

Desarrollo de Aplicación Móvil para el Metro de Lima



Realizado por:

Coras Condor Isaias Andres-Espinoza Noa Guiller Caru-Gonzales Ramirez Denis



Índice

A)	Situación Actual	
1.	Descripción del Metro de Lima.....	04
2.	Formulación de la problemática.....	05
3.	Resumen del sistema actual del metro de lima.....	06
B)	Propuesta de innovación	
1.	Detalles del nuevo proceso.....	11
2.	Desarrollo de App en plataforma Python.....	12
3.	Herramientas y tecnologías a usar.....	13
C)	Programación	
1.	Código fuente del programa desarrollado en Python.....	14
D)	Conclusiones y Recomendaciones	
1.	Conclusión 1.....	15
2.	Conclusión 2.....	15
3.	Conclusión 3.....	15
4.	Recomendación 1	15
	Referencia bibliográfica / URLs	16



Situación Actual

En el mundo moderno de hoy, con ciudades altamente urbanizadas, es imperativo contar con un sistema de transporte optimizado. El crecimiento de la población y la demanda de movilidad han llevado a congestiones viales, tiempos de desplazamiento prolongados y una disminución en la calidad de vida. Por ello un sistema de transporte eficiente, accesible y sostenible es esencial para abordar estos desafíos.

Además, un sistema de transporte optimizado garantiza la seguridad vial y proporciona a las personas más tiempo para sus actividades personales, mejorando así la calidad de vida en las ciudades urbanizadas de hoy, lo cual implica una mejora en el desarrollo cultural, social y económico del país.



Descripción del Metro de Lima

El Metro de Lima, también conocido como el Metro de Lima y Callao, es el sistema de transporte público que sirve a la ciudad de Lima y Callao.

La Línea 1 del Metro de Lima tiene una longitud de aproximadamente 34 kilómetros y cuenta con 26 estaciones. Recorre la ciudad de sur a norte, desde el distrito de Villa El Salvador hasta el distrito de San Juan de Lurigancho. Es una línea subterránea en su mayoría, aunque también tiene algunos tramos en superficie y elevados.

El Metro de Lima cuenta con modernos trenes y sistemas de señalización, lo que permite un transporte eficiente y seguro para los pasajeros. Además de la Línea 1, se están construyendo y planificando otras líneas para ampliar el sistema de metro y abarcar más áreas de la ciudad.

Cabe recalcar que el sistema de Metro de Lima ha sido una obra compleja y ha enfrentado diversos desafíos a lo largo de su historia, como problemas de financiamiento, demoras en la construcción y dificultades técnicas. Sin embargo, ha sido un proyecto importante para mejorar el transporte público en Lima y contribuir a la calidad de vida de los habitantes de la ciudad.



Formulación de la problemática

El acceso al metro de Lima en la actualidad involucra un proceso fragmentado y a menudo se traduce en largas colas, lo que implica un consumo adicional de tiempo para los usuarios, un recurso limitado en la vida de cada individuo.

El sistema del metro de lima se sustenta en el uso de tarjetas recargables.

En la práctica sin embargo, puede haber estaciones donde no haya suficientes máquinas de recarga disponibles, lo que puede generar filas largas y tiempos de espera prolongados.

A eso agregarle la existencia de las horas punta que suelen ser entre los días lunes y sábado de 05:00 a 10:00 horas, 12:00 a 14:00 horas y de las 18:00 a 23:00 horas.



Resumen del sistema actual del metro de lima

- 1) Para obtenerlo, se debe adquirir en una tienda dentro del mismo metro o a través del personal encargado. Sin embargo, estos lugares a veces pueden estar ocupados por otros usuarios, lo que resulta en una pérdida de tiempo para obtenerlo.





Resumen del sistema actual del metro de lima

2) Por lo general, al comprar tarjetas para el metro estas vienen recargadas, pero si el usuario necesita más saldo, debe recargarla en máquinas automáticas. Sin embargo, debido a la cantidad de usuarios y las horas pico, la congestión y el tiempo requerido para usar estas máquinas aumenta.

