



6-5-2014

Paola Solano.
Veronica R. Vargas.
Gabriela Reyes.

DOCUMENTACIÓN

Segunda Tarea Programada | Lenguajes de Programación

Tabla de contenido

Resumen Ejecutivo	2
Propósitos	3
Descripción del app	3
Requerimientos del sistema	3
Funcionalidades	5
Manejo de datos	5
Consulta de información	5
Descripción de diseño de alto nivel	6
Descripción detallada	8
Conexión Python-Prolog	8
Desarrollo del framework	10
Creación de HTML	12
Análisis de Resultados	14
Análisis de problemas encontrados	14
Instalación de librerías y programas	14
Escogencia del framework	14
Archivos html	14
Análisis de la aplicación	15
Manual de Usuario	16
Instalación de librerías y programas	16
Instrucciones de uso de la aplicación web.	19
Realizando consultas de Platos	20
Agregando Restaurantes.	22
Agregando Platos	22
Realizando búsqueda de Restaurantes	23
Conclusión	27

Resumen Ejecutivo

El proyecto consiste en el desarrollo de una aplicación que permite al usuario consultar y almacenar información sobre restaurantes y platillos.

A la aplicación web se puede acceder por medio de un teléfono móvil.

El usuario realizará diferentes consultas, las cuales permiten obtener la lista de los restaurantes registrados en el sistema, buscar información de un restaurante y de los platillos de este. También se podrá obtener el nombre de los platillos que pertenecen a un país, así como el nombre del restaurante que contiene dichos platillos en el menú. Si el usuario desea un ingrediente específico de un restaurante, la aplicación devolverá la lista de los platillos que contienen dicho ingrediente.

El sistema además permite que se ingresen nuevos datos los cuales pueden ser información de un restaurante o información de los platillos de los restaurantes, lo que promueve que los datos aumenten constantemente, y que las consultas presenten mayor información.

¿Qué es lo que se va a hacer?

Se desarrollará una aplicación móvil, que permita la consulta y almacenamiento de datos sobre restaurantes y los platillos de estos.

¿Quiénes lo van a hacer?

Paola Solano, Gabriela Reyes y Verónica R. Vargas, estudiantes del curso Lenguajes de Programación, de la carrera de Administración de Tecnología de la Información, del Instituto Tecnológico de Costa Rica

¿Cómo lo van a hacer?

Con los lenguajes de programación Prolog y Python se desarrollará el sistema de consultas y almacenamiento, asimismo se utilizará el framework flask para la implementación en la web.

¿Por qué se va a hacer?

La aplicación se desarrollará para facilitar la búsqueda de los restaurantes y sus platillos, obteniendo dicha información de forma eficiente.

Propósitos

En esta sección se presentará la descripción de la aplicación, y los requerimientos de diseño que presenta la misma.

Descripción del app

La aplicación está diseñada para que los usuarios realicen diversas consultas sobre restaurantes y sus platillos, asimismo ellos podrán almacenar nueva información que no se encuentre en el sistema.

Los datos que se almacenarán de los restaurantes serán: nombre del restaurante, tipo de comida que ofrece, ubicación, horario de atención y teléfono. Asimismo la información que se guardará de los platillos corresponderá a: nombre del restaurante en el cual se encuentra el platillo, nombre del platillo, tipo de platillo, país de origen y lista de ingredientes del platillo.

Las consultas que se realizarán a la aplicación serán:

- Lista de todos los restaurantes que se encuentran en el sistema.
- Lista de los platillos que pertenecen a un restaurante específico.
- Lista de los platillos que poseen un ingrediente en particular de acuerdo a un restaurante.
- Lista de restaurantes de acuerdo al tipo de comida.
- Información de un restaurante específico.
- Lista de platillos que pertenecen a un país específico.

El acceso al sistema será por medio de la web, los usuarios podrán ingresar a la aplicación desde un dispositivo móvil.

El usuario ingresará a la página web, la cual presentará el nombre de la aplicación y las diferentes funcionalidades del sistema. Con solo un click en el nombre de la acción que desee realizar la página presentará la información correspondiente.

Requerimientos del sistema

El sistema está estructurado en dos partes back-end y front-end.

El front-end presenta la interacción entre el usuario y la aplicación, permite la entrada y salida de los datos. Este módulo se desarrolló en Python 2.7 y se utilizó el framework flask para la implementación en la web. Además al acceder a la aplicación a través de un Smartphone la aplicación se debe adaptar a la pantalla

de dicho dispositivo, para lo cual se utilizó el modelo responsive, el cual es definido en el archivo ccs e invocado desde el archivo html de la aplicación.

El back-end es el encargado de realizar las consultas y almacenar la información en la base de conocimientos. El front-end se conecta con el back-end para enviar la información correspondiente.

La consulta de datos se realiza por medio de Prolog, el cual presenta una serie de reglas que permiten obtener la información, este programa envía la información necesaria a Python, por medio de una librería denominada Pyswip. La información es enviada al front-end.

En el almacenamiento de datos, la información recopilada del front-end es guardada en la base de conocimientos.

Funcionalidades

La aplicación presenta dos funcionalidades: Manejo de datos y consulta de información.

Manejo de datos

La aplicación permitirá al usuario agregar nuevos restaurantes, para lo cual se debe ingresar el nombre del restaurante, tipo de comida que ofrece el restaurante, la ubicación, el número de teléfono del restaurante y el horario de atención de dicho restaurante.

Asimismo el cliente podrá ingresar nuevos platillos, para lo cual la aplicación solicitará el nombre del platillo, el nombre del restaurante que ofrece dicho platillo, el tipo de platillo, el país de origen del platillo y la lista de ingredientes que presentan dicho platillo.

