



Ingeniería en Computación
Sede de Cartago
Lenguajes de Programación
Prof. Andrei Fuentes Leiva

Segundo proyecto

Kevin Sem Piedra Matamoras
2015069024

7 de Junio, 2019

Tabla de contenidos:

Tabla de contenidos:	2
Introducción:	3
Estructura del sistema	4
Diagrama de instalación	4
Software utilizado	4
Repositorio en Github	5
Estructura del repositorio:	5
Instalación del sistema	6
API	6
App	6
Diseño del API	7
GET:	7
POST:	7
Uso de la aplicación	8
Register y Log in	8
Filtros	9
Detalles de una receta	10
Registro de recetas nuevas	11
Log out	12
Referencias	13

Introducción:

El objetivo de este proyecto (PL02) es la creación de un sistema compuesto por una app de Android y un servicio web, que permite el manejo de recetas de cocina.

El servicio web es un API escrito en Python que hace uso de una Base de Conocimientos, escrita y manejada con Prolog, y una Base de Datos, escrita y manejada con PostgreSQL. La Base de Conocimientos se usa para manejar la información de las recetas, y la Base de Datos para el manejo de los usuarios y sus sesiones. Además, el servicio web utiliza tokens de sesión (session tokens) para controlar quién tiene derecho a usar el API.

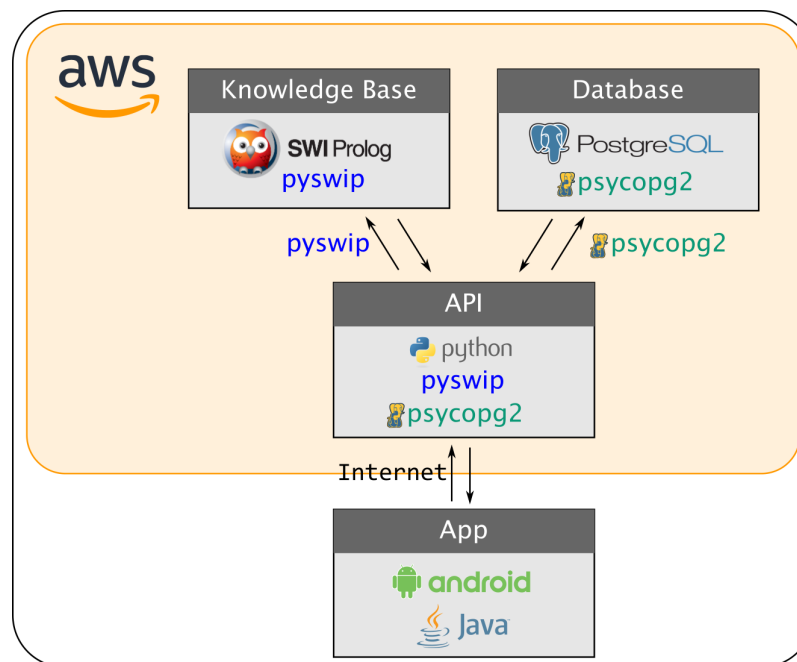
La app de Android está escrita en Java, y obtiene los datos que ocupa únicamente por medio del API.

Estructura del sistema

Diagrama de instalación

El sistema está implementado haciendo uso de Amazon Web Services (AWS), principalmente el servicio EC2 (Elastic Compute Cloud) para el web API. El app se conecta por medio del internet al API y accede a los datos que provee el servicio.

El API se encuentra en la dirección: <http://pl02api.sempiedram.com:35000>



Software utilizado

Para la implementación de la Base de Datos se utilizó PostgreSQL 11, con el IDE DataGrip 2019 de JetBrains, y también pgAdmin 4. PostgreSQL se usó en Linux (Ubuntu) y en Windows 10. Para acceder a la Base de Datos desde el API, se utilizó la librería psycopg2.

Para la implementación de la Base de Conocimientos se utilizó la librería Pyswip que permite la comunicación con SWI-Prolog, la implementación de Prolog que se usó.

