

# Inteligencia Artificial

## Proyecto 3: Diseño de un sistema de recomendación

Gutiérrez Sánchez Claudia Itzel

Hernández Aguilar Luis Alberto

Macías Gómez Jorge

Méndez Gallegos Ligia Natalia

Velázquez Nolasco Carlos Arturo

31 Enero 2021

## 1 Introducción

### 1.1 Introducción

Día a día usamos aplicaciones que usan sistemas de recomendación para brindarnos la mejor experiencia de usuario, desde redes sociales como Facebook, plataformas de streaming como Twitch e inclusive es usado en el internet de las cosas, como en los refrigeradores inteligentes para reemplazar los faltantes. Sin embargo, nosotros los estudiantes no contamos con tantas opciones, por lo que hemos decidido crear un recomendador de materias, específicamente para nosotros los computólogos.

Los sistemas de recomendación forman parte de un sistema de filtrado de información, los cuales presentan distintos tipos de temas o ítems de información que son del interés de un usuario en particular. Generalmente, un sistema recomendador compara el perfil del usuario con algunas características de referencia de los temas, y busca predecir el baremo o ponderación que el usuario le daría a un ítem que aún el sistema no ha considerado. Estas características pueden basarse en la relación o acercamiento del usuario con el tema o en el ambiente social del mismo

usuario.

### 1.2 Descripción del problema

Existen muchas materias en la carrera, y muchas de ellas involucran muchos temas diferentes, así pues llega a resultar complicado la selección de materias relacionadas con algunos intereses en específico.

En este proyecto se busca diseñar un sistema de recomendación de materias para los estudiantes de la carrera de Ciencias de la Computación utilizando una base de conocimientos y los sistemas de recomendación basados en contenido.

- **Objetivo:** Diseño de una inteligencia artificial(sistema de recomendación) capaz de recomendar materias a alumnos de Ciencias de la Computación con base en sus intereses.
- **Sistema de recomendación utilizado:** Para poder lograr dicho objetivo requerimos de un sistema de recomendación de contenido, ya que ocuparemos un agente que conoce la información de cada uno de los planes estudio de la carrera y busca e identifica palabras en cada uno de ellos que se asocian al interés del estudiante.

### 1.3 Algoritmo TF-IDF

Para lograr nuestro objetivo se plantea la utilización del algoritmo TF-IDF. Ya que basaremos nuestras recomendaciones en la información que se encuentra en distintos documentos y el área de interés que se nos haya dado, creemos que la utilización de dicho algoritmo sería útil para dar una recomendación.

El **Algoritmo TF-IDF** consiste en estimar la relevancia de un documento dependiendo de un término de interés. En nuestro caso podemos recibir N términos de interés y el algoritmo mide con qué frecuencia aparece un término o frase dentro de un documento y lo compara con el número de documentos en el que los términos aparecen.

Debido a su funcionamiento creemos que es el algoritmo más útil para cumplir nuestro objetivo, ya que encuentra los planes de estudio más afines a los términos de interés que se le haya dado al agente mediante una búsqueda por todos los planes de estudio de los términos que se hayan dado y devuelve los documentos más importantes basados en esa búsqueda.

### 1.4 Características del usuario

El usuario planeado para estas recomendaciones debe cumplir con las siguientes características:

- Ser estudiante de la carrera de Ciencias de la Computación o tener algún interés por tomar alguna materia de la carrera relacionada a algún tema de esta (Inteligencia Artificial, Programación, etc.)

### 1.5 Características del producto

Nuestro producto en este sistema de recomendación serían las materias de la carrera de Ciencias de la Computación, cada una cuenta con su clave, número de semestre y plan de estudio.

### 1.6 Enfoque del problema

Es un agente reactivo simple, ya que percibe su mundo actual -que son las materias de la licenciatura- y toma decisiones de acuerdo a estas -recomendar asignaturas-.

A través de los años la inteligencia artificial ha desarrollado enfoques distintos y cada uno ha sido concebido a partir de dotar a agentes de razonamiento similar al de un ser humano. Los enfoques son los siguientes:

- **Pensamiento Humano:** Cuyo objetivo es lograr que las computadoras piensen como humanos.
- **Pensamiento Racional:** Cuyo objetivo es lograr que las computadoras tomen decisiones racionales.
- **Actuar Humano:** Cuyo objetivo es crear máquinas que cumplan objetivos como lo harían los humanos.
- **Actuar Racionalmente:** Cuyo objetivo es crear máquinas que puedan cumplir objetivos racionalmente.