Consulta de información

El sistema de consultas que presenta la aplicación será el siguiente:

- Búsqueda de restaurante, el usuario ingresará en la aplicación el nombre del restaurante que desea consultar, y la aplicación desplegará una lista con los datos del restaurante, los cuales serán ubicación, número de teléfono, tipo de comida y horario de atención.
- Búsqueda de platillos de un restaurante, el sistema solicitará al usuario que ingrese el nombre del restaurante, el programa realizará la búsqueda y si encuentra la información, procederá a desplegar la lista de platillos que pertenecen al restaurante.
- Búsqueda de platillos por país de origen, la persona colocará el nombre del país, y el sistema le mostrara la lista de platillos que pertenecen a ese país, así como el nombre del restaurante que ofrece el platillo.
- Búsqueda de restaurantes, el sistema desplegará una lista con el nombre de todos los restaurantes que se encuentran registrados en la aplicación.
- Búsqueda de platillos por ingrediente, el sistema pedirá que se ingrese el nombre del ingrediente y el nombre del restaurante, la aplicación realizará la búsqueda y devolverá la lista de todos los platillos del restaurante, que poseen dentro de sus ingredientes, el ingrediente solicitado por el usuario.
- Búsqueda de restaurantes por tipo de comida, el usuario ingresará al sistema el tipo de comida, y la aplicación presentará una lista con el nombre de todos los restaurantes que presentan ese tipo de comida.

Descripción de diseño de alto nivel

Para la implementación de la aplicación se dividió el problema en partes más pequeñas, las cuales son la conexión Python-Prolog, el diseño responsive y el desarrollo del sistema en el framework.

La aplicación inicia al ejecutar el archivo con la extensión .py, el cual contiene el framework y las funciones de consulta y almacenamiento de datos.

Este archivo de Python, abre una página web por medio de una invocación a un archivo con la extensión .html, el cual es el responsable de la interacción cliente y aplicación.

Este archivo html contiene diferentes referencias que permiten el acceso a las diferentes funcionalidades de la aplicación. Asimismo el archivo html invoca a un archivo ccs el cual contiene la información del modelo responsive.

El archivo html que presenta las referencias a las diferentes funcionalidades de la aplicación envía al archivo de Python la información obtenida del usuario, e invoca la función indicada por el cliente en el sistema. La conexión entre la página web y Python se realiza por el framework denominado flask, el cual es importado en Python.

El framework contiene funciones propias que permiten obtener los datos del html y enviar los datos al html. Asimismo es el que genera las direcciones de la página en la web.

Las funciones en el archivo de Python permiten el acceso a la base de conocimientos de la aplicación, a través de la librería Pyswip el sistema realiza las consultas a la base de conocimientos. El sistema abre la base de conocimientos, con el query y las reglas de inferencia generadas en la aplicación, el sistema procede a realizar las consultas necesarias.

Para el almacenamiento de datos se recopila la información del html al archivo de Python, y este abre el archivo .pl y guarda los datos en la base de conocimientos.

Diagrama que presenta la estructura del sistema.

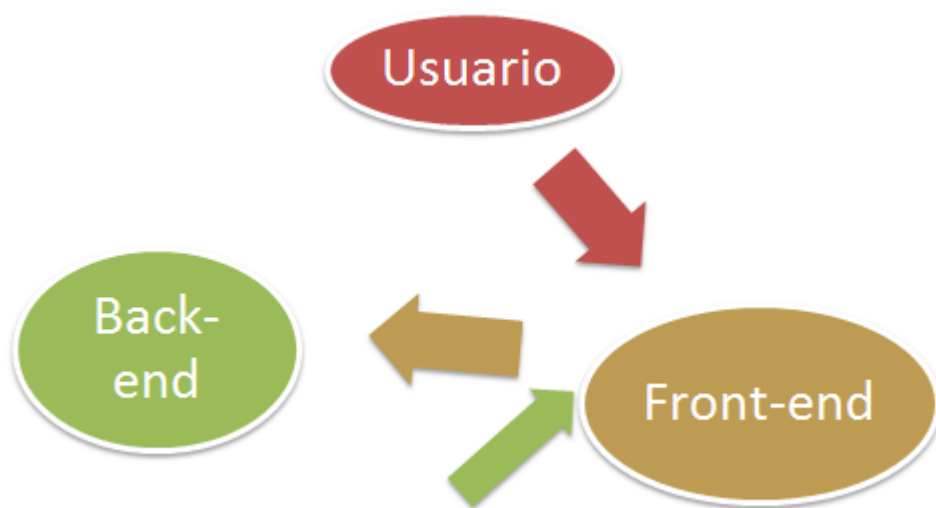
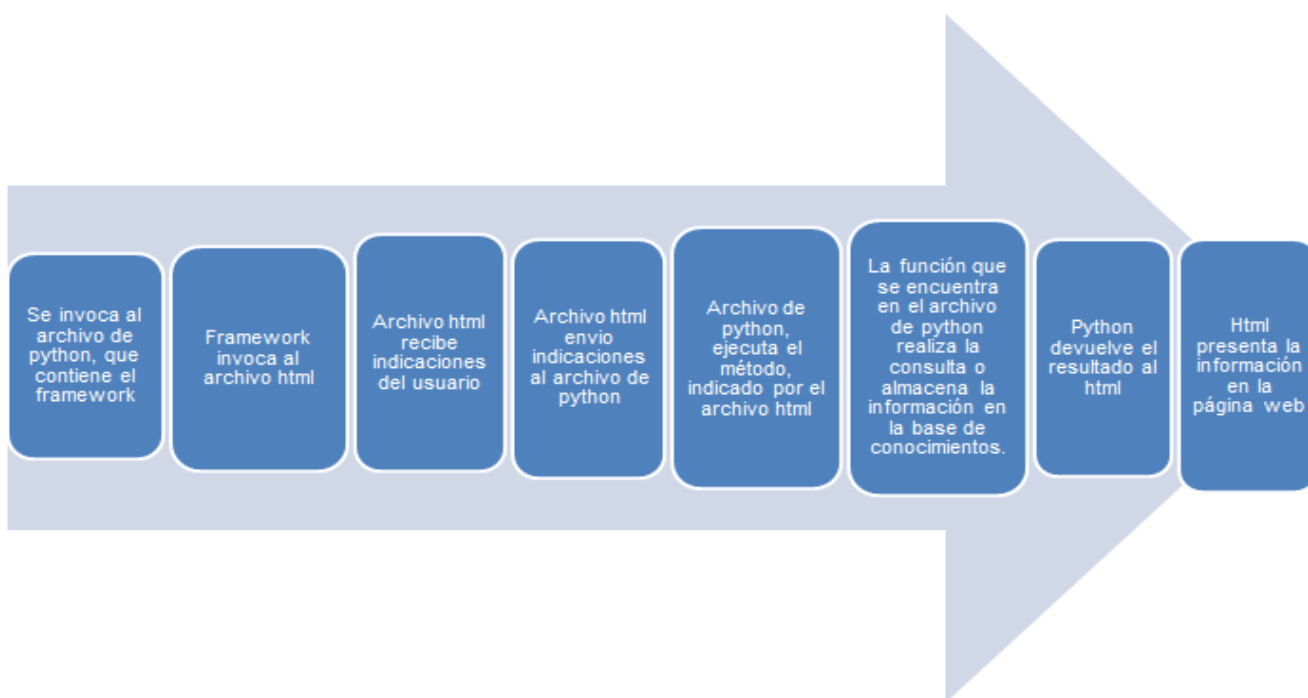


