

Instituto Tecnológico de Costa Rica



**Proyecto programado número 2**

**“Aplicación móvil para la consulta de restaurantes”**

**Carrera:**

Administración de tecnología de información

**Curso:**

Lenguajes de programación.

**Profesor:**

Andrei Fuentes Leiva.

**Estudiantes:**

Marianne Cordero.

Arlyn Lopez.

Coraima Fonseca.

**Semestre I, 2014**

## Contenido

Resumen ejecutivo .....	3
Propósito .....	5
Descripción detallada. ....	6
<i>Lenguajes de programación usados.</i> .....	10
<i>Tecnologías usadas.</i> .....	11
Problemas encontrados .....	13
Manual de usuario .....	14

## Resumen ejecutivo

El objetivo de “Foodie” es ser una aplicación móvil para la búsqueda e ingreso de restaurantes y platillos favoritos de un enano llamado Bombur, quien vive en Tierra Media.

La aplicación está compuesta por un conjunto de lenguajes de programación para distintos propósitos, así como otras tecnologías importantes. HTML es el lenguaje utilizado para que el diseño se adapte a las distintas dimensiones de los dispositivos móviles, además es lo que compone la interfaz. Python es el lenguaje de programación que tomará los datos y se los enviará a Prolog, el segundo tomándolos e infiriendo lógicamente con la serie de reglas propuestas.

La aplicación se encuentra compuesta de la siguiente manera:

- ❖ **Agregar restaurantes:** esta funcionalidad le permite al usuario agregar sus restaurantes preferidos para lo cual se le solicita el nombre del restaurante, el tipo de comida, ubicación, teléfono y horario del mismo.
- ❖ **Agregar platillos:** el usuario podrá agregar platillos para cada uno de los restaurantes, en donde se le solicitará el ingreso del restaurante al cual pertenece dicho platillo, el sabor que tiene (ya sea picante, salado, dulce, agri dulce o amargo), el país de origen y la lista de ingredientes del mismo.
- ❖ **Consultas:** el usuario podrá realizar una serie de consultas las cuales serán en base a los datos que ingresó acerca de los restaurantes y platillos. La lista de consultas que podrá hacer son:

- **Lista de restaurantes:** le muestra todos los restaurantes que ha agregado.

- **Lista de restaurantes filtrados por tipo de comida:** el usuario deberá indicar el tipo de comida que desea buscar y la aplicación le mostrará todos los restaurantes que posean dicho tipo de comida.
- **Búsqueda de restaurantes por nombre:** se le solicita al usuario que ingrese el nombre del restaurante que desea buscar y posteriormente se le indicará si ese restaurante se encuentra o no registrado.
- **Lista de restaurantes que tienen platillos de algún país específico:** en esta consulta se le solicita al usuario que ingrese el país del cual desea que se muestren sus platillos, con lo cual al final de la búsqueda se le mostrará los restaurantes que tienen platillos de dicho país ingresado.
- **Lista de platillos de un restaurante específico:** se le solicita al usuario que ingrese el nombre del restaurante y después se le mostrará los platillos que este posee.
- **Lista de platillos de un restaurante específico que tengan un ingrediente en particular:** esta consulta permite mostrarle al usuario los platillos que tienen un ingrediente en específico, para lo cual se le solicita que ingrese el ingrediente que desea buscar y el restaurante.

## Propósito

Bombur es un enano de Tierra Media que necesita acceder desde su teléfono inteligente a una aplicación que le permita consultar o agregar los platillos y restaurantes favoritos. Al realizar el registro de cada restaurante se deberá guardar: nombre del lugar, platillos, tipo de comida, ubicación, teléfono y horario. Los platillos a ingresar deberán contener: restaurante al que pertenece, nombre del platillo, sabor, país de origen y lista de ingredientes.

El diseño de la aplicación debe ser responsive, esto para que se adapte a la pantalla de distintos dispositivos móviles. Además, deberá estar compuesta por un Front End, este escrito en Python el cual deberá manejar la interacción con Bombur y un Back End escrito en Prolog que manejara la base de conocimientos para realizar las distintas consultas, dicha base será la encargada de contener los atributos de cada restaurante y platillo ingresado.

