



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

Segundo proyecto

**Kevin Sem Piedra Matamoros**  
2015069024

7 de Junio, 2019

# Tabla de contenidos:

<b>Tabla de contenidos:</b>	<b>2</b>
<b>Introducción:</b>	<b>3</b>
<b>Estructura del sistema</b>	<b>4</b>
Diagrama de instalación	4
Software utilizado	4
<b>Repositorio en Github</b>	<b>5</b>
Estructura del repositorio:	<b>5</b>
<b>Instalación del sistema</b>	<b>6</b>
<b>API</b>	<b>6</b>
App	6
<b>Diseño del API</b>	<b>7</b>
GET:	7
POST:	7
<b>Uso de la aplicación</b>	<b>8</b>
<b>Referencias</b>	<b>12</b>

# 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.

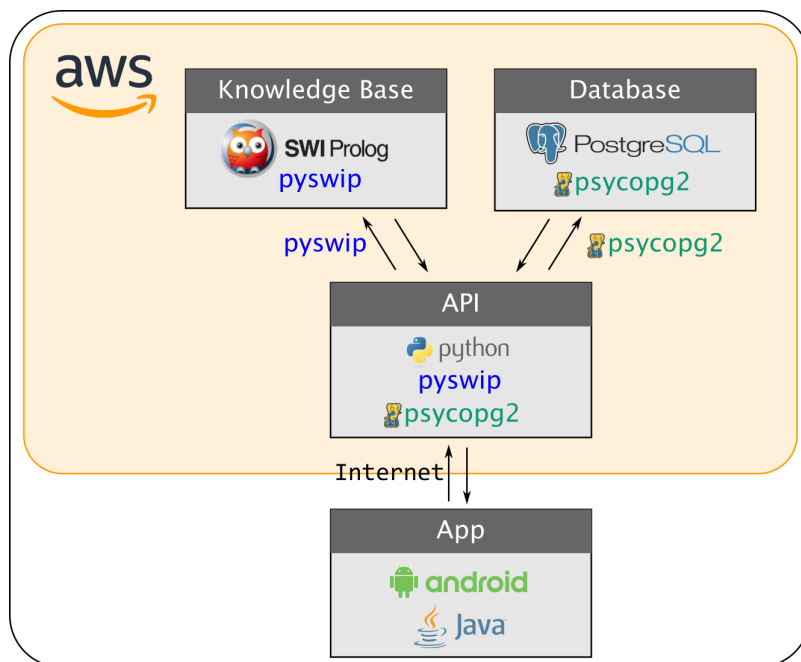
El servicio web es un API escrito en Python que hace uso de una Base de Conocimientos, escrita y manejada con Prolog, y una Base de Datos, escrita y manejada con PostgreSQL. La Base de Conocimientos se usa para manejar la información de las recetas, y la Base de Datos para el manejo de los usuarios y 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.



## 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

Al iniciar la aplicación se tiene la opción de ingresar con una cuenta existente, o de registrar una cuenta nueva:

The image shows two side-by-side screenshots of the application's login and registration screens. The left screen is the login page, featuring a teal header with 'LOG IN' and 'REGISTER' options. Below the header, there are two input fields labeled 'username' and 'password', and a 'LOG IN' button at the bottom. The right screen is the registration page, also with a teal header. It has input fields for 'username', 'email', 'name', 'password', and 'confirm password', and a 'REGISTER' button at the bottom.

Luego de registrarse e ingresar se llega a la lista de recetas, donde se puede buscar usando un filtro:

The image shows a screenshot of the MainActivity screen. At the top is a teal header with a hamburger menu icon and the title 'MainActivity'. Below the header is a 'filter' input field and a 'REFRESH' button. The list contains two items: 'cereal\_bowl' with a corn flakes icon and 'quick' as a tag, and 'chocolate\_drink' with a chocolate drink icon and 'tasty' as a tag.



El filtro es una lista separada por coma de especificadores de filtro. Un especificador de filtro empieza con “id”, “type”, o “ingredient”, seguido por dos puntos, y luego el valor para el especificador de filtro. Ejemplo de filtro: “type:quick, ingredient:milk”. Se remueven todas las recetas que no tengan esas propiedades.

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.





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

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

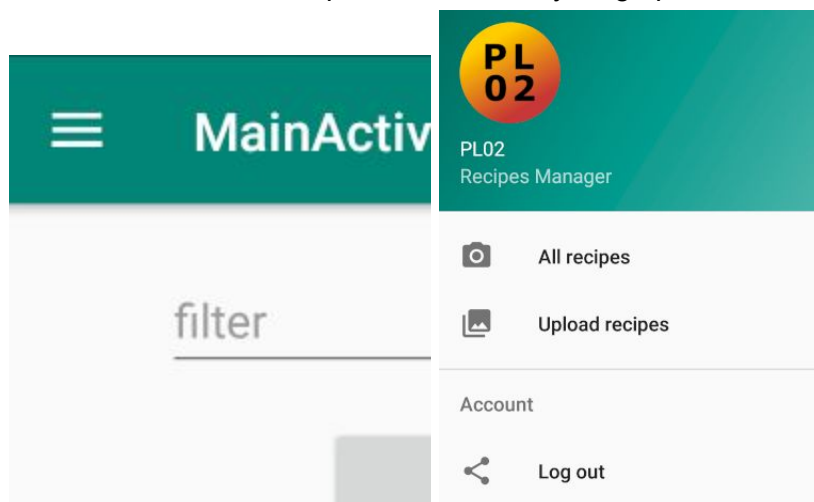
Aquí se pueden añadir los pasos e ingredientes escribiendolos en el campo relevante y presionando “añadir”. También se pueden remover seleccionandolos, y presionando “remover”.

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”:

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

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 psycopg2: <http://initd.org/psycopg/>
- Stackoverflow: <https://stackoverflow.com/>