

Bancolombia Prueba técnica Oct – 06.

DESARROLLO DE REQUISITOS.....	2
- INICIALIZACIÓN DEL PROYECTO DJANGO CON POETRY.....	2
- CREA UNA BASE DE DATOS CON UNA TABLA LLAMADA "MOVIMIENTOS". LA TABLA DEBE TENER CAMPOS PARA ALMACENAR INFORMACIÓN COMO EL VALOR DE LA TRANSACCIÓN, NÚMERO DE CUENTA, FECHA.....	2
Movimiento/views.py.....	2
Extractos/urls.py.....	3
Extractos/movimiento/models.py.....	3
- CREE UNA TABLA LLAMADA "CUENTAS" CON LA INFORMACIÓN DE NÚMERO DE CUENTA, TIPO DE CUENTA, SALDO Y TITULAR.	4
- INGRESE INFORMACIÓN DUMMY EN AMBAS TABLAS.	5
Ejecución de creación de datos en SQLite.....	6
TABLA usuario_usuarios.	6
Tabla cuenta_cuenta.....	6
Tabla movimiento_movimiento.....	7
- IMPLEMENTAR UNA FUNCIONALIDAD QUE PERMITA A LOS USUARIOS GENERAR UN ARCHIVO DE EXCEL QUE CONTENGA EL RESUMEN DE TODOS LOS MOVIMIENTOS DE SU CUENTA, CON UN DISEÑO AMIGABLE PARA EL USUARIO TIPO EXTRACTO CON LOS LOGOS DE LA ORGANIZACIÓN.....	8
Implementación.....	9
1. Realización de las migraciones.	9
2. Creación de datos.....	10
3. Creación del diseño del formato pdf.	10
4. Crear funcionalidades relacionadas al flujo básico del programa.	12
Función que regresa los movimientos dado la cuenta.	12
Secuencia (Flujo básico):.....	13
1. Usuario ingresa cédula (Para que el sistema valide si tiene cuenta.).....	13
2. Usuario solicita extracto (En este punto el usuario puede regresarse al home de la cuenta).	14
3. Sistema crea extracto.....	18
Posibles mensajes (Flujos alternativos).	18
- EL ARCHIVO EXCEL GENERADO DEBE ESTAR BLOQUEADO PARA MODIFICACIONES.	19
- EL ARCHIVO EXCEL DEBE TENER CONTRASEÑA DE APERTURA	19
ENTREGABLE:	21
SCRIPT DE CREACIÓN DE LA BASE DE DATOS	21
Tabla usuario_usuario.....	21
Tabla cuenta_cuenta.....	21
Tabla movimiento_movimiento.	22
SCRIPT DE INSERCIÓN DE LA DATA DUMMY.....	23
Insertar datos de usuario_usuario.....	23
Insertar datos en cuenta_cuenta.....	23
Insertar datos de movimientos.	24
EXTRA:	26
Al clonar el repositorio (Con git o github).....	26

Desarrollo de requisitos.

Se desarrollará la aplicación en django utilizando vistas.

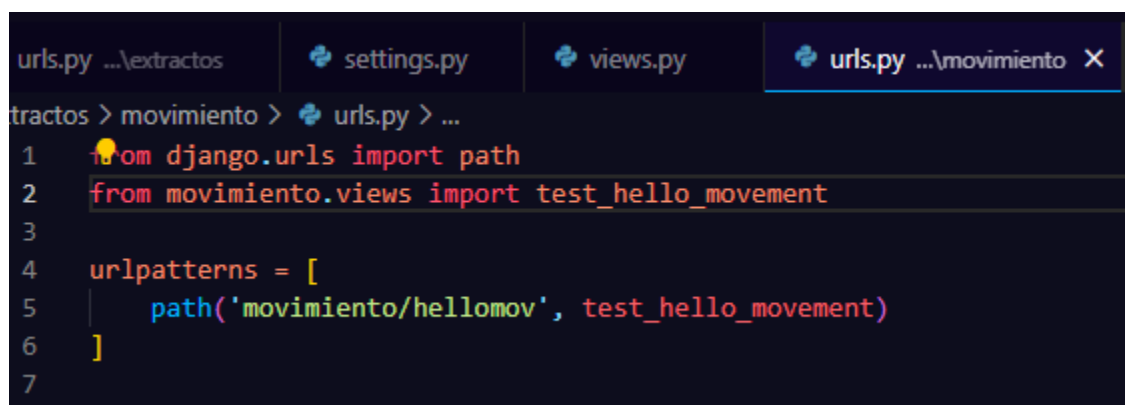
- *Inicialización del proyecto django con poetry.*
 - Primero se crea la carpeta del proyecto para inicializar el proyecto, en este caso la carpeta extractos.
 - Inicializar el archivo pyproject.toml: `poetry init --no-interaction --dependency django`.
 - Crear el proyecto: `poetry run django-admin startproject extractos`
 - Cada que se realice una instalación de dependencias, hacer `poetry Shell`.
 - Para añadir las dependencias, dentro de poetry Shell: `poetry add django`.
 - Seleccionar el interprete que tenga el django instalado
- *Crea una base de datos con una tabla llamada "Movimientos". La tabla debe tener campos para almacenar información como el valor de la transacción, número de cuenta, fecha.*
 - Ir a la carpeta extractos que contiene el archivo manage.py y realizar: `python manage.py startapp movimiento`
 - Probar un hola mundo en la carpeta para probar la configuración.

Movimiento/views.py

```
from django.shortcuts import render, HttpResponse

# Create your views here.
def test_hello_movement(request):
    return HttpResponse("Hola movimiento")
```

ERROR: Especificar app.views



```
urls.py ...\extractos  settings.py  views.py  urls.py ...\movimiento X
tractos > movimiento > urls.py > ...
1  from django.urls import path
2  from movimiento.views import test_hello_movement
3
4  urlpatterns = [
5      path('movimiento/hellomov', test_hello_movement)
6  ]
7
```

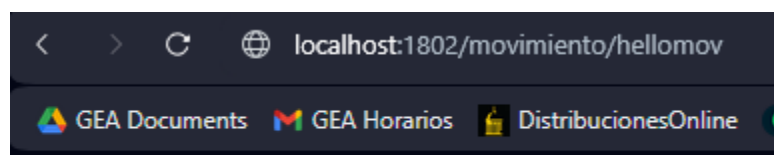
Bancolombia Prueba técnica Oct – 06.

Extractos/urls.py

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('', include('movimiento.urls')),
    path('admin/', admin.site.urls),
]
```

Revisión. Se harán en postman.



Hola movimiento

- Atributos que va a llevar la tabla Movimiento:
 - Fecha
 - Número de cuenta
 - Valor de la transacción
- Contexto del problema para la tabla movimiento.
- TABLA Movimiento
- NOTA: Estos datos no se usarán como dummy, es solo para mostrar la estructura.

Fecha	N° Cuenta	Valor
08/18	123	20
08/18	123	22
08/19	123	222
08/19	1234	1230
08/20	12345	123412
08/20	12345	1231235

Extractos/movimiento/models.py

Configuración del modelo en django.