Diagrama de flujo que presenta la forma en que se ejecuta la aplicación



Descripción detallada

Conexión Python-Prolog

El archivo de Python importara la librería pyswip, la cual permite la conexión entre Prolog y Python. Además se llama a Prolog, esta llamada es almacenada en la variable BaseConocimientos. Posteriormente a esa variable BaseConocimientos se le agrega la función .consult("Base_de_conocimientos.pl") la cual invoca la base de conocimientos y le indica el nombre del archivo con extensión .pl que debe abrir para obtener la información de la base de conocimientos.

Ingresar datos a la base de conocimiento.

1. **IngresarINFOrestaurante**, a este método se le ingresa el nombre del restaurante, el tipo de comida, el lugar, el número telefónico, y los días de atención, el método abre el archivo Base_de_conocimientos.pl e incluye en la base de conocimientos la nueva información del restaurante. El formato del hecho es el siguiente: restaurante(Nombre_restaurante, tipo_de_platillo, lugar, teléfono, días de atención). Si la información se almaceno correctamente el método retornará un mensaje indicando que los datos se guardaron correctamente.
2. **IngresarPlatillo**, el método recibe el nombre del restaurante, el nombre del platillo, el tipo de platillo, el país de origen del platillo y la lista de ingredientes del que conforman el platillo. El método abre la base de conocimientos y los inserta, el formato del hecho es el siguiente: platillos(Nombre_Restaurante, Nombre_Platillo, Tipo_de_Platillo, país_de_origen, lista_de_ingredientes). Si la información se almaceno con éxito la función devolverá un mensaje.

Consulta de datos

Las consultas a la base de conocimientos se realizarán con el método de Pyswip denominado query, el cual consulta en la base de conocimientos la información respectiva. La sintaxis del método es la siguiente BaseConocimientos.query(Definición de la regla que obtiene la información de Prolog). La información obtenida del método es almacenada en una lista.

1. **L_restaurantes**, función que obtiene la lista de todos los restaurantes que se encuentran en la base de conocimientos. La consulta es la siguiente: BaseConocimientos.query(listarestaurantes(Nombre)). En el archivo Base_de_conocimientos.pl se encuentra la siguiente regla listarestaurantes(NombreR):-restaurante(NombreR,_,_,_,_). La cual busca

todos los nombres de los restaurantes. Al finalizar la búsqueda la función retorna la lista de todos los restaurantes. Si no encontró información el sistema presentara el mensaje de que no se encuentran resultados.

2. **Buscar_restaurante_por_tipo_de_comida**, el método recibe el tipo de comida que se desea buscar, y se realiza la siguiente consulta `BaseConocimientos.query("restaurantes_tipo_comida(X," + tipoDeComida +"))`, el programa buscará en la base de conocimientos los hechos que cumplan la regla. La regla en el archivo de prolog se encuentra de la siguiente forma: `restaurantes_tipo_comida(NombreR,TipoComida):- restaurante(NombreR,TipoComida,_,_,_,_)`. Prolog devolverá la lista de todos los restaurantes que posean ese tipo de comida y esta misma lista la retornara python.
3. **Buscar_restaurante**, la función recibe el nombre del restaurante que desee buscar, y se enviará a la base de conocimientos la siguiente consulta: `BaseConocimientos.query("buscar_restaurante("+NombreR+",TipoComida,Lugar,Tel,DiasAtencion, HorasAtencion2)")`, el archivo de prolog realizará la consulta y retornara la información correspondiente al restaurante consultado, el tipo de comida que ofrece, el lugar, el número telefónico. En el archivo de prolog la regla se encuentra de la siguiente forma: `buscar_restaurante(NombreR,TipoComida,Lugar,Tel,DiasAtencion, HorasAtencion2):restaurante(NombreR,TipoComida,Lugar,Tel,DiasAtencion , HorasAtencion2)`.
4. **Buscar_platillos_pais_origen**, el método recibe el nombre de un país, y este deberá devolver la lista de los platillos que pertenecen a ese país. Asimismo se devolverá el nombre del restaurante que ofrece ese platillo. La consulta en Python será: `BaseConocimientos.query("platillos(NombreR,NombreP,_, "+NombrePais+", _)")`.
5. **Buscar_platillos_restaurante**, al método se le ingresa el nombre del restaurante, la consulta obtiene la lista de platillos que pertenecen a dicho restaurante, por lo que la consulta será: `BaseConocimientos.query("platillos_restaurantes("+NombreR+",NombrePlatillo)")`, la regla en el archivo de prolog será la siguiente: `platillos_restaurantes(NombreR,NombrePlatillo):- platillos(NombreR,NombrePlatillo,_,_,_)`.
6. **Buscar_ingredientes_platillo_restaurante**, el método recibe el nombre del restaurante y el ingrediente que se desea buscar. Y la función retornará la lista de platillos del restaurante que poseen ese ingrediente dentro de la receta. La consulta en Python es la siguiente: `BaseConocimientos.query("ingrediente_platillos_restaurante("+NombreR+", NombrePlatillo,"+Ingrediente+")")`. En el archivo de Prolog se utilizó la

