



Universidad
Nacional
de Loja

[Desarrollo de Software en Ambientes Cloud]

Desarrollo de caso de estudio:
Plataforma de pagos de transporte

Tutor:

ROBERTH GUSTAVO FIGUEROA DIAZ
roberth.figueroa@unl.edu.ec

Elaborado por:

FREDDY ALEXANDER CHAMORRO ENCALADA
fachamorro@unl.edu.ec

31 de octubre de 2022



Tabla de contenido

1. OBJETIVOS.....	3
2. DESARROLLO.....	3
2.1. Preparación del ambiente de trabajo.....	3
2.2. Comprensión de la temática de estudio y Ejecución de ProyectoSITU.....	4
2.2.1. Instalación y configuración del ProyectoSITU.....	4
2.2.2. Tecnologías utilizadas.....	4
2.2.3. Diagrama de clases.....	5
2.2.4. Explicación del Backend.....	6
2.2.5. Explicación del Frontend.....	7
2.2.6. Problema que pretende resolver el software implementado.....	5
2.2.7. Código fuente.....	9
3. CONCLUSIONES.....	9
4. BIBLIOGRAFIA.....	10
5. FIRMAS.....	10

1. OBJETIVOS

- Identificar los fundamentos y elementos necesarios para el desarrollo de software en ambientes cloud.

2. DESARROLLO

2.1. Preparación del ambiente de trabajo

Para el desarrollo de la presente actividad se utilizó las siguientes herramientas:

Herramienta	Versión	Comando para verificar
Python	3.9.15	python3.9 --version
pip	22.2.2	python -m pip --version
virtualenv	20.017	python -m virtualenv --version
Django	3.2.5	python -m django --version alternativo pip freeze
Git	2.25.1	git --version

```
freddy@freddy-Nitro-AN515-55: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
freddy@freddy-Nitro-AN515-55:~$ python3.9 --version  
Python 3.9.15  
freddy@freddy-Nitro-AN515-55:~$ python3.9 -m virtualenv --version  
virtualenv 20.0.17 from /usr/lib/python3/dist-packages/virtualenv/__init__.py  
freddy@freddy-Nitro-AN515-55:~$ python3.9 -m pip --version  
pip 22.2.2 from /home/freddy/.local/lib/python3.9/site-packages/pip (python 3.9)  
freddy@freddy-Nitro-AN515-55:~$
```

En caso de no disponer de python o las herramientas, en linux (mint) las instalamos con los siguientes comandos:

```
#instalar python3.9  
sudo apt install python3.9  
#instalar pip  
sudo apt install python3-pip  
#instalar virtualenv  
sudo apt install python3-virtualenv
```

2.2. Comprensión de la temática de estudio y Ejecución de ProyectoSITU

2.2.1. Instalación y configuración del ProyectoSITU

Para la instalación y pruebas del proyecto facilitado en el repositorio en github se configuró un entorno virtual de Python para lo cual se utilizó el siguiente comando:

```
#creamos el entorno virtual, se especifica la version de python por tener varias versiones instaladas en el anfitrión  
python3.9 -m venv RepoTransporteSITU
```

Dentro del entorno creado (directorio RepoTransporteSITU) podemos observar varios archivos y subdirectorios que se crean por defecto, estos nos permitirán hacer uso del entorno virtual. Para activar el entorno virtual utilizamos el siguiente comando:

