



APP WEB DE RESTAURANTES



Elaborado por:

Carlos Badilla.

Acsa Herrera.

Mónica Solano.

06 DE MAYO DE 2014

LENGUAJES DE PROGRAMACIÓN

TI 3404



TEC

Instituto Tecnológico de Costa Rica

Contenido

Resumen Ejecutivo	2
Propósito	2
Descripción del app:	2
Requerimientos:	3
Funcionalidades.....	3
Descripción de diseño de alto nivel	4
Descripción detallada del programa	6
Decisiones de diseño:	6
Lenguajes de Programación usados:	8
Tecnologías usadas:	8
Librerías usadas:	8
Problemas encontrados	9
Manual de Usuario	9
Conclusión personal	10

Resumen Ejecutivo

El proyecto es un sistema que permite obtener información de restaurantes con el teléfono móvil.

La idea es usar el móvil como si fuera una guía para conocer los datos de los restaurantes.

El sistema responde a la pregunta ¿Quiere ir a un restaurante? Enciende su teléfono móvil, abre la aplicación, selecciona la opción **lista de restaurantes** y podrá ver los nombres de los restaurantes que se encuentran registrados en la aplicación, si ingresa el nombre del restaurante y se mostrará en pantalla los datos pertinentes a ese restaurante. Puede realizar diversas consultas como lo es buscar restaurantes por el tipo de comida, platillos de un restaurante o de algún país específico, y platillos con un ingrediente en particular.

¿Qué es lo que vamos hacer? Desarrollar un servicio para obtener diferente información acerca de restaurantes en el teléfono móvil *Smartphone*.

¿Quiénes lo vamos hacer? Carlos Badilla Ureña, Acsa Herrera Madrigal, Mónica Solano Morales. Estudiantes de Administración de Tecnologías de Información del Instituto Tecnológico de Costa Rica.

¿Cómo lo vamos hacer? Creando una aplicación web a la cual se puede acceder desde cualquier *Smartphone*.

¿Por qué lo vamos hacer? Para facilitar la búsqueda de restaurantes, y así tener una forma más sencilla de realizar consultas sin tener que trasladarse largas distancias, simplemente obtener la información que se necesita desde un *Smartphone*.

Propósito

Descripción del app:

Es una aplicación web para restaurantes en la cual se pueden agregar restaurantes y platillos, además de realizar diferentes consultas sobre los mismos.

Agregar restaurantes y platillos es una modalidad en la que al usuario se le solicitan diferentes datos sobre el restaurante o el platillo, dependiendo de la opción que se haya seleccionado, y al finalizar el proceso los datos se almacenan y el usuario revisará un mensaje informado el éxito de la operación.

Las consultas se basan en la información ingresada por el usuario sobre los restaurantes y platillos favoritos, los cuales son almacenados en la base de conocimiento con el propósito de realizar futuras consultas y de esa manera darle mantenimiento a los datos. Por cada consulta realizada el usuario podrá ver en la pantalla los resultados obtenidos acerca de dicha consulta.

Requerimientos:

Para poder ejecutar la aplicación correctamente se deben de tener en cuenta los siguientes requerimientos:

- Python 2.7 o posterior.
- ctypes 1.0 o posterior.
- SWI-Prolog 6.0.2 o posterior.
- Libpl como librería compartida (shared library)
- Flask 0.10.1 última versión.
- Trabajar en Linux de 32bits.
- Si es una *Virtual Machine* trabajar desde un emulador de *Smartphones*.

Funcionalidades

La aplicación cuenta con diferentes funcionalidades, las modalidades principales son mantenimiento de datos y consulta, de estas se despliegan otras funciones las cuales se explicarán a continuación:

Mantenimiento de Datos:

- Agregar restaurantes: esta funcionalidad le permite al usuario ingresar un nuevo restaurante de su preferencia, para ello se le solicita que ingrese diversos datos sobre el restaurante (nombre, tipo de comida, ubicación, teléfono, horario de atención) y luego este será almacenado en la base de conocimiento.
- Agregar platillos: esta funcionalidad le permite al usuario ingresar un nuevo platillo de un restaurante de su preferencia, para ello se le solicita que ingrese diversos datos sobre el platillo (nombre del restaurante, nombre del platillo, el sabor, país de origen y lista de ingredientes) y luego este será almacenado en la base de conocimiento.