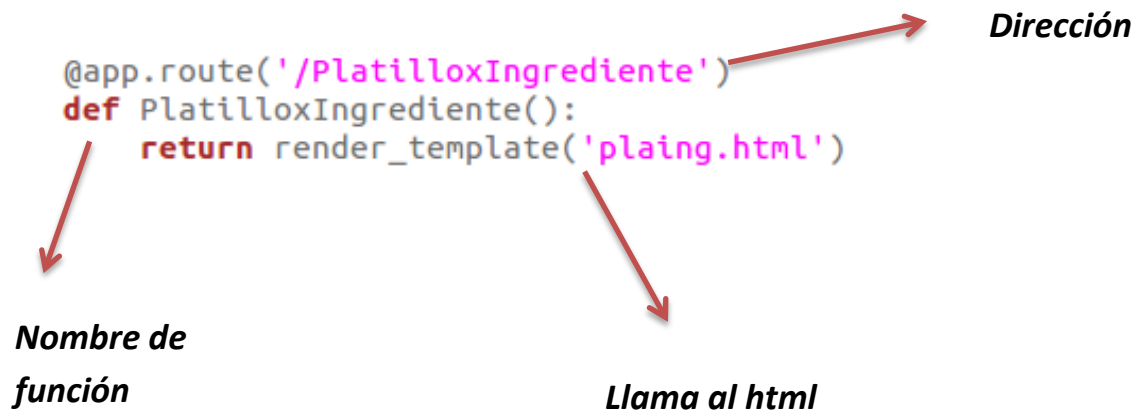
función `member`, la cual determina si un elemento se encuentra en la lista.
`ingrediente_platillos_restaurante(NombreR,NombrePlatillo,Ingrediente):-`
`platillos(NombreR,NombrePlatillo,_,_,[IngredienteL|Lista]),`
`member(Ingrediente,[IngredienteL|Lista]).`

Desarrollo del framework

Para el diseño de la aplicación web, se utilizó el framework flask, esto debido a su facilidad de uso, y además de que con este framework se lograron acceder a las consultas en la base de datos, ya que con los utilizados anteriormente no se logró. Para utilizar flask se importa la librería flask en Python, además, las diferentes funciones y comandos que se utilizan para las consultas y obtención de datos. Para la implementación de la web, además de flask, se utilizó como complementó `jinja2`; motor de plantillas + html.

Las funciones que se encuentran realizadas con flask, poseen una estructura muy similar entre todas.

Primero se asigna una dirección url a la función que se desea crear, mediante el comando `app.route` seguidamente se define la función; se indica el nombre por medio del comando `def`, luego se utiliza un `return` y el comando `render_template` de la librería `flask` para llamar a un html que contiene el diseño web de la aplicación para cada función.



Funciones web

- **Home:** Función que se encuentra el menú principal de la aplicación y es la encargada de llamar a la página principal.
- **AgregaPlatillo:** Función que se encuentra el menú principal de la aplicación y es la encargada de llamar a la página (html) para que el usuario pueda agregar los datos de un platillo a la base de conocimientos.