Se requiere que las consultas se puedan realizar de la siguiente manera, por lista de restaurantes ingresados, lista de restaurantes basándose en el tipo de comida, búsqueda de restaurantes solo por nombre, lista de restaurantes por país de origen, ahora bien, se debe realizar una serie de consultas para los platillos en los cuales se podrá, obtener la lista de platillos por restaurante y lista de platillos por ingrediente en particular.

## Descripción detallada

### Decisiones de diseño:

La aplicación está compuesta por el front end y el back end. A continuación se describirá de manera detallada el proceso para diseñar cada una de las partes mencionadas:

❖ **Back-End:** es la parte de la aplicación que se encarga de manejar la base de conocimientos y con esto realizar las consultas que indique el usuario. Se utilizó Pyswip, el cual consiste en un puente que permite conectar el lenguaje de programación Python con Prolog. Desde Python se puede realizar la inclusión de hechos y reglas en la base de conocimientos de Prolog, gracias a Pyswip. Desde el back end se realizó lo siguiente:


- **Agregar restaurante:** en esta funcionalidad se le solicita al usuario que ingrese el nombre del restaurante, el tipo de comida, la ubicación, el teléfono y el horario. Los datos ingresados se almacenan en un archivo.pl de manera que no se pierda la información cuando la aplicación cierre.
- **Agregar platillo:** el usuario puede ingresar platillos de algún restaurante, para lo cual se le solicita que ingrese el restaurante al que pertenece ese platillo, el nombre de este, el sabor (si es picante, salado, dulce, agridulce o amargo), el país de procedencia y la lista de ingredientes. Al igual que los datos del restaurante, los datos del platillo se almacenan en un archivo.pl, esto para su posterior uso a la hora de realizar las consultas.
- **Consultas:** la aplicación cuenta con la siguiente lista de consultas:
  - ❖ Lista de restaurantes.
  - ❖ Lista de restaurantes filtrados por tipo de comida.
  - ❖ Búsqueda de restaurantes por nombre.
  - ❖ Lista de restaurantes que tienen platillos de algún país específico.
  - ❖ Lista de platillos de un restaurante específico.
  - ❖ Lista de platillos de un restaurante específico que tengan un ingrediente en particular.

Algoritmo general para realizar las consultas:

Lo primero que se realiza es la lectura del archivo donde se encuentran los datos del restaurante (estos datos están separados por “,”), dicha lectura almacena cada restaurante con sus respectivos datos en una lista, y al final retorna una lista con sublistas, en donde en cada sublista irá toda la información de cada restaurante, es decir, de la siguiente manera: [['mac,chatarra,cartago,2551-87-90,lunes a viernes'], ['pizza hut,chatarra,guanacaste,2637-90-23,lunes a domingo']].

Posterior a la lectura del archivo, se crea un ciclo en el cual el propósito es introducir los datos del restaurante en la base de conocimientos; se toma cada elemento de la lista que devolvió la lectura del archivo (llamada **lista** en el programa), dicho elemento corresponde a una lista con los datos de cada restaurante, esa lista tiene como nombre **lista\_restaurantes** (es una lista que almacena un solo string, el cual es la información del restaurante, por ejemplo ['mac,chatarra,cartago,2551.87.90,lunes a viernes']). Después se crea una lista que se llama **lista\_restaurantesFinal** en la cual se almacena cada dato por separado de la lista, es decir, una lista que contenga por ejemplo: ["mac", "chatarra", "cartago", "2551-87-90", "lunes a viernes"]. Cada elemento de esta lista se almacenará en una variable, la cuales serán los datos que irán almacenados en el hecho restaurante(A,B,C,D,E), por ejemplo se crea el hecho restaurante("mac", "chatarra", "cartago", "2551-87-90", "lunes a viernes") y una vez hecho este, se ingresa en la base de conocimientos mediante `prolog.assertz(restaurante("mac", "chatarra", "cartago", "2551-87-90", "lunes a viernes"))`, este procedimiento se realiza con cada restaurante que se encuentre en el archivo. La búsqueda se hace mediante `prolog.query`, la cual permite hacer la consulta en la base de conocimientos.

