RestFinder

Aplicación web de restaurante realizada en los lenguajes de programación Python, Prolog y HTML

Tarea Programada II

Realizada por:

Jimmy Granados

Fabían Fernández

Esteban Leandro

Table of Contents

[Propósito de la App (descripción y requerimientos) 2](#_Toc386496118)

[Lenguajes de programación usados. 2](#_Toc386496119)

[Librerías usadas. 2](#_Toc386496120)

[Decisiones de diseño. 3](#_Toc386496121)

[Algoritmos Implementados. 3](#_Toc386496122)

[Análisis de resultados. 3](#_Toc386496123)

[Manual de usuario. 3](#_Toc386496124)

[Conclusión. 4](#_Toc386496125)

Documentación RestFinder

# Propósito de la App (descripción y requerimientos)

Se nos pidió crear una aplicación web que permitiera agregar restaurantes junto con detalles de los mismos como ubicación, tipo de comida, horario y teléfono, de igual forma debería agregar platillos e igualmente sus detalles como los ingredientes, sabor, país de origen. Todo lo anterior se debía almacenar en una base de conocimientos e internamente debía de consultarse desde el lenguaje de programación Python y presentar el resultado por medio de un framework a elegir.

Era necesario que la aplicación web fuera de diseño responsive por lo tanto se debía adaptar al tamaño de la pantalla del dispositivo donde se verá.

# Descripcion de diseño de alto nivel

# Lenguajes de programación usados.

* **Python**

Python es un lenguaje de programación interpretado y orientado a objetos.

* **Prolog**

Es un lenguaje para programar que utiliza el paradigma lógico con técnicas de producción final interpretada.

# Librerías usadas.

Para la realización de nuestro programa, utilizamos las siguientes librerías:

* **Pyswip**

Librería que permite la lectura y comprensión de archivos escritos en el lenguaje de programación Prolog.

* **Jinja2**

Motor de plantillas basado en texto y puede ser utilizado para generar cualquier tipo de código, así como el código fuente.

* **CherryPy**

Es un framework de aplicaciones web orientado a objetos.

# Decisiones de diseño.

Para la realización de la tarea programada, tomamos distintas decisiones de diseño las cuales afectaron las formas en las que se comporta el programa, entre ellas están:

* Escoger Cherrypy sobre Django ya que Django trabaja con una base de datos y sobre Flask ya que es mucho más efectivo.
* Separar el programa en distintos archivos, como restaurante e input separados, para así aumentar la portabilidad del código escrito

# Algoritmos Implementados.

Los algoritmos creados se detallan a continuación:

* Cuando se hacen registros de restaurantes y platillos, se tiene una función la cual recibe todos los detalles de estos y son guardados directamente a los archivos .pl correspondientes.
* Cada ventana de la aplicación web tiene su función dentro de una clase, en la cual la función recibe como parámetro los datos ingresados en la aplicación web y dentro de estas se llaman a las funciones en el archivo restaurante.py.

# Manual de usuario.

1. Necesitan clonar el repositorio de github, para esto utilizan el comando git clone dirección/de/repositorio.
2. Una vez que el repositorio se clono y los archivos se descargaron, se dirigen a la carpeta por medio del comando cd dirección/de/la/carpeta
3. Ingresamos en la consola *hostname –I* y el resultado será nuestra IP, la ingresamos en el archivo Input.py en la línea donde dice cherrypycherrypy**.**server**.**socket\_host = ‘aquí\_resultado’
4. Ahora ingresamos *python Input.py* en la consola, lo que hará que se ejecute la aplicación web.
5. Ahora, entramos al navegador y escribimos nuestra IP junto con “:8080”, y al cargar tendremos nuestra página lista para hacer consultas, agregar restaurantes y platillos.

# Conclusión.

Durante la realización de la tarea programada, llegamos a darnos cuenta que el lenguaje de programación Prolog es un lenguaje muy poderoso, que permite la realización de diversos tipos de programas, y que está muy bien documentado en internet y libros debido a los más de 35 años de existencia del mismo.

También concluimos que el paradigma lógico, funciona de una manera muy eficiente y que realizar tareas simples, a veces se puede tornarse complicado si no se tiene conocimiento previo del lenguaje de programación.

Además, durante la realización de la tarea programada, aprendimos conceptos bastante importantes como lo son:

* Manejo de frameworks.
* Conexión con otros lenguajes de programación.
* Front-End y Back-End.
* Uso de sistemas operativos alternativos a MS Windows
* Diversidad de frameworks

Por lo que consideramos que la tarea programada nos dejó mucho conocimiento técnico que podemos utilizar más adelante en nuestras carreras y en el ámbito profesional.