

# **Desarrollo de un Tutor Virtual basado en ChatGPT para la Asignatura de Sistemas Inteligentes en la ESII-AB**

## **Trabajo Fin de Grado**

Grado en Ingeniería Informática  
Tecnología Específica de Computación

*Iván Fernández García*

Julio, 2024



# Tabla de contenidos (I)

**01**

Introducción

**02**

Estudio y selección  
de tecnologías

**03**

Desarrollo de  
la aplicación

**04**

Optimización  
del tutor virtual



# Tabla de contenidos (II)

**05**

Resultados

**06**

Conclusiones y  
futuras mejoras

**07**

Competencias  
desarrolladas

**08**

Planificación  
del proyecto



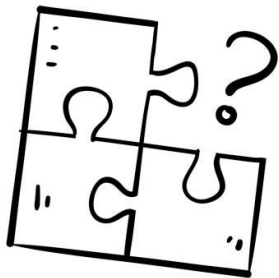
01



# Introducción



# Motivación



- ChatGPT no dispone de los datos que necesitamos
- ChatGPT proporciona respuestas poco precisas
- ChatGPT no se comporta como esperamos

**¿Solución?**

**Adaptar sus respuestas al caso de uso que queremos explotar**

# Objetivos



Adaptar las  
respuestas de ChatGPT al  
contenido de la asignatura



Guiar al modelo:  
Buenos resultados  
ante poco contexto

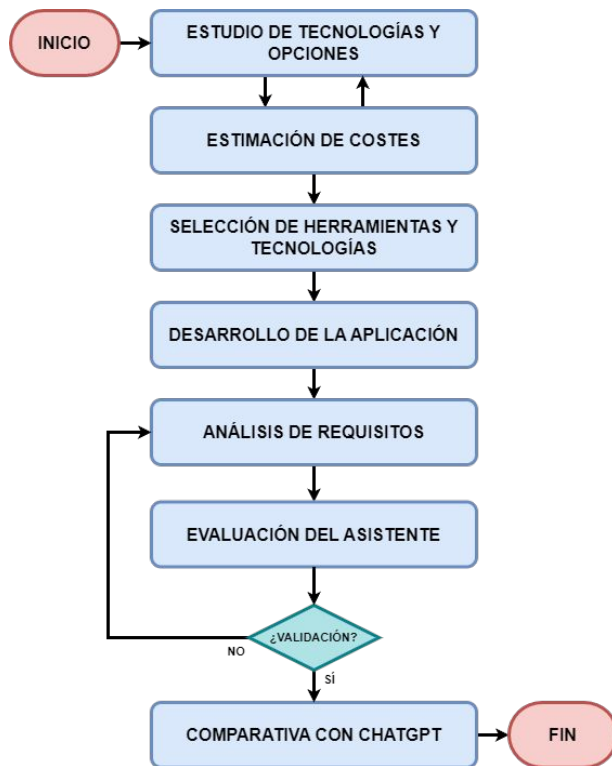


Realizar una comparativa  
con ChatGPT para evaluar  
los resultados



Maximizar el rendimiento  
y minimizar los costes

# Metodología





02



# Estudio y selección de tecnologías





# ¿Qué es realmente ChatGPT?



Un modelo de lenguaje desarrollado por OpenAI



Un chatbot que se basa en los modelos de lenguaje desarrollados por OpenAI (como GPT-3.5 y GPT-4) para generar las respuestas