```
# Creacion del modelo de movimiento.
class Movimiento(models.Model):
    fecha = models.DateField(default=timezone.now())
    n_cuenta = models.ForeignKey(Cuenta, on_delete=models.CASCADE)
    valor = models.FloatField()
```

NOTA: Django establece que no puede usarse `datetime.date.today()`, sino la propia fecha de hoy de django.

Bancolombia Prueba técnica Oct – 06.

Instead, you should pass a callable function to `default`, that is executed each time a model instance needs a default value. The hint wants to convey that you should literally use `DateField(default=django.utils.timezone.now)` without the parentheses.

Se establece la fecha por defecto el reloj del sistema.

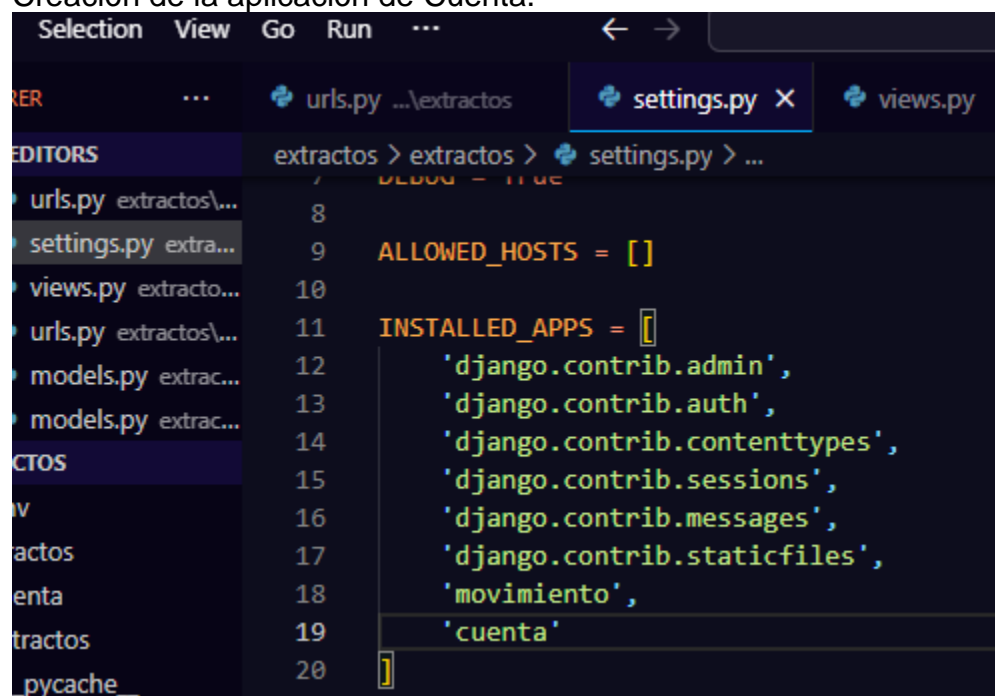
- Cree una tabla llamada “Cuentas” con la información de número de cuenta, tipo de cuenta, saldo y titular.

- Atributos para la tabla Cuenta.
 - Número de cuenta.
 - Tipo de cuenta.
 - Saldo.
 - Titular.
- Analizando el contexto del software.

TABLA Cuenta

Nº Cuenta	Titular	Tipo_cuenta	Saldo
123	Yo		
1234	Yo2		
12345	Yo5		

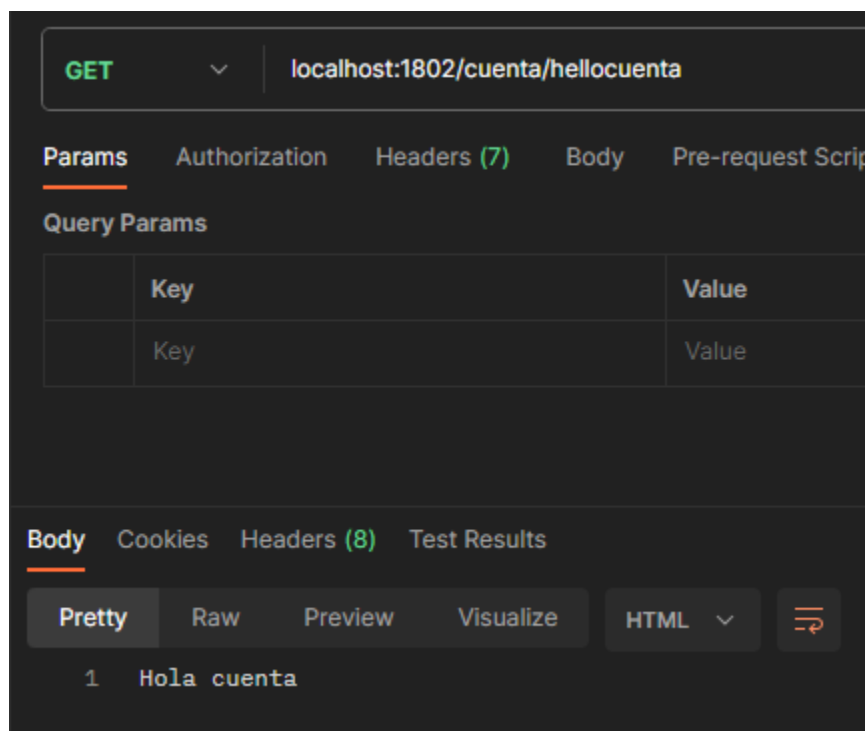
- Creación de la aplicación de Cuenta.



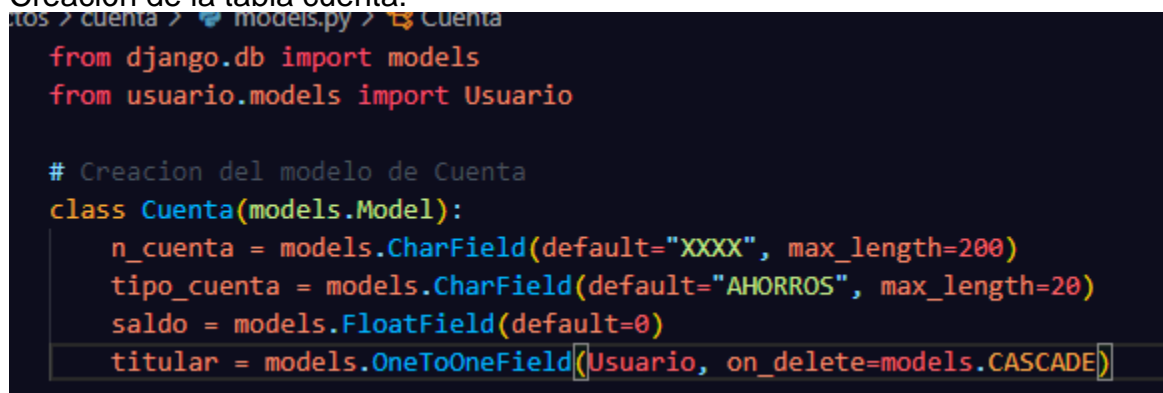
```

8
9 ALLOWED_HOSTS = []
10
11 INSTALLED_APPS = [
12     'django.contrib.admin',
13     'django.contrib.auth',
14     'django.contrib.contenttypes',
15     'django.contrib.sessions',
16     'django.contrib.messages',
17     'django.contrib.staticfiles',
18     'movimiento',
19     'cuenta'
20 ]
  
```

- Se realiza el mismo proceso para movimiento, para probar una ruta “Hola mundo” que corrobora la correcta conexión con el programa.



- Creación de la tabla cuenta.



- Ingrese información Dummy en ambas tablas.

El ingreso de los datos Dummy, estarán dados en la sección “Script de inserción de la data dummy”, se crearán un total de 10 registros por tabla (Iniciamente en postgres para probar los comandos, luego en SQLite para usarlo en el proyecto), variando distintas cédulas, cuentas y movimientos.

- Notemos que, existen discrepancia en el manejo de SQL entre sqlite y postgres. En este caso, Postgres se usó para probar los comandos SQL de creación de tablas e inserción de datos (Lo mismo se hará para **SQLite en este caso, ya que está conectado al proyecto creado en django y es el que se trabajará**).

Bancolombia Prueba técnica Oct – 06.

Ejecución de creación de datos en SQLite.

NOTA: Los códigos para crear las tablas e insertar los datos están en la sección “Entregable” al final de trabajo.

TABLA usuario_usuario

```
CREATE TABLE "usuario_usuario" (
    "identificacion" varchar(50) NOT NULL,
    PRIMARY KEY("identificacion")
);
```

Table: usuario_usuario

	identificacion
	Filter
1	1004
2	1005
3	1006
4	1083
5	1099
6	xgb777
7	Anue12
8	0462
9	0910
10	al18

```
INSERT INTO "usuario_usuario" (identificacion)
VALUES ('1004'),
       ('1005'),
       ('1006'),
       ('1083'),
       ('1099'),
       ('xgb777'),
       ('Anue12'),
       ('0462'),
       ('0910'),
       ('al18');
```

Tabla cuenta_cuenta

```
CREATE TABLE "cuenta_cuenta" (
    "n_cuenta" varchar(50) NOT NULL,
    "tipo_cuenta" varchar(20) NOT NULL,
    "saldo" real NOT NULL,
    "titular_id" varchar(50) NOT NULL UNIQUE,
    FOREIGN KEY("titular_id") REFERENCES "usuario_usuario"("identificaci
        ON DELETE CASCADE
    DEFERRABLE INITIALLY DEFERRED,
    PRIMARY KEY("n_cuenta")
);
```

Bancolombia Prueba técnica Oct – 06.

table: **cuenta_cuenta**

n_cuenta	tipo_cuenta	saldo	titular_id
Filter	Filter	Filter	Filter
1 5318	AHORROS	10.0	1004
2 6518	AHORROS	-10.0	1006
3 8518	AHORROS	0.0	1083
4 9518	CORRIENTE	10.0	1099
5 0518	CORRIENTE	10.0	xgb777
6 1518		2.0	Anue12
7 2518		10.0	0462
8 3518	CORRIENTE	9.0	0910

```
INSERT INTO cuenta_cuenta ("n_cuenta", "tipo_cuenta", "saldo", "titular_id")
VALUES ('5318', 'AHORROS', 10, '1004'),
('6518', 'AHORROS', -10, '1006'),
('8518', 'AHORROS', 0, '1083'),
('9518', 'CORRIENTE', 10, '1099'),
('0518', 'CORRIENTE', 10, 'xgb777'),
('1518', '', 2, 'Anue12'),
('2518', '', 10, '0462'),
('3518', 'CORRIENTE', 9, '0910');
```

Tabla movimiento_movimiento.