Uso del Tren (o cualquier Metro) con tarjeta

LATAM





Resumen del sistema actual del metro de lima

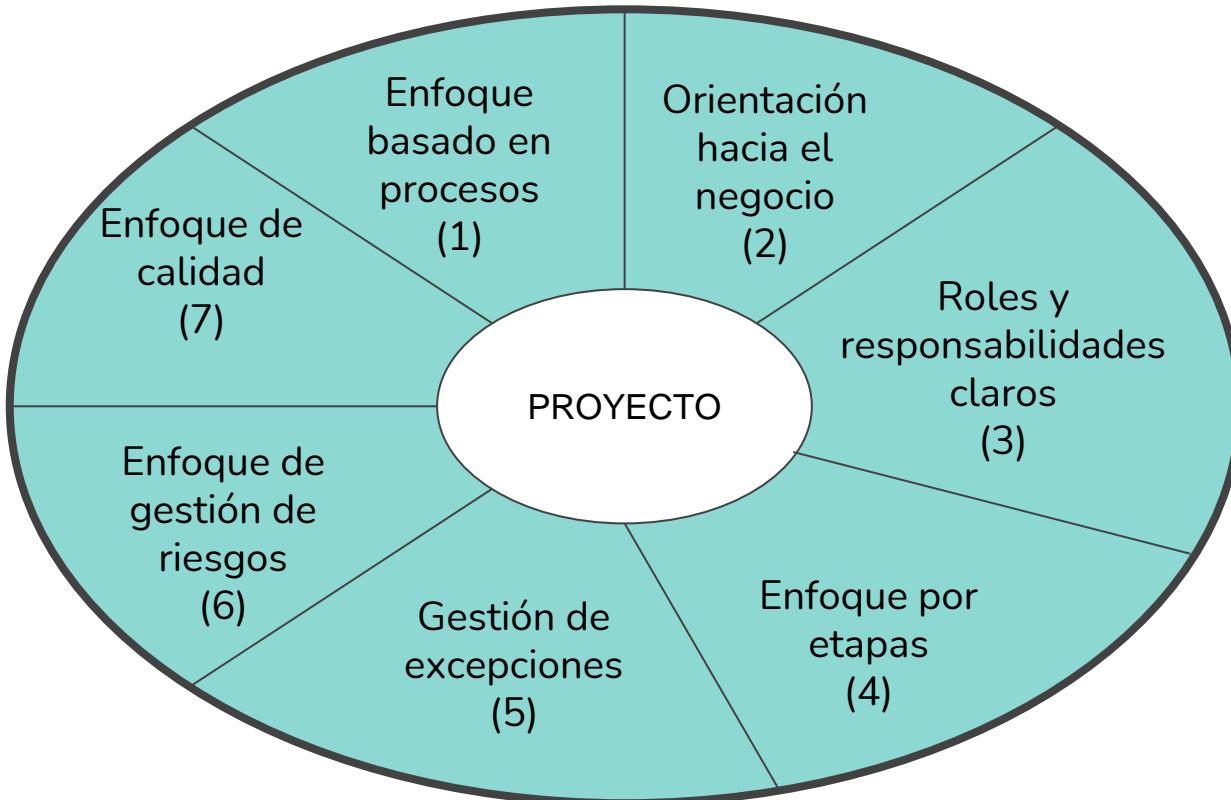
3) Finalmente, solo resta dirigirse al tren, mostrar la tarjeta a una máquina automática y abordar el tren.

Este sistema consta de tres partes que a menudo implican largas colas, consumiendo el tiempo del usuario y causando posibles retrasos en su destino. Esto puede requerir que los usuarios salgan más temprano o, en algunos casos, llegar tarde a su lugar de trabajo o destino deseado.



Diseño metodológico (PRINCE2)

Se realiza la gestión del proyecto en base a 7 principios bien establecidos



(1) Enfoque estructurado de gestión de proyectos que se divide en procesos claros y definidos.

Administración----Programación-----Implementación-----Capacitación

(2) Se asegura de que el proyecto tenga una justificación comercial sólida y se mantenga en línea con las necesidades y prioridades del negocio

La optimización del sistema de cobros aumenta el flujo de pasajeros.

(4) Se divide el proyecto en etapas gestionables. Cada etapa tiene su propio objetivo, alcance y planes específicos.

Propuesta -----> Desarrollo -----> Aplicación -----> Mantenimiento

(5) Se definen límites claros para la toma de decisiones y establece mecanismos para gestionar desviaciones significativas del plan original.

Ejemplo: Planteamos la situación en la que entren en operación las demás líneas del metro

(6) Se proporciona una gestión para identificar, evaluar y/o cuantificar los riesgos en todas las etapas del proyecto.

Ejemplo: Prevención de errores en la base de datos al momento de una operación

(7) Se centra en la definición clara de los criterios de calidad y en la verificación continua de la calidad de, en este caso, el software

Se busca la total aprobación de la autoridad de transporte urbano, así como del MTC



(3)

JUNTA DIRECTIVA DEL PROYECTO:

Project Manager: Coras Condor Isaias Andres

Team Manager: Gonzales Ramirez Denis

Project Support: Espinoza Noa Guiller Caru

Por lo anterior visto; Los beneficios más importantes que ofrece esta metodología son:

- Flexibilidad en la toma de decisiones
- Tener una buena comunicación que implique a la dirección
- Revisión del progreso
- Justificación comercial continua del proyecto durante todo su ciclo de vida