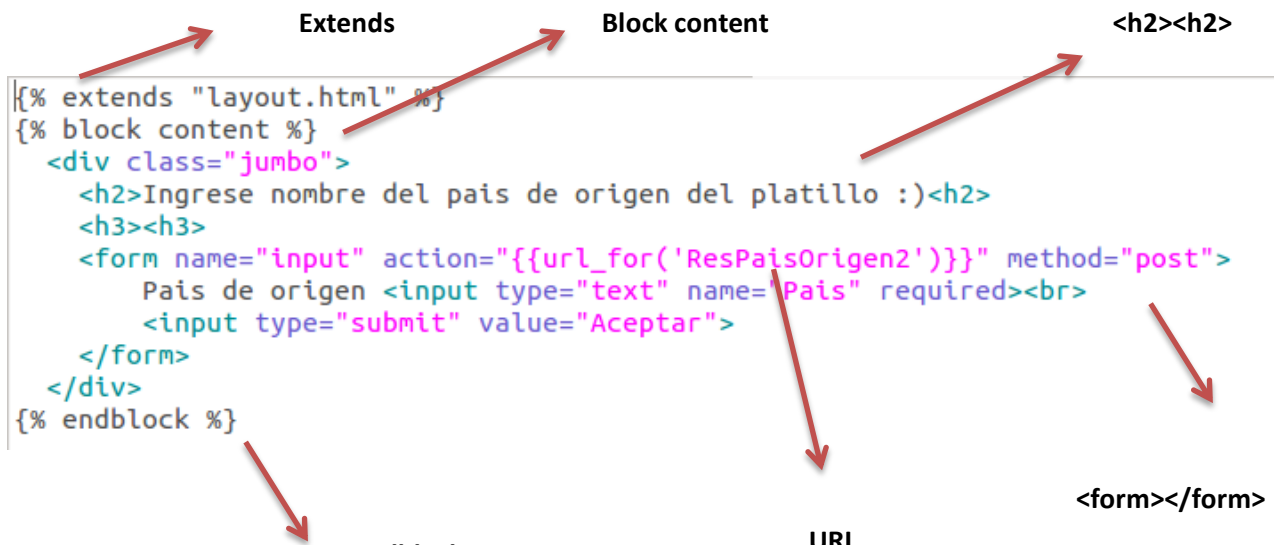
- **AgregaPlatillo2:** Recibe los datos ingresados por el usuario para la inserción del platillo en la base de conocimientos. Se utilizó el comando *request.form[""]* para recibir los datos con mayor facilidad, los datos son guardados en una variable cada uno, y después son ingresados en la función de IngresarPlatillo que escribe en la base de conocimiento. Se es permitido ingresar 5 ingredientes por platillo. Una vez terminada la inserción se dirige nuevamente a la página por si desea agregar otro platillo.
- **AgregaRestaurante:** Función que se encuentra el menú principal de la aplicación y es la encargada de llamar a la página (html) para que el usuario pueda agregar los datos de un restaurante a la base de conocimientos.
- **AgregaRestaurante2:** Recibe los datos ingresados por el usuario para la inserción de un restaurante en la base de conocimientos. Se utilizó el comando *request.form[""]* para recibir los datos, estos son guardados en una variable cada uno, y después son ingresados en la función de IngresarINFORestaurante que escribe en la base de conocimiento.
- **PlatilloxRestaurante:** Método que se encuentra el menú principal de la aplicación y es el encargado de llamar a la página (html) para que el usuario pueda consultar los platillos que ofrece un restaurante.
- **PlatilloxRestaurante2:** Recibe el nombre del restaurante del cual se desea obtener la información de la lista de platillos que posee. Retorna a una página en la que se encuentra la lista de platillos.
- **PlatilloxIngrediente:** Método que se encuentra el menú principal de la aplicación y es el encargado de llamar a la página (html) para que el usuario pueda consultar los restaurantes que tienen platillos con un ingrediente específico.
- **PlatilloxIngrediente2:** Recibe el nombre del restaurante y el nombre del ingrediente que se desea consultar en la base de conocimientos, realiza la llamada a la función que consulta y retorna una página con la lista de platillos que contienen el ingrediente que ha sido consultado.
- **ResPaisOrigen:** Función que se encuentra el menú principal de la aplicación y es la encargada de llamar a la página (html) para que el usuario pueda consultar los platillos que pertenecen a un país en específico.
- **ResPaisOrigen2:** Recibe el nombre del país, llama a la función que realiza la búsqueda de los restaurantes, y retorna una página con los datos seleccionados.
- **RestauranteEspecifico:** Esta función se encuentra en el menú principal, es la encargada de realizar la búsqueda de los restaurantes que se encuentran almacenados en la base de conocimientos.
- **RestauranteEspecifico2:** Función encargada de recibir el nombre del restaurante que es enviado como parámetro a la función que realiza la

consulta. Retorna una página con la lista de restaurantes que se encuentran almacenados.

- **Platillo:** Función que se encuentra en el menú y es la encargada de buscar los restaurantes por un tipo de comida, hace referencia a una página que contiene la estructura para ingresar los datos.
- **Platillo2:** Método que recibe el tipo de comida que el usuario desea consultar y lo pasa como parámetro a la función que realiza la consulta. Retorna una página con la lista de restaurantes que poseen el platillo indicado.
- **Principal:** Función que posee el menú principal con todas las acciones que puede realizar el usuario.

Creación de HTML

La estructura de los HTML es la misma para todos los archivos de este tipo que son utilizados en el programa. Debido a que son muchos los documentos de este mismo tipo, a continuación se explicará la estructura de uno de ellos que facilitará el entendimiento de los demás. Primero se indicarán las partes del archivo y luego se dará una breve descripción de la funcionalidad que tienen.



Extends: El `{%extends endblock` etiqueta `class="jumbo"` indica que esa plantilla se entiende de la plantilla "layout.html". Entonces, cuando el sistema de plantillas evalúa esta plantilla, primero se localiza el padre en este caso "layout.html".

Block content: El `{%block content%}` es utilizado para agregar el contenido de la plantilla.

<h2><h2>: Estos comandos son para agregar líneas al html el número 2 indica el tamaño de la letra que se coloque entre los dos comandos. Entre más bajo sea el número más grande es la letra.

<form></form>: Dentro de estos comandos lo que se realiza en la inserción de datos por medio de los inputs.

URL: este comando indica la dirección a la que se debe ingresar cuando se presione el botón de aceptar. Esa dirección es referenciada a una de las funciones que se encuentra dentro del archivo .py

Endblock: El {% endblock %} es utilizado para indicar que el bloque de esa plantilla ha finalizado

.

Análisis de Resultados

Análisis de problemas encontrados

Instalación de librerías y programas.

Al realizar la instalación de Pyswip la librería presento problemas a las personas del grupo de trabajo que poseían el SO de 64 bits.

Asimismo la instalación de flask presento problemas, ya que en internet se encuentran diferentes formas de instalarlo. Algunos miembros del equipo de trabajo no descargaron todos los paquetes que se deben instalar. Por lo que al crear los archivos y ejecutarlos el programa presento problemas que indicaban que faltaban ciertas librerías de flask.

También al realizar la creación de los documentos html, se encontró que los programas en flask deben seguir cierta estructura, por lo se realizaron varias pruebas y búsquedas en internet, para determinar cuál es la forma en que los archivos se almacenan.

Escogencia del framework

Se investigaron los diferentes framework que presenta Python, y se analizaron cuáles eran los más convenientes. Al inicio se escogió Django, pero se descartó por que presentaba problemas al instalarlo. Se procedió a utilizar Apache, pero este presento problemas al realizar el virtualhost, el cual es un requerimiento de este framework. Luego se implementó el programa con Cherrypy, se empezaron a desarrollar los archivos html y el diseño responsive, pero al realizar las consultas a la base de conocimientos la página web se caía. Por lo que finalmente se decidió trabajar con Flask, el cual presentó unos pequeños problemas, los cuales se lograron solucionar.