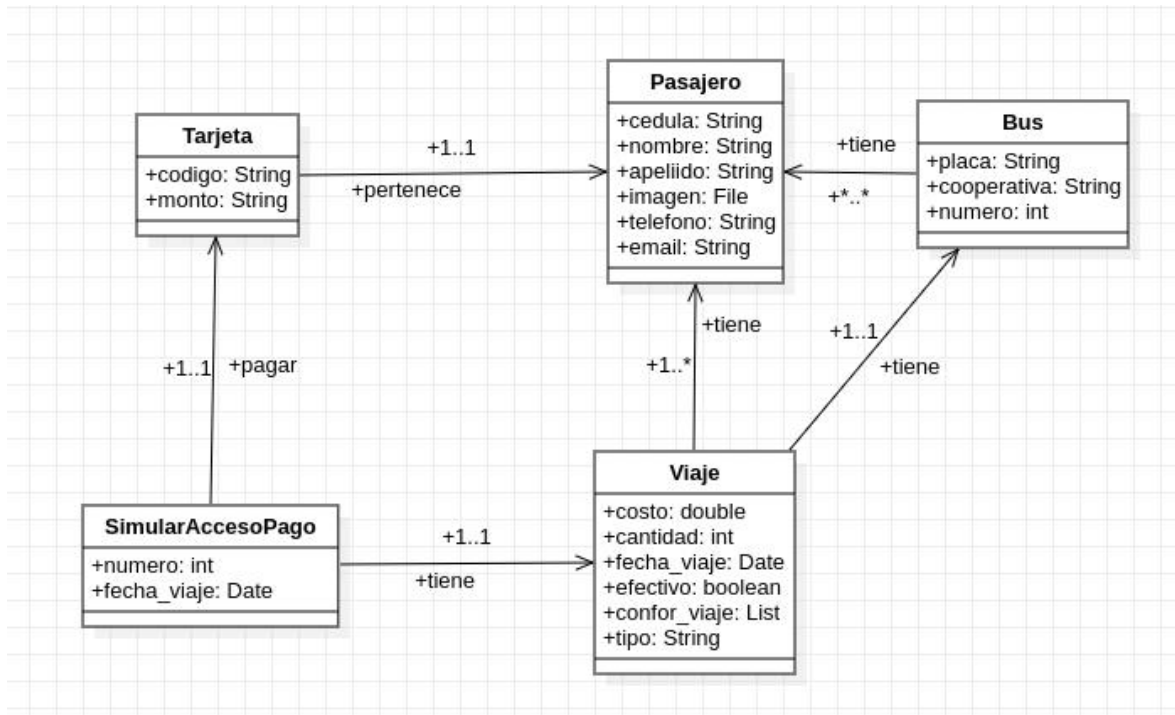
```
#activar entorno virtual  
source RepoTransporteSITU/bin/activate  
#para verificar los paquetes instalados utilizamos el comando  
pip freeze
```

2.2.2. Tecnologías utilizadas

En el entorno virtual instalamos django en su version 3.2.5 y creamos un proyecto de Django, posteriormente creamos el superusuario y luego una aplicación donde copiaremos el código del del proyecto facilitado en github(todas las herramientas y versiones utilizadas en la presente actividad se encuentran en el apartado 2.1).

```
#comando para instalar dentro del entorno las herramientas a utilizar  
pip install Django==3.2.5  
#Para crear el proyecto de Django se utilizó el siguiente comando  
django-admin startproject ProyectoTransporte  
#Para crear la aplicación de Django se utilizó el siguiente comando  
python manage.py startapp SITU
```

