



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA



DEPARTAMENTO DE
ELECTRONICA

IWG101: INTRODUCCIÓN A LA INGENIERÍA

Primera Entrega Proyecto U Bike - Introducción a la Ingeniería

Estudiantes: Sebastián Saldías - Acza Rifo - María Poblete - Josué Leiva
Profesor: Luis Lizama
Ayudante: Andrés Pino
Fecha: 08 de Mayo, 2024

Contents

1	Resumen	2
2	Introducción	2
3	Desarrollo	2
3.1	Método de Resolución	2
3.2	Cambios en el método	3
3.3	Planificación	3
4	Diagramas	4
4.0.1	Nota	6
5	Conclusiones	7
6	Referencias	7

1 Resumen

En esta entrega del proyecto UBIke —University Bike—, se presentarán diversos aspectos, se detallará la problemática que solucionaremos y por sobre todo se ahondará en el método de solución que estamos utilizando. Junto a esto se presenta un poco de la planificación y las tareas que estamos próximas a completar. Complementamos esta información con algunos diagramas que ilustran los procesos más técnicos.

2 Introducción

En esta primera entrega, se abordarán diversas problemáticas con respecto al proyecto llamado *UBike*. Se tratarán elementos tales como la planificación del proyecto, sus áreas, y el método de solución de la problemática. Esto además de presentar los avances ya existentes, en progreso y por realizar; incluída una calendarización con fechas estimadas.

Recordando un poco, se tiene que el problema a analizar, e idealmente resolver de manera inicial, trata (en pocas palabras) de abordar el dilema de la inseguridad e ineficiencia en cuanto al uso de los ciclistas en las instalaciones de la universidad.

3 Desarrollo

3.1 Método de Resolución

Para resolver este problema, junto al equipo, se pensó en crear un sistema que permita al estudiantado utilizar algún medio de autenticación para así poder ingresar y en su defecto, retirar, la bicicleta que el mismo utilice en la USM.

Abordando la idea, se ideó utilizar alguna tarjeta o llavero con tecnología RFID para la autenticación y algún tipo de módulo que permita leer este objeto, y que pueda enviar esta información a algún servidor para ser procesada. Este dicho módulo, se encontraría en cada ciclistero, permitiendo así que cada uno de estos, fuese independiente de otro, tanto en energía como en libertad de locación.

Luego de discutirlo, se decidió proseguir de la siguiente manera: Utilizar un microcontrolador Raspberry Pi Pico W [1] como dispositivo central del módulo, una tarjeta lectora de RFID [2] modelo RC522 [3] la cual se conectaría a este microcontrolador, un diodo LED de color rojo para efectos de indicar el funcionamiento del módulo. Se escogió utilizar este microcontrolador inicialmente, ya que permite conectividad Wifi y Bluetooth, lo que nos otorga flexibilidad para desarrollar esta solución.

La tarjeta lectora se conectaría a este microcontrolador, al igual que el LED, mediante cables, hacia los puertos GPIO [4] de este. Este microcontrolador permite ser programado oficialmente mediante el soporte de micropython entregado por Raspberry Pi Foundation, o mediante un SDK de C/C++ también soportado por la misma institución [5], para mayor facilidad se optó por el uso de micropython para programar el dispositivo.

Para poder procesar los datos y realizar procesos más exigentes, además de favorecer la escalabilidad, se decidió por establecer un servidor centralizado, en donde cada módulo (por cada ciclistero) se conecte mediante Wifi a este servidor, y pudiesen comunicarse para que así cada módulo pueda enviar el Unique ID (UID) de cada llavero o tarjeta (dato leído por la tarjeta RFID), hacia este servidor.

Con el fin de crear este servidor, se decidió desarrollar una página web mediante el robusto framework Django [6] de desarrollo backend, esto permite crear un backend robusto y rápido, fácilmente desplegable, además de seguro. Al tratarse de una plataforma web, se facilita tanto la comunicación módulo-servidor mediante la creación de APIs, como el acceso del usuario a la solución que se está planteando mediante una interfaz web.

Django es útil como solución backend, pero es necesario también una solución frontend, por simplicidad y agilidad, se utilizará la famosa librería de desarrollo Bootstrap [7] la cual permite el desarrollo responsive de páginas web.

Antes de continuar, se presenta un diagrama esencial el cual tiene como finalidad sintetizar las ideas de este proyecto. No tiene como aspecto principal lograr precisión técnica, sino ilustrar un inicio del proyecto.



Figura. 1: Diagrama esencial del proyecto.

3.2 Cambios en el método

Lo visto en el ítem anterior es un desarrollo de la idea inicial, pero en vista del rápido avance necesario para cumplir con los plazos (más información posteriormente), este desarrollo afectó a algunos cambios. El más notorio de estos, el cambio de microcontrolador.

Anteriormente se propuso utilizar el microcontrolador Raspberry Pi Pico W, pero se encontraron algunas limitaciones en el uso de este, algunas de estas siendo:

- Lentitud de ejecución de código.
- Falta de documentación robusta (micropython y SDK C/C++).
- Poca madurez en el desarrollo con micropython.