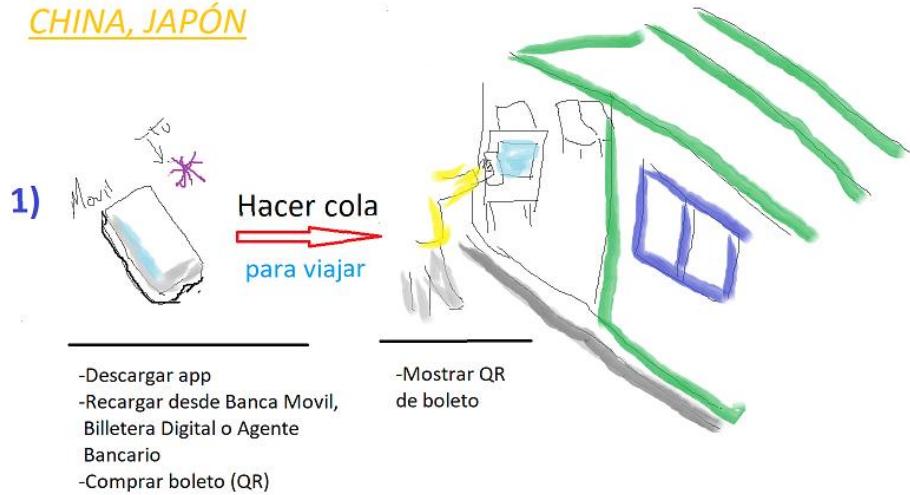
Propuesta de innovación

En el nuevo sistema de obtención de boletos, el proceso se simplificaría mediante una aplicación móvil disponible tanto para Android como para iOS, que cualquier persona podría instalar en su dispositivo celular.

1) Inicialmente, se deberá descargar la aplicación, crear un usuario y seleccionar el monto a recargar, lo cual se puede realizar a través de la Banca móvil, Billetera digital o un agente bancario.

Uso del Tren (o cualquier Metro) con app

CHINA, JAPÓN



1 proceso, 1 vez hacer cola (directo al metro)



Propuesta de innovación

2) Finalmente, al dirigirse al metro, el usuario simplemente mostrará en la pantalla de su celular el código QR recibido, el cual será escaneado por una máquina automática para permitirle abordar el tren.

En este novedoso sistema, el usuario puede realizar los primeros pasos desde cualquier ubicación, incluso mientras realiza otras actividades, y solo necesita acercarse al tren para mostrar su boleto en la pantalla del celular y abordar. Esto resulta en un ahorro significativo de tiempo para el usuario, optimizando así el servicio ofrecido

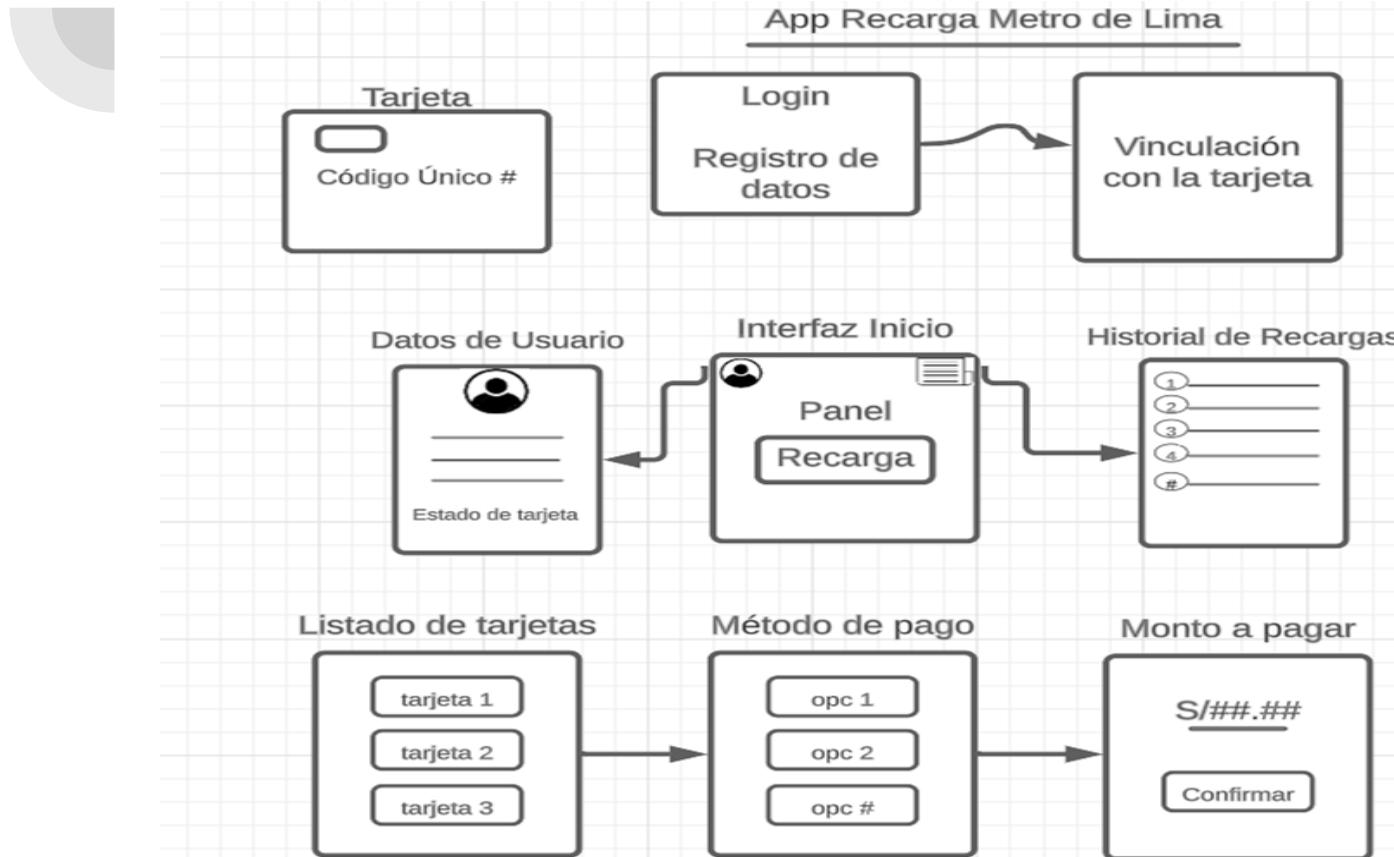
Uso del Tren (o cualquier Metro) con app