```
CREATE TABLE "movimiento_movimiento" (
    "id" integer NOT NULL,
    "fecha" date NOT NULL,
    "valor" real NOT NULL,
    "n_cuenta_id" varchar(50) NOT NULL,
    PRIMARY KEY("id" AUTOINCREMENT),
    FOREIGN KEY("n_cuenta_id")
REFERENCES "cuenta_cuenta" ("n_cuenta")
ON DELETE CASCADE
DEFERRABLE INITIALLY DEFERRED
);
```

```
INSERT INTO movimiento_movimiento ("fecha", "valor", "n_cuenta_id")
VALUES ('2023-1-1', RANDOM(), '5318'),
('2022-3-15', RANDOM(), '5318'),
('2021-6-30', RANDOM(), '5318'),
('2020-9-1', RANDOM(), '6418'),
('2023-12-15', RANDOM(), '6418'),
('2022-1-30', RANDOM(), '6418'),
('2021-3-1', RANDOM(), '6518'),
('2020-6-15', RANDOM(), '6518'),
('2023-9-1', RANDOM(), '6518'),
('2022-12-30', RANDOM(), '8518'),
('2021-1-2', RANDOM(), '8518'),
('2020-3-14', RANDOM(), '8518');
```

Bancolombia Prueba técnica Oct – 06.

	id	fecha	valor	n_cuenta_id
	Filter	Filter	Filter	Filter
15	15	2022-9-2	-6.05530314967126e...	9518
16	16	2021-1...	5.16896225406394e+18	9518
17	17	2020-1...	2.9974804243934e+18	0518
18	18	2023-3-3	-1.82260300808306e...	0518
19	19	2022-6...	1.29050186800268e+18	0518
20	20	2021-9...	-8.11678892387533e...	1518
21	21	2020-1...	-6.7545534822662e+18	1518
22	22	2023-1...	4.8723143208468e+18	1518
23	23	2022-3...	4.8485807133989e+18	2518
24	24	2021-6-4	-4.16558084088306e...	2518
25	25	2020-9...	5.72220248104425e+18	2518
26	26	2023-1...	-5.29998019807257e...	3518
27	27	2022-1-4	-6.43993998354066e...	3518
28	28	2021-3...	7.67181673447636e+18	3518
29	29	2020-6...	-2.73163523058277e...	5318
30	30	2023-9-1	5.71380418445808e+18	5318
31	31	2022-1...	6.67522603435993e+18	5318

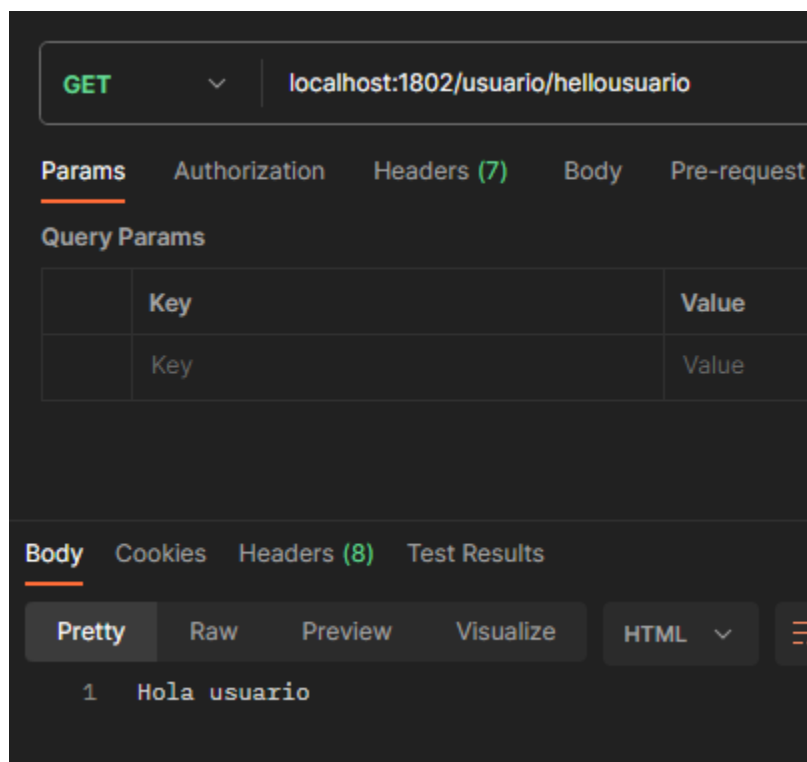
- *Implementar una funcionalidad que permita a los usuarios generar un archivo de Excel que contenga el resumen de todos los movimientos de su cuenta, con un diseño amigable para el usuario tipo extracto con los logos de la organización.*
- Como se menciona un usuario este tendrá una tabla donde, por el contexto del ejercicio, contendrá su cuenta y su cédula, donde la cédula es su llave primaria.

TABLA Usuario.

Identificación
1004
1083

- Creación de la app de usuario: `python manage.py startapp usuario`
- Se pondrá en las apps instaladas y se probará una ruta “Hello world” para probar si está bien configurando.

Bancolombia Prueba técnica Oct – 06.



- Modelo del usuario.

```

os > usuario > models.py > ...
from django.db import models

# Creacion del modelo usuario.
class Usuario(models.Model):
    identificacion = models.CharField(max_length=50, primary_key=True)

```

- NOTA: Dado que el corazón del ejercicio es funcionalidad de la generación del documento de extractos, no se creará un AbstractUser, sino un modelo sencillo, suficiente para la implementación de este caso.

Implementación.

1. Realización de las migraciones.
- `python manage.py makemigrations`

```

Migrations for 'usuario':
  usuario\migrations\0001_initial.py
    - Create model Usuario
Migrations for 'cuenta':
  cuenta\migrations\0001_initial.py
    - Create model Cuenta
Migrations for 'movimiento':
  movimiento\migrations\0001_initial.py
    - Create model Movimiento

```

- `Python manage.py migrate.`

Bancolombia Prueba técnica Oct – 06.