Ahora se explicará que retorna cada consulta que utiliza este algoritmo.

 **Lista restaurantes:** una vez ingresado el hecho en la base de conocimientos se procedió a realizar un ciclo for de la siguiente manera:

```
for soln in prolog.query("restaurante(X,Y,Z,D,C)":  
    print "Nombre restaurante:" , soln["X"]
```

En dicho ciclo se busca en la base de conocimientos el hecho **restaurante(X,Y,Z,D,C)** y cada vez que lo encuentre entonces imprimirá el nombre del restaurante.

- ✚ **Lista de restaurantes filtrados por tipo de comida:** en esta consulta, aparte de crear el hecho **restaurante(A,B,C,D,E)** se ingresa en la base de conocimientos la regla **comida(X,A):-restaurante(A,X,\_,\_,\_)**, en la cual el encabezado se cumple si se cumple el hecho de que A es restaurante

Se realiza la consulta de la siguiente manera:

```
for soln in prolog.query("comida("+ X + "," + "A"+")"):
    print "Nombre restaurante:" , soln["A"]
```

Al final imprime todos los restaurantes que tengan X tipo de comida.

- ✚ **Búsqueda de restaurantes por nombre:** se utiliza el algoritmo general, se le solicita al usuario que ingrese el nombre del restaurante que desea buscar y al final se le indica si el restaurante se encuentra o no en la base de conocimientos.

- ✚ **Lista de platillos de un restaurante específico:** aquí se crea el hecho **nombre\_platillo(Restaurante,Platillo,Sabor,Pais\_Origen,Ingredientes)** que sigue la misma lógica del algoritmo general y se ingresa en la base de conocimientos. Los ingredientes se almacenan en una lista. Al final se ingresa a la base la regla **platillos(A,B):- restaurante(A,\_,\_,\_,\_),nombre\_platillo(A,B,\_,\_,\_)**. El ciclo para la búsqueda es el siguiente:

```
for soln in prolog.query(platillos(A,X)):

    print "Los platillos son: " + soln["X"]
```

Y se imprimen todos los platillos de un restaurante en particular.

- ✚ **Lista de restaurantes que tienen platillos de un país específico:** a diferencia de las demás consultas, se crea la regla **lista\_pais(X,Y):- restaurante(Y,B,C,D,E),nombre\_platillo(Y,F,G,X,I)** y la búsqueda se realiza de la siguiente manera:

```
for soln in prolog.query(lista_pais(X,Y)):

    print "Los restaurantes son: " + soln["Y"]
```



✚ **Lista de platillos de un restaurante específico que tengan un ingrediente en particular:** se ingresa la siguiente regla nueva a la base de conocimientos **ingrediente\_particular(X,Y,A):-**  
**restaurante(X,\_,\_,\_),nombre\_platillo(X,A,\_,\_),miembro(Y,B),**  
**miembro(X,[X|Cola]) miembro(X,[Cabeza|Cola]):-miembro(X,Cola),**  
estas últimas dos funcionan para ver si el ingrediente ingresado por el usuario pertenece a la lista donde están almacenados los ingredientes de cada platillo. **La búsqueda se realiza de la siguiente manera:**

```
for soln in prolog.query("ingrediente_particular(X,Y,Z)"):
    print "Los platillos con dicho ingrediente son: " + soln["Z"]
```

#### ❖ **Front-End:**

##### ➤ **Diseño de la página web:**

Este corresponde al front end de la aplicación, es decir, la parte que permite que el usuario interaccione con la misma.

El diseño se realiza a partir del lenguaje HTML, este será el encargado de proporcionar un diseño que se adapte al entorno de cada dispositivo móvil en que sea usada. Si bien es cierto este se encuentra conectado con el framework de “Foodie” y además utiliza una librería llamada jinja2 que proporciona la inclusión de código en el framework.