CHINA, JAPÓN

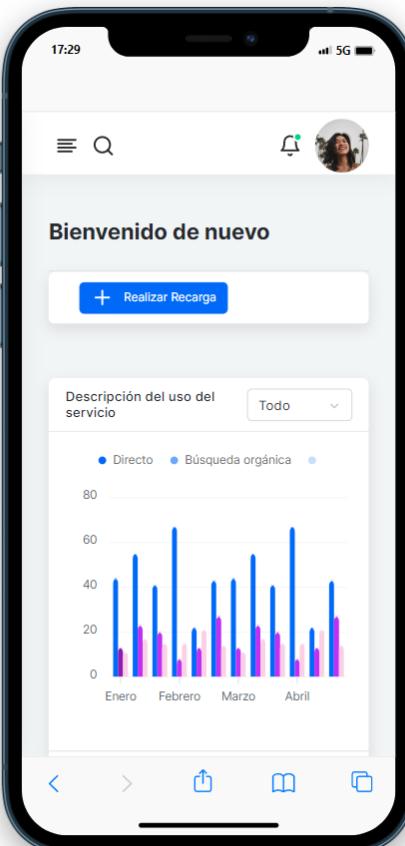
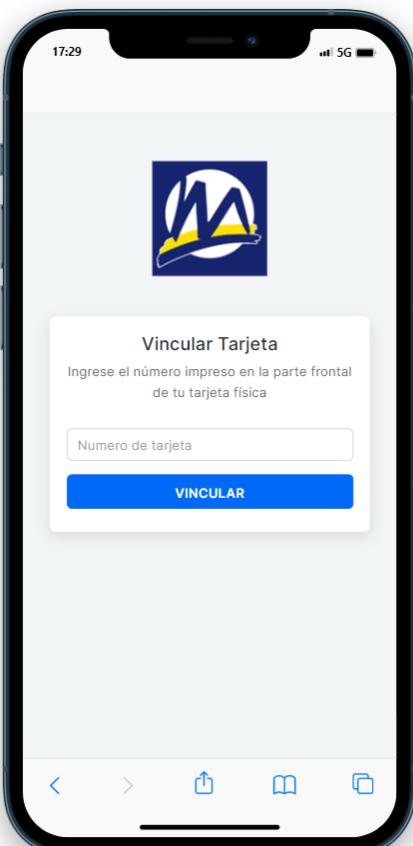
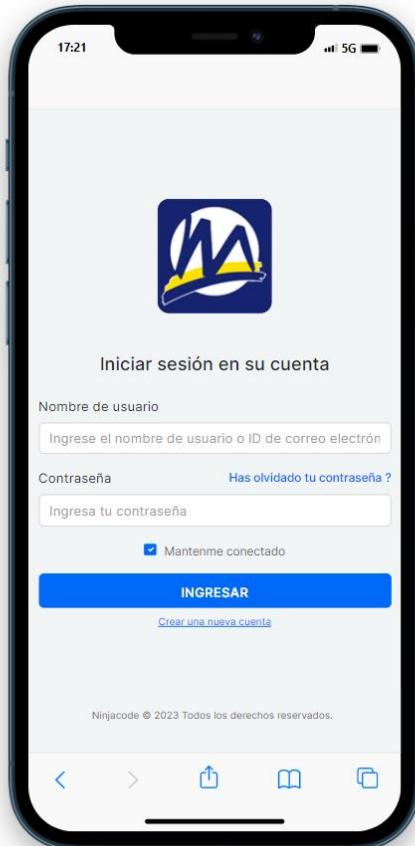