Para el desarrollo del API se utilizó PyCharm 2019 de JetBrains, junto a las librerías psycopg2, pyswip, bcrypt, y otras librerías estándar de Python 3.

Para la creación de la app se utilizó el IDE Android Studio.

Repositorio en Github

El proyecto se realizó haciendo uso del sistema de control de versiones Git, y con Github como proveedor del repositorio en línea.

La dirección del repositorio de este proyecto es: <https://github.com/sempiedram/PL02>

Estructura del repositorio

- Carpeta **PL02API**: Contiene el código fuente del API.
 - **api_main.py**: El servidor del API escrito en Python.
 - **prolog_api.pl**: Las reglas de Prolog iniciales que usa el API.
- Archivo **api_database_script.sql**: Script para la creación de la Base de Datos, con PostgreSQL.
- Carpeta **PL02App**: El proyecto completo de Android Studio del app de recetas.
- Archivo **installation_diagram.png**: Diagrama de instalación del sistema.

Instalación del sistema

API

Para iniciar el API se debe seguir los siguientes pasos:

1. Instalar y crear la Base de Datos:
 - a. Instalar PostgreSQL 11.
 - b. Ejecutar el script de la base de datos con el usuario "postgres":
api_database_script.sql.
 - c. Asignarle la contraseña "12345678" al usuario "postgres".
2. Instalar la Base de Conocimientos:
 - a. Instalar SWI-Prolog.
 - b. Instalar pyswip.
3. Colocar el archivo prolog_api.pl en la misma carpeta que api_main.py.
4. Ejecutar el servidor (**api_main.py**).

Para que el API sea accesible desde fuera del localhost, se debe abrir el puerto 35000 para los protocolos TCP y UDP.

App

Para usar el app se debe:

1. Abrir y compilar el proyecto de Android Studio de la carpeta PL02App.
2. Instalar el app en un dispositivo y ejecutar.

Diseño del API

El API solo provee métodos GET y POST para toda la funcionalidad. Las respuestas a la mayoría de los requests son en JSON, y describen si hubo error o no mediante el campo "outcome". Por ejemplo: {"outcome": "error", "error": "invalid_recipe_id", "error_msg": "Invalid recipe id."}

El API provee los siguientes endpoints:

GET:

Todos los métodos GET utilizan el HTTP header "Authentication", de donde obtienen el session_token. Si el session_token no se provee, o no es válido, entonces se da un error.

- **/recipes/all?filter=<filter>**: Retorna en un JSON los IDs de las recetas disponibles que calzan con el filtro. Ejemplo: {"recipes_ids": ["cereal", "fries"]}
- **/recipes/get?recipe_id=<recipe_id>**: Retorna en un JSON la información de una receta dado su ID. Ejemplo: {"recipe": {"id": "cereal", "type": "quick", "ingredients": ["milk", "cereal"], "steps": [[1, "Add cereal to the bowl."], [2, "Pour milk on the bowl."], [3, "Enjoy."]], "photos": ["xxx", "yyy", "zzz"]}}
- **/photos/get?photo_id=<photo_id>**: Retorna la imagen especificada.

POST:

- **/users/login**: Recibe un JSON y utiliza los campos "username", y "password" (que es un hash del password real). Retorna un session_token si los datos eran correctos.
- **/users/register**: Recibe un JSON con los campos "username", "email", "name", "password". Retorna información en un JSON de si sí se registro el usuario.
- **/users/logout**: Utiliza el session_token dado mediante el HTTP header "Authentication" para invalidar ese token.
- **/photos/new**: Recibe una imagen y retorna un JSON con el ID ("photo_id") asignado a esa imagen. Ejemplo: {"photo_id": "xxx"}
- **/recipes/new**: Registra la receta dada. Recibe la información en un JSON con los campos "id", "type", "ingredients", "steps", "photos".
 - "ingredients" es una lista de ingredientes de la receta, por ejemplo: ["milk", "oranges"].
 - "steps" es una lista de pasos de la receta con la forma [posición, paso]. Ejemplo: ["2", "este es el segundo paso"].
 - "photos" es una lista de ids de imágenes que fueron subidas al servidor antes. Ejemplo: ["xxx", "yyy", "zzz"].