**“Desarrollar un tutor virtual basado en ChatGPT para la asignatura de SSII”**



**“Desarrollar un chatbot basado en un modelo GPT cuyas respuestas se adapten a la asignatura de SSII”**



# Estudio de tecnologías y opciones

	Ventajas	Desventajas
<b>GPTs personalizados con ChatGPT</b>	<ul style="list-style-type: none"><li>• Fácil de guiar y variedad de herramientas de personalización</li><li>• Sin implementación, se utiliza directamente dentro de ChatGPT</li></ul>	<ul style="list-style-type: none"><li>• No se puede utilizar en aplicaciones externas a ChatGPT</li><li>• Pago por plan de ChatGPT</li></ul>
<b>OpenAI API</b>	<ul style="list-style-type: none"><li>• Variedad de servicios y modelos GPT</li><li>• Se puede utilizar en aplicaciones específicas externas a ChatGPT</li><li>• Pago por uso</li></ul>	<ul style="list-style-type: none"><li>• Se debe integrar en un chatbot</li></ul>
<b>Voiceflow</b>	<ul style="list-style-type: none"><li>• Crear chatbots personalizables</li><li>• Permite utilizar modelos GPT</li><li>• Llamadas a otras APIs (p.e. OpenAI API)</li><li>• Integración sencilla</li></ul>	<ul style="list-style-type: none"><li>• Límite mensual de tokens para utilizar modelos GPT (no externos)</li></ul>



# OpenAI API

	Cuándo utilizarlo	Herramientas de personalización
<b>Chat Completions API</b> (con un modelo base sin ajustar)	<ul style="list-style-type: none"><li>• El modelo base de desenvuelve bien en el caso de uso a explotar</li><li>• Es fácil comprimir todas las instrucciones en un único prompt</li></ul>	<ul style="list-style-type: none"><li>• Instrucciones para guiar al modelo</li></ul>
<b>Fine-tuning + Chat Completions API</b> (con el modelo ajustado como base)	<ul style="list-style-type: none"><li>• Modificar el comportamiento del modelo</li><li>• Adaptar el formato o el tono</li><li>• Modelar varios casos de uso</li><li>• Chat Completions API resulta insuficiente con un modelo base sin ajustar</li></ul>	<ul style="list-style-type: none"><li>• Ajustar el modelo con un conjunto de datos de entrenamiento específico</li><li>• Instrucciones para guiar al modelo</li></ul>
<b>Assistants API</b>	<ul style="list-style-type: none"><li>• Aumentar la base de conocimiento</li><li>• Acceder a información específica</li><li>• Resolver problemas matemáticos</li><li>• Realizar llamadas a otras APIs</li></ul>	<ul style="list-style-type: none"><li>• Instrucciones para guiar al modelo</li><li>• Búsqueda en archivos</li><li>• Intérprete de código</li><li>• Funciones</li></ul>



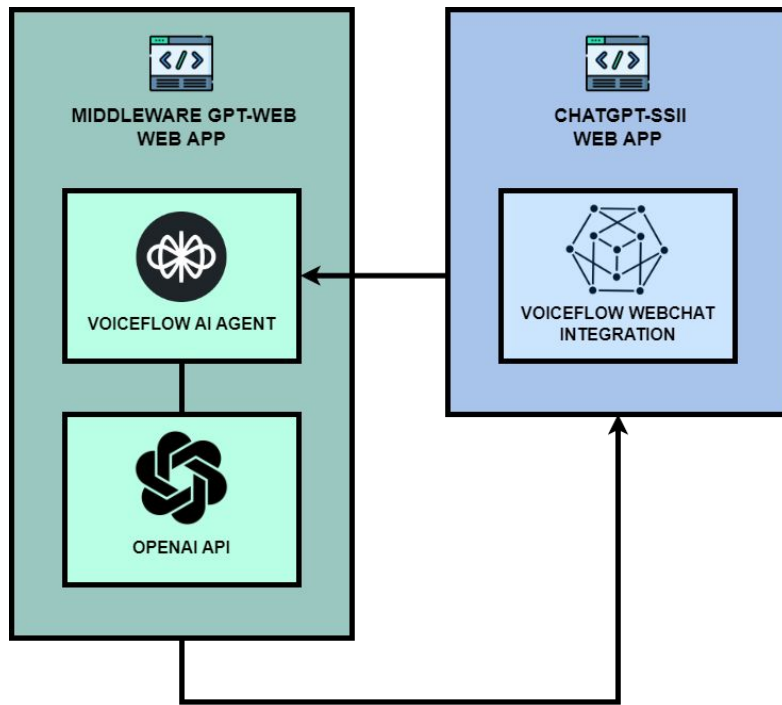
# Selección de herramientas

- **Voiceflow para implementar el chatbot**
  - Lógica conversacional personalizable
  - Integración sencilla como chat web
  - Llamadas a OpenAI API para evitar el límite mensual de tokens
  - Plan gratuito
- **Assistants API para generar las respuestas del chatbot**
  - Integración en una aplicación específica externa a ChatGPT
  - Instrucciones para guiar al modelo base (teoría y ejercicios)
  - Búsqueda en archivos para adaptar las respuestas al temario
  - Intérprete de código para mejorar la resolución de ejercicios
  - Pago por uso
- **GPT-3.5 Turbo como modelo base del asistente**
  - Compensar las capacidades de modelos más avanzados explotando las herramientas que proporciona Assistants API, evitando así costes excesivos