2.2.3. Diagrama de clases



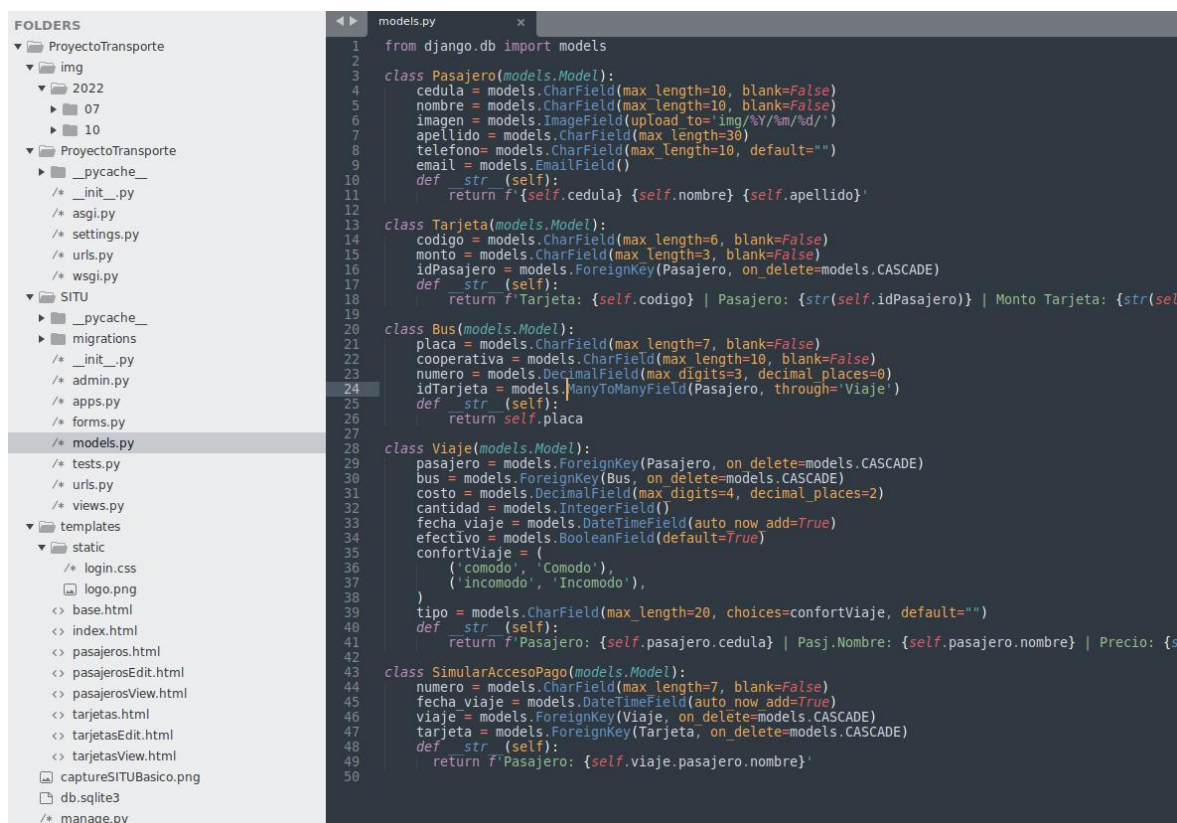
2.2.4. Problema que pretende resolver el software implementado

El proyecto aborda la gestión pago de transporte público, en el modelado se puede observar lo siguiente:

- En el sistema se registra los datos personales del usuario o pasajero del transporte.
- De la entidad bus se registra su placa, cooperativa y numero de bus.
- Cuando el pasajero realiza un viaje se registra sus datos y el número de bus, además se registra el costo de viaje, la fecha, si pagará con efectivo o con tarjeta.
- Al realiza el pago con tarjeta el sistema genera el acceso para el pago y registra en número y fecha de viaje.
- Seguido el sistema resta del saldo de la tarjeta del pasajero el monto del valor del viaje y asigna un código a la transacción.

2.2.5. Explicación del Backend

La estructura del proyecto en la parte Backend es la siguiente:



Explicación:

- En el archivo **settings.py** del proyecto registramos nuestras aplicaciones en la opción llamada `INSTALLED_APPS`. Este archivo además no permite realizar ajustes/configuración para este proyecto Django; nos define un conjunto de configuraciones que podemos aplicar sobre el proyecto de manera global.
- El archivo **urls.py** nos permite realizar las declaraciones URL para el proyecto Django; desde aquí podemos cargar más archivos de URLs de otras aplicaciones o dependencias.
- La base de datos tipo SQLite se encuentra en el archivo **db.sqlite3**, es donde se almacenarán todos los datos ingresados en el sistema.

