

ALGORITMIA PARA LA INTELIGENCIA ARTIFICIAL

Ingeniería de software con Inteligencia Artificial

Visión general

Creación de Algoritmos Aplicados a la Inteligencia Artificial

Objetivos

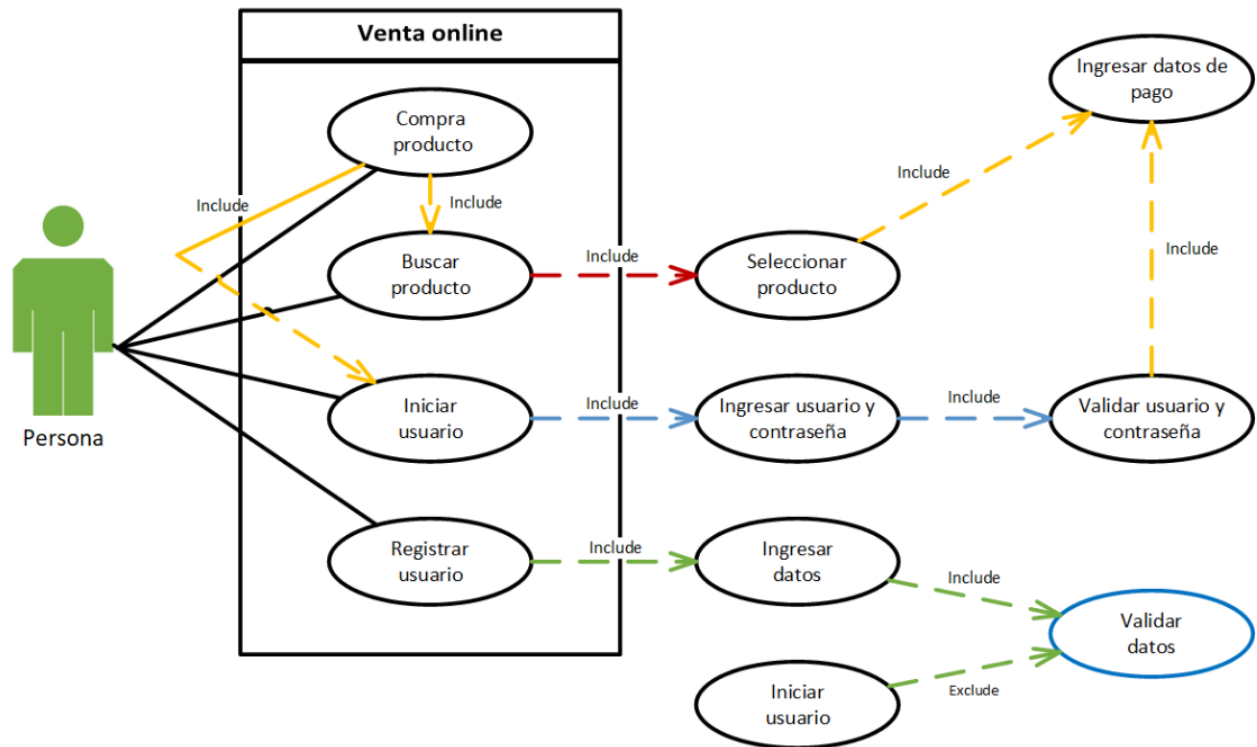
1. Evaluar la capacidad del estudiante para desarrollar algoritmos aplicados a la Inteligencia Artificial.

Especificaciones

Debido a la Pandemia una tienda de libros desea implementar un módulo de registros de pedidos por internet, para lo cual un asistente virtual se hará cargo de registrar los pedidos, recomendar productos y llevar un seguimiento pedidos de los clientes.

Este sistema debe permitir generar reportes de los pedidos en forma automática para que el repartidor los lleve sin problema.

Se adjunta referencia a través del siguiente diagrama de caso de uso



Hitos

1. El algoritmo aplicando Python para cumplir con las exigencias del usuario
2. Un informe del procedimiento desarrollado para resolver el problema planteado.