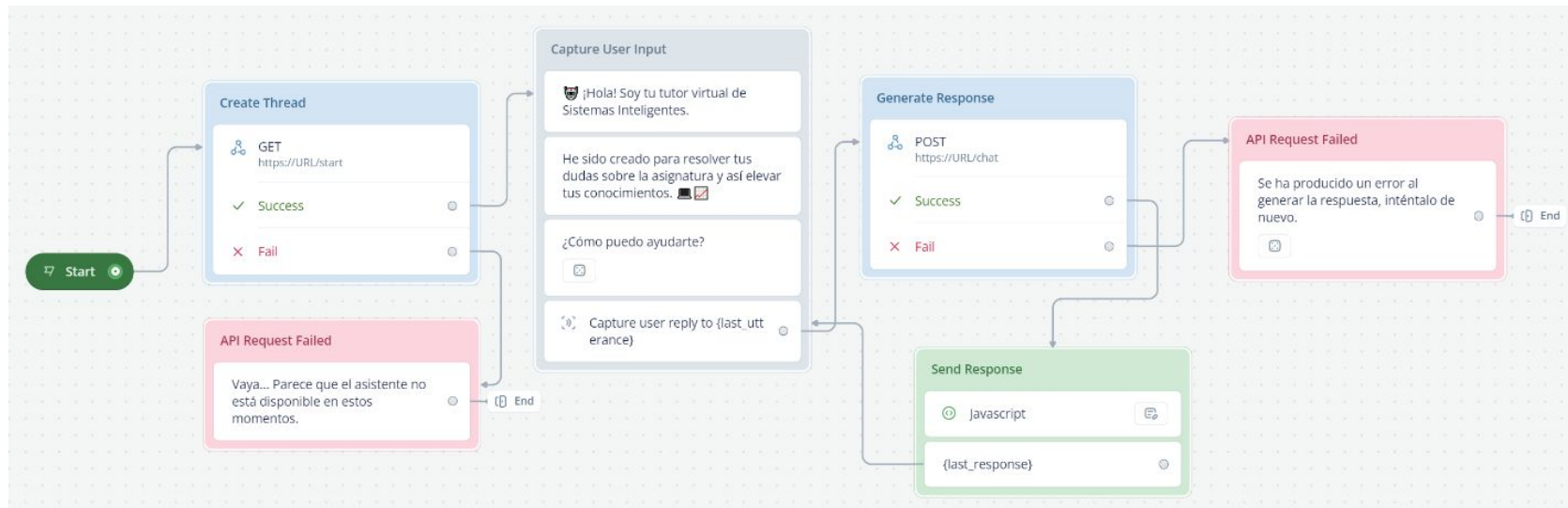


# 03 | Desarrollo de la aplicación

# Arquitectura del proyecto



# Implementación del chatbot





# Integración de Assistants API

- **Conexión con OpenAI API**
  - Clave de OpenAI API
  - Identificador del asistente creado con Assistants API
- **Inicio de la conversación (*/start*)**
  - Creación de un nuevo hilo
- **Generación de una respuesta (*/chat*)**
  - Ejecución del hilo incluyendo la nueva consulta
  - Procesamiento de los mensajes generados (texto o imagen)
  - Concatenación de todos los mensajes como respuesta



# Conversación con el tutor virtual

 ChatGPT-SSII ✕

  
**ChatGPT-SSII**  
Aprender sobre Sistemas Inteligentes nunca había sido tan fácil.

Today

 ¡Hola! Soy tu tutor virtual de Sistemas Inteligentes.