Para diseñar una página web HTML usa la siguiente estructura:

```

< 1 >
<HEAD> < 2 >Mi primera página web< 3 TITLE> < 4 >
<BODY>
< 5 > <h1> TÍTULO DE MI PÁGINA </h1> </center>
<hr> Este es el cuerpo de mi primera página web
<p> Este es el segundo párrafo del cuerpo
<!--Estas etiquetas no necesitan final 6 >
</ 7 >
< 8 >

```

Cada número representa una etiqueta en las cuales se define por ejemplo, el tipo de letra, color o bien formas de la zona, márgenes y tamaños del silde. Se realiza una redirección interna de cada página con el comando “href” para cada una. Las etiquetas pueden empezar con cualquier letra y deben preceder por un “#”.

En el archivo Conexión\_html.py se encuentran los métodos que permiten la conexión desde Python a los html donde se encuentra el código de la página.

### **Lenguajes de programación usados**

- **Python:** este lenguaje de programación fue utilizado en el Front-End de la aplicación para realizar la interacción con el usuario y por consecuente envía los datos al Back end (Prolog). Python es un lenguaje multiparadigma y con una sintaxis muy sencilla lo que hace que la curva de aprendizaje sea muy pequeña. Además, es un lenguaje multiplataforma y se encuentra tipado dinámicamente.
- **Prolog:** este lenguaje de programación fue utilizado en el Back-End de la aplicación, recibe todos los datos del Front end y los almacena en la base de conocimientos por consecuente cuando se realiza una consulta el

proceso es realizado inversamente. Prolog hace uso del paradigma lógico por lo que su sintaxis es muy particular.

- **HTML:** este lenguaje es especial para la elaboración de páginas web, utilizado en el Front-End de la aplicación, define una estructura básica de referencia ya que cada objeto creado debe contenerla. Gracias a este lenguaje de programación es posible crear un diseño que se adapte a los distintos entornos de los dispositivos. Estructura base:

```
<nombre-de-elemento atributo="valor">Contenido</nombre-de-  
elemento>
```

### **Tecnologías usadas**

- **PySWIP:** es una licencia GPL de Python la cual permite establecer un puente entre este y Prolog. Es necesario que se importe SWI-Prolog como biblioteca compartida así mismo Ctypes para que se establezca la conexión.
- **Ctypes:** es un módulo que en UNIX permite el acceso a las distintas librerías instaladas y las permite que sean manipuladas a lo largo del programa, en este caso cumplen la función de acceder a las librerías de Prolog.
- **CherryPy:** es un entorno HTTP orientado a objetos el cual permite crear aplicaciones web con el uso casi nulo de protocolos y haciendo más sencillo el proceso de compilación. Posee un servidor web que sirve en cualquier ambiente en donde se utilice Python.
- **Jinja2:** es un motor especial para el manejo de plantillas en Python, crea un entorno de ejecución integrado para la conexión de las páginas HTML y el

front end de la aplicación web. Está basado en el sistema de plantillas de Django y es capaz de heredar plantillas. Se basa internamente en Unicode y es prácticamente compatible con las versiones de Python 2.4 a posteriores.

## **Problemas encontrados**

### ***Instalación de Pyswip:***

En internet se encuentran tutoriales o bien información incompleta sobre el tema. La instalación de Pyswip requería la instalación de libpl y Ctypes, para lo cual no venía información lo suficientemente explicada como para haberlo instalado de una manera más rápido.

### ***Curva de aprendizaje con respecto al uso del framework:***

Primeramente se decidió utilizar Django como framework ya que venía información suficiente sobre cómo hacer aplicaciones web en Python por medio de este, sin embargo a lo largo del desarrollo de la misma se dificultó el manejo de este debido a que utilizaba muchos archivos y configuraciones, con lo cual se decidió cambiar de framework con el propósito de encontrar uno más fácil de manejar. Para ello investigamos sobre el framework Cherrypy, el cual lo pudimos adaptar más fácilmente y el entendimiento sobre cómo funcionaba fue más mejor que Django.