```
Applying usuario.0001_initial... OK
Applying cuenta.0001_initial... OK
Applying movimiento.0001_initial... OK
Applying transaccion.0001_initial... OK
```

- La conexión a la base de datos está dado por sqlite3. Que debe abrirse desde la app dbBrowser.
- En la sección de “*Script de creación de la base de datos*” se probarán los códigos de creación de tablas usando SQL en PostgreSQL.
- Tabla intersección de la relación de cuenta y transacción.

Table: movimiento_movimiento_r


id	movimiento_id	cuenta_id
Filter	Filter	Filter


2. Creación de datos.

- En la sección “*Ejecución de creación de datos en SQLite*” se crearon los datos de prueba para generar los documentos.

3. Creación del diseño del formato pdf.

- Se usarán extractos actuales para estudiar cómo es su diseño.





Producto protegido por
el Seguro de Depósitos
www.fogafin.gov.co

ESTADO DE CUENTA

DESDE: 2023/03/31 HASTA: 2023/06/30
CUENTA DE AHORROS

En Tu360Compras no tienes que esperar las rebajas para comprar.

RESUMEN

SALDO ANTERIOR		SALDO PROMEDIO	
TOTAL ABONOS		CUENTAS X COBRAR	
TOTAL CARGOS		VALOR INTERESES PAGADOS	
SALDO ACTUAL		RETEFUENTE	

FECHA	DESCRIPCIÓN	SUCURSAL	DCTO.	VALOR	SALDO
1/04					
2/04					
2/04					
3/04					
5/04					
6/04					
6/04					

Bancolombia Prueba técnica Oct – 06.

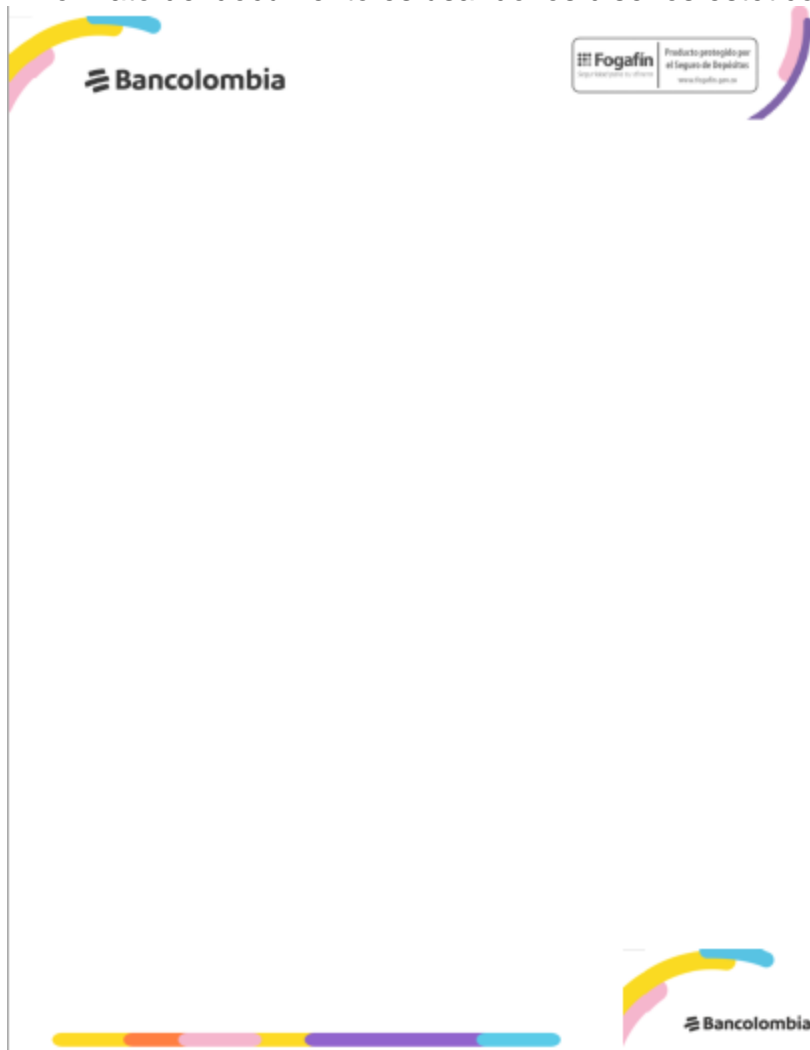
- Para los datos que poseemos, usaremos, la Fecha, en Descripción usaremos el tipo de cuenta, el valor y el saldo (de la tabla cuentas), inicialmente se mostrarán todos los movimientos realizados hasta el momento por la cuenta.
 - Se piensa hacer extractos por mes.
 - Se piensa hacer extractos por año.
- Fuente usada: Solicitud de documentos, Soluciones de movilidad privada (OpenSans-Regular).
- La tabla de los movimientos será de la forma:

FECHA	DESCRIPCION	VALOR	SALDO
YYYY-MM-DD	Retiro Bancario	-1231	2

NOTA: Ya que en los requisitos no se establece una descripción, la descripción será una lista aleatoria entre los valores “Retiro Bancario”, “Cuota de Manejo”, “Pago Agua”, “Retiro Cajero,”

NOTA: Si el extracto es por mes, la fecha solo mostrará DD, Si el extracto es por año, la fecha solo mostrará MM-DD.

- El formato del documento es usando los diseños estéticos de Bancolombia.



Bancolombia Prueba técnica Oct – 06.

4. Crear funcionalidades relacionadas al flujo básico del programa.

Función que regresa los movimientos dado la cuenta.

- En las vistas de la carpeta movimiento se crea el controlador que retorna todos los movimientos. Esta función es la primera iteración de este controlador, que luego, realizará lo mismo, pero recibiendo el dato por un formulario y no por un parámetro de la ruta.

```
#Obtener los movimientos dado una cuenta.
def get_movimientos_por_cuenta(request, n_cuenta):
    movimientos = Movimiento.objects.filter(n_cuenta=n_cuenta)
    return HttpResponse(movimientos)
```

NOTA: Se utiliza filter y no get, porque se retornarán muchos objetos.

- Se configura su respectivo urls.

```
urlpatterns = [
    path('movimiento/hellomov', test_hello_movement),
    path('movimiento/movimientosCuentas/<str:n_cuenta>', get_movimientos_por_cuenta, name="movimiento_cuenta"),
]
```

- Configuración de la creación del extracto: Cálculo del saldo.

NOTA: Dado que los registros fueron creados a partir de sentencias SQL, este saldo se le conocerá como el “saldo inicial” o “saldo anterior”.

Utilización de los estados de respuestas.

Respuestas informativas (100–199),

Respuestas satisfactorias (200–299),

Redirecciones (300–399),

Errores de los clientes (400–499),

y errores de los servidores (500–599). (<https://developer.mozilla.org/es/docs/Web/HTTP/Status>)

- Se creará un archivo de constante por app por si se crean valores a randomizar en la respuesta de la request.
- Hacer que la descripción sea random.

```
<td>{{ descripcion|random }}</td>
```

- NOTA: Para usar jinja, se deben tener la sintaxis estricta de usar los “=” pegadas.
- Creación de la tabla dado la cuenta ingresada.

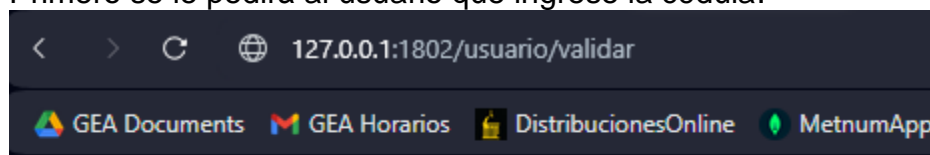
```
<table border="1">
    <tr>
        <th>FECHA</th>
        <th>DESCRIPCION</th>
        <th>VALOR</th>
        <th>SALDO</th>
    </tr>
    {% for movimiento in movimientos %}
    <tr>
        <td>{{ movimiento.fecha }}</td>
        <td>{{ descripcion|random }}</td>
        <td>{{ movimiento.valor }}</td>
        <td>{{ movimiento.saldo }}</td>
    </tr>
    {% endfor %}
```

FECHA	DESCRIPCION	VALOR	SALDO
Dec. 3, 2020	Cuota de Manejo	-57500.0	-57498.0
Dec. 3, 2020	Retiro Bancario	-57500.0	-114998.0
Sept. 25, 2021	Cuota de Manejo	-7500.0	-122498.0
Sept. 25, 2021		-7500.0	-129998.0
Jan. 17, 2023	Retiro Bancario	47500.0	-82498.0
Jan. 17, 2023	Retiro Bancario	47500.0	-34998.0

Secuencia (Flujo básico):

1. *Usuario ingresa cédula (Para que el sistema valide si tiene cuenta.).*

- Se crea un index que contendrán las paginas necesarias para descargar el extracto, este se usará basado en un marco html, que es la base de dicho index.
- Primero se le pedirá al usuario que ingrese la cédula.



Usuario

Ingrese su documento de identificacion:

Validate

1004353159

NOTA: Primero se concentrará el código en la realización del flujo básico (caso optimista), en caso de que falle algún envío de la entrada, la página lo envirá nuevamente a usuario/validar.

- Se creó una clase forms de django y se creó un validador que verifique la cédula exista en la base de datos.