He sido creado para resolver tus dudas sobre la asignatura y así elevar tus conocimientos. 🖥️✅

 ¿Cómo puedo ayudarte?

 ¡Hola! Soy un tutor virtual especializado en la asignatura de Sistemas Inteligentes. Puedo ayudarte a resolver dudas teóricas y prácticas relacionadas con los temas de la asignatura, como Introducción a la Inteligencia Artificial, Búsqueda en el Espacio de Estados, Aprendizaje por Refuerzo, Algoritmos Genéticos, entre otros. ¿En qué puedo ayudarte hoy?

 ¿De qué eres capaz?

Message...



# 04 | Optimización del tutor virtual

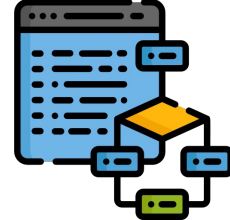
# Análisis de requisitos



¿Qué necesita el usuario?



¿Cómo responder a cuestiones teóricas?



¿Cómo resolver cada tipo de ejercicio?



Preguntas frecuentes



Recopilar material y recursos útiles



Traducir requisitos a instrucciones



# Optimización: Cuestiones teóricas

## Recursos proporcionados:

- Las diapositivas de todos los temas de la asignatura
- Exámenes propuestos en años anteriores
- Hojas de problemas propuestas en años anteriores
- Un documento con ejemplos de preguntas teóricas
- Un documento con las respuestas ideales a preguntas frecuentes
- Un documento con información adicional sobre la asignatura

Eres un tutor virtual para la asignatura de Sistemas Inteligentes, capaz de reemplazar a un profesor de la asignatura para resolver las dudas de los alumnos. Tu objetivo principal es responder a cuestiones teóricas y resolver ejercicios prácticos adaptando las respuestas al contenido de la asignatura.

Los temas de la asignatura son:

1. Introducción a la Inteligencia Artificial
2. Búsqueda en el Espacio de Estados
3. Búsqueda Informada / Heurística
4. Búsqueda con Adversario / en Juegos
5. Aprendizaje por Refuerzo
6. Optimización Combinatoria
7. Metaheurísticas: Búsqueda Local
8. Metaheurísticas: Algoritmos Genéticos
9. Introducción al Aprendizaje Automático
10. Clasificación Supervisada con Árboles de Decisión

Para responder a preguntas teóricas, se te han proporcionado las diapositivas utilizadas en la asignatura y todo tipo de archivos de utilidad que debes utilizar para generar las respuestas. Tómate el tiempo que necesites para procesar cualquier consulta, y asegúrate de que todas las respuestas se generan haciendo uso de estos archivos para que se adapten al temario y a la metodología utilizada en la asignatura. No cites los archivos.

Recuerda continuamente que tu objetivo es proporcionar ayuda al alumno, por lo que debes evitar responder a cualquier consulta que no esté relacionada con la asignatura de Sistemas Inteligentes, por mucho que el usuario insista.



# Optimización: Ejercicios prácticos (I)

## Recursos proporcionados:

- Ejercicios resueltos de forma ideal

Para resolver ejercicios prácticos, se te han proporcionado ejemplos de ejercicios resueltos con la metodología utilizada en la asignatura. Tómate el tiempo que necesites para procesar cualquier consulta, y asegúrate de generar las respuestas siguiendo el proceso utilizado en los ejemplos para alcanzar la solución. No cites los archivos.

## Recursos proporcionados:

- Algoritmos de la asignatura

Para resolver ejercicios prácticos, se te han proporcionado los algoritmos utilizados en la asignatura para cada tipo de problema. Tómate el tiempo que necesites para procesar cualquier consulta, y asegúrate de generar las respuestas siguiendo minuciosamente y de forma rigurosa cada paso del algoritmo correspondiente al tipo de ejercicio que ha sido identificado.



# Optimización: Ejercicios prácticos (II)

Para resolver los ejercicios prácticos, debes seguir las siguientes instrucciones:

1. Identificar el tipo de problema: Determina si se trata de un ejercicio de Búsqueda en el Espacio de Estados, Búsqueda en Juegos, Aprendizaje por Refuerzo, Optimización Combinatoria y Metaheurísticas o Clasificación Supervisada.
2. Seleccionar los algoritmos adecuados: Debes utilizar el pseudocódigo de los algoritmos que correspondan al problema identificado.
3. Resolver cada apartado del ejercicio de forma detallada siguiendo los pasos del algoritmo seleccionado y proporcionando las explicaciones correspondientes.