Consultas:

- Restaurantes: en esta consulta el usuario podrá ver en pantalla una lista con los nombres de todos los restaurantes que se encuentran almacenados en la base de conocimiento.
- Restaurantes por tipo de comida: para realizar esta consulta el usuario debe ingresar un tipo de comida y con este dato el programa procederá a realizar la búsqueda de los restaurantes que poseen el tipo de comida indicado, luego le mostrará al usuario una lista con los nombres de los restaurantes encontrados que cumplen con el parámetro de consulta.

- Restaurantes por nombre: esta consulta le permite al usuario ingresar el nombre de un restaurante, de acuerdo a ese nombre se realiza una búsqueda de todos los restaurantes que tienen ese parámetro, con los datos encontrados el usuario podrá ver en la aplicación la ubicación de los diferentes restaurantes, el número de teléfono, el tipo de comida y el horario de atención al cliente.
- Restaurantes que tienen platillos de algún país específico: cuando se realiza esta consulta se le solicita al usuario que ingrese el nombre del país de origen de los platillos, con base en la información dada el programa ejecuta esta función y procede a mostrar en pantalla una lista con los restaurantes obtenidos en la búsqueda.
- Platillos de un restaurante específico: para realizar esta consulta el usuario debe ingresar el nombre de un restaurante específico y con este dato el programa procederá a realizar la búsqueda de los platillos que pertenecen al restaurante indicado, luego le mostrará al usuario una lista con los nombres de los platillos encontrados que cumplen con el parámetro de consulta.
- Platillos de un restaurante específico que tengan un ingrediente en particular: en esta consulta el usuario podrá ver en pantalla una lista con los nombres de todos los platillos de un restaurante que se encuentran almacenados en la base de conocimiento de acuerdo a un ingrediente en particular. Para esto se deben ingresar el nombre del restaurante y el nombre del ingrediente, con estos datos se realiza primeramente una búsqueda del restaurante, luego efectúa otra búsqueda en los ingrediente de los platillos, si uno de los ingredientes coincide con el ingresado por el usuario entonces se verán en pantalla los nombres de los platillos que tienen ese ingrediente en particular.

Descripción de diseño de alto nivel

La aplicación está estructurada en dos componentes Front-end y Back-end.

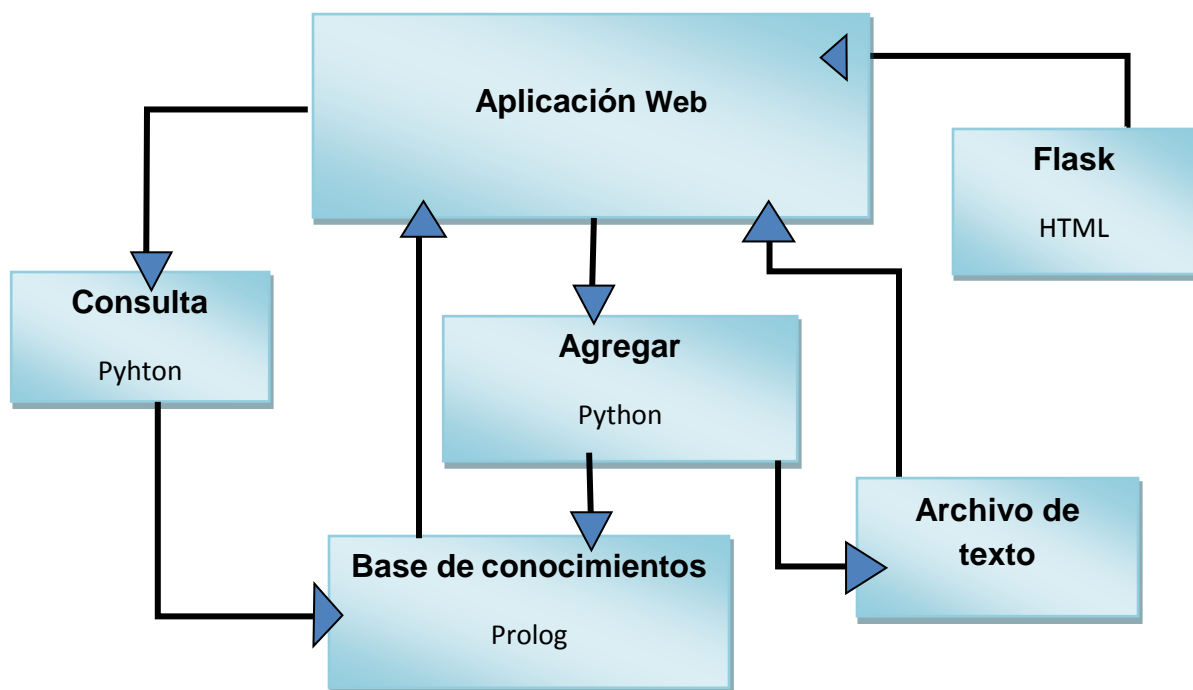
Front-end: es la estructura que se encarga de manejar la interacción con el usuario, esta será por medio de la interfaz que se ha creado en HTML que forma parte de la aplicación que ha sido creada en Python. Al usuario le permite el ingresar datos y la opción de observar los datos por medio de salidas.