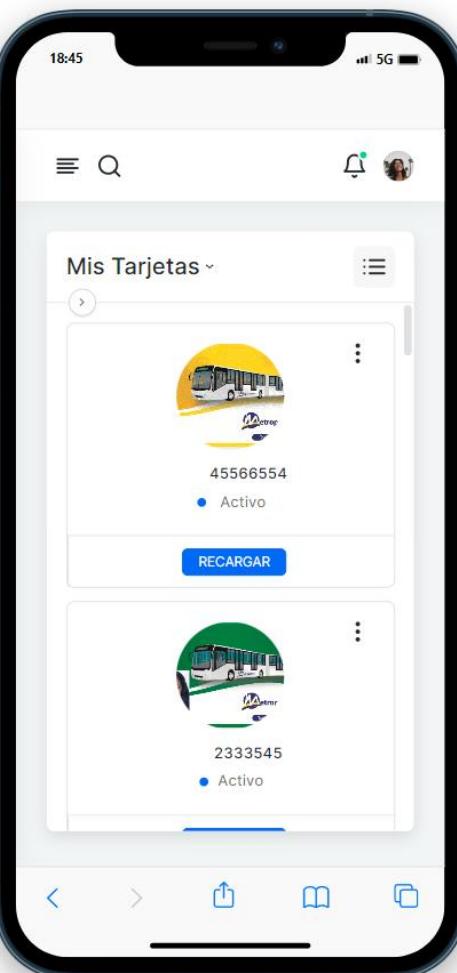
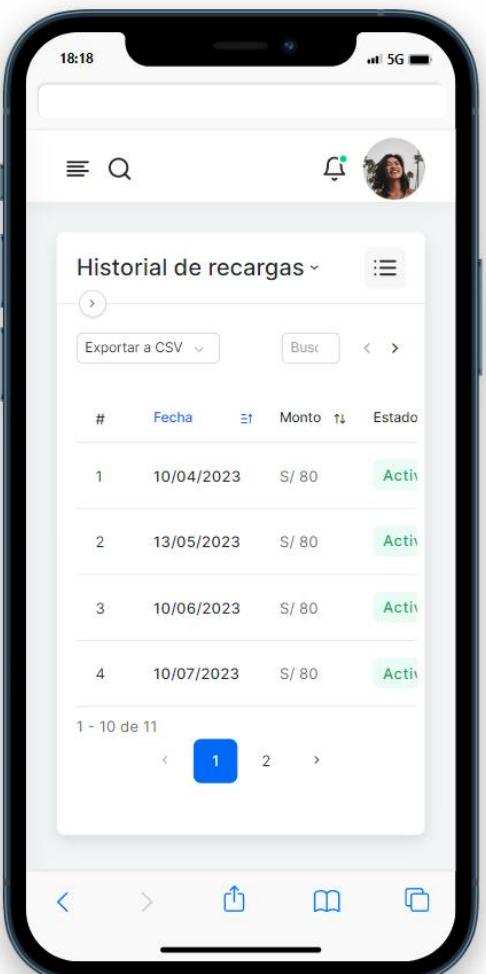
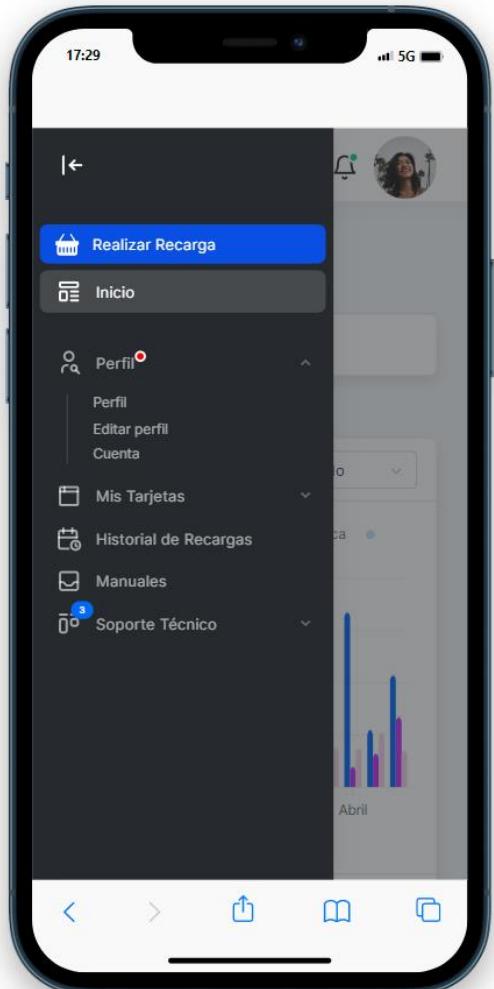
1 proceso, 1 vez hacer cola (directo al metro)