Archivos html

Al inicio de la implementación en la página web, se presentaron inconvenientes con la lectura de los archivos html, sin embargo, esto se resolvió creando una carpeta denominada “templates” en la misma dirección que contenía el archivo .py. El error se dio debido a que flask a la hora de leer un archivo html dentro de una función, busca automáticamente dentro de una carpeta “templates”, la cual no fue creada a la hora de realizar la instalación de flask.

Análisis de la aplicación

A la hora de realizar consultas a la base de conocimientos, el servidor se caía, después de varios intentos de arreglarlo el programa funcionó. Cuando se realizaba una consulta a la base de conocimientos, lo que retornaba la función que hacía la consulta, era una lista, por lo que a la hora de imprimirlo, se imprimía en una sola línea, sin embargo, por medio de la investigación en internet, se logró encontrar una función que separara los argumentos. Por lo cual al realizar una consulta se logra ver una lista con los datos retornados.

La aplicación funciona correctamente siguiendo el manual de usuario, no obstante, algunas validaciones no pudieron ser implementadas. Algunas funciones como el `.lower()` y el `cambiar_espacios()` no funcionaron con los datos ingresados, por lo que se tuvieron que comentar.

Para la implementación de los html se utilizó jinja2, que resultó muy fácil de utilizar para la creación de los diseños de la página y la obtención de datos que ingresa el usuario.

No se ha podido implementar el css que da formato a la aplicación.

Manual de Usuario

Este manual permitirá aprender a utilizar la aplicación de búsqueda de restaurantes y platillos, en él se las funcionalidades el mismo y las instrucciones de uso.

Para facilidad en la ejecución y el uso de la aplicación el manual de usuario se compone de dos secciones:

- Instalación de librerías y programas.
- Instrucciones de uso de la aplicación web.

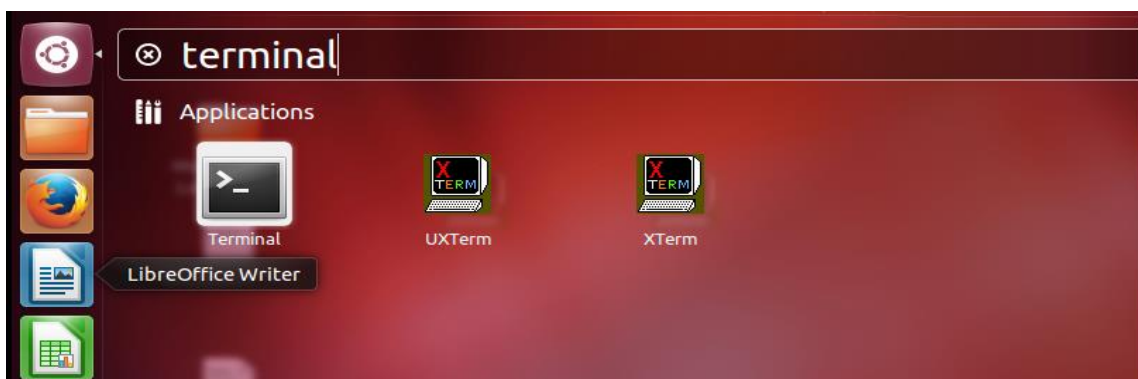
Instalación de librerías y programas.

Para la ejecución correcta del programa, el usuario deberá instalar las siguientes librerías y programas, la aplicación funcionará en el sistema operativo de Linux.

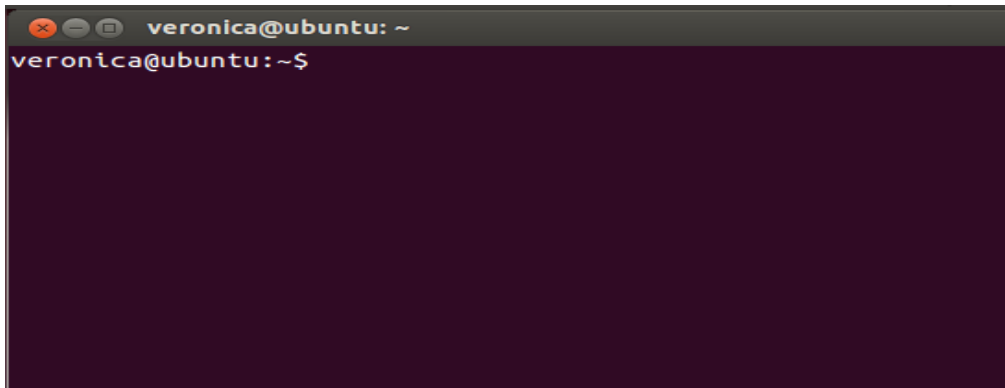
La instalación de los programas y librerías en el sistema operativo Ubuntu se realizará por medio de líneas de comando. Para ello se procederá a abrir la terminal.

1) Abrir la terminal.

Ir al cuadro de “Dash Home” y colocar en la barra de búsqueda la palabra “Terminal”.



Ingresa a la terminal, después del símbolo “\$” se colocan las líneas que permiten la instalación de los diferentes archivos.



2) Instalación de Python 2.7.

Si el programa no se encuentra instalado en la computadora se procederá a abrir la terminal y a colocar la siguiente información.

- 3) **Instalación de Python 2.7.** Si el programa no se encuentra instalado en la computadora se procederá a abrir la terminal y a colocar la siguiente información.

```
wget http://www.python.org/ftp/python/2.7.6/Python-2.7.6.tgz
tar -xzf Python-2.7.6.tgz
cd Python-2.7.6
./configure
make
sudo make install
```

Estas líneas permiten ingresar al sitio web de la aplicación, descargar Python 2.7, abrir el archivo y configurar el archivo para que se pueda usar en el sistema operativo.

3) Instalación de Prolog.

El programa que se utilizará será SWI-Prolog, para lo que se ingresa a la terminal y se coloca la siguiente línea.

```
sudo apt-get install swi-prolog
```

4) Instalación de Pyswip.