Planificación del trabajo

N°	ACTIVIDADES/ ENTREGABLES	CRONOGRAMA/ FECHA DE ENTREGA									
1	Maquetado de la aplicación	28	02	22	-	28	02	22			
2	Implementación de la vista	01	03	22	-	01	03	22			
3	Implementación del modelo	01	03	22	-	01	03	22			
4	Implementación del controlador	02	03	22	-	02	03	22			
5	Realizar el <i>testing</i> de la aplicación	03	03	22	-	03	03	22			
6											

Preguntas guía

Durante la investigación de estudio, debes obtener las respuestas a las siguientes interrogantes:

1.	¿Cuáles son los medios de expresión de un algoritmo en inteligencia artificial?
2.	¿Cómo se desarrolla y analiza los algoritmos en IA?
3.	¿Describe en qué consiste la algoritmia aplicada a la IA?.
4.	¿Cuáles son técnicas de Diseño de algoritmos IA?
5.	¿Describe en qué consiste aprendizaje evolutivo?

Hoja de respuestas a las preguntas guía

1.	Los algoritmos pueden ser expresados de muchas maneras, incluyendo al lenguaje natural, pseudocódigo, diagramas de flujo y lenguajes de programación entre otros. Las descripciones en lenguaje natural tienden a ser ambiguas y extensas. El usar
----	--

	pseudocódigo y diagramas de flujo evita muchas ambigüedades del lenguaje natural. Dichas expresiones son formas más estructuradas para representar algoritmos; no obstante, se mantienen independientes de un lenguaje de programación específico.
2.	Para crear Inteligencia Artificial, hay dos materias primas fundamentales: los algoritmos y los datos para entrenarlos. Digamos que el algoritmo proporciona las instrucciones para la máquina y los datos permiten a la máquina aprender a utilizar esas instrucciones y perfeccionar su uso.
3.	La Inteligencia Artificial utiliza algoritmos para crear máquinas que piensan como personas. Por supuesto, es un concepto mucho más complejo, pero, si tuviéramos que hacer un resumen del vínculo entre un algoritmo y la Inteligencia Artificial, sería ese.
4.	Algoritmo divide y vencerás: El método está basado en la resolución recursiva de un problema dividiéndolo en dos o más subproblemas de igual tipo o similar. Algoritmos voraces (greedy): seleccionan los elementos más prometedores del conjunto de candidatos hasta encontrar una solución. En la mayoría de los casos la solución no es óptima. Algoritmos paralelos: permiten la división de un problema en subproblemas de forma que se puedan ejecutar de forma simultánea en varios procesadores. Algoritmos probabilísticos: algunos de los pasos de este tipo de algoritmos están en función de valores pseudoaleatorios
5.	Los algoritmos evolutivos son métodos de optimización y búsqueda de soluciones basados en los postulados de la evolución biológica. En ellos se mantiene un conjunto de entidades que representan posibles soluciones, las cuales se mezclan, y compiten entre sí, de tal manera que las más aptas son capaces de prevalecer a lo largo del tiempo, evolucionando hacia mejores soluciones cada vez.