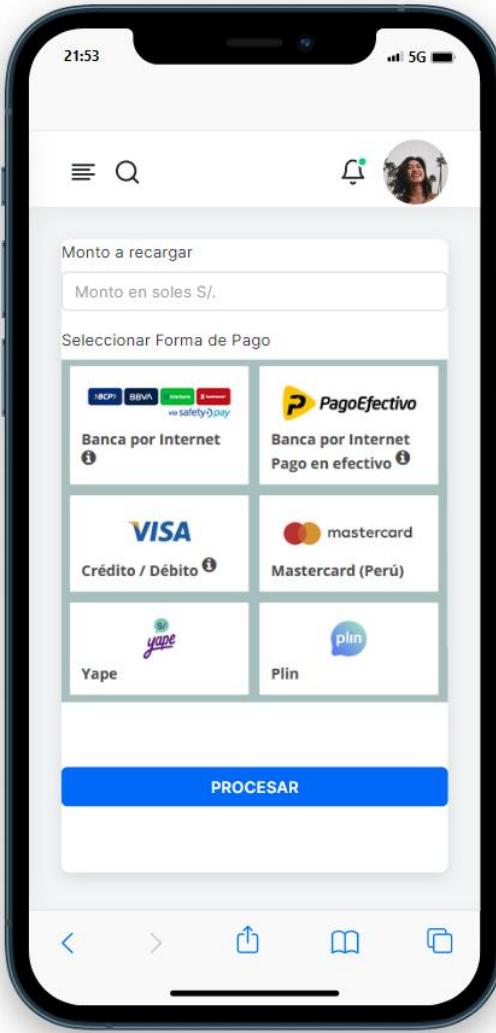
Detalles del nuevo proceso



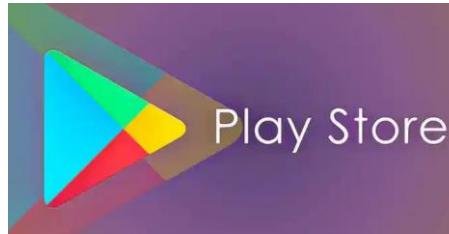
Desarrollo de App con Flutter y Dart







Herramientas y tecnologías a usar



Código Fuente en Python

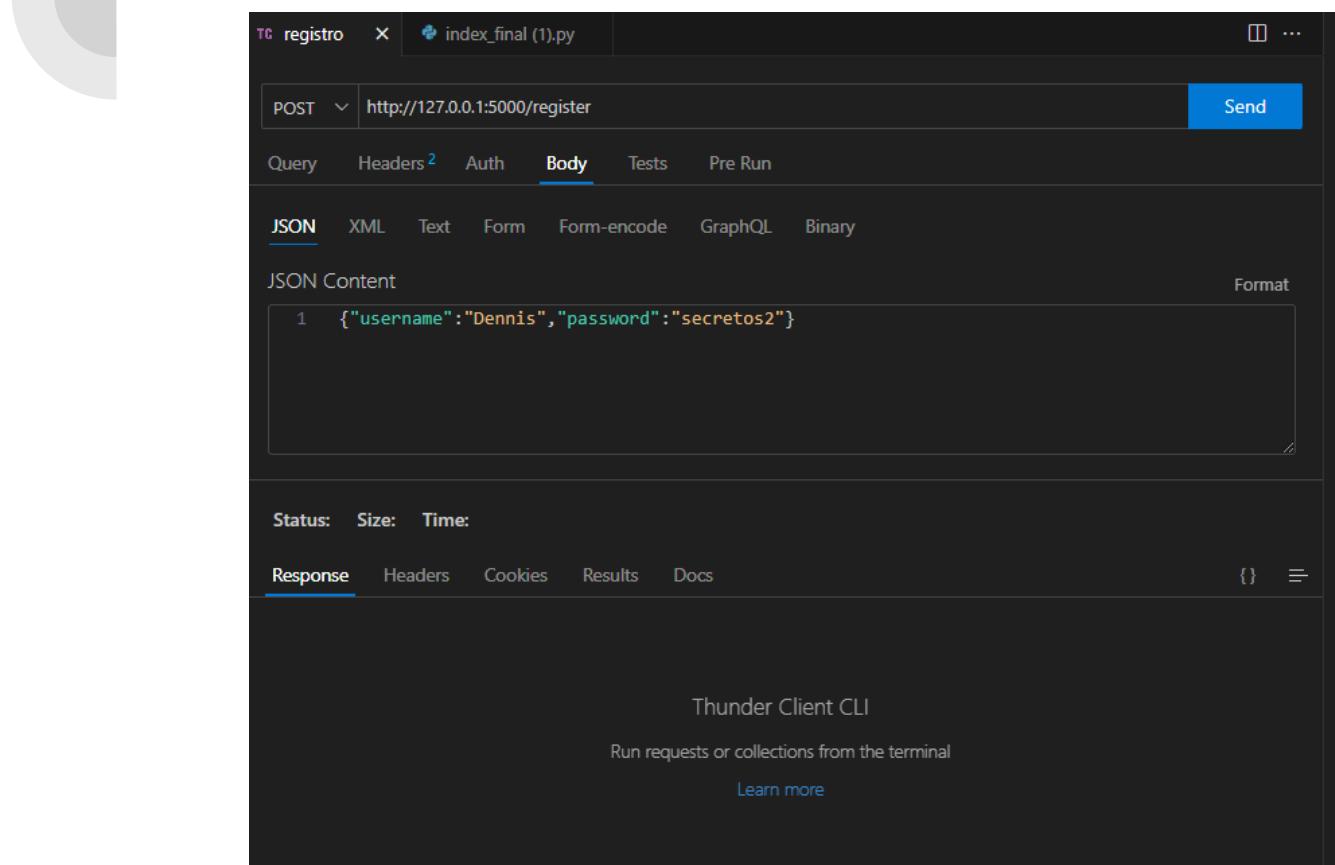
```
demo01.py > ...
1  from flask import Flask, jsonify, request #necesitamos flask para implementar servicio web, jsonify codifica json - consulta al servidor
2
3  app = Flask(__name__) #declaramos el metodo en la variable app
4
5  users = [] # Lista para almacenar usuarios
6
7  @app.route('/register', methods=['POST']) #declaramos las urls de comunicacion con el metodo POST
8  def register():
9      data = request.get_json() #capturamos los parametros que envia el aplic.
10     username = data['username']
11     password = data['password']
12     users.append({'username': username, 'password': password})
13     return jsonify({'message': 'Usuario registrado exitosamente'}) #me devuelve una respuesta en el aplic.
14
15 @app.route('/login', methods=['POST']) #declaramos las urls de comunicacion con el metodo POST
16 def login():
17     data = request.get_json()
18     username = data['username']
19     password = data['password']
20     for user in users:
21         if user['username'] == username and user['password'] == password:
22             return jsonify({'message': 'Inicio de sesión exitoso'})
23     return jsonify({'message': 'Usuario o contraseña incorrectos'})
24
25 @app.route('/list', methods=['GET']) # metodo para listar users
26 def list():
27     return jsonify({'usuarios': users})
28
29 if __name__ == '__main__':
30     app.run(debug=True) #inicializamos la aplicacion
31
32
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL
PS C:\Users\Ninjocode\Desktop\Proyecto App> & C:/Users/Ninjocode/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Ninjocode/Desktop/Proyecto App/demo01.py"
* Serving Flask app 'demo01'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 375-110-352
```