Uso de la aplicación

Register y Log in

Al iniciar la aplicación se tiene la opción de ingresar con una cuenta existente, o de registrar una cuenta nueva. Para lograr esto, se ingresan los datos requeridos, y se presiona el botón correspondiente:

The image displays two distinct user interface forms for application authentication. The form on the left is for logging in, featuring a teal header with 'LOG IN' and 'REGISTER' tabs, where 'LOG IN' is selected. It contains input fields for 'username' and 'password', and a 'LOG IN' button at the bottom. The form on the right is for registration, also with a teal header, but with 'REGISTER' selected. It includes input fields for 'username', 'email', 'name', 'password', and 'confirm password', along with a 'REGISTER' button at the bottom. Both forms have a light gray background and a thin red underline under the active tab in the header.

LOG INREGISTER

username

password

LOG IN

LOG INREGISTER

username

email

name

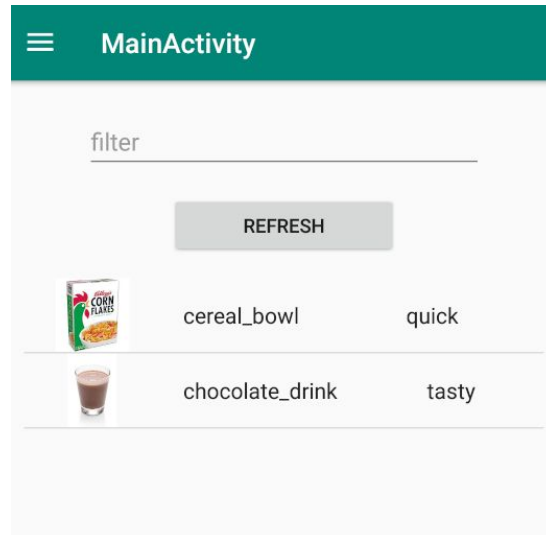
password

confirm password

REGISTER

Filtros

Luego de registrarse e ingresar se llega a la lista de recetas, donde se pueden ver todas las recetas, presionando “Refresh”, o buscar usando un filtro:



Los filtros permiten solo mostrar las recetas que tengan propiedades específicas. Solo se puede aplicar un filtro a la vez y tienen el formato “<tipo filtro>=<valor filtro>” (sin los “<>”).

Ejemplo de filtro: “type=quick”, que solo retornaría las recetas que tienen tipo “quick”.

Los filtros fueron implementados con Prolog, con los predicados “recipe_filtered_type”, “recipe_filtered_id”, y “recipe_filtered_ingredient” que están implementados en el archivo “**prolog_api.pl**”.

Los posibles tipos de filtro son:

- ingredient=<ingrediente>: filtra las recetas que no tienen el ingrediente dado.
- type=<tipo>: filtra las recetas que no tienen el tipo dado.
- id=<id>: filtra las recetas que no tienen el id dado.

Detalles de una receta

Las recetas mostradas en la lista pueden ser presionadas para cargar detalles de ellas. Por ejemplo, aquí se muestran los detalles de la receta con id `cereal_bowl`:

cereal_bowl

quick

0: cereal

1: milk

1: Add cereal to the bowl.

2: Pour milk on the bowl.

3: Enjoy.

0	
1	
2	

Para retornar a la lista de recetas, se utiliza el botón de regreso.