Es una librería de Python que permite la conexión entre Prolog y Python. Por lo que se descarga e instala desde la terminal de la siguiente forma.

```
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
```

5) Instalación de Flask.

Este es el framework que permite la conexión de Python a la página web. Flask se descarga desde la terminal de la siguiente forma.

```
sudo easy_install virtualenv
mkdir myproject
cd myproject
virtualenv venv
New python executable in venv/bin/python
Installing distribute.....done.
. venv/bin/activate
pip install Flask
```

Instrucciones de uso de la aplicación web.

Una vez realizadas la instalaciones anteriores se procesa a compilar el programa para lo cual es necesario acceder a la terminal, y ubicarse a la carpeta donde se ha guardado el archivo del programa y desde ahí dar el comando “python dok.py”. Este comando genera una dirección web, como se muestra a continuación.

```
andrepao096@ubuntu: ~/Downloads
andrepao096@ubuntu:~$ cd Downloads
andrepao096@ubuntu:~/Downloads$ python dok.py
Warning: /home/andrepao096/Downloads/Base_de_conocimientos.pl:23:
      Clauses of restaurante/6 are not together in the source-file
* Running on http://127.0.0.1:5000/
```

Con la tecla “control” y realizando un click derecho en la dirección web nos aparecerá la aplicación con el siguiente menú.



Busqueda de Restaurantes

- [Inicio](#)
- [Menu](#)
 - [Consulta de platillos](#)
 - [De acuerdo al restaurante](#)
 - [De acuerdo a ingrediente especifico](#)
 - [Agregar Restaurantes](#)
 - [Agregar Platillos](#)
 - [Busqueda de restaurantes](#)
 - [Lista General](#)
 - [Informacion de restaurante](#)
 - [De acuerdo al pais de origen del platillo](#)
 - [Lista tipo de comida](#)

BIENVENIDO

Realizando consultas de Platos.

Al presionar la opción “De acuerdo al restaurante” nos referenciará a la siguiente ventana.



127.0.0.1:5000/PlatilloxRestaurante

Busqueda de Restaurantes

- [Inicio](#)
- Menu
 - Consulta de platos
 - [De acuerdo al restaurante](#)
 - [De acuerdo a ingrediente especifico](#)
 - [Agregar Restaurantes](#)
 - [Agregar Platos](#)
 - Busqueda de restaurantes
 - [Lista General](#)
 - [Informacion de restaurante](#)
 - [De acuerdo al pais de origen del platillo](#)
 - [Lista tipo de comida](#)

Ingrese nombre del platillo

Nombre del Restaurante:

Esta ventana nos pide un nombre de restaurante, una vez escrito el nombre de restaurante presione el botón “Aceptar”. Este botón responderá la consulta mostrando una lista con los platos que hay en el restaurante consultado, como se muestra en la siguiente imagen, siempre y cuando este se encuentre guardado en la aplicación.



127.0.0.1:5000/PlatilloxRestaurante2

Busqueda de Restaurantes

- [Inicio](#)
- Menu
 - Consulta de platos
 - [De acuerdo al restaurante](#)
 - [De acuerdo a ingrediente especifico](#)
 - [Agregar Restaurantes](#)
 - [Agregar Platos](#)
 - Busqueda de restaurantes
 - [Lista General](#)
 - [Informacion de restaurante](#)
 - [De acuerdo al pais de origen del platillo](#)
 - [Lista tipo de comida](#)

Lista de platos del restaurante kfc

- Nombre del Plato: wrap

En este caso el restaurante consultado fue kfc y el platillo que existe en la aplicación es wrap.

Al presionar la opción “**De acuerdo a ingrediente específico**” nos referenciará a la siguiente ventana.



127.0.0.1:5000/PlatilloxIngrediente

Busqueda de Restaurantes

- [Inicio](#)
- Menu
 - Consulta de platillos
 - [De acuerdo al restaurante](#)
 - [De acuerdo a ingrediente específico](#)
 - [Agregar Restaurantes](#)
 - [Agregar Platillos](#)
 - Busqueda de restaurantes
 - [Lista General](#)
 - [Informacion de restaurante](#)
 - [De acuerdo al pais de origen del platillo](#)
 - [Lista tipo de comida](#)

Ingrese la informacion solicitada

Nombre de Restaurante

Nombre de Ingrediente

Esta ventana permite realizar consultas de los platillos de un restaurante específico que un contienen cierto ingrediente, para ello se debe ingresar los datos solicitados. Una vez ingresados los datos y presionado el botón “Aceptar”, se responderá la consulta mostrando una lista con los platillos que hay en el restaurante consultado y que contienen ese ingrediente, como se muestra en la siguiente imagen.



127.0.0.1:5000/PlatilloxIngrediente2

Busqueda de Restaurantes

- [Inicio](#)
- Menu
 - Consulta de platillos
 - [De acuerdo al restaurante](#)
 - [De acuerdo a ingrediente específico](#)
 - [Agregar Restaurantes](#)
 - [Agregar Platillos](#)
 - Busqueda de restaurantes
 - [Lista General](#)
 - [Informacion de restaurante](#)
 - [De acuerdo al pais de origen del platillo](#)
 - [Lista tipo de comida](#)

Lista de platillos del restaurante kfc con el ingrediente tomate

- Nombre del platillo: wrap

En este caso el restaurante consultado fue kfc con el ingrediente tomate y el platillo que existe en la aplicación es wrap.

Agregando Restaurantes.

Al seleccionar la opción “**Agregar Restaurantes**” se referenciará a la siguiente página.