Para enfrentar este problema, se decidió cambiar el microcontrolador, esta vez utilizando la famosa placa de la compañía Espressif, en particular se trata de una ESP32 DevKit v1. [8] (o también llamado DevKitC).

Este microcontrolador es altamente utilizado en la industria, es robusto y tiene alta compatibilidad con diversidad de sistemas, librerías, protocolos, etc. También posee conectividad Wifi y Bluetooth, permitiendo al proyecto no desviar su curso en cuanto al desarrollo de software se trata, con una pequeña excepción.

Esta placa, se desarrolla utilizando C++, tal y como Espressif lo indica en su documentación [8], pero también se puede programar utilizando el famoso IDE de Arduino [9], lo cual facilita la tarea.

Cabe destacar que todo el avance, en cuanto a software se trata, se encuentra disponible en el repositorio de GitHub del proyecto UBIke [10].

3.3 Planificación

Con el propósito de facilitar el avance, el proyecto se ha dividido en diversas áreas, en donde cada uno de los estudiantes del grupo trabaja, aprende y se especializa, esta son:

- Desarrollo Frontend (María Poblete, Acza Riffo)
- Modelación 3D (Sebastián Saldías)
- Electrónica (Sebastián Saldías)
- Desarrollo Embebido (Josué Leiva)
- Desarrollo Backend (Josué Leiva)

Como plataforma de organización en cuanto a tareas, avances y fechas se trata, se utiliza Microsoft Planner [11], una aplicación que permite el ordenamiento de los proyectos mediante la metodología Kanban.

Debido a la complejidad de la planificación, y a las limitaciones de la plataforma, en caso de desear obtener acceso a esta, favor enviar un correo solicitando este acceso a la siguiente dirección: jleivap@usm.cl. El equipo se reserva el derecho y libertad de autorizar a quien considere competente, el acceso a la plataforma.

A modo de resumen se pueden mencionar las siguientes tareas por realizar o completar:

- Entrega Nro. 1.
- Definir fuente de poder para prototipo.
- Definir material y diseño para soporte prototipo de bicicletas.
- Crear vistas de usuario (registro y común) (frontend).
- Crear vista para guardias (frontend).
- Implementar apertura de solenoide según Response de API.
- Realizar diagrama de cascada de los puntos críticos del proyecto.

Si bien existen más tareas por mencionar, se recalca nuevamente el uso de la plataforma Ms. Planner.

4 Diagramas

En este proyecto de alta complejidad, teniendo en cuenta de que se trata de un ramo de primer semestre, se destaca la importancia de entender conceptualmente el funcionamiento de la solución a desarrollar. En este punto es útil el uso y creación de diagramas de flujo, es por ello que se facilitan dos diagramas en particular, un primer diagrama de flujo que indica la lógica de funcionamiento del sistema, y un segundo diagrama que muestra el proceso de comunicación entre el módulo bicicletero y la API del servidor, esto mediante el protocolo HTTP y el uso de Wifi; figuras 2 y 3 respectivamente.

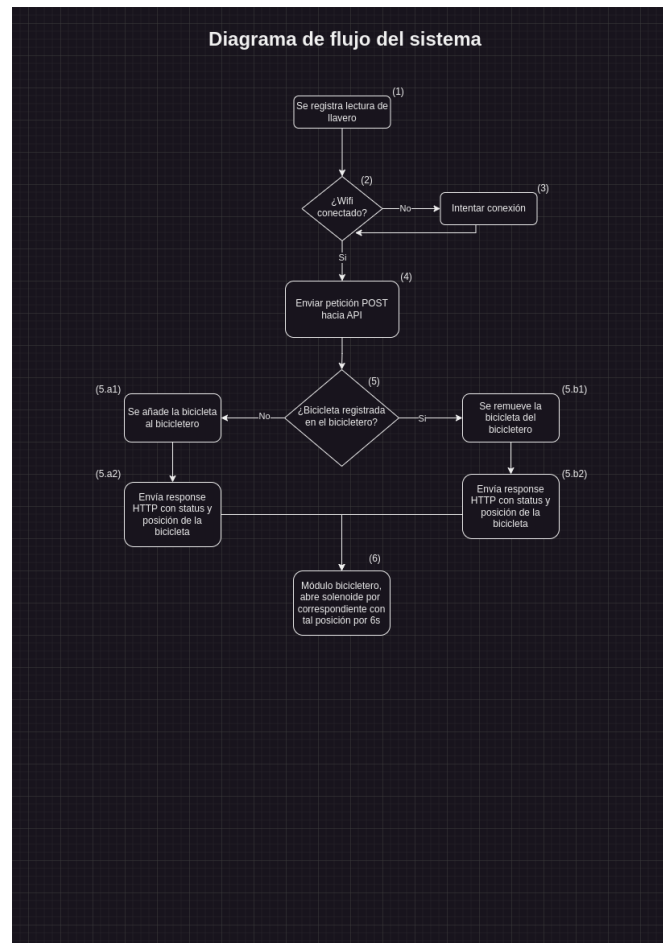


Figura. 2: Diagrama de flujo de lógica del funcionamiento.

