TAREA PROGRAMADA II
TI-3404

Integrantes:

- Edgar Solórzano Pacheco
- Daniel Mora Chacón
- Gerardo Calderón Badilla

Profesor:

Andrei Fuentes Leiva

Documentación Externa

Contenido

Resumen ejecutivo

El desarrollo de aplicaciones móviles para smartphones está en crecimiento, es una nueva área de interés que nace debido a la necesidad de las personas de mantenerse activas e informadas. Este proyecto engloba el desarrollo de una app descrita por el profesor, se basa en la idea de la localización de ciertos elementos. El sistema a desarrollar tendrá dos modalidades principales, agregar datos y consultarlos. Se deberá desarrollar un programa que permita y sea flexible a la hora de ingresar los datos de un restaurante, este contará con ciertas características especificadas por el profesor. A su vez el programa deberá de ser capaz de proporcionar datos solicitados por el usuario. El proyecto constará de diferentes capas o niveles, primero el Front-end que será el encargado de manejar toda la interacción con el usuario, incluyendo entrada y salida de datos. También se deberá comunicará con el Back-end para las tareas relacionadas con mantenimiento de datos y consulta. El front-end será una aplicación web escrita en el lenguaje de programación Python, utilizando una seleccionada framework para su uso. El Back-end a su vez se encargará de manejar la base de conocimientos de restaurantes, y de responder las consultas hechas por el usuario, esto será de manera indirecta, ya que el Front-end es el que maneja la interacción con el usuario, y hace a su vez las consultas al Back-end, el cual contará con dos funcionalidades principales, manejar la base de conocimientos y responder las consultas. La base de conocimientos será una serie de declaraciones en el lenguaje de programación Prolog que definirá todos los atributos de los restaurantes y los platillos, el Back-end deberá actualizar esta base de conocimientos usando los datos que el usuario ingrese al sistema por medio del Front-end.

Propósito: descripción del app, requerimientos

Este proyecto de programación nos ayudará a conocer cómo funciona el desarrollo de aplicaciones para smartphones, asimismo la investigación ayudará a entender cómo se pueden utilizar diferentes herramientas de programación para obtener un resultado final y consistente. La aplicación deberá estar formada por un front-end el cual estará encargado de interactuar con el usuario por medio de una aplicación web, solicitará datos y mostrará información en pantalla. A su vez contará con un back-end el cual se encargará de modificar o agregar datos de manera interna a los archivos escritos en Prolog, los cuales contienen la información que el usuario ingresará y almacenará. En otras palabras, el front-end pasa los datos al back-end, el cual guarda los datos en la base de conocimientos, y le devuelve el resultado de la operación al front-end. El front-end muestra al usuario un mensaje que le indique que ya se terminó de realizar el almacenamiento de datos en la base de conocimientos. El Back-end deberá estar escrito en el lenguaje de programación Prolog como se mencionó anteriormente. Con respecto al Front-end, deberá ser una aplicación web controlada utilizando Python, que además deberá de contar con un diseño web adaptativo (responsive design). Esto le permitirá tener un layout adaptado para smartphones, esto quiere decir que su interfaz deberá de ser flexible dependiendo del tamaño del dispositivo.

Funcionalidades

El programa tiene ciertas funcionalidades especificadas por el profesor, algunas de las consultas con las que cuenta la aplicación son la lista de restaurantes, restaurantes filtrados por tipo de comida, búsqueda de restaurantes por nombre, restaurantes que tienen platillos de algún país específico, platillos de un restaurante específico, platillos específicos que tengan un ingrediente en particular. A su vez el programa puede almacenar cierta información solicitada al usuario, por ejemplo los datos de un restaurante como el nombre, tipo de comida, ubicación, teléfono y horario. Adicionalmente, para cada restaurante se pueden ingresar los platillos favoritos. La información que se podrá ingresar de cada platillo es el nombre, el sabor (si es picante, salado, dulce, agridulce o amargo), país de origen del platillo, y la lista de ingredientes que este posee.

Descripción de diseño de alto nivel

Aplicación web escrita en HTML, con hojas de estilo y archivos javascript que permiten el diseño web adaptativo (responsive design). Controlada mediante el uso de un archivo escrito en Python en el cual se hace uso de la librería Flask y sus funciones de servidor web y demás. Cada formulario dentro de las páginas web están ligados a funciones escritas en dicho archivo, las cuales le pasan los datos a las funciones encargadas del manejo de Prolog ubicadas en otro archivo también escrito en Python, el cual utilizando la librería pyswip permite el manejo de la base de conocimientos. El resultado de las consultas o procesos de mantenimiento se muestra en un archivo de plantilla con extensión .htm

Descripción detallada

Lenguajes de programación usados:

HTML = diseño de las páginas web

CSS3 = estilos para las páginas web y control del responsive design de la interfaz

Javascript = funcionalidades animadas y control del responsive design de la interfaz

Python 2.7 = Manejo de la aplicación web y base de conocimientos.

Prolog = manejo de la base de conocimientos.

Frameworks utilizados:

Flask 0.10.1 = servidor web y manejo de la interfaz desde Python.

Pyswip 0.2.3 = conexión entre Python y Prolog.