127.0.0.1:5000/AgregaRestaurante

Busqueda de Restaurantes

- [Inicio](#)
- [Menu](#)
 - Consulta de platillos
 - [De acuerdo al restaurante](#)
 - [De acuerdo a ingrediente especifico](#)
 - [Agregar Restaurantes](#)
 - [Agregar Platillos](#)
 - Busqueda de restaurantes
 - [Lista General](#)
 - [Informacion de restaurante](#)
 - [De acuerdo al pais de origen del platillo](#)
 - [Lista tipo de comida](#)

Agregando Restaurantes

Ingrese la informacion solicitada

Nombre de Restaurante

Ubicacion

Telefono

Tipo de comida del restaurante

Horario: Dia Inicio

Dia Final

Para realizar la operación deberá ingresar los datos solicitados. Al presionar el botón “Aceptar”, aparecerá una nueva página, la cual le permitirá agregar nuevos restaurantes y además le informará que la información anterior fue guarda con éxito.

Agregando Platillos.

Al seleccionar la opción “**Agregar Platillos**” se referenciará a la siguiente página.

127.0.0.1:5000/AgregaPlatillo

Lista tipo de comida

Agregando Platillos

Ingrese la informacion solicitada

Nombre de Restaurante

Nombre Platillo

Tipo Platillo

Pais de Origen

Ingrese sus cinco ingredientes favoritos

Ingrediente1

Ingrediente2

Ingrediente3

Ingrediente4

Ingrediente5

Para realizar la operación deberá ingresar los datos solicitados. Al presionar el botón “Aceptar”, aparecerá una nueva página, la cual le permitirá agregar nuevos platillos y además le informará que la información anterior fue guarda con éxito.

Realizando búsqueda de Restaurantes.

Al seleccionar la opción “**Lista General**” se referenciará a la siguiente página. La cual muestra una lista de los restaurantes guardados en la aplicación.



Al seleccionar la opción “**Información de Restaurante**” se referenciará a la siguiente página.



Para realizar la operación deberá ingresar los datos solicitados. Al presionar el botón “Aceptar”, aparecerá la siguiente ventana, la cual muestra la información general de restaurante.



127.0.0.1:5000/RestauranteEspecifico2

Busqueda de Restaurantes

- [Inicio](#)
- [Menu](#)
 - [Consulta de platillos](#)
 - [De acuerdo al restaurante](#)
 - [De acuerdo a ingrediente especifico](#)
 - [Agregar Restaurantes](#)
 - [Agregar Platillos](#)
 - [Busqueda de restaurantes](#)
 - [Lista General](#)
 - [Informacion de restaurante](#)
 - [De acuerdo al pais de origen del platillo](#)
 - [Lista tipo de comida](#)

Informacion del restaurante

- **Nombre de Restaurante terramall**
- **Tipo de Comida: chatarra**
- **Telefono: 22334455**
- **Horario de atencion: domingo_viernes**

En este caso el restaurante consultado fue kfc y la página muestra a ubicación del lugar, el tipo de comida, el teléfono y el horario de atención.

Al seleccionar la opción “**De acuerdo al país de origen del platillo**” se referenciará a la siguiente página.



127.0.0.1:5000/ResPaisOrigen

Busqueda de Restaurantes

- [Inicio](#)
- [Menu](#)
 - [Consulta de platillos](#)
 - [De acuerdo al restaurante](#)
 - [De acuerdo a ingrediente especifico](#)
 - [Agregar Restaurantes](#)
 - [Agregar Platillos](#)
 - [Busqueda de restaurantes](#)
 - [Lista General](#)
 - [Informacion de restaurante](#)
 - [De acuerdo al pais de origen del platillo](#)
 - [Lista tipo de comida](#)

Ingrese nombre del pais de origen del platillo :)

Pais de origen

Una vez ingresados los datos y presionado el botón aceptar, aparecerá la siguiente página, la cual muestra el nombre del restaurante que contiene platillos originarios de ese país y nombre del platillo.



Al seleccionar la opción “**Lista tipo de comida**” se referenciará a la siguiente página.



Una vez ingresados los datos y presionado el botón aceptar, aparecerá la siguiente página, la cual muestra el nombre de los restaurantes que ofrecen el tipo de comida consultado.

Busqueda de Restaurantes

- [Inicio](#)
- Menu
 - Consulta de platillos
 - [De acuerdo al restaurante](#)
 - [De acuerdo a ingrediente especifico](#)
 - [Agregar Restaurantes](#)
 - [Agregar Platillos](#)
 - Busqueda de restaurantes
 - [Lista General](#)
 - [Informacion de restaurante](#)
 - [De acuerdo al pais de origen del platillo](#)
 - [Lista tipo de comida](#)

Lista de restaurantes por tipo de comida

- **Nombre de Restaurante mussio**
- **Nombre de Restaurante mc_donalds**
- **Nombre de Restaurante kfc**

Para volver al menú principal se presiona “Inicio” esta opción la puede realizar desde cualquier página a la cual será referenciado. Las consultas realizadas se responderá siempre y cuando la información este almacenada en la aplicación. Además los datos que se ingresen deberán estar en minúscula y sin espacios, para el buen funcionamiento de la aplicación.

Conclusión

El proyecto permitió ampliar el conocimiento sobre el uso de las tecnologías en el desarrollo de aplicaciones.

Durante la ejecución del proyecto se desarrolló la investigación, ya que se buscaron diferentes frameworks y se entendió el funcionamiento de cada uno de estos. Asimismo el trabajo permitió que los miembros del equipo conocieran las diferentes librerías que permiten la conexión entre el lenguaje de programación Python y el lenguaje de Prolog.

Al realizar la aplicación en Prolog se aplicaron los conocimientos adquiridos en las clases de Lenguajes de Programación. Además se observó la forma en que un lenguaje de programación puede realizar acciones específicas de otro lenguaje de programación.

En cuanto al desarrollo de la aplicación web se entendió cuáles son los requerimientos para el desarrollo de ésta, asimismo se trabajó con el modelo responsive, el cual permite que la página se adapte al tamaño de cualquier dispositivo, que acceda a la aplicación.

Finalmente el desarrollo de la aplicación nos permitió mejorar las habilidades de búsqueda, análisis e implementación de sistemas informáticos.