Este diagrama de flujo, representa la lógica que nuestra solución plantea, brindando así una clara referencia con respecto a los objetivos que cada pieza de nuestro planteamiento, debe lograr.

El evento principal, el cual gatilla la lógica descrita, es la lectura del llavero del usuario, mediante el lector RFID ya mencionado anteriormente; y termina cuando el módulo finaliza la conexión con el servidor central y actúa con respecto a la respuesta de este.

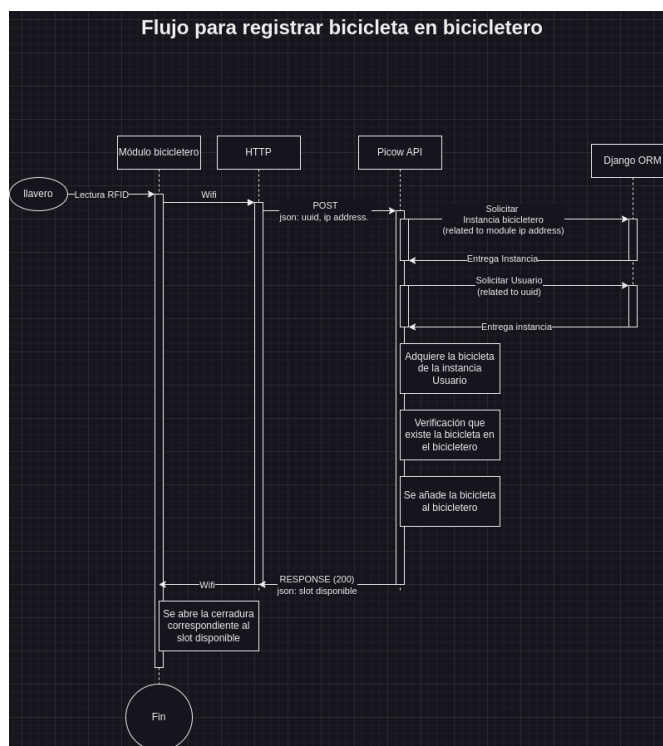


Figura. 3: Flujo de comunicación entre módulo bicicletero y servidor.

Esta última figura (diagrama de cascada), muestra el proceso de comunicación entre el módulo bicicletero y el servidor central, mediante el protocolo HTTP, utilizando WiFi de por medio.

A medida que avanza el proyecto, estas estructuras pueden variar, por lo que estos diagramas no son definitivos. Para esta entrega estos elementos sólo tienen la finalidad de ilustrar al lector cómo se está desarrollando esta solución.

4.0.1 Nota

Cabe destacar que hemos decidido no incluir en esta entrega algunos otros detalles de la solución. Entre estos elementos se incluyen: Método de seguridad primaria, planificación respecto al frontend y la lógica del sistema cuando ocurren excepciones en el funcionamiento de los distintos puntos críticos de la infraestructura digital que se está desarrollando.

Esto se ha decidido así, ya que creemos fundamental sentar bases robustas dentro del proyecto, para así no tener mayores problemas más adelante en el transcurso del desarrollo.

5 Conclusiones

Luego de presentar esta información, concluimos como equipo, en que es imperativo trabajar activamente entre integrantes, compartiendo información con las otras áreas, para así mantener un trabajo uniforme, que logre mantener la visión y objetivo de la solución a implementar.

Además de esto, reconocemos la necesidad de mantener una organización y estructuras claras, creando objetivos tanto a corto como a largo plazo, logrando así los avances necesarios para evitar retrasos.

6 Referencias

- [1] R. P. Foundation, “Raspberry pi pico,” Disponible en <https://www.raspberrypi.com/products/raspberry-pi-pico/>.
- [2] B. Fennani, H. Hamam, and A. O. Dahmane, “Rfid overview,” in *ICM 2011 Proceeding*, 2011, pp. 1–5.
- [3] NXP, “Mfrc522 standard performance mifare and ntag frontend,” Disponible en <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf> (2024/04/4).
- [4] R. P. Foundation, “Physical computing with python,” Disponible en <https://projects.raspberrypi.org/en/projects/physical-computing/1>.
- [5] —, “Software development,” Disponible en (<https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html#software-development>).
- [6] D. S. Foundation, “Django, the web framework for perfectionists with deadlines.” Disponible en <https://www.djangoproject.com/>.
- [7] B. Team, “Build fast, responsive sites with bootstrap,” Disponible en <https://getbootstrap.com/>.
- [8] E. Systems, “Esp32-devkitc.”
- [9] Arduino, “Arduino ide 2.3.2,” Disponible en <https://www.arduino.cc/en/software> (20/02/2024).
- [10] J. Leiva and S. Saldías, “Ubike iwg101 project,” Disponible en <https://github.com/Technopy311/UBike>.
- [11] M. Office, “Microsoft office,” Disponible en <https://tasks.office.com/>.