### ***Conexión de Cherrypy con Prolog***

No se pudo conectar la parte de consultas con la página. Se generó un error el cual no pudimos corregir debido a que el tiempo que teníamos para hacerlo fue muy limitado ya que teníamos un examen un día antes de la entrega del proyecto y el mismo día también, para lo cual no podíamos cambiar de framework, que hubiera sido la opción más oportuna.

Se mostrará la parte de las consultas y la página por separado.

## Manual de usuario

### *Instalación de Python en Ubuntu*

Es importante indicar que la versión de Python elegida es 2.7.

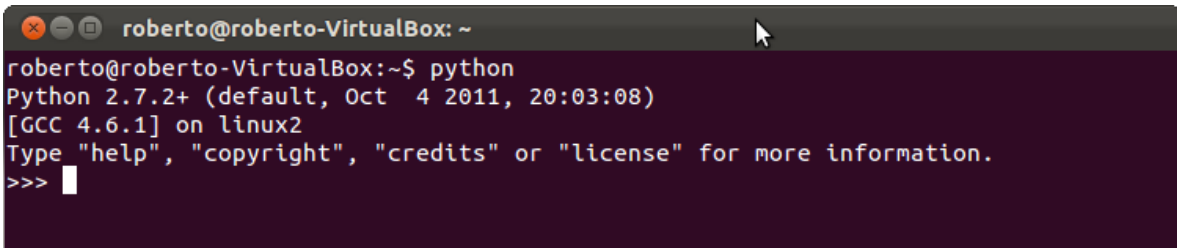
1. Ingrese a la terminal de Ubuntu.
2. Ingrese el siguiente comando:

*>> sudo apt-get install python*

3. Compruebe que el paquete ha sido instalado ingresando la siguiente línea de comandos.

*>> python*

- En caso de que la operación se ha realizado con éxito aparecerá la siguiente información. La cual indica que se tiene instalado Python 2.7.2+



```
roberto@roberto-VirtualBox: ~  
roberto@roberto-VirtualBox:~$ python  
Python 2.7.2+ (default, Oct  4 2011, 20:03:08)  
[GCC 4.6.1] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> 
```

## *Instalación de PySWIP en Ubuntu*

1. Ingrese al siguiente enlace:

<https://code.google.com/p/pyswip/wiki/INSTALL>

2. Ingrese a la terminal de Ubuntu.
3. Ingrese los siguientes comandos para extraer el archivo.

```
$ wget http://gollem.science.uva.nl/cgi-bin/nph-download/SWI-Prolog/pl-5.6.34.tar.gz
$ tar xzvf pl-5.6.34.tar.gz
$ cd pl-5.6.34
```

4. Ingrese el siguiente comando, este configurara el entorno de Python.

```
$ ./configure --prefix=/usr --enable-shared
```

5. Compile el código.

```
$ make
```

6. Instale el código con el siguiente comando.

```
$ sudo make install
```

7. En caso de que ocurra algún problema en la instalación ingrese la siguiente línea de comandos.

```
$ cd packages/clpqr
$ ./configure --prefix=/usr --enable-shared
$ make && sudo make install
```