Registro de recetas nuevas



Ya estando en la lista de recetas, se puede deslizar la pantalla hacia la izquierda para abrir la pantalla de subir recetas:

The image shows two side-by-side screenshots of a recipe creation interface. The left screenshot displays the main form with fields for 'recipe id', 'recipe type', 'ingredient', 'step', and 'photo'. It includes buttons for 'ADD INGREDIENT', 'REMOVE INGREDIENT', 'ADD STEP', 'REMOVE STEP', 'ADD PHOTO', 'REMOVE PHOTO', and an 'UPLOAD' button at the bottom. The right screenshot shows a detailed view of a recipe titled 'rice_and_beans' with the flavor 'tasty'. It lists ingredients: '0 rice', '1 and', and '2 beans', and a step: '0 mix well'. Buttons for 'ADD INGREDIENT', 'REMOVE INGREDIENT', 'ADD STEP', and 'REMOVE STEP' are visible.

Aquí se pueden añadir los pasos e ingredientes escribiendolos en el campo relevante y presionando “añadir”. También se pueden remover seleccionandolos, y presionando “remover”.

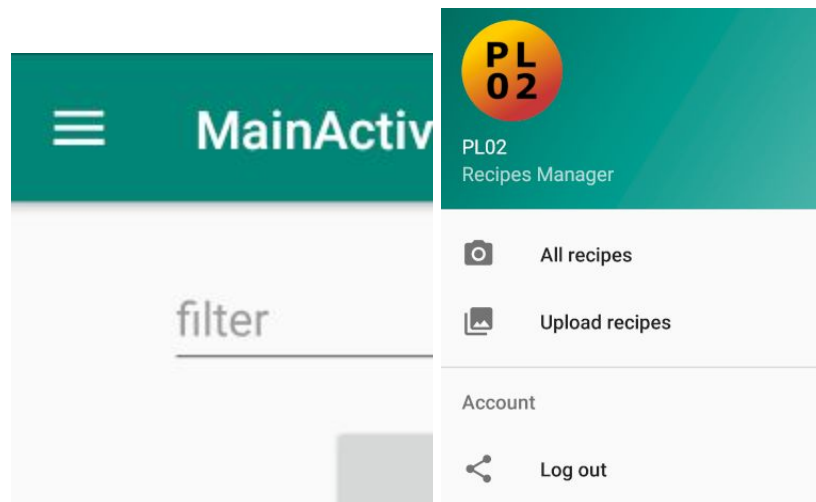
Para añadir una foto, se presiona “añadir foto”, y se abre la Galería de Android, en donde se puede seleccionar qué foto importar. Después de ingresar la información necesaria, y presionar “subir”, debería aparecer un mensaje que dice si se pudo registrar la receta.

Una vez registrada, la receta se puede ver en la lista de recetas, presionando “refrescar” de nuevo:

REFRESH			rice_and_beans
	cereal_bowl	quick	tasty
	chocolate_drink	tasty	0: rice 1: and 2: beans
	rice_and_beans	tasty	0: mix well

Log out

Para cerrar la sesión, se tiene que abrir el menú y luego presionar “Log Out”:



Referencias

- Documentación de SWI-Prolog: <http://www.swi-prolog.org/>
- Documentación de PostgreSQL: <https://www.postgresql.org/docs/11/index.html>
- Documentación de las librerías de Python 3:
<https://docs.python.org/3.7/library/json.html#json.load>
- Librería pyswip: <https://github.com/yuce/pyswip>
- Librería psycpg2: <http://initd.org/psycpg/>
- Stackoverflow: <https://stackoverflow.com/>
- Android reference and documentation: <https://developer.android.com>

Algunas preguntas específicas en StackOverflow que utilicé o me ayudaron a implementar el sistema:

- Generación de MD5 hashes en Java:
<https://stackoverflow.com/questions/415953/how-can-i-generate-an-md5-hash>
- Resizing of bitmaps in Android:
<https://stackoverflow.com/questions/10413659/how-to-resize-image-in-android>
- Opening ports on Ubuntu:
<https://stackoverflow.com/questions/30251889/how-to-open-some-ports-on-ubuntu>
- Restful APIs and user sessions:
<https://stackoverflow.com/questions/6068113/do-sessions-really-violate-restfulness>