A través del front-end de la aplicación el usuario conocerá la información que ha sido ingresada al haber utilizado la aplicación, al mismo tiempo que podrá añadir nuevos restaurantes y platillos.

Back-end: se encarga del manejo de la base de conocimientos y dar respuesta a las consultas.

En la base de conocimientos se encuentran todos los predicados que han sido agregados por medio del Front-end. Como respaldo, el predicado es agregado a un archivo de texto según el formato de Prolog. A través de las funciones que se han declarado en Python y forman parte de la aplicación se pueden realizar las consultas.

A continuación se presenta un diagrama del diseño del programa de acuerdo a su funcionalidad.



Descripción detallada del programa

Decisiones de diseño:

Para el manejo de los datos se crearon las siguientes funciones:

AgregarRestaurante ()

Entrada: recibe la información del restaurante detallada por nombre, tipo de comida, ubicación, teléfono y horario.

Funcionalidad: la información que ha sido ingresada será convertida en un predicado, el cual será agregado en la base de conocimientos, el funtor será "restaurante()". Esta función llama a Escribir_Archivo ().

Salida: agregará el predicado a la base de conocimientos

Agregar_Platillo ()

Entrada: recibe la información que se debe tener para registrar un platillo

Funcionalidad: la información que ha sido ingresada será convertida en un predicado, el cual será agregado en la base de conocimientos, el funtor será "platillo()". Esta función llama a Escribir_Archivo().

Salida: agregará el predicado a la base de conocimientos

Además de tener una base de conocimientos se usa como respaldo un archivo de texto, ya que, cuando se cierra la aplicación se eliminan por defecto todos los predicados, cuando algún platillo o restaurante es ingresado se almacenará en el archivo, el cual se maneja con dos funciones:

Escribir_Archivo ():

Entrada: recibe un string, nombrado predicado.

Funcionalidad: se encarga de almacenar el predicado en el archivo de texto, se utiliza para respaldo de la información.

Salidas: escribe en el archivo de texto.

Leer_Archivo (): esta función es utilizada cuando se abra la aplicación, así tendrá todos los datos que han sido ingresados.

Entrada: vacío.

Funcionalidad: se encarga de cargar los predicados que han sido almacenados en el archivo de texto a la base de conocimientos.

Salida: cargar predicados.

Con respecto a la parte de consultas se implementaron distintas funciones las cuales permiten dar la información que ha sido solicitada.

NoEsta_enLista (): es usada para validación.

Entrada: recibe una lista y un string.

Funcionalidad: se encarga de validar si el elemento que ha sido ingresado forma parte de la lista.

Salida: true o false dependiendo si el elemento forma parte de la lista

Imprimir_Lista_Rest ()

Entrada: vacío.

Funcionalidad: crea la lista de restaurantes que se encuentran en la base de conocimientos.

Salida: retorna una lista con los nombres de los restaurantes.

Rest_por_Tipo ()

Entrada: recibe un string definido por tipo de comida.

Funcionalidad: busca los restaurantes en la base de conocimientos según el tipo de comidas que ha sido ingresado y los agrega a una lista.

Salida: retorna la lista de restaurantes según el tipo de comida ingresado

Rest_por_Nombre ()

Entrada: recibe un string definido por el nombre del restaurante.

Funcionalidad: busca en la base de conocimientos los restaurantes según el nombre que fue ingresado y crea una lista con la información de los restaurantes.

Salida: lista con la información de los restaurantes que tienen el mismo nombre

Rest_Platillo_Pais ()

Entrada: recibe un string definido por nombre de país.

Funcionalidad: busca en la base de conocimientos los platillos por el país ingresado y los agrega a una lista.

Salida: retorna una lista de platillos según el país.

Platillos_Rest ()

Entrada: recibe un string definido por nombre de restaurante.

Funcionalidad: busca en la base de conocimientos los platillos de un restaurante en específico y los agrega a una lista.