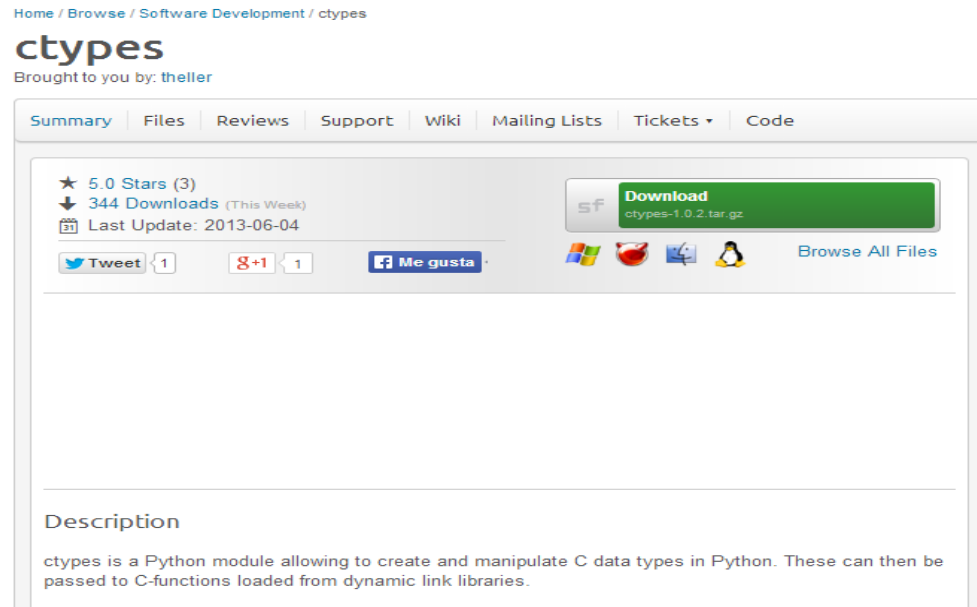
8. Ingrese el siguiente comando para descargar una librería que no se encuentra en el paquete.

```
$ sudo ln -s /usr/lib/pl-5.6.34/lib/i686-linux/libpl.so.5.6.34
/usr/lib/libpl.so
```

Descargue del siguiente link Ctypes:

<http://sourceforge.net/projects/ctypes/?source=dlp>

8.1 Al ingresar presione download.



9. Ingrese los siguientes comandos para desempaquetar Ctypes:

```
$ tar xzvf pyswip-X.X.X
$ sudo python setup.py install
```

10. Ingrese Python en la terminal y pruebe la instalación con las siguientes líneas de código:

```
>>> from pyswip import Prolog
>>> prolog = Prolog()
>>> prolog.assertz("father(michael,john)")
```

El programa procederá a realizar la inferencia.



### **Instalación de CherryPy en Ubuntu**

1. Ingrese a la terminal de Ubuntu.
2. Ingrese la siguiente línea de comandos.

```
$ apt-get install python-cherrypy3
```

3. Cree un archivo llamado "prueba .py" copie las siguientes líneas y guárdelo.

```
import cherrypy

class HelloWorld:
    def index(self):
        return "Hola Mundo!"
    index.exposed = True

cherrypy.quickstart(HelloWorld())
```

4. Ingrese Python desde la terminal y ejecute el archivo de prueba para comprobar que se ha instalado correctamente.

```
Python prueba.py
```

5. Ingrese a la dirección indicada desde el navegador web.

### **Instalación de Jinja2 en Ubuntu**

1. Ingrese a la terminal de Ubuntu.
2. Ingrese la siguiente línea de comandos.

```
easy_install Jinja2
```

3. Compruebe que el paquete ha sido instalado ingresando la siguiente línea de comandos.

```
>> python
```

```
>>import Jinja2
```