El enfoque que sigue nuestro agente es el de **Pensamiento Racional**, ya que queremos que el agente sea capaz de dar las mejores recomendaciones analizando los distintos planes de estudio.

## 2 Agente

### 2.1 Entorno de trabajo

La especificación del entorno de trabajo está dada por el *REAS* de nuestro agente recomendador, que se desglosa en cuatro puntos: *Rendimiento*, *Entorno*, *Actuadores* y *Sensores*:

REAS: Agente Recomendador			
Rendimiento	Entorno	Actuadores	Sensores
Coincidencia entre las recomendaciones dadas y el área de interés, tiempo de respuesta del agente	Interfaz	Base de conocimientos de planes de estudio, palabras a tomar en cuenta	Lector de materias.

- **Medidas de rendimiento** Son los criterios que determinan el éxito en el comportamiento del agente.
  - Recomendaciones dadas:  
El objetivo del agente es dar recomendaciones al usuario de cualquier materia que le interese.
  - Tiempo de respuesta dada del agente:  
La respuesta de recomendador que da el agente es instantánea.
- **Entorno** Es parte del ambiente donde el agente va a actuar y percibir.
  - Interfaz:  
Es donde el estudiante preguntará por una recomendación.
- **Actuadores** Elementos del agente que reaccionan al ambiente para hacer algo.
  - Base de conocimiento de planes de estudio:  
Es donde se tendrán los planes de estudio de cada materia
  - Palabras a tomar en cuenta:  
El agente cuenta con un recurso de palabras de escapa, palabras que se van a

ignorar en la búsqueda de sugerencias. También llamadas "stop-words" consiste en un diccionario de palabras irrelevantes para nuestras sugerencias, por ejemplo las conjunciones o preposiciones, a este conjunto de palabras le agregaremos palabras que encontramos en todos los documentos que no aportan a los contenidos de los planes, son palabras que son parte de plantilla que siguen los planes; ejemplos de estas palabras son : "CARRERA", "MATEMATICO", "SERIACION", "INDICATIVA", "ANTECEDENTE", "MODALIDAD", "CURSO", "CARACTER", "TEORICAS", "PRACTICAS", "CREDITOS", "HORAS", "SEMANA", "SEMESTRE", "CLAVE", "FACULTAD", "CIENCIAS", "OBJETIVO", "BIBLIOGRAFIA", "BASICA", "COMPLEMENTARIA", "SUGERENCIA", "PARA", "LA", "EVALUACION", "DE", "LA", "ASIGNATURA"

- **Sensores** Los sensores es el medio por el cual el agente recibe las percepciones.
  - Lector de materias:  
Es lo que el agente leerá para poder analizarlo y recomendar una materia.

## 2.2 Conocimiento

La información que usa el agente para recomendar son los planes de estudio de las materias y el conocimiento es la recomendación que da como respuesta.

- **Esquema:** Contamos con una base de conocimientos similar a una base de datos, el mapa curricular de la carrera junto con los planes de estudios de cada una de las materias.
- **Propiedades del esquema:** El esquema tiene **adecuación representacional**, ya que contiene toda la información necesaria para la toma de decisiones del agente cuando hace una recomendación, y **eficiencia en la adquisición**

ya que solo es necesario agregar al mapa curricular que tiene en su base de conocimientos la nueva materia a tomar en cuenta, su clave, nombre y semestre, junto con el pdf del plan de estudio.

- **Tipo de Conocimiento:** Conocimiento relacional simple, ya que nuestro agente solo se encarga de buscar en los planes de estudios palabras afines al área de interés y devuelve el nombre de la materia, el semestre y el clave de esa materia.
- **Estrategias para manipular el conocimiento:** La única forma que tenemos de enriquecer nuestro conocimiento es agregando información a nuestra base de conocimientos inicial.

## 2.3 Propiedades del entorno

- **Entorno de trabajo:** Es la interfaz con la que el usuario interactuará.
- **Observable:** El entorno de trabajo es totalmente observable ya que los sensores detectan todos los aspectos que son relevantes en la toma de decisiones. Es decir, las recomendaciones dadas por el agente.
- **Determinista:** El entorno es determinista ya que el siguiente estado del medio está totalmente determinado por el estado actual y la acción ejecutada por el agente.
- **Episódico:** El entorno es episódico ya que no se toma en cuenta las decisiones previas, por lo que la decisión actual no le afecta que la próxima fase sea defectuosa
- **Discreto:** El entorno es discreto ya que el agente tiene un número discreto de palabras claves en la base de conocimiento y reglas para dar una recomendación.
- **Multiagente:** El entorno es multiagente ya que tenemos dos agentes, el estudiante y el recomendador.

