

# **Archivo de Documentación**

**para**

## **LRecipe**

**Preparado por:**

**Paul Esteban Gutierrez Salas  
Andrés Eduardo Gutiérrez Salas  
Daniel Martínez Bonilla**

**Organización:**

**Instituto Tecnológico de Costa Rica**

**11 de Mayo del 2018**

# Índice

<b>Índice</b>	<b>2</b>
<b>1. Introducción</b>	<b>2</b>
<b>2. Bitácora</b>	<b>3</b>
2.1 Preparaciones Iniciales	3
2.2 Investigación	3
2.2.1 Integración de Prolog con Python	3
2.2.2 Manejo de Archivos con Prolog	3
2.2.3 Framework Flask	4
2.2.4 Heroku	4
2.2.5 Aplicaciones Móviles	4
2.3 Implementación	4
<b>3. Anexos</b>	<b>7</b>
3.1 Modelos	7
<b>4. Referencias</b>	<b>7</b>

## 1. Introducción

El siguiente proyecto trata de una aplicación móvil con el objetivo de manejar una base de datos de recetas de comida, para el curso Lenguajes de Programación impartido en el Instituto Tecnológico de Costa Rica.

- La aplicación móvil está construida en el Lenguaje de Programación Java
- La API está construida en el Lenguaje de Programación Python, se decidió usar el Framework Flask, para este objetivo.
- La base de datos administrativa está construida en SQLite3
- La base de datos de las recetas está construida en una base de conocimientos del Lenguaje de Programación Prolog.

El código fuente del proyecto se encuentra en un repositorio privado dentro de Github a nombre del usuario UnopposedQuill el cual pertenece a Paúl Esteban Gutiérrez Salas encontrable bajo el enlace:

- <https://github.com/UnopposedQuill/II-Proyecto-Lenguajes>

## 2. Bitácora

### 2.1 Preparaciones Iniciales

- 11 de Mayo: 2 horas - Paúl, Andrés, Daniel

Se crea el repositorio y se toma la decisión de trabajar utilizando:

- Python por ser más fácil de instalar y trabajar que Ruby, y por la cantidad de herramientas de desarrollo disponibles.
- Framework Flask por ser ligeramente más sencillo de trabajar comparado con Django.
- IDE PyCharm por su su integración nativa con el Framework “Flask”.
- Android Studio por su relativa “sencillez” comparada con otras herramientas para poder crear aplicaciones móviles.
- SQLite3 por su facilidad de integración con Python, facilidad de trabajo en múltiples sistemas operativos (uno de los integrantes trabaja en Windows).

### 2.2 Investigación

#### 2.2.1 Integración de Prolog con Python

- 11 de Mayo: 4 horas - Paúl

Se investigan varios métodos para unir Prolog junto con Python, entre los cuales figuran:

- PyLog: Descontinuado, no se pudieron encontrar ni binarios ni archivos fuente.
- SwiPy: Descontinuado, no compatible con las versiones actuales de Swi-Prolog ni con las de Python.
- PySwip: Versión alpha (0.2.8), pero funcional.

Se encontró un ejemplo funcional utilizando PySwip en el repositorio:

→ [https://github.com/2rs2ts/pyswip\\_envctrl](https://github.com/2rs2ts/pyswip_envctrl)

Se tomó la decisión de hacer la unión entre Python y Prolog usando PySwip, y se empezó a construir la base de conocimientos para el proyecto

#### 2.2.2 Manejo de Archivos con Prolog

- 15 de Mayo: 1 hora - Paúl

Se investiga acerca del manejo de archivos con Prolog con el objetivo de asegurar una persistencia en los datos. La información usada fue:

<http://www.learnprolognow.org/lpnpag.php?pagetype=html&pageid=lpn-htmlse54>

### 2.2.3 Framework Flask

- 19 de Mayo: 5 horas

Se investigó cómo hacer una API utilizando el framework para Python “Flask”, se descubrió que para poder hacer una API utilizable por una aplicación móvil esta API debe ser RESTful, por lo cual se empezó a investigar cómo hacer una API RESTful utilizando Flask, la documentación fue encontrada en:

<https://flask-restful.readthedocs.io/en/latest/>

Se tomó la decisión de hacer una pequeña prueba utilizando la documentación encontrada en:

<https://flask-restful.readthedocs.io/en/latest/quickstart.html>

### 2.2.4 Heroku

- 4 de Junio: 1 hora

Se investiga como implementar una app mixta de Python y Prolog en heroku.