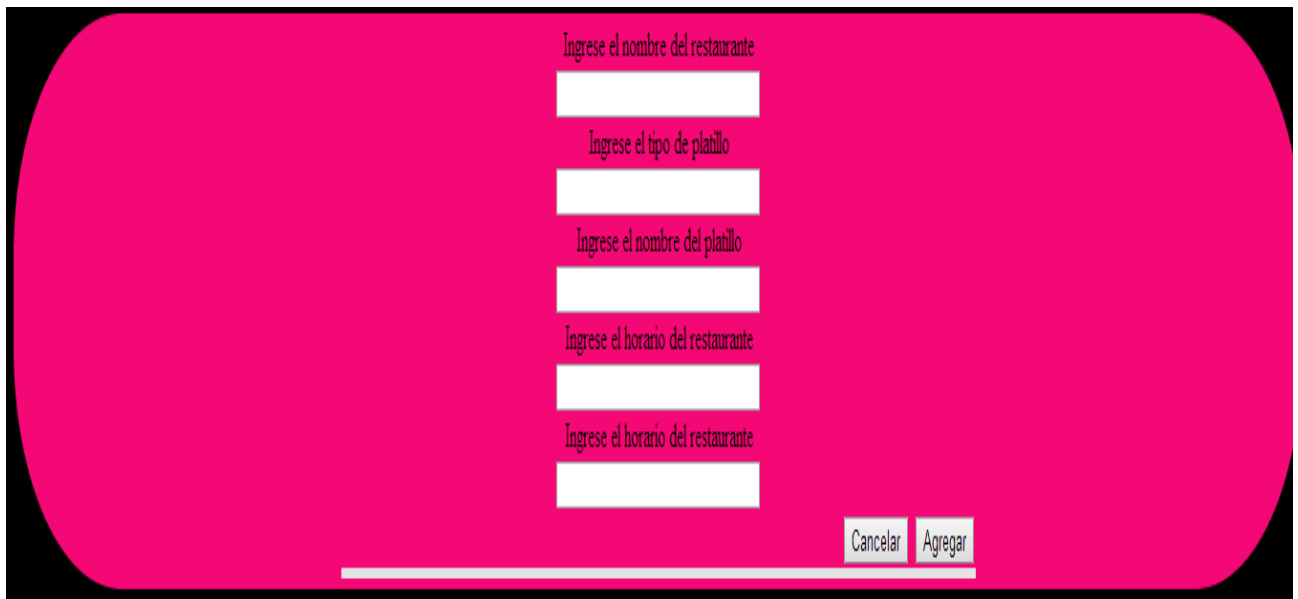
- En caso de que la operación se ha realizado con éxito no aparecerá el error Tradeback.

🚦 Diseño de la página web:

### ***Página de inicio***



### **Agregar restaurante**



Ingrese el nombre del restaurante

Ingrese el tipo de platillo

Ingrese el nombre del platillo

Ingrese el horario del restaurante

Ingrese el horario del restaurante

Cancelar Agregar

### **Agregar platillo**



## Agregar Platillo

Ingrese el nombre del platillo

Ingrese el país de origen del platillo

Ingrese los ingredientes del platillo

Ingrese el tipo de platillo

Cancelar Agregar

## Consultas

### Consulta por lista de restaurantes

Seleccione el restaurante

Cancelar Aceptar

### Consulta de restaurantes por tipo de comida

Ingrese el tipo de comida

Cancelar Aceptar

### Consulta de restaurantes por pais

Ingrese el pais de origen

Cancelar Aceptar

### Consulta de restaurantes por nombre

Ingrese el nombre del restaurante

Cancelar Aceptar

### Consulta de platillos por restaurante especifico

Seleccione el restaurante

Cancelar Aceptar

### Consulta de platillos por ingrediente especifico

Ingrese el ingrediente del platillo

Cancelar Aceptar

### ***Funcionamiento desde la terminal***

```
*****FOODIE*****

1-Agregar restaurante
2-Agregar platillo
3-Consultar la ista de restaurantes
4-Consultar la lista de restaurante por tipo de comida
5-Realizar busqueda de restaurantes por nombre
6-Consultar restaurantes que tienen platillos de un pais especifico
7-Consultar los platillos de un restaurante especifico
8-Consultar los platillos de un restaurante especifico con un ingrediente en particular

Que desea realizar: █
```

***Este corresponde al menú***

```
Que desea realizar: 1

Datos del restaurante

Ingresa el nombre del restaurante:mac
Ingresa el tipo de comida del restaurante:chatarra
Ingresa la ubicacion del restaurante:limon
Ingresa el telefono del restaurante:2789-12-34
Ingresa el horario del restaurante:lunes-domingo

Los datos del restaurante se han ingresado exitosamente
```

***Si elige la primera opción entonces se le solicitarán los datos del restaurante que desea agregar***

```
Que desea realizar: 2

Datos del platillo

Ingrese el nombre del restaurante:mac
Ingrese el nombre del platillo:hamburguesa
Ingrese el sabor del platillo (picante /salado /dulce /agridulce /amargo):salado
Ingrese el pais de origen del platillo:francia
Ingrese los ingredientes del platillo:carne,lechuga,tomate,cebolla
Los datos del platillo se han ingresado exitosamente
```