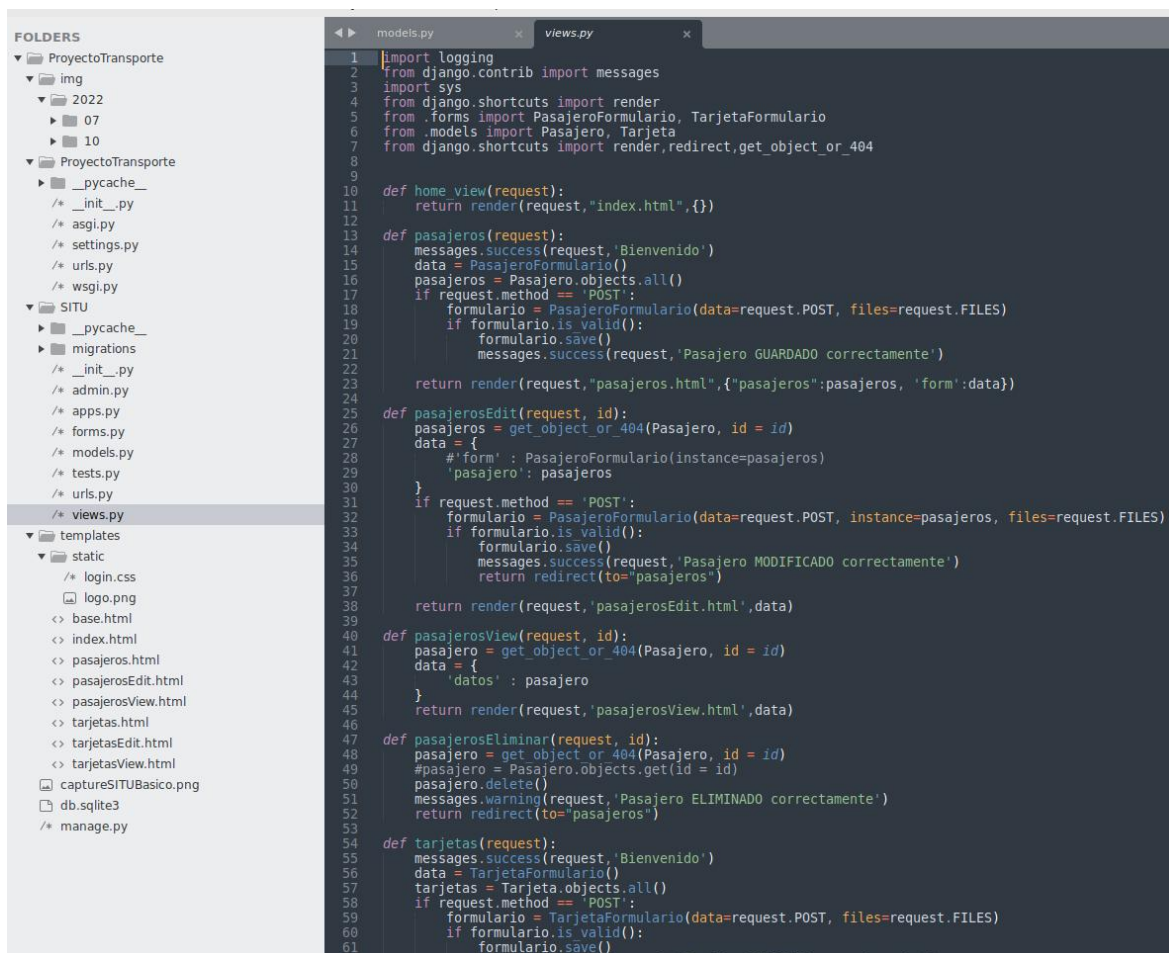
Dentro de las aplicaciones creadas tenemos algunos archivos importantes, se nombrarán los que tienen mas relevancia y en los que se realizaron modificaciones:

- El modelado de clases se realizó en el archivo **models.py**. A cada clase se agregó el método especial llamado **__str__** que debe devolver una cadena de caracteres con lo que queremos mostrar. Ese método se invoca cada vez que se llama a la función **str**.
- En el archivo **admin.py** podemos registrar modelos para emplearlos en una aplicación que nos ofrece Django para realizar la gestión de datos.

Al momento actual se encuentra implementado las funcionalidades pasajero y tarjeta en un CRUD básico.

