

Instituto Tecnológico de Costa Rica



Proyecto programado número dos

“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	4
Descripción detallada.	5
<i>Lenguajes de programación usados.</i>	6
<i>Tecnologías usadas.</i>	7
Problemas encontrados	8
Manual de usuario	9

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 tomaría los datos y se los enviaría a Prolog, el segundo tomándolos e infiriendo lógicamente con la serie de reglas propuestas.

La aplicación se encuentra compuesta en tres partes:

- Inicio: donde se direcciona a la página principal de la aplicación.
- Agregar: tomara los datos escritos por el usuario y los enviara al back end de la aplicación.
- Consulta: con respecto a los datos guardados se hará la consulta de platillos o bien restaurantes.

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: nombre, ingredientes, sabor y país de origen.

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

Implementación de clases: trabajando directamente con el front end es necesario implementar una serie de clases en las que se dividirá el trabajo. *Class Home*, contiene la configuración principal de la aplicación web, como lo son la ruta principal, la carga de los archivos de interfaz y las direcciones de las siguientes pestañas además contiene la llamada de los métodos agregar y consultas estás haciendo referencia a una redirección a ellas mismas.

Class Gallerys: en esta clase se encuentra el formulario para realizar el registro de restaurantes, mediante la función *Restaurante*. Se crean una serie de *TextFields* para el ingreso de: nombre, tipo de comida, ubicación, teléfono y horario. Se utiliza el comando *GET* para que el programa obtenga cada dato. Es necesario definir para cada objeto otra función con el nombre de cada uno, esta recibirá el parámetro enviado por la función principal *Restaurante* y posteriormente lo escribirá en un archivo para mantener la persistencia de los datos en la aplicación.

Lenguajes de programación usados.

Python: este lenguaje de programación es utilizado en el Front end de la aplicación para realizar la interacción con el usuario y por consecuente envía los datos al Back end. 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 es 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 deja ser 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 ya que no solo con la descarga de algunas librerías o bien la ejecución de comandos espaciales, se encontró un tutorial un poco más completo que indicaba la instalación de Ctypes en Python.

Curva de aprendizaje con respecto al uso del framework:

Posterior a la investigación de los diferentes frameworks para el diseño de la aplicación móvil se ha elegido como primera opción ya que este se presentaba como el más abierto además en internet existen infinidad de documentación sin embargo este no resulto como se esperaba ya que la base de conocimientos posiblemente afectaría el desempeño del Back end, la curva de aprendizaje es un poco elevada ya que a diferencia de otros frameworks este maneja muchos archivos para la personalización del mismo. En última instancia después de distintas pruebas y errores se ha decidido usar CherryPy, volviéndose más sencillo para las personas que tienen conocimiento de orientación a objetos con una curva de aprendizaje más baja.

Manual de usuario

Instalación de Python en Ubuntu.

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

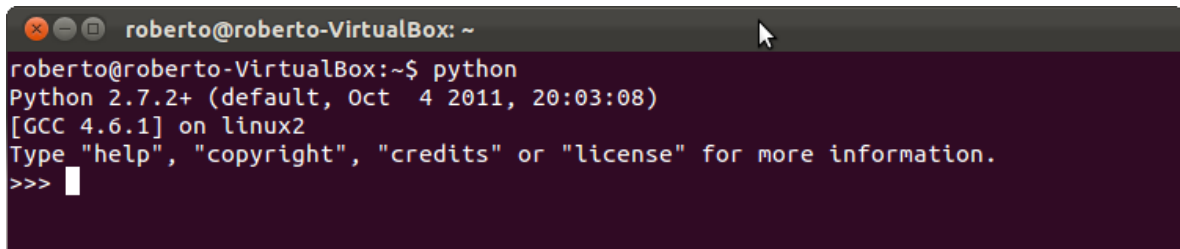
1. Ingrese a la terminal de Ubuntu.
2. Ingrese el siguiente comando:

>> 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.



```
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
```

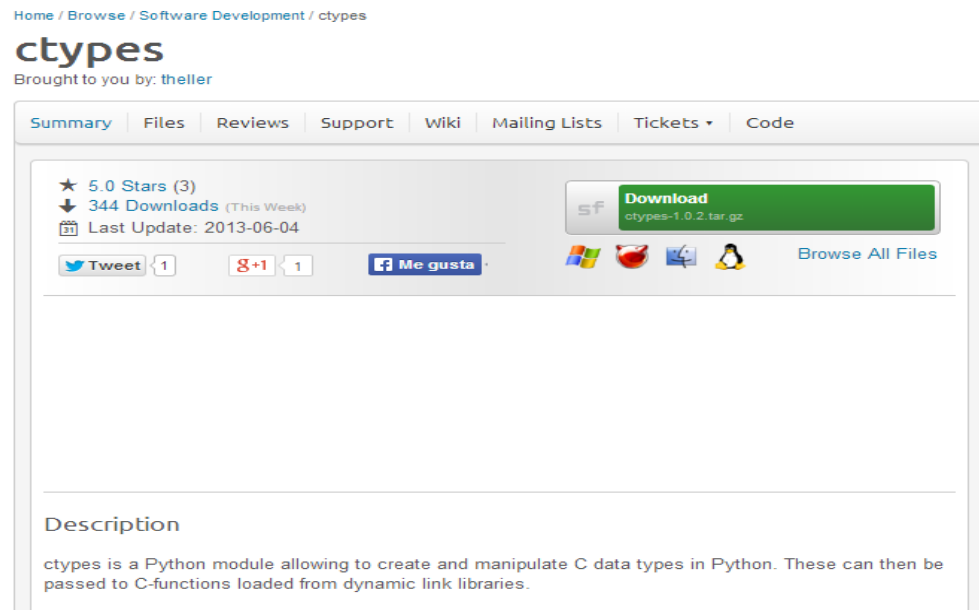
8. Ingrese el siguiente comando para descargar una librería que no se encuentra en el paquete.

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

9. Descargue del siguiente link Ctypes:

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

9.1 Al ingresar presione download.



10. Ingrese los siguientes comandos para desempaquetar Ctypes:

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

11. 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.