```
<h1>Usuario</h1>
<form method="POST">
    {% csrf_token %}
    {{form.as_p}}

    <button>
        Validate
    </button>
</form >
```

```
#Ingresar la cédula.
def validar_usuario(request):
    if(request.method=='GET'):
        return render(request,
                        'validar_usuario.html',
                        {'form': LogUsuario()})
    else:
        try:
            user=Usuario.objects.get(identificacion=request.POST['identificacion'])
            return HttpResponse({"message OK": f"Bienvenido a Bancolombia con {user.identificacion}"})
        except:
            return redirect('validar_usuario')
```

```
path('usuario/validar', validar_usuario, name="validar_usuario")

<body>
  <nav>
    <ul>
      <li> <a href="/"> Home </a></li>
      <li> <a href="{% url 'validar_usuario' %}">Datos de Cliente Bancolombia</a></li>
    </ul>
  </nav>
```

- Lo anterior se hará por cada vez que se haga un formulario, es decir. Crear el formulario, crear el controlador relacionado al formulario, su respectiva ruta y añadirlo al home o al página donde tenga sentido.

2. Usuario solicita extracto (En este punto el usuario puede regresarse al home de la cuenta).

- En este apartado, se implementará el botón que genera el código dado la cuenta, esta es mandando con un parámetro en la solicitud (no es una buena practica ya que es algo sensible, a no ser que se cifre).

```
#Generar el extracto a partir de la cuenta, aca solo devuelve la tabla, no el archivo.
def generate_extracto(request, n_cuenta):
    if request.method=='GET':
        return render(request, 'generate_extracto.html')
    else:
        movimientos = Movimiento.objects.filter(n_cuenta=n_cuenta).order_by('fecha')

        #Calculo del saldo:
        inst_cuenta = Cuenta.objects.get(n_cuenta=n_cuenta)

        #Calcular saldos dado los movimientos.
        saldo_inicial=inst_cuenta.saldo

        list_movimientos_saldos=[]
        for movimiento in movimientos:
            #Saldo tras el movimiento
            saldo_inicial+=movimiento.valor

            list_movimientos_saldos.append({'fecha': movimiento.fecha,
                                           'valor': movimiento.valor,
                                           'saldo': saldo_inicial})
```

- Se cambió la ruta que había inicialmente.

```
path('movimiento/getExtracto/<str:n_cuenta>', generate_extracto, name="extracto_cuenta"),
```

- El formulario solo tiene un botón.

Bancolombia Prueba técnica Oct – 06.

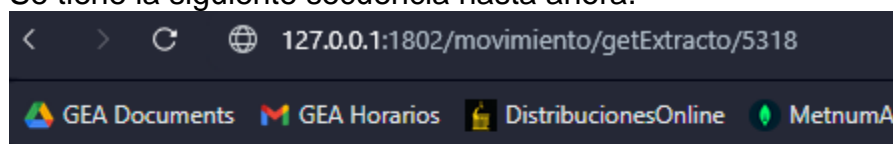
```

<h1>Generar Extracto</h1>
<form method="POST">
  {% csrf_token %}

  <button>
    Generar Extracto
  </button>
</form>

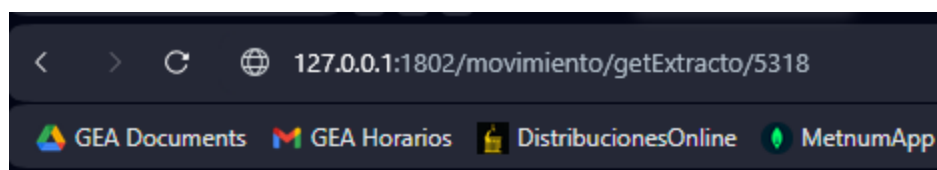
```

- Se tiene la siguiente secuencia hasta ahora.



Generar Extracto

Generar Extracto



FECHA	DESCRIPCION	VALOR	SALDO
June 26, 2020		18022022.0	18022032.0
June 26, 2020	Pago Agua	18022022.0	36044054.0
June 30, 2021	Retiro Bancario	30000.0	36074054.0
June 30, 2021		30000.0	36104054.0
Dec. 1, 2022	Pago Agua	455.0	36104509.0
Dec. 1, 2022	Retiro Cajero	455.0	36104964.0
March 15, 2022	Retiro Cajero	-40000.0	36064964.0
March 15, 2022	Pago Agua	-40000.0	36024964.0
Jan. 1, 2023	Retiro Bancario	50000.0	36074964.0
Jan. 1, 2023	Cuota de Manejo	50000.0	36124964.0
Sept. 1, 2023	Retiro Cajero	-123.0	36124841.0
Sept. 1, 2023	Pago Agua	-123.0	36124718.0

- Para generar el archivo pdf, se creará un script aparte para probar como unir la tabla del html en el archivo pdf,
- Usando PyPDF2 directamente, no funcionó.

Bancolombia Prueba técnica Oct – 06.

- Usando pdfkit, hay que instalar un programa aparte para pasar html para pdf.
 - Usando Reportlab, para pasar html para pdf.
- La solución definitiva está en la carpeta “funcion_def”, no considero que sea la más óptima (falta investigar más), pero considero que es aceptable dado las librerías que estaban disponibles.
 - NOTA: Como se observa anteriormente, se usará el HTML anterior para generar dicho pdf.
- El proceso general es, tomar la tabla que está en el documento HTML, convertirlo pdf usando un archivo auxiliar (que posteriormente es borrado) y luego unir el pdf que contiene la tabla junto con uno que tiene un formato estético en los márgenes con la marca “Bancolombia” (Usando imágenes tomado de la página y extractos reales).

1. Abrir el documento HTML.

```
# Lee el contenido de extracto.html
with open(html_content_path, "r", encoding="utf-8") as html_file:
    html_content = html_file.read()

html_content
```

2. Se utiliza BeautifulSoup para parsear el htm y busca la tabla del contenido.

```
# Se convierte en un objeto BeautifulSoup
soup = BeautifulSoup(html_content, 'html.parser')

# Tomar la tabla del contenido
table = soup.find('table')
```

3. Se toman los datos de la tabla y se guardan en una lista.

```
# pdf auxiliar para realizar la unión.
pdf_file = 'output.pdf'
doc = SimpleDocTemplate(pdf_file, pagesize=letter)

elements = []

# Tomar los datos de la tabla y guardarlos en una lista por filas
data = []
for row in table.find_all('tr'):
    data_row = []
    for cell in row.find_all(['th', 'td']):
        data_row.append(cell.get_text())
    data.append(data_row)

# Table es una clase de Reportlab.
pdf_table = Table(data)
```

4. Darle estilo a la tabla. NOTA: los comentarios o estilos puede ser que cambien, ya que en ese momento aun se revisan los mejores colores y fuentes.


```
# Table es una clase de Reportlab.
pdf_table = Table(data)

# Estilo de la tabla
pdf_table.setStyle(TableStyle([
    ('BACKGROUND', (0, 0), (-1, 0), (0.8, 0.8, 0.8)), # Fondo de la fila de encabezado
    ('TEXTCOLOR', (0, 0), (-1, 0), (1, 1, 1)), # Color del texto en la fila de encabezado
    ('ALIGN', (0, 0), (-1, -1), 'CENTER'), # Alineación del texto en el centro
    ('FONTNAME', (0, 0), (-1, 0), 'Times-Roman'), # Fuente en negrita para la fila de encab
    ('BOTTOMPADDING', (0, 0), (-1, 0), 12), # Espacio inferior en la fila de encabezado
    ('BACKGROUND', (0, 1), (-1, -1), (0.95, 0.95, 0.95)), # Fondo de las filas de datos
]))
```

5. Agregar la tabla a la lista de elementos y crear el documento auxiliar.

```
# Agregar la tabla al contenido del PDF
elements.append(pdf_table)

# Generar el PDF
doc.build(elements)
```

6. Tomar el archivo auxiliar, unir las paginas y se cifra con una contraseña que en la app web será el numero de identificación del usuario que será pasado como parámetro (no es buena práctica a no ser que se cifre).