2.2.6. Explicación del Frontend

La estructura del proyecto en la parte Frontend es la siguiente:



```

FOLDERS
▼ ProyectoTransporte
  ▼ img
    ▼ 2022
      ▼ 07
        ▼ 10
  ▼ ProyectoTransporte
    ▼ __pycache__
    /* __init__.py
    /* asgi.py
    /* settings.py
    /* urls.py
    /* wsgi.py
  ▼ SITU
    ▼ __pycache__
    ▼ migrations
    /* __init__.py
    /* admin.py
    /* apps.py
    /* forms.py
    /* models.py
    /* tests.py
    /* urls.py
    /* views.py
  ▼ templates
    ▼ static
      /* login.css
      logo.png
      base.html
      index.html
      pasajeros.html
      pasajerosEdit.html
      pasajerosView.html
      tarjetas.html
      tarjetasEdit.html
      tarjetasView.html
    captureSITUBasico.png
    db.sqlite3
    /* manage.py

models.py
1 import logging
2 from django.contrib import messages
3 import sys
4 from django.shortcuts import render
5 from .forms import PasajeroFormulario, TarjetaFormulario
6 from .models import Pasajero, Tarjeta
7 from django.shortcuts import render, redirect, get_object_or_404
8
9
10 def home_view(request):
11     return render(request, "index.html", {})
12
13 def pasajeros(request):
14     messages.success(request, 'Bienvenido')
15     data = PasajeroFormulario()
16     pasajeros = Pasajero.objects.all()
17     if request.method == 'POST':
18         formulario = PasajeroFormulario(data=request.POST, files=request.FILES)
19         if formulario.is_valid():
20             formulario.save()
21             messages.success(request, 'Pasajero GUARDADO correctamente')
22
23     return render(request, "pasajeros.html", {"pasajeros": pasajeros, 'form': data})
24
25 def pasajerosEdit(request, id):
26     pasajero = get_object_or_404(Pasajero, id = id)
27     data = {
28         #'form' : PasajeroFormulario(instance=pasajeros)
29         'pasajero': pasajero
30     }
31     if request.method == 'POST':
32         formulario = PasajeroFormulario(data=request.POST, instance=pasajeros, files=request.FILES)
33         if formulario.is_valid():
34             formulario.save()
35             messages.success(request, 'Pasajero MODIFICADO correctamente')
36             return redirect(to="pasajeros")
37
38     return render(request, "pasajerosEdit.html", data)
39
40 def pasajerosView(request, id):
41     pasajero = get_object_or_404(Pasajero, id = id)
42     data = {
43         'datos': pasajero
44     }
45     return render(request, "pasajerosView.html", data)
46
47 def pasajerosEliminar(request, id):
48     pasajero = get_object_or_404(Pasajero, id = id)
49     #pasajero = Pasajero.objects.get(id = id)
50     pasajero.delete()
51     messages.warning(request, 'Pasajero ELIMINADO correctamente')
52     return redirect(to="pasajeros")
53
54 def tarjetas(request):
55     messages.success(request, 'Bienvenido')
56     data = TarjetaFormulario()
57     tarjetas = Tarjeta.objects.all()
58     if request.method == 'POST':
59         formulario = TarjetaFormulario(data=request.POST, files=request.FILES)
60         if formulario.is_valid():
61             formulario.save()

```

Explicación:

- El archivo **views.py** nos permite crear las funciones o vistas, es la capa de la vista, aquí especificamos las funciones del CRUD de nuestra aplicación.
- En el fichero **forms.py** es donde especificamos los formularios que queremos utilizar en nuestra página web. Estos formularios permiten al usuario interactuar con el sistema.
- Adicionalmente debemos crear una carpeta en el directorio raíz llamada **templates** donde ubicaremos todos los archivos HTML y estáticos que usaremos en la aplicación como imágenes, iconos, entre otros.