Salida: lista de los platillos de un restaurante.

Platillos_porIngrediente ()

Entrada: recibe dos string uno definido por el nombre del restaurante y el otro por el ingrediente que desea buscar.

Funcionalidad: busca en la base de conocimientos los platillos del restaurante que tiene el ingrediente ingresado.

Salida: lista de platillos del restaurante que tienen el ingrediente.

Lenguajes de Programación usados:

Los lenguajes utilizados son:

Prolog: presenta un tipo de paradigma lógico el cual trabaja con predicados, hechos y reglas, esto permite un proceso de inferencia. Es utilizado para la creación de la base de conocimientos donde se encontrarán una serie de declaraciones, entre ellas los predicados que se ingresan durante la ejecución de la aplicación.

Python: se utiliza para la definición de las funciones de agregar y las funciones para las consultas, se utilizan sentencias propias de este lenguaje para el manejo lógico de los datos, lo indiferente a estas funciones es la sintaxis con la que se construyen los predicados y las consultas dentro de estas funciones

HTML: se utiliza para la elaboración de la página web, en esta se utiliza las funciones que fueron declaradas en Python, se utiliza para la construcción de la interfaz y para el manejo de los datos cuando se agregan y cuando se debe consultar

Tecnologías usadas:

Para el desarrollo del framework se utilizó Flask.

Flask: me permite crear la conexión entre Python y HTML, o sea, la utilidad que se le da a esta es para la extracción de los datos de Python a HTML, entre los framework del mercado Flask es el medio con el que puedo crear la aplicación web

Librerías usadas:

Las librerías que se utilizaron fueron:

PySwip: es un puente entre Python y SWI-Prolog el cual permite almacenar datos en una base de conocimiento y realizar consultas en prolog usando SWI-Prolog en programas del Python.

```
from pyswip import Prolog
```

Flask: es una pequeña librería web, que se encargará de generar contenido dinámico a partir de plantillas. Este es un potente microframework web para Python, fácil de usar y con bastantes módulos para implementar numerosas funcionalidades webs.

```
from flask import Flask, render_template, request, redirect, url_for, abort, session
```

Problemas encontrados

1. Retraso en la instalación de PySwip: al principio desconocíamos los requerimientos para el funcionamiento de este por lo que nos tomó varios tiempo poder instalarlo exitosamente
2. Framework seleccionado: nos tomó muchas pruebas poder escoger cual era la mejor opción a utilizar para el desarrollo de la conexión.
3. Respaldo de la información: desconocíamos que la base de conocimientos se almacenaba temporalmente lo cual nos llevó a crear una solución para el almacenamiento estático de los predicados.
4. Mayúsculas: el uso de variado de mayúsculas y minúsculas genera desacierto en el momento de realizar la búsqueda para las consultas, ya que ciertas palabras pueden ser ingresadas en minúsculas y otras en mayúsculas. Como solución, internamente se convierten los datos que son ingresados en minúsculas para la facilidad de manipulación.
5. Implementación css:

Manual de Usuario

1. Instale el programa Python 2.7 en su computadora. (En caso de tenerlo omita esta instrucción).
2. Instale SWI-Prolog, seguidamente debe instalar la librería de PySwip. Para esto puede seguir las siguientes líneas de comandos: (En caso de tenerlo omita esta instrucción).

```
wget http://www.swi-prolog.org/download/stable/src/pl-6.0.2.tar.gz
tar xzvf pl-6.0.2.tar.gz
cd pl-6.0.2/
./configure --prefix=/usr --enable-shared
make && sudo make install
cd packages/clpqr/
./configure --prefix=/usr
make && sudo make install
sudo ln -s /usr/lib/swipl-6.0.2/lib/i686-linux/libswipl.so
/usr/lib/libpl.so
sudo ln -s /usr/lib/swipl-6.0.2/lib/i686-linux/libswipl.so.6.0.2
/usr/lib/.
wget http://pyswip.googlecode.com/files/pyswip-0.2.2.zip
unzip pyswip-0.2.2.zip
cd pyswip-0.2.2/
sudo python setup.py install
```

3. Si desea puede instalar el IDLE de Python2.7 para generar el código desde allí, esto lo podrá encontrar en el centro de software de Ubuntu, ingrese y siga los pasos de instalación.