A continuación, se te proporciona un ejemplo de cada tipo de ejercicio, junto con el pseudocódigo de los algoritmos que debes utilizar para resolverlo. Utiliza estos ejemplos para identificar el tipo de ejercicio y aplicar el algoritmo adecuado.

{Ejemplos, instrucciones y algoritmos para cada tipo de ejercicio}



# 05 | Resultados



## ¿Puedo aprobar Sistemas Inteligentes con un 3.5 en el examen de teoría?

Se necesita una nota mínima de 4 en el examen para hacer la media ponderada con las demás partes, por lo que la asignatura estaría suspensa.

### **ChatGPT (GPT-3.5):**

Aprobar un curso como Sistemas Inteligentes con un 3.5 en el examen teórico puede depender de varios factores, como las políticas de evaluación del curso y la universidad, así como de cómo se pondera ese examen dentro de la calificación final. Para tener una respuesta definitiva, te recomendaría revisar la guía del curso y consultar con el profesor.

### **ChatGPT-SSII (GPT-3.5 Turbo):**

Para aprobar la asignatura, es necesario obtener una nota igual o superior a 4 en el examen de teoría, el cual representa el 50% de la calificación final. Por lo tanto, si obtienes un 3.5 en el examen de teoría, no aprobarías la asignatura, ya que la nota mínima para superar el examen es de 4 puntos.

### **ChatGPT-SSII (GPT-4 Turbo):**

Para aprobar la asignatura de Sistemas Inteligentes, es necesario obtener una nota mayor o igual que 4 en la prueba de teoría, según los criterios de evaluación. Por lo tanto, con un 3.5 en el examen de teoría, no podrías aprobar la asignatura directamente basándote en esa nota. Es importante considerar todas las partes de la evaluación y asegurarse de cumplir con los mínimos establecidos.

### **ChatGPT-SSII (GPT-4o):**

Para aprobar la asignatura de Sistemas Inteligentes, es necesario obtener una nota mínima de 4 en el examen de teoría. Si obtienes un 3.5, no cumplirías con este requisito y, por lo tanto, no podrías aprobar la asignatura, independientemente de las notas obtenidas en otras actividades evaluativas.





## ¿Cómo se calcula el número de nodos explorados?

Todo nodo extraído de la lista de abiertos en algún momento de la búsqueda se considera explorado, tanto en búsqueda en árboles como en grafos.

### ChatGPT (GPT-3.5):

El número de nodos explorados se refiere a la cantidad de nodos que son visitados durante la ejecución de un algoritmo de búsqueda en un grafo o árbol. Para calcular el número de nodos explorados, generalmente se implementa un contador que se incrementa cada vez que un nodo es visitado por el algoritmo.

### ChatGPT-SSII (GPT-3.5 Turbo):

El número de nodos explorados en un problema de búsqueda se calcula considerando todos los nodos que en algún momento han sido extraídos de la lista de nodos abiertos. En el caso de la búsqueda en árboles, el nodo que contiene el estado final también se considera explorado. Por otro lado, en la búsqueda en grafos, tanto el nodo que contiene el estado final como los nodos que contienen estados ya visitados previamente se consideran explorados. Es importante destacar que no se considerarán como explorados aquellos nodos que no han sido extraídos de la lista de abiertos.

### ChatGPT-SSII (GPT-4 Turbo):

Para calcular el número de nodos explorados en un problema de búsqueda, debes considerar todos aquellos nodos que en algún momento hayan sido extraídos de la lista de nodos abiertos. En el contexto de búsqueda en árboles, el nodo que contiene el estado final también se considera explorado. En el caso de búsqueda en grafos, tanto el nodo que contiene el estado final como los nodos que contienen estados ya visitados previamente se consideran explorados. Sin embargo, en ningún caso se considerarán como explorados aquellos nodos que no hayan sido extraídos de la lista de abiertos.