### 2.2.5 Aplicaciones Móviles

- X de Mayo: 5 horas

## 2.3 Implementación

- 12 de Mayo: 5 Horas

Se empezó a trabajar un archivo de pruebas para manejar Prolog desde Python, se descubrió que hay dos maneras de usar PySwip: Crear un objeto prolog, o bien crear funtores y hechos como si Python fuera Prolog, se tomó la decisión de usar la primera, pues se tuvo problemas al crear reglas con la segunda.

- 14 de Mayo: 4 Horas - Paúl

Se logró realizar una consulta en base a reglas y hechos en Prolog, se tuvo problemas debido a una regla mal hecha, pero logró corregirse, y se logró crear una base de conocimientos funcional, pero no persistente.

- 15 de Mayo: 4 Horas - Paúl

Se empezó a modificar la base de conocimientos y el archivo de prueba de Python para que guardara los datos en un archivo .pl, se tuvieron problemas para averiguar si el archivo existe o no, lo cual requirió ser investigado, y fue encontrado en

[http://www.swi-prolog.org/pldoc/man?predicate=exists\\_file/1](http://www.swi-prolog.org/pldoc/man?predicate=exists_file/1)

[http://www.swi-prolog.org/pldoc/man?predicate=access\\_file/2](http://www.swi-prolog.org/pldoc/man?predicate=access_file/2)

Se tomó la decisión de dejar un archivo permanente con las reglas respecto a las cuales se desea que Prolog siempre trabaje, mientras que los datos quedan en otro archivo.

Se tuvo problemas al crear nuevos hechos en la base de conocimientos debido al formato en el que Prolog guarda los términos, tuvo que leerse la documentación para saber cómo cambiar este formato por defecto, esta documentación se puede encontrar en:

[http://www.swi-prolog.org/pldoc/doc\\_for?object=write\\_term/2](http://www.swi-prolog.org/pldoc/doc_for?object=write_term/2)

- 19 de Mayo: 1 Hora - Paúl  
Se tomó la decisión de hacer que cada entrada fuera individual, en lugar de hacerlo por reuniones como se hacía anteriormente, esto para permitir un mejor conteo de horas.
- 20 de Mayo: 1 Hora - Andrés  
Se crea un nuevo proyecto de Android Studio con un esqueleto para la sección de login necesaria para el proyecto.
- 22 de Mayo: 7 Horas - Paúl - Administrativo  
Se renombra el proyecto de Android Studio, se refactorizan los nombre de los directorios de los archivos fuente para que ambos proyectos (de software separado) no choquen entre sí. Se tuvo que mejorar el .gitignore para reducir la cantidad de archivos que se suben al repositorio que no son necesarios para el proyecto. Todo esto para poder unir los trabajos en la misma versión del proyecto con menos problemas. Se tuvo amplios problemas pues Andrés realizó el proyecto de Android Studio usando una versión experimental de Gradle. Se tuvo que cambiar la estructura del proyecto con el objetivo de poder retroceder la versión del Gradle y poder compilar el proyecto.
- 23 de Mayo: 2 Horas - Paúl  
Se mejora la regla de consultas para abarcar consultas con listas de pasos y listas de imágenes, también se actualiza la versión a utilizar en el gradle. También se agrega una imagen de prueba, de un pan.
- 23 de Mayo: 5 horas Daniel  
Se trabajó en implementar una conexión con entre Flask-RESTful y pyswip, lo cual se complicó dado que pyswip no funciona correctamente con múltiples threads, por lo que se llegó a la solución de usar subprocesos para el manejo de consultas a prolog.
- 24 de Mayo: 2 horas - Daniel  
Se trabajó en implementar los endpoints para agregar, y recuperar recetas, y la lista completa de las recetas.
- 2 de Junio: 2 horas - Daniel  
Agregados los endpoints para las búsquedas de recetas y mejorados las reglas de prolog para la recuperación de información de recetas.

- 4 de Junio: 5 Horas - Andrés

Se abandona el esqueleto anterior de la aplicación móvil con el objetivo de empezar con un esqueleto de un proyecto fallido anterior de Paúl con el objetivo de ahorrar tiempo. Para esto, Andrés busca tutoriales en Youtube. Específicamente:

- <https://www.youtube.com/watch?v=jFYAp42rMEA>
- [https://www.youtube.com/watch?v=0Aoz5F\\_pkkq](https://www.youtube.com/watch?v=0Aoz5F_pkkq)

- 5 de Junio: 10 Horas - Paúl

Se mejora el esqueleto creado por Andrés, se tuvo problemas al tomar fotos con el teléfono debido a que el manejo de archivos cambió desde que se hizo el tutorial que utilizó Andrés, cuya solución fue encontrada en:

- <https://stackoverflow.com/questions/38200282/android-os-fileuriexposedexception-file-storage-emulated-0-test-txt-exposed>

También debido a que hay permisos que deben ser solicitados durante la ejecución de la aplicación (como el uso de la cámara). La solución se encontró en:

- <https://developer.android.com/training/permissions/requesting?hl=es-419>

También debido a que por algún motivo el archivo temporal no se estaba creando en el teléfono causado por falta de permisos de escritura de archivos en el manifiesto:

- <https://stackoverflow.com/questions/20202966/android-not-creating-file/20204954>

- 5 de Junio: 5 horas - Daniel

Se procede a subir la api a Heroku, pero se encuentra un error en la comunicación de prolog, a parte de eso la api funciona.