Flask se usa para aplicaciones web con python

Levantamos el servicio con flask para que nos genere el enrutador URL

Haciendo testing con plugin “thunder client”



Enviamos los parámetros a la url register mediante el método post

File Edit Selection View Go Run Terminal Help ➔ Proyecto App

THUNDER CLIENT ⚡ ...

New Request

Activity Collections Env

filter collections

Testing App

POST registro just now

POST registro

http://127.0.0.1:5000/register

Send

Query Headers² Auth Body¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {"username": "Dennis", "password": "secretos2"}
```

Status: 200 OK Size: 51 Bytes Time: 7 ms

Response Headers⁵ Cookies Results Docs

```
1 {
2   "message": "Usuario registrado exitosamente"
3 }
```

Copy ↵ ↓

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

THUNDER CLIENT

New Request

Activity Collections Env

filter collections

Testing App

POST registro just now

GET Listar usuarios just now

registro Listar usuarios index_final (1).py

GET http://127.0.0.1:5000/list Send

Query Headers 2 Auth Body Tests Pre Run

Query Parameters

parameter value

Status: 200 OK Size: 164 Bytes Time: 3 ms

Response Headers 5 Cookies Results Docs

```
1 {
2     "usuarios": [
3         {
4             "password": "secretos2",
5             "username": "Dennis"
6         },
7         {
8             "password": "secretos3",
9             "username": "Cesar"
10        }
11    ]
12 }
```

Copy ↵ ↓

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

The screenshot shows the Thunder Client interface. On the left, there's a sidebar with 'Testing App' expanded, showing a POST request to 'registro' and a GET request to 'Listar usuarios'. The main area has tabs for 'registro', 'Listar usuarios', and 'index_final (1).py'. A 'New Request' button is at the top. Below it are tabs for 'Activity', 'Collections' (which is selected), and 'Env'. A search bar says 'filter collections'. Under 'Testing App', the 'Listar usuarios' entry is highlighted. The main panel shows a 'Query Parameters' section with a table for adding parameters. Below that is a status bar with 'Status: 200 OK', 'Size: 164 Bytes', and 'Time: 3 ms'. The 'Response' tab is selected, showing a JSON response with two user objects. The JSON is formatted with line numbers and copy/cut/clear buttons.

Hacemos una consulta a la Url list, con Get recibimos datos

```

class Tarjeta:
    def vincular(self):
        codigo_tarjeta = input("Ingrese el código de la tarjeta: ")
        # Lógica para vincular la tarjeta
        print("La tarjeta ha sido vinculada exitosamente.")

class Recargas:
    def __init__(self):
        self.recargas = []
        self.saldo = 0

    def realizar_recarga(self):
        monto = float(input("Ingrese el monto a recargar: "))
        tipo_pago = ''

        print("\n----- Forma de pago -----")
        print("1. Yape")
        print("2. Plin")
        print("3. Tarjeta")
        print("4. Volver al Menú Principal")

        opcion = input("Seleccione una opción: ")

        if opcion == "1":
            tipo_pago = 'Yape'
        elif opcion == "2":
            tipo_pago = 'Plin'
        elif opcion == "3":
            tipo_pago = 'Tarjeta'
        elif opcion == "4":
            return

        self.recargas.append({'monto': monto, 'tipo_pago': tipo_pago})
        self.saldo += monto
        print("Recarga realizada exitosamente.")