***Si elige la segunda opción entonces se le solicitarán los datos del platillo que desea agregar***

```
Desea realizar algo mas:

1-Agregar restaurante
2-Agregar platillo
3-Consultar la ista de restaurantes
4-Consultar la lista de restaurante por tipo de comida
5-Realizar busqueda de restaurantes por nombre
6-Consultar restaurantes que tienen platillos de un pais especifico
7-Consultar los platillos de un restaurante especifico
8-Consultar los platillos de un restaurante especifico con un ingrediente en particular
3

Los restaurantes son:

['mac', 'taco', 'pizza']
```

***Si elige la tercera opción entonces se le mostrarán todos los restaurantes registrados***

```

1-Agregar restaurante
2-Agregar platillo
3-Consultar la lista de restaurantes
4-Consultar la lista de restaurante por tipo de comida
5-Realizar busqueda de restaurantes por nombre
6-Consultar restaurantes que tienen platillos de un pais especifico
7-Consultar los platillos de un restaurante especifico
8-Consultar los platillos de un restaurante especifico con un ingrediente en particular

Que desea realizar: 4

Ingrese el tipo de comida para realizar la consulta:
chatarra

Los restaurantes con ese tipo de comida son:
['mac', 'taco']

```

***Si elige la cuarta opción se le solicitará el tipo de comida que desea buscar y posterior a eso se le mostrará los restaurantes con dicho tipo de comida***

```

1-Agregar restaurante
2-Agregar platillo
3-Consultar la lista de restaurantes
4-Consultar la lista de restaurante por tipo de comida
5-Realizar busqueda de restaurantes por nombre
6-Consultar restaurantes que tienen platillos de un pais especifico
7-Consultar los platillos de un restaurante especifico
8-Consultar los platillos de un restaurante especifico con un ingrediente en particular
5

Ingrese el nombre del resturante para realizar la consulta:mac

El restaurante si se encuentra

```

***La quinta opción permite averiguar si el restaurante que el usuario busca se encuentra registrado***

```

1-Agregar restaurante
2-Agregar platillo
3-Consultar la ista de restaurantes
4-Consultar la lista de restaurante por tipo de comida
5-Realizar busqueda de restaurantes por nombre
6-Consultar restaurantes que tienen platillos de un pais especifico
7-Consultar los platillos de un restaurante especifico
8-Consultar los platillos de un restaurante especifico con un ingrediente en par
ticular

Que desea realizar: 6

Ingrese el pais para realizar la consulta:francia

Los restaurantes con platillos de ese pais son:
['mac']

```

***Co la opción seis se le solicita al usuario el país para después mostrarle los restaurantes con platillos de dicho país***

```

Desea realizar algo mas:

1-Agregar restaurante
2-Agregar platillo
3-Consultar la ista de restaurantes
4-Consultar la lista de restaurante por tipo de comida
5-Realizar busqueda de restaurantes por nombre
6-Consultar restaurantes que tienen platillos de un pais especifico
7-Consultar los platillos de un restaurante especifico
8-Consultar los platillos de un restaurante especifico con un ingrediente en par
ticular
7

Ingrese el nombre del resturante para realizar la consulta:mac
Los platillos son:

['hamburguesa']

```

***La opción siete le solicitará al usuario el nombre del restaurante y después se le mostrará los platillos de dicho restaurante***



```
Desea realizar algo mas:

1-Agregar restaurante
2-Agregar platillo
3-Consultar la lista de restaurantes
4-Consultar la lista de restaurantes por tipo de comida
5-Realizar busqueda de restaurantes por nombre
6-Consultar restaurantes que tienen platillos de un pais especifico
7-Consultar los platillos de un restaurante especifico
8-Consultar los platillos de un restaurante especifico con un ingrediente en particular
8

Ingrese el restaurante:mac
Ingrese el ingrediente a buscar:lechuga

['hamburguesa']

Desea realizar algo mas:
```

***La última opción le solicita al usuario que ingrese el nombre del restaurante y el ingrediente que anda buscando, para posteriormente mostrarle todos los platillos con dicho ingrediente***