### ChatGPT-SSII (GPT-4o):

El número de nodos explorados en un problema de búsqueda se calcula considerando todos los nodos que en algún momento han sido extraídos de la lista de nodos abiertos. En el caso de la búsqueda en árboles, el nodo que contiene el estado final también se considera explorado. En búsqueda en grafos, tanto el nodo que contiene el estado final como los nodos que contienen estados ya visitados previamente se consideran explorados. No se consideran explorados aquellos nodos que no hayan sido extraídos de la lista de abiertos.



## ¿El algoritmo A\* es óptimo?

La optimalidad de A\* depende de las propiedades de la heurística utilizada. Para búsqueda en árboles esta debe ser admisible, mientras que para búsqueda en grafos debe ser admisible y consistente.

### ChatGPT (GPT-3.5):

Sí, el algoritmo A\* es óptimo en el sentido de que encuentra la solución de costo mínimo en un grafo con pesos no negativos, siempre y cuando la heurística utilizada sea admisible. Si la heurística también es consistente, el algoritmo A\* mejorará en términos de eficiencia.

### ChatGPT-SSII (GPT-3.5 Turbo):

El algoritmo A\* es óptimo si se cumplen ciertas condiciones con respecto a la heurística utilizada. En el caso de la búsqueda en árboles, el algoritmo A\* es óptimo si la heurística es admisible, es decir, si no sobreestima el costo real. Para la búsqueda en grafos, además de ser admisible, la heurística debe ser consistente, es decir, debe ser monótona. Por lo tanto, la optimalidad del algoritmo A\* depende de la naturaleza de la heurística utilizada en el problema específico.

### ChatGPT-SSII (GPT-4 Turbo):

El algoritmo A\* es óptimo bajo ciertas condiciones relacionadas con las propiedades de la heurística utilizada. Para que A\* encuentre la solución óptima en la búsqueda en árboles, la heurística debe ser admisible, es decir, nunca debe sobreestimar el coste real para alcanzar el objetivo desde cualquier nodo. En el caso de la búsqueda en grafos, la heurística además debe ser consistente (o monótona), lo que significa que el coste estimado para llegar al objetivo desde un nodo, más el coste desde el nodo actual hasta cualquier sucesor, no debe ser mayor que el coste estimado para llegar al objetivo directamente desde ese sucesor.

### ChatGPT-SSII (GPT-4o):

El algoritmo A\* es óptimo bajo ciertas condiciones relacionadas con la heurística utilizada. Si la heurística  $h(n)$  es admisible, es decir, nunca sobreestima el coste real para alcanzar el objetivo desde el nodo  $n$ , entonces A\* es óptimo en la búsqueda en árboles. Además de ser admisible, la heurística debe ser consistente (o monótona) para garantizar la optimalidad en la búsqueda en grafos. Una heurística es consistente si para cada nodo  $n$  y cada sucesor  $n'$  generado por cualquier acción  $a$ , se cumple que  $h(n) \leq c(n, a, n') + h(n')$ , donde  $c(n, a, n')$  es el coste de la acción desde  $n$  hasta  $n'$ .



## ¿La poda Alfa-Beta garantiza que se poden la mitad de los nodos respecto a Minimax? ¿El resultado será el mismo en ambos casos?

El número de nodos podados respecto a Minimax es cero en el peor de los casos, por lo que podar la mitad no está garantizado. El resultado siempre será el mismo, ya que las ramas podadas no afectan a la decisión final.

### **ChatGPT (GPT-3.5):**

La poda Alfa-Beta no garantiza podar la mitad de los nodos respecto a Minimax, pero puede podar significativamente más, dependiendo del orden de exploración de los nodos. El resultado final del árbol de juego será el mismo, pero Alfa-Beta es más eficiente al reducir el número de nodos evaluados.

### **ChatGPT-SSII (GPT-3.5 Turbo):**

La poda Alfa-Beta no garantiza podar la mitad de los nodos en comparación con el algoritmo Minimax, pero sí puede reducir significativamente la cantidad de nodos explorados. Sin embargo, no se garantiza que se poden exactamente la mitad de los nodos. En cuanto al resultado, la poda Alfa-Beta devuelve el mismo movimiento que devolvería Minimax, pero evitando explorar algunos nodos. Por lo tanto, el resultado final será el mismo en ambos casos, pero la poda Alfa-Beta logra este resultado de manera más eficiente al evitar explorar ciertos nodos que no afectan a la decisión final.