    def listar_recargas(self):
        print("\n----- Listado de Recargas -----")

        if len(self.recargas) == 0:
            print("No se han realizado recargas aún.")
        else:
            for recarga in self.recargas:
                print(f"Monto: {recarga['monto']} | Tipo de Pago: {recarga['tipo_pago']}")

    def mostrar_saldo(self):
        print(f"Saldo: S/ {self.saldo}")

```

```

class Menu:
    def __init__(self):
        self.tarjeta = Tarjeta()
        self.recargas = Recargas()

    def menu_interno(self):
        while True:
            print("\n----- Menú Interno -----")
            print("1. Realizar Recarga")
            print("2. Listar Recargas")
            print("3. Mostrar Saldo")
            print("4. Volver al Menú Principal")

            opcion = input("Seleccione una opción: ")

            if opcion == "1":
                self.recargas.realizar_recarga()
            elif opcion == "2":
                self.recargas.listar_recargas()
            elif opcion == "3":
                self.recargas.mostrar_saldo()
            elif opcion == "4":
                return
            else:
                print("Opción inválida. Por favor, seleccione nuevamente.")

    def login(self):
        while True:
            print("===== BIENVENIDOS AL APP METRO DE LIMA =====")
            username = input("Ingrese su nombre de usuario: ")
            password = input("Ingrese su contraseña: ")

            if username == "admin" and password == "admin":
                print("Inicio de sesión exitoso.")
                self.menu_principal()
                break
            else:
                print("Credenciales incorrectas. Por favor, intente nuevamente.")

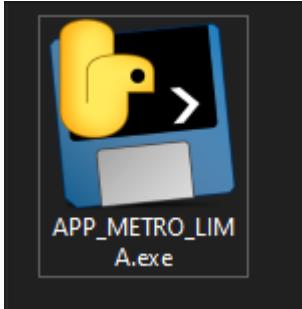
    def menu_principal(self):
        while True:
            print("\n----- Menú Principal -----")
            print("1. Vincular Tarjeta")
            print("2. Menú Interno")
            print("3. Salir")

            opcion = input("Seleccione una opción: ")

            if opcion == "1":
                self.tarjeta.vincular()
            elif opcion == "2":
                self.menu_interno()
            elif opcion == "3":
                print("Hasta luego!")
                break
            else:
                print("Opción inválida. Por favor, seleccione nuevamente.")

```

Compilado en .exe



```
C:\Users\Ninjocode\Desktop\Proyecto App\dist\APP_METRO_LIMA.exe
=====
BIENVENIDOS AL APP METRO DE LIMA
=====
Ingresese su nombre de usuario: admin
Ingresese su contraseña: admin
Inicio de sesión exitoso.

----- Menú Principal -----
1. Vincular Tarjeta
2. Menú Interno
3. Salir
Seleccione una opción: 1
Ingresese el código de la tarjeta: 333333
La tarjeta ha sido vinculada exitosamente.

----- Menú Principal -----
1. Vincular Tarjeta
2. Menú Interno
3. Salir
Seleccione una opción: 2

----- Menú Interno -----
1. Realizar Recarga
2. Listar Recargas
3. Mostrar Saldo
4. Volver al Menú Principal
Seleccione una opción: 1
Ingresese el monto a recargar: 30

----- Forma de pago -----
1. Yape
2. Plin
3. Tarjeta
4. Volver al Menú Principal
Seleccione una opción: 1
Recarga realizada exitosamente.

----- Menú Interno -----
1. Realizar Recarga
2. Listar Recargas
3. Mostrar Saldo
4. Volver al Menú Principal
Seleccione una opción: ■
```



Programación código fuente

[https://drive.google.com/file/d/1bhMStT8POcQLLbjtA2TzK0820ltGQQO
M/view?usp=sharing](https://drive.google.com/file/d/1bhMStT8POcQLLbjtA2TzK0820ltGQQOM/view?usp=sharing)



Conclusiones y Recomendaciones

- La digitalización debe ser un recurso para el desarrollo de la sociedad, así como la solución a los problemas de la misma.
- La base de la organización y el correcto funcionamiento de un servicio de transporte masivo como el metro de Lima y Callao radica en el desarrollo de sistemas eficientes de gestión y registro.
- Se cumplió modelado de una solución viable para solventar la demora en el registro de tarjetas del metro de Lima con el objetivo de beneficiar a los usuarios del transporte.
- Se denota el hecho que los hipotéticos servidores orientados a la gestión de transacciones para el pago de los boleto deben tener constante monitoreo en orden de mantener fluidez y rapidez.



Bibliografía

El Comercio

-<https://elcomercio.pe/lima/transporte/metropolitano-caos-desesperacion-abordar-buses-fotos-noticia-514609-noticia/>

Wikipedia

-https://es.wikipedia.org/wiki/Metro_de_Lima_y_Callao

Youtube

-https://www.youtube.com/watch?v=E6s_UUto9Aw

Chat GPT

-<https://chat.openai.com/>



GRACIAS