Proceso de ejecución

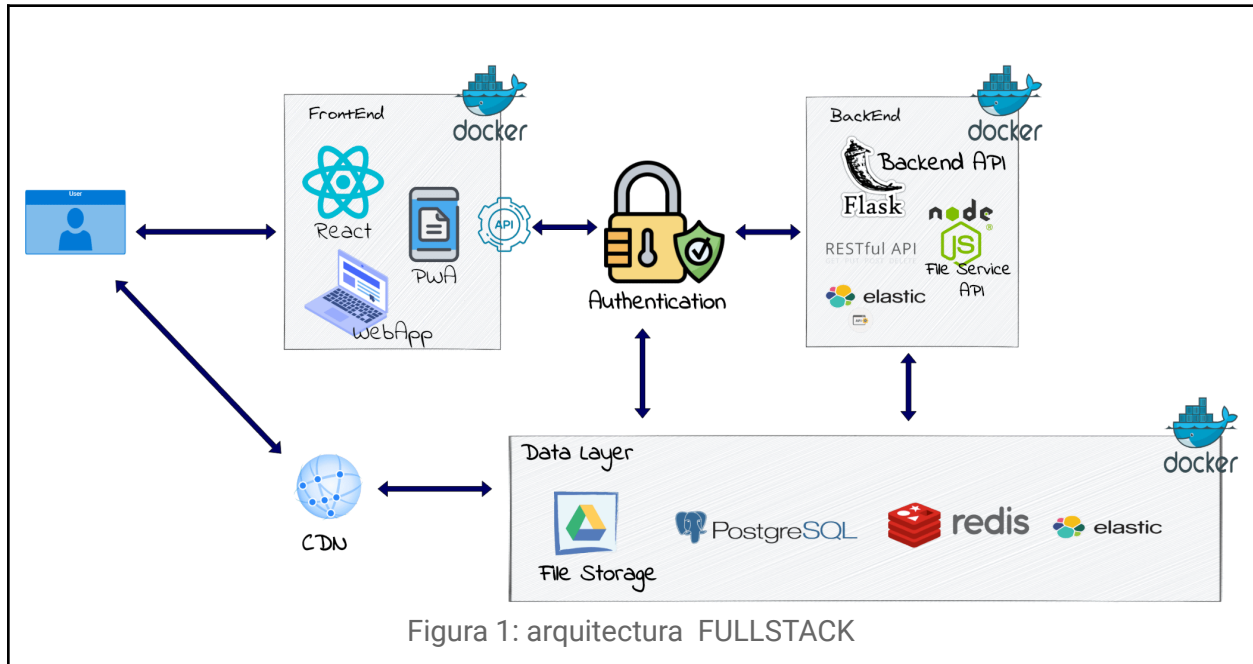
OPERACIONES / PASOS /SUBPASOS	NORMAS ESTÁNDARES
Parte 1: back-end (Flask) <ul style="list-style-type: none"> - Según tus propios requisitos, puede que tengas que instalar los paquetes de python necesarios de antemano. Yo utilicé pip para instalar todos los paquetes que necesitaba y no utilicé un entorno virtual (aunque es muy recomendable). Para crear un 	Adecuar el brillo del equipo de acuerdo con la iluminación.

<p>nuevo entorno virtual de python llamado venv (puedes llamarlo de otra manera - sólo reemplaza el último venv en el comando de abajo con tu propio nombre de venv), ejecuta</p> <ul style="list-style-type: none"> - \$ python -m venv venv 	
<ul style="list-style-type: none"> - Esto es lo que se ejecuta cuando ejecutamos flask run. Yo llamé al mío app.py, pero puedes elegir cualquier otro nombre que no sea flask.py porque esto entraría en conflicto con el propio Flask. 	<p>Realizar descansos cada intervalo de 25 minutos.</p>
<ul style="list-style-type: none"> - El error CORS que siempre obtenemos cuando hacemos una solicitud de API a un dominio diferente. En esta etapa actual (antes del despliegue), React se ejecuta en el puerto 3000 y Flask en el puerto 5000. - En mi ApiHandler.py sólo hay dos funciones simples GET y POST para manejar las solicitudes GET y POST a este punto final de la API: - \$ flask run 	<p>Mantener tu espacio de trabajo limpio, ventilado y con iluminación adecuada.</p> <p>Revisar que la conexión de los equipos electrónicos sea correcta y no estén dañados</p>
<p>Parte 2 – Front-end (React)</p> <ul style="list-style-type: none"> - A continuación, crearemos una carpeta en el directorio del proyecto para el front-end (React). Yo he llamado a la mía "frontend". <p>\$ npx create-react-app .</p> <ul style="list-style-type: none"> - Esto creará un nuevo proyecto React. - En tu archivo app.js, haz los siguientes cambios para probar nuestra petición GET a nuestro back-end de flask. 	<p>Tener una postura adecuada para realizar el trabajo:</p> <p>Columna recta.</p>

- A continuación, para generar el requirements.txt, cd en su directorio raíz de este proyecto y ejecutar:
\$ pip freeze >> requirements.txt

Distancia de un brazo
con respecto a la
computadora.