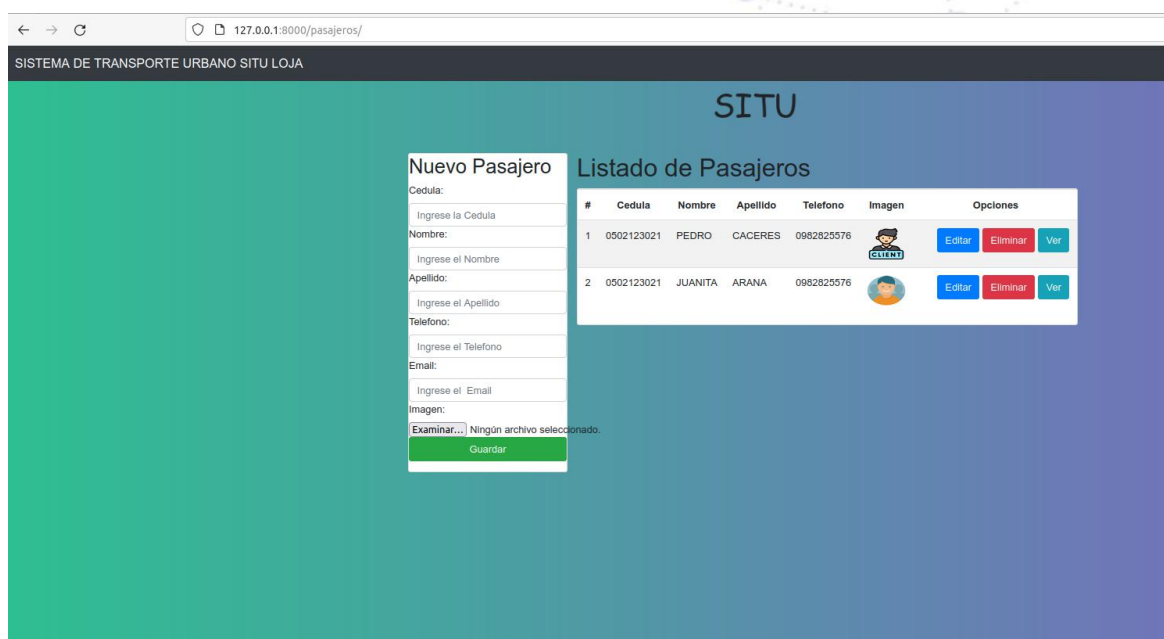
Al momento actual se encuentra implementado las funcionalidades pasajero y tarjeta en un CRUD básico.

2.2.7. Ejecución del proyecto

Para ejecutar el proyecto utilizamos el siguiente comando:

```
#para la ejecución del proyecto utilizamos el comando el proyecto por defecto se ejecuta en el puerto 8000  
python manage.py runserver
```

A continuación se muestra la ejecución del proyecto con el CRUD creado para el pasajero.