```
with open(pdf_contenido, 'rb') as pdf_content_file, open(pdf_formato, 'rb') as pdf_style_f
    content_pdf = PyPDF2.PdfReader(pdf_content_file)
    style_pdf = PyPDF2.PdfReader(pdf_style_file)

# Crea un nuevo PDF como salida
result_pdf = PyPDF2.PdfWriter()

# Contraseña
identifiacion='1004'

# Combina el contenido del PDF de contenido con el formato del PDF de estilo
for page_num in range(len(content_pdf.pages)):
    content_page = content_pdf.pages[page_num]
    style_page = style_pdf.pages[page_num]

    # Copia el contenido de la página de estilo (márgenes) a la página de contenido
    content_page.merge_page(style_page)

    # Agrega la página resultante al PDF de resultado
    result_pdf.add_page(content_page)

# Contraseña
result_pdf.encrypt(identifiacion)
```

```
# Guarda el PDF resultante en un nuevo archivo
output_pdf_path = os.path.join(os.path.expanduser("~"), f"Downloads\\contenido_con_formato1.p
with open(output_pdf_path, 'wb') as output_pdf_file:
    result_pdf.write(output_pdf_file)

output_pdf_file.close()
```

NOTA: Mandar el archivo a descargar, pero el nombre del archivo será la identificación con el número de cuenta.

7. Eliminar el archivo auxiliar (No es buena práctica).

```
# Ruta del archivo que deseas eliminar
archivo_a_eliminar = 'output.pdf'

# Verificar si el archivo existe
if os.path.exists(archivo_a_eliminar):

    #Cerrar el archivo auxiliar
    pdf_content_file.close()

    # Eliminar el archivo
    os.remove(archivo_a_eliminar)
else:
    print("Archivo no encontrado")
    print(f"Intenta de nuevo: {archivo_a_eliminar}")
```

NOTA: Este código está en la parte de pruebas, hay elementos que no son buenas prácticas, pero se usan por efectos del funcionamiento del ejercicio.

- NOTA: Las rutas en el código del django, así esté en la carpeta movimiento/views.py. Realmente a la hora de buscar archivos lo busca en carpeta_proyecto/, en este caso el proyecto se llama extractos: `pdf_formato = './static/Formato-Doc-Bancolombia.pdf'`.

Static está en la carpeta extractos, no movimientos.

3. Sistema crea extracto.

- a. Sistema selecciona todos los movimientos de esa cuenta y los ordena por fecha.
- b. Sistema crea el pdf con formato Bancolombia.
- c. Sistema cifra el archivo con una contraseña que será su cédula.
- d. Sistema descarga el archivo automáticamente, indicando que la contraseña es su número de documento.
- e. Sistema regresa al usuario a la página para generar extracto.

Posibles mensajes (Flujos alternativos).

1. Al acceder a alguna ruta que no sea:
 - /cuenta/hellocuenta: Ruta usada para probar si se conectó la app con el proyecto.
 - /movimiento/hellomov: Ruta usada para probar si se conectó la app con el proyecto.
 - /movimiento/movimientosCuentas/n_cuenta:

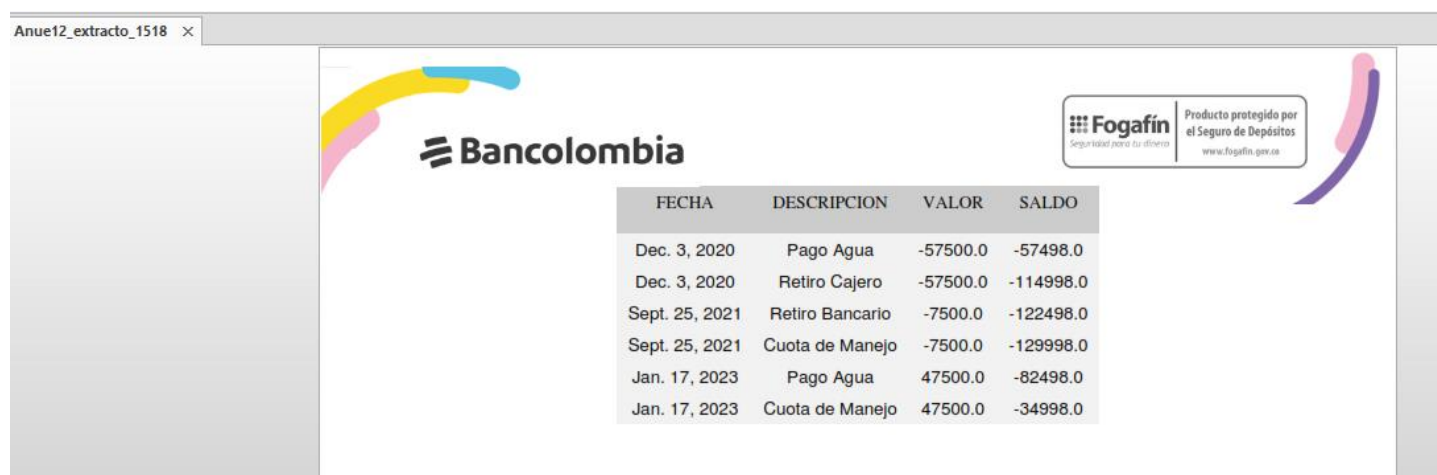
Bancolombia Prueba técnica Oct – 06.

- Dado que solo se usa para revisar si se están guardando los registros, en la página no mostrará nada o mostrarán los objetos de tipo movimiento.
- Ejemplos:
 - /movimiento/movimientosCuentas/1004
 - /movimiento/movimientosCuentas/1005
 - ...
- /movimiento/getExtracto/n_cuenta
 - En esta ruta se llega al botón que genera el extracto con todos los movimientos de la cuenta asociada a n_cuenta.
 - Ejemplos:
 - /movimiento/getExtracto/5318
 - /movimiento/getExtracto/6518
 - /movimiento/getExtracto/8518
 - /movimiento/getExtracto/9518
 - ...
- /: Ruta home para ir hacia generar el extracto (Esto es por si hace parte del menú).
- /usuario/hellousuario: Ruta usada para probar si se conectó la app con el proyecto.
- /usuario/validar: Se ingresará la cédula para extraer los extractos asociados a su cuenta.

- *El Archivo Excel generado debe estar bloqueado para modificaciones.*

Cómo el documento es un pdf, este está bloqueado para modificaciones.

Anue12_extracto_1518 x



FECHA	DESCRIPCION	VALOR	SALDO
Dec. 3, 2020	Pago Agua	-57500.0	-57498.0
Dec. 3, 2020	Retiro Cajero	-57500.0	-114998.0
Sept. 25, 2021	Retiro Bancario	-7500.0	-122498.0
Sept. 25, 2021	Cuota de Manejo	-7500.0	-129998.0
Jan. 17, 2023	Pago Agua	47500.0	-82498.0
Jan. 17, 2023	Cuota de Manejo	47500.0	-34998.0

- *El Archivo Excel debe tener contraseña de apertura*

- En este apartado se encripta el documento.

Bancolombia Prueba técnica Oct – 06.

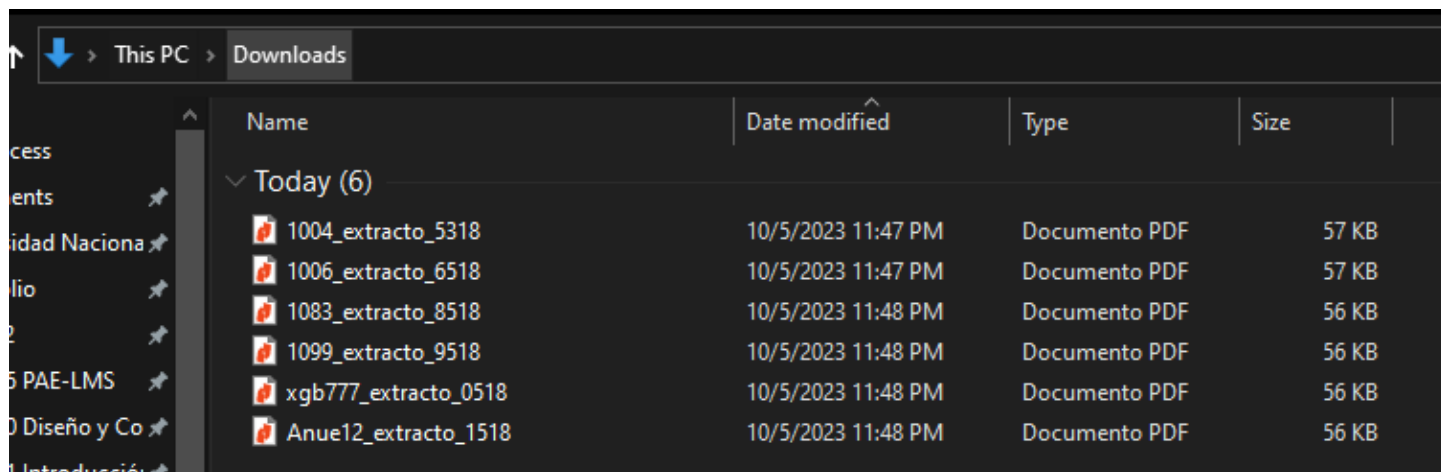
```
# Combina el contenido del PDF de contenido con el formato del PDF de estilo
for page_num in range(len(content_pdf.pages)):
    content_page = content_pdf.pages[page_num]
    style_page = style_pdf.pages[page_num]