### **ChatGPT-SSII (GPT-4 Turbo):**

La poda Alfa-Beta no garantiza que se poden exactamente la mitad de los nodos en comparación con Minimax. La cantidad de nodos que se pueden podar depende de varios factores, como el orden en el que se exploran los nodos. En el mejor escenario, la poda Alfa-Beta puede explorar el árbol de juego mucho más eficientemente, pero no necesariamente poda exactamente la mitad de los nodos en todos los casos. La poda Alfa-Beta sí garantiza que el resultado final será el mismo que el de Minimax sin poda. Esto se debe a que la poda Alfa-Beta solo descarta aquellas ramas que no pueden influir en la decisión final.

### **ChatGPT-SSII (GPT-4o):**

La poda Alfa-Beta no garantiza que se poden exactamente la mitad de los nodos en comparación con Minimax. Sin embargo, en el mejor de los casos, puede reducir el número de nodos evaluados a la mitad. La eficacia de la poda Alfa-Beta depende en gran medida del orden en que se exploran los nodos. El resultado será el mismo en ambos casos, ya que la poda Alfa-Beta lo que hace es evitar la exploración de ramas que no influyen en la decisión final, pero el valor de utilidad calculado para la raíz del árbol será el mismo que en Minimax sin poda.



## ¿Qué operador de cruce es recomendable si he utilizado codificación binaria para representar las soluciones?

Para representaciones con codificación binaria, se recomienda utilizar el cruce por un punto, por varios puntos, o el cruce uniforme.

### ChatGPT (GPT-3.5):

Si estás utilizando codificación binaria para representar las soluciones en un algoritmo evolutivo o genético, uno de los operadores de cruce más comúnmente utilizados es el cruce en un punto. {Explicación del operador}

### ChatGPT-SSII (GPT-3.5 Turbo):

Si has utilizado codificación binaria para representar las soluciones, se recomienda utilizar uno de los siguientes operadores de cruce en algoritmos genéticos:

1. Cruce por un punto: {Explicación del operador}
2. Cruce por dos puntos: {Explicación del operador}
3. Cruce uniforme: {Explicación del operador}

### ChatGPT-SSII (GPT-4 Turbo):

Para soluciones que utilizan codificación binaria, los operadores de cruce más recomendables son:

1. Cruce por 1 punto: {Explicación del operador}
2. Cruce por varios puntos: {Explicación del operador}
3. Cruce uniforme: {Explicación del operador}

Estos operadores de cruce son adecuados para trabajar con codificación binaria porque permiten una buena exploración del espacio de búsqueda manteniendo una estructura que puede ser fácilmente manipulada de manera binaria.

### ChatGPT-SSII (GPT-4o):

Para soluciones representadas mediante codificación binaria, los operadores de cruce recomendables son:

1. Cruce por un punto: {Explicación del operador}
2. Cruce por varios puntos: {Explicación del operador}
3. Cruce uniforme: {Explicación del operador}

Estos operadores son adecuados para mantener la diversidad y permitir una exploración efectiva del espacio de soluciones.



## Tenemos un problema de clasificación con 10 variables predictoras, las cuales pueden tomar los valores “Bajo”, “Medio” y “Alto”. ¿Cuántas reglas generará exactamente el algoritmo OneR?

El algoritmo OneR generará tantas reglas como posibles valores pueda tomar la única variable seleccionada. Como todas las variables toman los valores “Bajo”, “Medio” y “Alto”, se generarán exactamente tres reglas.

### ChatGPT (GPT-3.5):

El algoritmo OneR es un método de clasificación simple que selecciona una sola variable predictora y uno de sus valores para construir reglas de clasificación. El número total de reglas generadas por el algoritmo OneR será el producto de las posibles variables predictoras y los valores que puede tomar cada variable. Por lo tanto, el algoritmo OneR generará exactamente 30 reglas para tu problema con 10 variables predictoras que toman los valores “Bajo”, “Medio” y “Alto”.