## 2.4 Propósito del Agente

Devolver una serie de materias recomendadas basadas en el área de interés que se le haya dado y la información con la que cuenta sobre las distintas materias que hay en la carrera de Ciencias de la Computación. Mediante la búsqueda de palabras asociadas al área de interés en los planes de estudio de cada materia.

## 2.5 Función del agente

El agente al ser llamado por el usuario, recibe los intereses de este y los procesa junto con el algoritmo para desplegar una lista de materias recomendadas.

## 2.6 Implementación del algoritmo TF-IDF

Como se comentó en la descripción del problema, se diseñó un sistema de recomendaciones de contenido, para lograr esto se utilizó el algoritmo TF-IDF.

Para la realización del agente recomendador se decidió utilizar la función `TfidfVectorizer` de la biblioteca `scikit-learn`, la cuál es utilizada para machine learning; esta función nos ayuda a crear una matriz de similitudes ocupando las palabras de los planes de estudio de todas las materias de la carrera y los intereses del estudiante. Finalmente con base a las puntuaciones de similitud, el agente le muestra al estudiante las materias afines a sus intereses.

Se debe tomar en cuenta que se consideraron una serie de palabras que el TF-IDF Vectorizer ignora ya que eran palabras que se repetían en la mayoría de los planes de estudio.

Todos los detalles sobre las funciones utilizadas están presentes en la implementación y debidamente documentadas para explicar su necesidad.

## 2.7 Ejemplo de Funcionamiento

A continuación un ejemplo para explicar detalladamente el funcionamiento del sistema de recomendación. En este caso le solicitaremos al agente que nos dé materias recomendadas para Inteligencia Artificial.

1. Para que el agente haga uso de la base de conocimientos, es necesario obtener los planes de estudio de las materias, esto se hace descargando los archivos pdf en la página web de la facultad de Ciencias.
2. Para dar inicio al main y como ejecución única, convertimos los archivos pdf descargados a txt para que Python pueda con mayor facilidad.
3. Definimos las "step-words" de las que hablamos con anterioridad.
4. Se llama al método con los términos de interés:

*recomendaciones(intereses del usuario)*

Este método recibe como parámetro una cadena de texto, la cual contendrá los intereses del usuario separado por espacios, por ejemplo, si me gusta Inteligencia Artificial usare

*get\_recommendations('Inteligencia Artificial')*

Para recibir recomendaciones que se relacionen a ese gusto.

5. El agente comienza a leer los archivos y ejecuta la función `train('Inteligencia Artificial')`, la cual crea el objeto TF-IDF Vectorizer que regresa una matriz de similitudes donde cada entrada es uno de los documentos comparado contra los demás.
6. Finalmente toma la primera entrada de la matriz, que contiene los gustos del usuario, esta entrada contiene las similitudes vs los planes, los ordenamos y selecciona las 10 sugerencias mas próximas y las muestra al usuario.

## 2.8 Representación del estado del entorno

Necesitamos que el entorno, que es la interfaz, se encuentre activo y a la espera de la instrucción del usuario para poder ejecutar el funcionamiento del agente de acuerdo a la solicitud recibida.

## 2.9 Representación del estado interno del agente

- Estudiante: El estado interno del estudiante debe ser que tenga el deseo de conocer las recomendaciones proporcionadas por el sistema, pues así solo falta que ejecute la función para obtener las recomendaciones y no necesita nada más.
- Recomendador: El estado interno del recomendador sería que no hayan materias a recomendar, pues así no necesita saber nada más y no actuaría.

## 2.10 Ventajas del agente

El agente puede ser adaptado para cualquier otra carrera de la Facultad de Ciencias. Sin embargo, también podría utilizarse para otras licenciaturas, sólo habría que cambiar los archivos originales de los temarios de cada materia y las palabras "stop-words". Por otra parte, el tiempo de respuesta por parte del agente es muy eficiente, ya que con más de 100 materias, el usuario obtiene una recomendación al instante.

## 2.11 Desventajas del agente

Creemos que nuestro proyecto tiene algunas desventajas, al basarse en el conocimiento de los planes de estudio únicamente, no podrá dar recomendaciones más allá de los contenidos de los mismos. Un ejemplo de esto es que si al usuario le interesa aprender Python, no podremos sugerirle materias, dado que ningún plan habla sobre este lenguaje de programación. Este no es el caso para Java, dado que aparece en algunos planes, pero ejemplifica una limitante de nuestro proyecto. Por otro lado, tampoco aprende de los gustos del usuario dado que está pensado para consultas únicas. Si queremos ampliar el conocimiento brindado a nuestro agente debemos incluir toda la información en una misma consulta que será ajena a cualquier otra.