    # Copia el contenido de la página de estilo (márgenes) a la página de contenido
    content_page.merge_page(style_page)

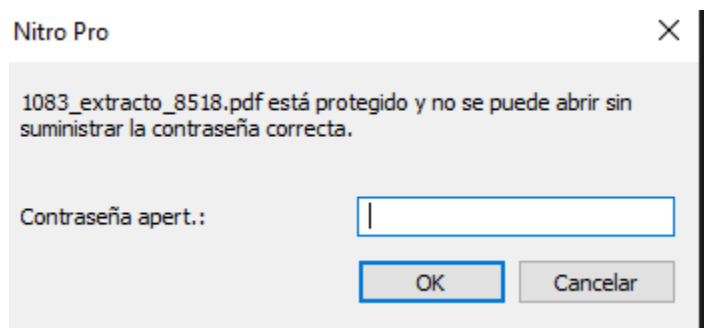
    # Agrega la página resultante al PDF de resultado
    result_pdf.add_page(content_page)

# Contraseña
result_pdf.encrypt(identifiacion)
```

- Se deberían mandar los extractos a la carpeta descargas.



Name	Date modified	Type	Size
Today (6)			
1004_extracto_5318	10/5/2023 11:47 PM	Documento PDF	57 KB
1006_extracto_6518	10/5/2023 11:47 PM	Documento PDF	57 KB
1083_extracto_8518	10/5/2023 11:48 PM	Documento PDF	56 KB
1099_extracto_9518	10/5/2023 11:48 PM	Documento PDF	56 KB
xgb777_extracto_0518	10/5/2023 11:48 PM	Documento PDF	56 KB
Anue12_extracto_1518	10/5/2023 11:48 PM	Documento PDF	56 KB



Nitro Pro

1083_extracto_8518.pdf está protegido y no se puede abrir sin suministrar la contraseña correcta.

Contraseña apert.:

NOTA: Recordar que la contraseña es el número de identificación.

- Usuarios con cuenta.

Bancolombia Prueba técnica Oct – 06.

n_cuenta	tipo_cuenta	saldo	titular_id
Filter	Filter	Filter	Filter
5318	AHORROS	10.0	1004
6518	AHORROS	-10.0	1006
8518	AHORROS	0.0	1083
9518	CORRIENTE	10.0	1099
0518	CORRIENTE	10.0	xgb777
1518		2.0	Anue12
2518		10.0	0462
3518	CORRIENTE	9.0	0910

Entregable.

Script de creación de la base de datos

Copiar los códigos tal cuál se envían.

NOTA IMPORTANTÍSIMA: Django crea la eliminación en cascada no más para su ORM pero no se define en la migración de la tabla, por ende, toca establecerla uno mismo desde el gestor de la base de datos (<https://stackoverflow.com/questions/54751466/django-does-not-honor-on-delete-cascade>).

Tabla usuario_usuario.

```
CREATE TABLE "usuario_usuario" (
    "identificacion"      varchar(50) NOT NULL,
    PRIMARY KEY("identificacion")
);
```

Tabla cuenta_cuenta.

```
CREATE TABLE "cuenta_cuenta" (
    "n_cuenta"    varchar(50) NOT NULL,
    "tipo_cuenta" varchar(20) NOT NULL,
    "saldo"       real NOT NULL,
    "titular_id"  varchar(50) NOT NULL UNIQUE,
    FOREIGN KEY("titular_id") REFERENCES "usuario_usuario"("identificacion")
                                ON DELETE CASCADE
    DEFERRABLE INITIALLY DEFERRED,
    PRIMARY KEY("n_cuenta")
);
```

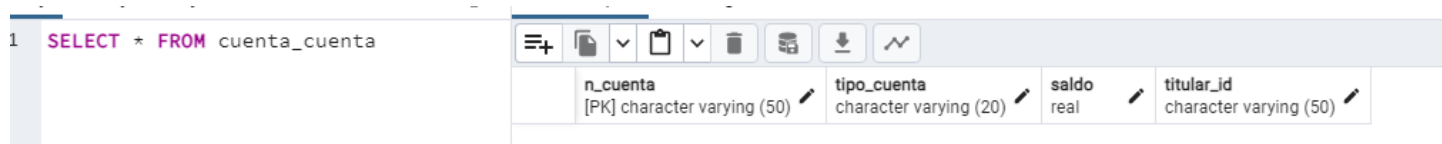


Tabla movimiento movimiento.

En postgres.

```
CREATE TABLE "movimiento_movimiento" (  
    "id" SERIAL NOT NULL,  
    "fecha" date NOT NULL,  
    "valor" real NOT NULL,  
    "n_cuenta_id" varchar(50) NOT NULL,  
    PRIMARY KEY("id"),  
    FOREIGN KEY("n_cuenta_id")  
        REFERENCES "cuenta_cuenta"("n_cuenta")  
        ON DELETE CASCADE  
        DEFERRABLE INITIALLY DEFERRED  
);
```

En SQLite.

```
CREATE TABLE "movimiento_movimiento" (  
    "id"    integer NOT NULL,  
    "fecha"    date NOT NULL,  
    "valor" real NOT NULL,  
    "n_cuenta_id"    varchar(50) NOT NULL,  
    PRIMARY KEY("id" AUTOINCREMENT),  
    FOREIGN KEY("n_cuenta_id")  
REFERENCES "cuenta_cuenta"("n_cuenta")  
    ON DELETE CASCADE  
DEFERRABLE INITIALLY DEFERRED  
);
```

Bancolombia Prueba técnica Oct – 06.

Script de inserción de la data dummy

Insertar datos de usuario_usuario.

```
INSERT INTO "usuario_usuario" (identificacion)
```

```
VALUES ('1004'),
       ('1005'),
       ('1006'),
       ('1083'),
       ('1099'),
       ('xgb777'),
       ('Anue12'),
       ('0462'),
       ('0910'),
       ('al18');
```

```
SELECT * from usuario_usuario|
```

	identificacion [PK] character varying (50)
1	1004
2	1005
3	1006
4	1083
5	1099
6	xgb777
7	Anue12
8	0462
9	0910
10	al18

Insertar datos en cuenta_cuenta.

- Valores en comilla simple.

```
INSERT INTO cuenta_cuenta ("n_cuenta","tipo_cuei
VALUES ('5318','AHORROS',10,'1004')
```

```
INSERT INTO cuenta_cuenta ("n_cuenta","tipo_cuenta","saldo",titular_id)
VALUES ('5318','AHORROS',10,'1004'),
```

Bancolombia Prueba técnica Oct – 06.

```
(('6518','AHORROS',-10,'1006'),
 ('8518','AHORROS',0,'1083'),
 ('9518','CORRIENTE',10,'1099'),
 ('0518','CORRIENTE',10,'xgb777'),
 ('1518','',2,'Anue12'),
 ('2518','',10,'0462'),
 ('3518','CORRIENTE',9,'0910'));
```

```
SELECT * from cuenta_cuenta
```

	n_cuenta [PK] character varying (50)	tipo_cuenta character varying (20)	saldo real	titular_id character varying (50)
1	5318	AHORROS	10	1004
2	6418	AHORROS	-10	1005
3	6518	AHORROS	-10	1006
4	8518	AHORROS	0	1083
5	9518	CORRIENTE	10	1099
6	0518	CORRIENTE	10	xgb777
7	1518		2	Anue12
8	2518		10	0462
9	3518	CORRIENTE	9	0910

Insertar datos de movimientos.

- Se van a crear 30 registros para intentar cubrir la mayoría de casos posibles y crear el extracto con información suficiente para ilustrar lo pedido en el enunciado.
- Para el valor, se utilizarán números aleatorios entre menos un millón y diez millones [-1'000.000,10'000.000].

En SQLite.

- Se tomarán valores entre [-50000,2000000] escogidos arbitrariamente.