IDLE (using Python-2.7)

Integrated Development Environment for Python (using Python-2.7)

4. Instale el framework Flask, para realizar la instalación debe de ingresar a la terminal e ingresar las siguiente líneas de comandos:

```
$ sudo apt-get install python-virtualenv
$ mkdir myproject
$ cd myproject
$ virtualenv venv
New python executable in venv/bin/python
Installing distribute.....done.
```

5. Ingrese a la terminal y luego ingrese a la ubicación donde se encuentran los archivos del programa. Ejecute el archivo con el nombre progra.py.

```
acsa@ubuntu:~/Desktop/CherryPy-3.2.4$ python progra.py
```

6. Posicione el mouse sobre la dirección http, luego presione clic derecho y open link.

```
* Running on http://127.0.0.1:5000/
```

7. Se inicia la aplicación, en pantalla se mostrará el siguiente menú:



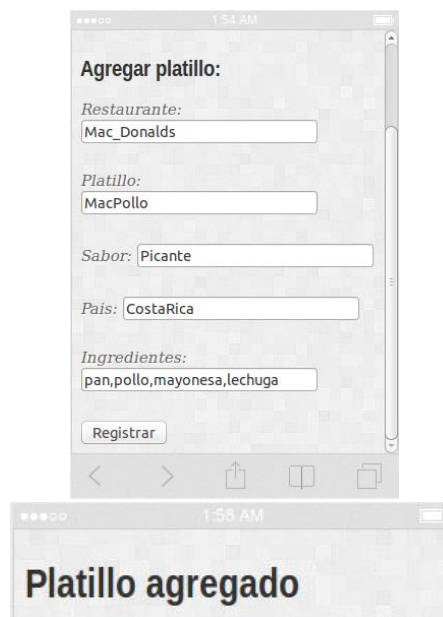
8. Opción Ingresa: se muestran tres opciones, ingresar **Restaurante** o **Platillo**, y Menu que es para regresar al menú de inicio.



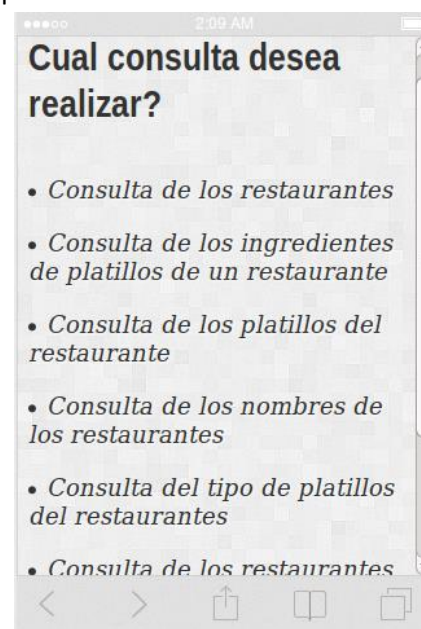
9. Ingresa por Restaurante: debe de ingresar el nombre del restaurante, el tipo de comida, la ubicación, el teléfono y el horario.



10. Ingresa por Platillo: debe de ingresar el nombre del restaurante, el nombre del platillo, el sabor, el país de origen y los ingredientes.



11. Opción Consultar: se muestran las diferentes consultas que se pueden realizar sobre los restaurantes y sus platillos.



12. Consulta de los restaurantes: se muestran en pantalla una lista de todos los restaurantes que están registrados en la aplicación.



13. Consulta de los ingredientes de platillos de un restaurante: debe ingresar un restaurante y el nombre de un ingrediente.



14. Consultar los platillos del restaurante: se debe ingresar el restaurante al que se le van a consultar los platillos.



15. Consulta de los nombres de los restaurantes: se debe ingresar el nombre del restaurante al que se va hacer la consulta.



16. Consulta del tipo de platillos del restaurante: se debe ingresar el tipo de comida de los restaurantes para realizar la consulta.



17. Consulta del restaurante según el país del cual es el platillo: se debe ingresar el nombre del país de origen del platillo.



Conclusión personal

Con la realización de este proyecto se pudo concluir que es de suma importancia, antes de empezar a programar, tener conocimiento sobre el campo en el que se está trabajando, investigar sobre diferentes funciones que son útiles para la realización del trabajo, y comprender su funcionamiento. Analizar y entender cuáles son las necesidades, qué es lo que se busca y el resultado al que se quiere llegar.

Para finalizar, es relevante mencionar que para facilitar la confección del programa, un método que se puede poner en práctica es la realización de un plan de trabajo que permita ir modelando un esquema o un diagrama de las ideas para la elaboración del sistema, y así poder ir implementando los algoritmos que se establecieron y afinar los detalles para una buena producción del programa deseado.