Bootstrap 3.1.1 = permite la característica de responsive design de la aplicación web.

SWI-Prolog 6.6.4 = versión de Prolog utilizada. Manejo de funciones.

Entornos de programación utilizados:

Adobe Dreamweaver = diseño de la aplicación web y manejo de hojas de estilo

Python IDLE = prueba de funciones en Python.

Geany = escritura y depuración de código.

Decisiones de diseño:

Aplicación web:

Interfaz: 8 páginas con código HTML y CSS.

'rest.manager.htm' : utilizada para el inicio de la aplicación/menú principal

'rest.manager.consulta.htm', **'rest.manager.consulta.platillos.htm'** y **'rest.manager.consulta.restaurantes.htm'** : utilizadas para mostrar las consultas disponibles

'rest.manager.mantenimiento.htm', **'rest.manager.mantenimiento.restaurantes.htm'** y **'rest.manager.mantenimiento.platillos.htm'** : utilizadas para mostrar las funcionalidades de mantenimiento

‘rest.manager.resultado.htm’ : utilizada como plantilla para mostrar el resultado de las consultas y los procesos de mantenimiento

‘rest.manager.informacion’ : utilizada para mostrar información extra del proyecto (guía de navegación en el sitio y descarga de documentación externa).

Se decidió que cada archivo tuviese rest.manager al inicio para evitar confusiones con otros archivos. Como es un proyecto de desarrollo en grupo, pensamos que era una buena opción para mantener el orden.
La navegación por la aplicación web se realiza por medio de una barra de Menú ubicada en la parte superior, desde esta se puede acceder a las distintas funcionalidades. Se utilizaron Dropdowns para aprovechar el espacio en la barra de menú, uno para las Consultas y otro para el Mantenimiento. Estos permiten a su vez, una fácil visualización en dispositivos móviles.

Control de página web desde Python:

Como se mencionó anteriormente, se utilizó la librería Flask para todo lo que es el manejo de una aplicación web desde Python. El archivo ‘rest.manager.py’ contiene todo lo necesario para levantar el servidor web, comunicar la aplicación con Python y enviarle estos datos a las funciones encargadas del manejo de la Base de Conocimiento. A continuación se explican las funciones encontradas en ‘rest.manager.py’:

** Nota: Todas las funciones envían su resultado a la plantilla resultado.htm para desplegarlo gráficamente.

consultar_restaurantes_todos(): No obtiene nada del usuario, solo se encarga de obtener una lista con todos los restaurantes.

consultar_restaurantes_tipo(): Obtiene el tipo de comida por parte del usuario, se encarga de obtener una lista con todos los restaurantes cuyo tipo de comida sea el mismo que ingresó el usuario.

consultar_restaurantes_nombre(): Recibe un nombre por parte del usuario, en encarga de buscar los restaurantes que satisfacen el nombre ingresado.

consultar_restaurantes_pais(): Obtiene el país deseado, se encarga de buscar los restaurantes que tengan platillos de un determinado país.

consultar_platillos_restaurante(): Obtiene el nombre de un restaurante, se encarga de obtener la lista de platillos de dicho restaurante.

agregar_restaurante(): Obtiene los valores nombre, tipo de comida, ubicacion, telefono y horario del usuario; añade entonces un restaurante a la base de conocimientos. Devuelve el mensaje de si el proceso fue exitoso o no.

asignar_platillo(): Obtiene el nombre del restaurante y el nombre del platillo. Se encarga de asignarle el platillo ingresado al restaurante deseado.

agregar_platillo(): Obtiene los valores nombre, sabor, pais e ingredientes del usuario. Se encarga de agregar un nuevo platillo a la Base de conocimientos.

agregar_ingrediente(): Obtiene el nombre del platillo y un ingrediente. Se encarga de agregarle el ingrediente ingresado al platillo deseado.

```
if __name__ == '__main__':
        app.run(debug=True)
```

Con esta parte del código es que se corre la aplicación en el servidor local con el debugger activado. Cambiar a debug=False si no se desea utilizar dicha funcionalidad.

Problemas encontrados

Los principales problemas encontrados fueron a la hora de instalar la librería de pyswip, esto es necesario para poder entrelazar los lenguajes de programación de Python y Prolog, utilizando así comandos para leer o escribir especificaciones en el back-end. La comunicación entre Python y Prolog presentó problemas ya que el archivo de la base de conocimientos no se leía según los parámetros establecidos, generando así errores o respuestas incorrectas a las consultas hechas por el usuario. También al inicio del desarrollo de la aplicación se decidió usar el framework Django, sin embargo surgieron problemas de implementación en el servidor, el cual no funcionaba según lo esperado, por lo tanto se decidió cambiar el framework utilizado por Flask, el cual facilita su uso a la hora de enlazarlo con la interfaz de la aplicación web creada. El uso de pip y virtualenv necesitó un poco de cuidado para crear el entorno virtual, sin embargo flask posee dependencias mínimas lo cual permitió su implementación de manera correcta. Una vez que la aplicación estaba preparada se creó la plantilla para que se pudiera desplegar el resultado de las consultas, para ello se creó un archivo html en el que se dió referencia a los formularios en las diferentes páginas.

Manual de usuario