SISTEMA DE TRANSPORTE URBANO SITU LOJA

SITU

Nuevo Pasajero

Cedula:
Ingrese la Cedula

Nombre:
Ingrese el Nombre

Apellido:
Ingrese el Apellido



Telefono:
Ingrese el Telefono

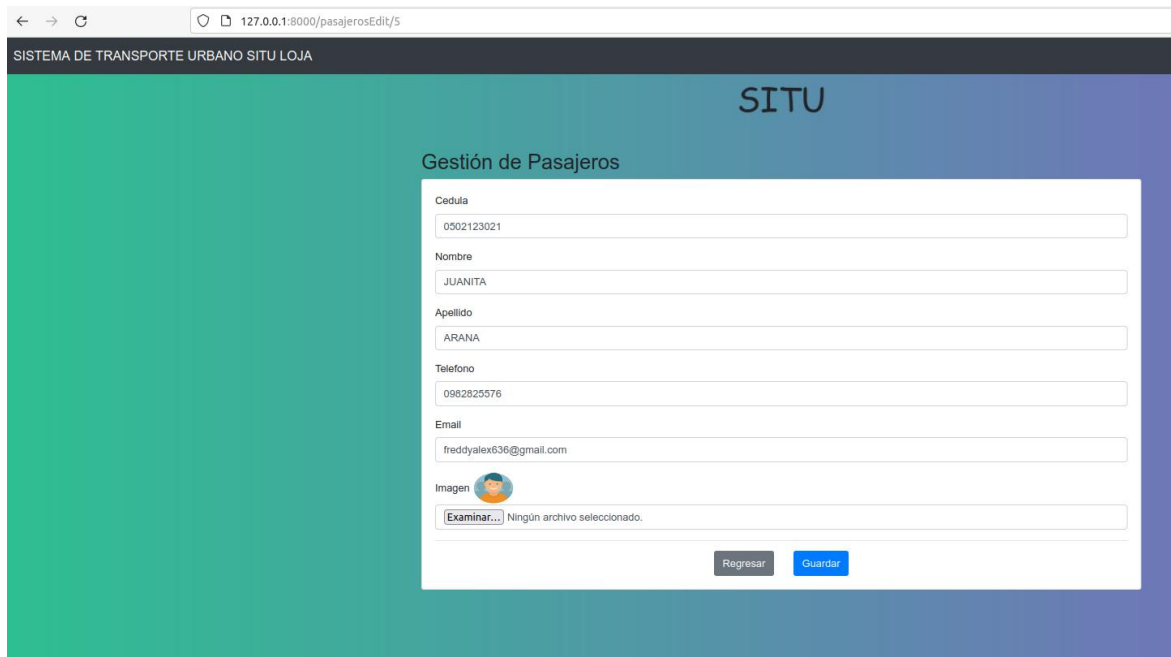
Email:
Ingrese el Email

Imagen:
Examinar... Ningún archivo seleccionado.

Guardar

Listado de Pasajeros

#	Cedula	Nombre	Apellido	Telefono	Imagen	Opciones
1	0502123021	PEDRO	CACERES	0982825576		Editar Eliminar Ver
2	0502123021	JUANITA	ARANA	0982825576		Editar Eliminar Ver



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/pasajerosEdit/5'. The page title is 'SISTEMA DE TRANSPORTE URBANO SITU LOJA'. The main content area has a green and blue gradient background with the text 'SITU' in the top right. A white form titled 'Gestión de Pasajeros' is centered on the page. The form contains the following fields: 'Cedula' (0502123021), 'Nombre' (JUANITA), 'Apellido' (ARANA), 'Telefono' (0982825576), 'Email' (freddyalex636@gmail.com), and 'Imagen' (a small profile picture icon). Below the 'Imagen' field is a button labeled 'Examinar...' and the text 'Ningún archivo seleccionado.'. At the bottom of the form are two buttons: 'Regresar' and 'Guardar'.

2.2.8. Código fuente

Repositorio en GitHub

<https://github.com/fachamorro/ProyectoSITU.git>

3. CONCLUSIONES

- Django es un framework de desarrollo web que nos permite crear aplicaciones de manera rápida en el patrón de arquitectura de software MVC. Además permite crear un sitio de administración de Django donde los administradores del sitio pueden usar la página para crear, editar o ver los modelos de datos del sitio.
- Virtualenv nos permite crear entornos de trabajo aislados con herramientas y dependencias separadas de las instaladas de manera global en el sistema operativo del computador, nos permite instalar dependencias acorde a las necesidades para uso solo en el proyecto que estamos trabajando.

4. BIBLIOGRAFIA

- Barranquilla, P. (s/f-a). Una guía completa de Django para principiantes - Parte 2. <https://pybaq.co/>. Recuperado el 25 de octubre de 2022, de <https://pybaq.co/blog/una-guia-completa-de-django-para-principiantes-parte-2/>
- Barranquilla, P. (s/f-b). Una guía completa de Django para principiantes- Parte 1. <https://pybaq.co/>. Recuperado el 25 de octubre de 2022, de <https://pybaq.co/blog/una-guia-completa-de-django-para-principiantes-parte-1/>
- Django tutorial part 9: Working with forms. (s/f). Mozilla.org. Recuperado el 27 de octubre de 2022, de <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Forms>
- Writing your first Django app, part 1. (s/f). Django project.com. Recuperado el 27 de octubre de 2022, de <https://docs.djangoproject.com/en/4.1/intro/tutorial01/>

5. FIRMAS

Nombres y Apellidos	Firma
Freddy Alexander Chamorro Encalada	