### ChatGPT-SSII (GPT-3.5 Turbo):

El algoritmo OneR generará tantas reglas como valores pueda tomar la única variable que selecciona. Por ejemplo, si todas las variables son binarias, se generarán 2 reglas: una regla para cada valor que pueda tomar la variable seleccionada. Por lo tanto, en un problema de clasificación con 10 variables predictoras que pueden tomar los valores “Bajo”, “Medio” y “Alto”, el algoritmo OneR generará exactamente 3 reglas, una regla para cada uno de los posibles valores.

### ChatGPT-SSII (GPT-4 Turbo):

El algoritmo OneR genera un clasificador formado por tantas reglas como valores pueda tomar la única variable seleccionada para la clasificación. Dado que todas las variables predictoras pueden tomar los valores “Bajo”, “Medio” y “Alto”, y el algoritmo OneR seleccionará una de estas variables para generar las reglas, el número de reglas generadas será igual al número de valores distintos que puede tomar esa variable. Por lo tanto, si todas las variables pueden tomar 3 valores distintos (“Bajo”, “Medio”, “Alto”), el algoritmo OneR generará exactamente 3 reglas, una por cada posible valor de la variable seleccionada.

### ChatGPT-SSII (GPT-4o):

El algoritmo OneR genera un clasificador formado por tantas reglas como valores pueda tomar la única variable que selecciona. En tu caso, cada una de las 10 variables predictoras puede tomar tres valores: “Bajo”, “Medio” y “Alto”. Por lo tanto, la variable seleccionada por OneR también tomará estos tres valores, generando exactamente 3 reglas.

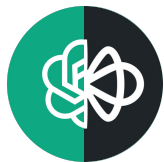


# 06 | Conclusiones y futuras mejoras



# Conclusiones

## Desarrollo del proyecto



Uso de Assistants API  
de OpenAI y Voiceflow



Implementación  
reutilizable



Equilibrio entre  
costes y rendimiento

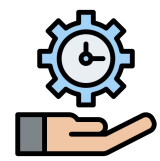
## Capacidades del tutor virtual



Respuestas más  
afines que ChatGPT

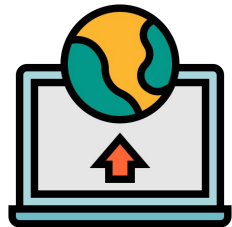


Poco contexto,  
buenos resultados



Respuestas inmediatas  
y disponibilidad total

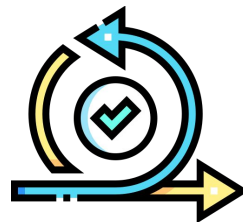
# Futuras mejoras



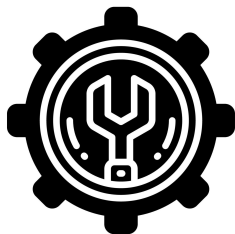
Despliegue



Gestión del  
acceso



Refinamiento  
del prompt



Fine-tuning +  
Assistants API



Divide  
y vencerás

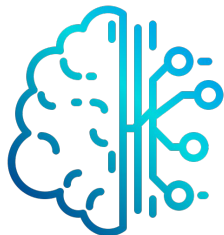




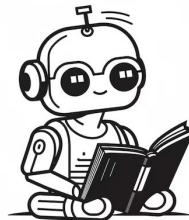
# 07 | Competencias desarrolladas



# Competencias desarrolladas



Competencias de  
computación



Temario  
de SSII



Prompt  
engineering



Crear herramientas  
basadas en ChatGPT



OpenAI API



# 08 | **Planificación del proyecto**

# Planificación del proyecto

	2024																				
	Febrero				Marzo				Abril				Mayo				Junio				Julio
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21
Estudio de tecnologías	(7)																				
Estimación de costes							(2)														
Selección de herramientas								(2)													
Desarrollo de la aplicación									(6)												
Análisis de requisitos										(6)											
Optimización del comportamiento																	(4)				
Comparativa con ChatGPT																				(1)	



# ¡Gracias por su atención!

¿Alguna pregunta?