- 6 de Junio: 5 horas - Daniel

Se logra solucionar el error con la comunicación de prolog, y se procede a probar la api, con lo cual queda completa.

- 6 de Junio: 12 Horas - Paúl

Se mejora la interfaz para ser capaz de manejar los N instrucciones, M ingredientes y O listas de imágenes (de la misma manera que lo maneja la API).

Para colocar las N imágenes se tomó la decisión de utilizar el siguiente tutorial para crear un ImageSlider:

- <https://www.youtube.com/watch?v=Q2FPDI99-as>

Con esto se tuvo problemas debido a que se necesitaba agregar imágenes de manera dinámica, luego de un rato de búsqueda se encontró:

- <https://stackoverflow.com/questions/13664155/dynamically-add-and-remove-view-to-viewpager>

También se tomó la decisión de remover del proyecto los archivos no necesarios del proyecto sobre el cual se empezó a trabajar.

Finalmente se empezó a trabajar en cómo colocar todo en listas dentro de la aplicación de Android Studio, se tomó la decisión de seguir trabajando con los RecyclerViews, cuya documentación se encuentra en:

- <https://developer.android.com/guide/topics/ui/layout/recyclerview>

Se agregaron los Actividades para observar detalladamente una receta dada, para ver todas las recetas así como para crear las recetas, se logró elaborar una lista utilizando RecyclerViews así como sus respectivos adaptadores.

También se cambiaron todos los datos placeholders del registro y login para utilizar la API, se utilizó:

- <https://www.baeldung.com/java-http-request>
- <https://github.com/eugenp/tutorials/tree/master/core-java-modules/core-java-networking>

- 7 de Junio: 6 Horas - Paúl

Se tuvieron problemas debido a que todas las operaciones de networking deben hacerse en un hilo que no sea el principal, se solucionó con:

- <https://developer.android.com/training/articles/perf-anr>

Se tuvo que aprender cómo leer JSON en Java puesto que la API retorna ese tipo de dato para todo, se usó:

- [https://www.tutorialspoint.com/android/android\\_json\\_parser.htm](https://www.tutorialspoint.com/android/android_json_parser.htm)

Se tuvo que aprender a usar SharedPreferences para almacenar por un buen tiempo el token de la aplicación, se usó:

- <https://stackoverflow.com/questions/23024831/android-shared-preferences-example>
- <https://stackoverflow.com/questions/3687315/deleting-shared-preferences>

Se implementó la funcionalidad de agregado de recetas con la API.

Se implementó la funcionalidad de detalle de recetas con la API.

## 4. Referencias

<https://wiki.python.org/moin/IntegratingPythonWithOtherLanguages#Prolog>  
<https://stackoverflow.com/questions/17326621/python-interface-with-swi-prolog>  
[https://github.com/2rs2ts/pyswip\\_envctrl](https://github.com/2rs2ts/pyswip_envctrl)  
<https://pypi.org/project/pyswip/>  
<https://github.com/yuce/pyswip>  
<http://www.learnprolognow.org/lpnpagel.php?pagetype=html&pageid=lpn-htmlse54>  
[http://www.swi-prolog.org/pldoc/man?predicate=exists\\_file/1](http://www.swi-prolog.org/pldoc/man?predicate=exists_file/1)  
[http://www.swi-prolog.org/pldoc/doc\\_for?object=write\\_term/2](http://www.swi-prolog.org/pldoc/doc_for?object=write_term/2)  
<https://developer.android.com/training/permissions/requesting?hl=es-419>  
<https://www.youtube.com/watch?v=jFYAp42rMEA>  
[https://www.youtube.com/watch?v=0Aoz5F\\_pkkq](https://www.youtube.com/watch?v=0Aoz5F_pkkq)  
<https://stackoverflow.com/questions/13664155/dynamically-add-and-remove-view-to-viewpager>  
<https://developer.android.com/guide/topics/ui/layout/recyclerview>  
<https://www.baeldung.com/java-http-request>  
<https://github.com/eugenp/tutorials/tree/master/core-java-modules/core-java-networking>