## 3 Conclusiones

### 3.1 Conclusiones

Para realizar el sistema de recomendación, primero se tuvo que decidir cuál sería el producto de nuestro sistema de recomendación y una problemática que pudimos identificar fue que muchos estudiantes no saben qué materias elegir que sean acordes a sus intereses, y no se toman el tiempo de revisar cada plan de estudio para descubrirlo, por lo que el realizar un sistema que les pueda recomendar materias, de acuerdo a sus intereses, sin tomar más de unos minutos de su tiempo, nos pareció una buena idea para el proyecto.

Una vez elegido nuestro producto, nos dimos a la tarea de investigar cómo lo implementaríamos (elegir el algoritmo adecuado y aprender a implementarlo).

Después de concluir el proyecto podemos asegurar que hemos aprendido cómo implementar un sistema recomendador básico, el cual puede ser usado como base para un sistema más complejo. Por ejemplo, se le podría agregar una interfaz humano-máquina o incluso conectarlo a un servicio y subirlo a la web para que sea más accesible para el usuario, también podríamos volver más complejo al sistema permitiendo que reciba no solo los intereses sino también las materias ya cursadas para que dé una mejor recomendación. Todas estas mejoras utilizando nuestro código como base.

Una de las cosas más importantes de las que nos dimos cuenta en este proyecto, es que un sistema de recomendaciones no solo se utiliza en el campo del entretenimiento (Netflix, Google, Spotify), sino que también puede ayudar con decisiones del día a día.

### 3.2 Sobre la implementación

Llevar a cabo la implementación del recomendador resultó una tarea sencilla. Una de las partes más complicadas fue encontrar una biblioteca que lograra leer los pdf para poder convertirlos en archivos txt. Esto se realizó para agilizar las consultas una vez se convierten en txt. Fue tardado el juntar todos los pdf, ya que tuvimos que descargarlos de uno en uno, y crear un registro de las materias con su clave.

## 4 Bibliografía y contenidos

### References

- [1] Russell, S, Norving, P. *Artificial Intelligence A Modern Approach*. Prentice-Hall., New Jersey, 1995.
- [2] Prashant Gupta. HACKERNOON. 21 Sep 2017. <https://hackernoon.com/rational-agents-for-artificial-intelligence-caf94af2cec5>
- [3] Marc Erich Latoschik. TECHFAK. <https://www.techfak.uni-bielefeld.de/ags/wbski/lehre/digiSA/WS0506/MDKI/Vorlesung/vl11.knowledgerep.pdf>
- [4] Area de Computación e Inteligencia Artificial. NEBRIJA. [https://www.nebrija.es/~cmalagon/ia/transparencias/busqueda\\_general\\_ia.pdf](https://www.nebrija.es/~cmalagon/ia/transparencias/busqueda_general_ia.pdf)
- [5] Hamdaoui, Y. (2021, 12 diciembre). TF(Term Frequency)-IDF(Inverse Document Frequency) from scratch in python . Medium. <https://towardsdatascience.com/tf-term-frequency-idf-inverse-document-frequency-from-scratch-in-python-6c2b61b78558>

### Contents

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Introducción . . . . .	1
1.2	Descripción del problema . . . . .	1
1.3	Algoritmo TF-IDF . . . . .	2
1.4	Características del usuario . . . . .	2
1.5	Características del producto . . . . .	2
1.6	Enfoque del problema . . . . .	2
<b>2</b>	<b>Agente</b>	<b>2</b>
2.1	Entorno de trabajo . . . . .	2
2.2	Conocimiento . . . . .	3
2.3	Propiedades del entorno . . . . .	4
2.4	Propósito del Agente . . . . .	4
2.5	Función del agente . . . . .	4
2.6	Implementación del algoritmo TF-IDF	4

2.7	Ejemplo de Funcionamiento . . . . .	4
2.8	Representación del estado del entorno	5
2.9	Representación del estado interno del agente . . . . .	5
2.10	Ventajas del agente . . . . .	5
2.11	Desventajas del agente . . . . .	5
<b>3</b>	<b>Conclusiones</b>	<b>6</b>
3.1	Conclusiones . . . . .	6
3.2	Sobre la implementación . . . . .	6
<b>4</b>	<b>Bibliografía y contenidos</b>	<b>6</b>