor aptitud para ser recomendada al usuario. Dicho algoritmo trabajará con poblaciones de 200 individuos, y obtenida la población final se procederá a realizar una selección elitista para quedarse con el individuo más apto. Una vez obtenida la cursada ideal para el usuario, se procederá a mostrar la misma en forma de un calendario semanal, para que el usuario pueda visualizar de forma más amigable la recomendación de cursada.

	Cursada Recomendada					
	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Mañana					Análisis de Sistemas	
Tarde						
Noche	Automatas y Gramáticas	Virtualización de Hardware	Ciencia de Datos	Gestión de las Organizaciones		

### Link a vídeo de presentación realizado

Video de presentación: <https://youtu.be/FghZMJRiPNg>

### Referencias:

1. Arranz de la Peña, Jorge y Parra Truyol, Antonio. Algoritmos Genéticos. Disponible en [https://mielhistorico.unlam.edu.ar/data7/data2/contenido/3664/1127\\_articulo\\_AG.pdf](https://mielhistorico.unlam.edu.ar/data7/data2/contenido/3664/1127_articulo_AG.pdf). Accedido en Mayo 2024
2. Cátedra Inteligencia Artificial Unlam. Algoritmos Genéticos Complemento. Disponible en <https://mielhistorico.unlam.edu.ar/data7/data2/contenido/3664/Algoritmos-Geneticos-Complemento.pdf>. Accedido en Mayo 2024

3. Cátedra Inteligencia Artificial Unlam. Resumen Búsqueda y Optimización. Disponible en <https://mielhistorico.unlam.edu.ar/data7/data2/contenido/3664/Resumen-Busqueda,-Optimizacion-y-Algoritmos-Geneticos.pdf>. Accedido en Mayo 2024
4. Fernando Sancho Caparrini. Algoritmos Genéticos. Disponible en [https://www.cs.us.es/~fsancho/Blog/posts/Algoritmos\\_Geneticos.md.html](https://www.cs.us.es/~fsancho/Blog/posts/Algoritmos_Geneticos.md.html). Accedido en Mayo de 2024
5. Keras. API de capas de Keras. Disponible en <https://keras.io/api/layers/>. Accedido en Mayo 2024.
6. Keras. Funciones de Pérdidas. Disponible en <https://keras.io/api/losses/>. Accedido en Mayo 2024.
7. Keras. Optimizadores para Keras. Disponible en <https://keras.io/api/optimizers/>. Accedido en Mayo 2024.
8. Marcos Gestal Pose. Introducción a los Algoritmos Genéticos. Disponible en <https://cursa.ihmc.us/rid=1KNKMJ4LN-11XXFSG-1KV5/Algoritmos%20de%20Terminos.pdf>. Accedido en Mayo de 2024
9. Pablo Estévez Valencia. Optimización mediante Algoritmos Genéticos. Disponible en [https://www.researchgate.net/profile/Pablo-Estevez-2/publication/228708779\\_Optimizacion\\_Mediante\\_Algoritmos\\_Geneticos/links/0912f51111f82b2a61000000/Optimizacion-Mediante-Algoritmos-Geneticos.pdf](https://www.researchgate.net/profile/Pablo-Estevez-2/publication/228708779_Optimizacion_Mediante_Algoritmos_Geneticos/links/0912f51111f82b2a61000000/Optimizacion-Mediante-Algoritmos-Geneticos.pdf). Accedido en Mayo de 2024
10. Santos, Carlos; Velez-Langs, Oswaldo. Sistemas Recomendadores: Un enfoque desde los algoritmos genéticos. Disponible en <https://www.redalyc.org/pdf/816/81690104.pdf>. Accedido en Mayo de 2024
11. SEDICI UNLP. ALGORITMOS GENÉTICOS. Disponible en [https://sedici.unlp.edu.ar/bitstream/handle/10915/4060/IV\\_-\\_Algoritmos\\_gen%C3%A9ticos.pdf?sequence=8&isAllowed=y](https://sedici.unlp.edu.ar/bitstream/handle/10915/4060/IV_-_Algoritmos_gen%C3%A9ticos.pdf?sequence=8&isAllowed=y). Accedido en Mayo 2024
12. TensorFlow. Clasifique datos estructurados utilizando capas de preprocesamiento de Keras. Disponible en [https://www.tensorflow.org/tutorials/structured\\_data/preprocessing\\_layers](https://www.tensorflow.org/tutorials/structured_data/preprocessing_layers). Accedido en Mayo 2024