Dibujo / Esquema / Diagrama



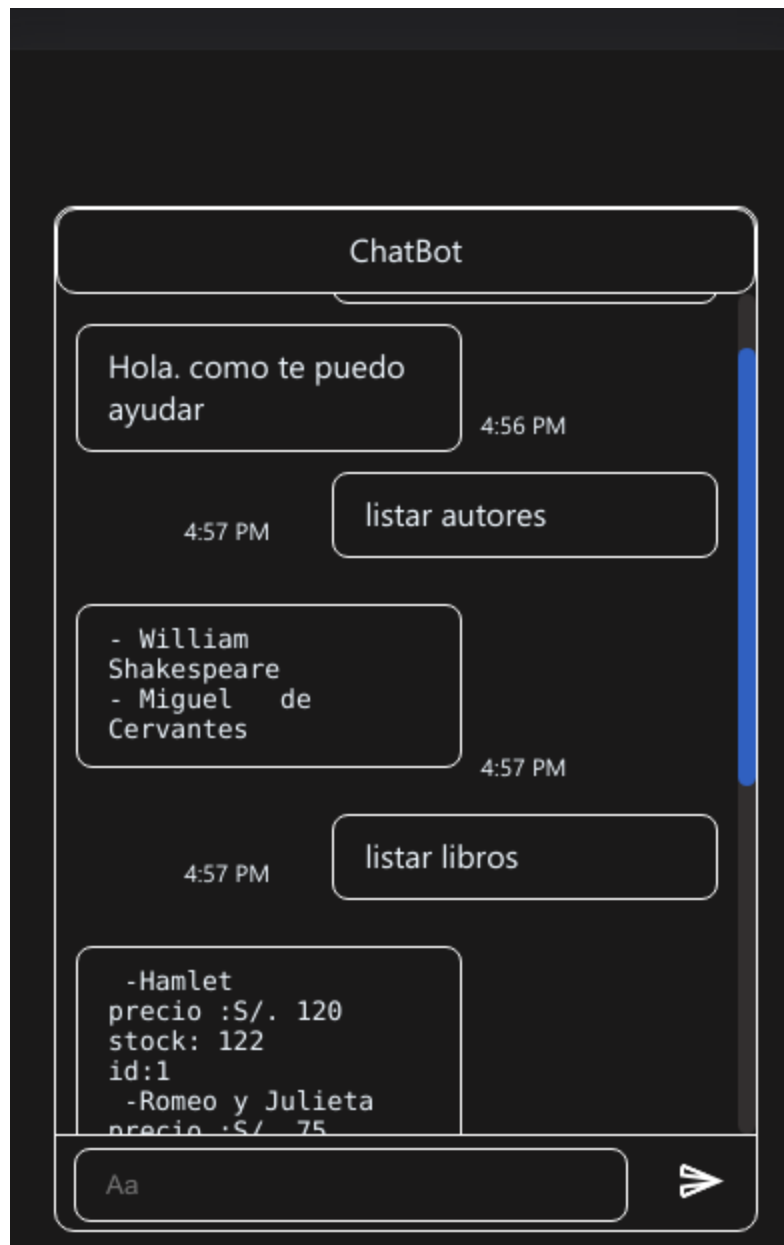


Figura 1: Interfaz gráfica

Listas de recursos

INSTRUCCIONES: completa la lista de recursos necesarios para la ejecución del trabajo.

MÁQUINAS Y EQUIPOS

Computadora: El recurso principal para ejecutar el trabajo, debido a que en este
Ejecutaremos los programas como el editor de código, el compilador, etc.

Modem: Aquel punto de conexión con la red que nos permitirá descargar los
Programas necesarios, y realizar las búsquedas para resolver dudas e inconvenientes.

HERRAMIENTAS E INSTRUMENTOS

Navegador: Chrome tiene un buen conjunto de herramientas para
desarrolladores gracias Chrome DevTools , que facilitan bastante la creación
y la depuración de proyectos y aplicaciones.

Visual Studio Code: Esta herramienta multi-código gratuita de edición python
viene lista para usarse con una gran variedad de funciones personalizables.

Sqlite3: Gestor de base de datos.
Heroku Git