```
INSERT INTO movimiento_movimiento ("fecha","valor","n_cuenta_id")
```

```
VALUES ('2023-1-1',50000,'5318'),
 ('2022-3-15',-40000,'5318'),
 ('2021-6-30',30000,'5318'),
 ('2020-9-1',-20000,'6518'),
 ('2023-12-15',10000,'6518'),
 ('2022-1-30',-50,'6518'),
 ('2021-3-1',-55000,'6518'),
 ('2020-6-15',45000,'6518'),
 ('2023-9-1',-35000,'6518'),
 ('2022-12-30',25000,'8518'),
```


Bancolombia Prueba técnica Oct – 06.

```
('2021-1-2',-15000,'8518'),  
('2020-3-14',50000,'8518'),  
('2023-6-29',-2500,'9518'),  
('2022-9-2',52500,'9518'),  
('2021-12-14',-42500,'9518'),  
('2020-1-29',32500,'0518'),  
('2023-3-3',-22500,'0518'),  
('2022-6-17',12500,'0518'),  
('2021-9-25',-7500,'1518'),  
('2020-12-3',-57500,'1518'),  
('2023-1-17',47500,'1518'),  
('2022-3-25',-37500,'2518'),  
('2021-6-4',27500,'2518'),  
('2020-9-18',-17500,'2518'),  
('2023-12-26',18500,'3518'),  
('2022-1-4',-1852,'3518'),  
('2021-3-18',1802,'3518'),  
('2020-6-26',18022022,'5318'),  
('2023-9-1',-123,'5318'),  
('2022-12-1',455,'5318');
```

NOTA: Todos los datos que se creen en n_cuenta_id deben estar creados previamente, sino nos se violan restricciones de las claves foráneas.

En postgres.

```
INSERT INTO movimiento_movimiento ("fecha","valor","n_cuenta_id")
```

```
VALUES ('2023-1-1',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'5318'),  
('2022-3-15',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'5318'),  
('2021-6-30',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'5318'),  
('2020-9-1',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'6418'),  
('2023-12-15',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'6418'),  
('2022-1-30',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'6418'),
```

Bancolombia Prueba técnica Oct – 06.

('2021-3-1',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'6518'),
('2020-6-15',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'6518'),
('2023-9-1',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'6518'),
('2022-12-30',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'8518'),
('2021-1-2',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'8518'),
('2020-3-14',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'8518'),
('2023-6-29',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'9518'),
('2022-9-2',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'9518'),
('2021-12-14',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'9518'),
('2020-1-29',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'0518'),
('2023-3-3',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'0518'),
('2022-6-17',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'0518'),
('2021-9-25',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'1518'),
('2020-12-3',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'1518'),
('2023-1-17',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'1518'),
('2022-3-25',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'2518'),
('2021-6-4',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'2518'),
('2020-9-18',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'2518'),
('2023-12-26',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'3518'),
('2022-1-4',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'3518'),
('2021-3-18',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'3518'),
('2020-6-26',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'5318'),
('2023-9-1',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'5318'),
('2022-12-1',FLOOR(RANDOM()*(10000000-(-100000)+1))+(-1000000),'5318');

EXTRA:

Al clonar el repositorio (Con git o github).

1. pip install poetry
2. Cd nombre_repositorio
3. Poetry Shell
4. Poetry install.
5. Cd extractos.

Bancolombia Prueba técnica Oct – 06.

6. Python manage.py runserver. NOTA: Por defecto es el puerto 8000.

Estilos.

- La carpeta static debe tener '/' para que enlace con la carpeta estilos.

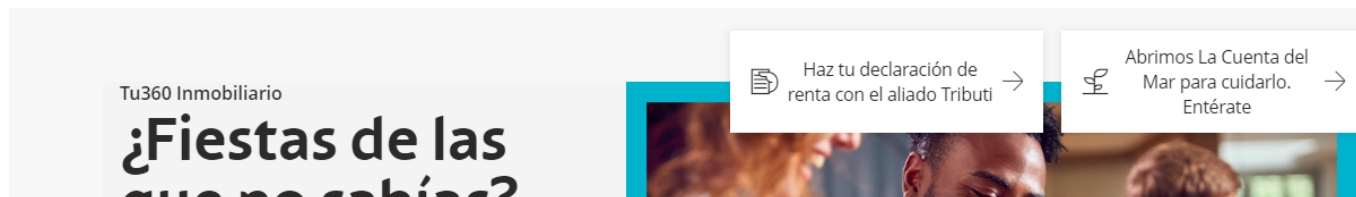
```

s.py 1 ● # s1.css <> base.html settings.py X
tos > settings.py > ...
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.2/howto/static-files/

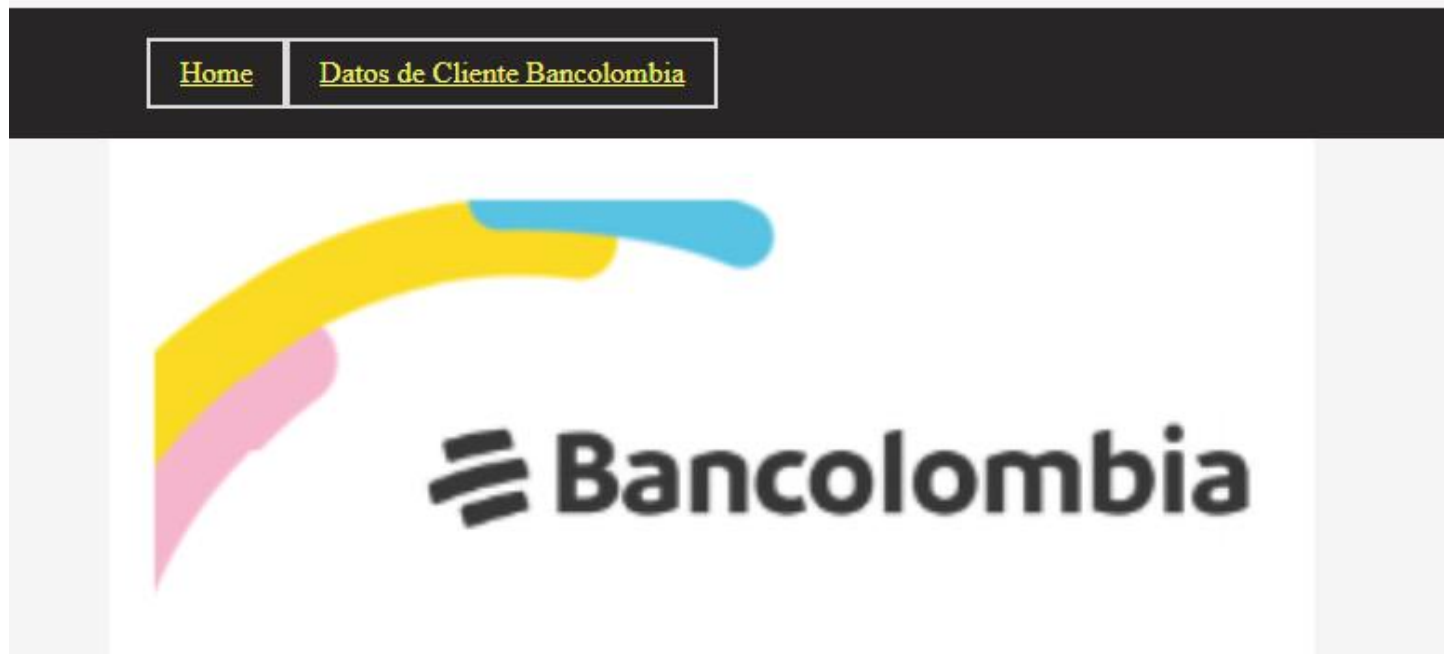
STATIC_URL = '/static/'


```

- Se observa la página de Bancolombia para obtener estilos “parecidos” aunque solo es para dare un poco de estilo, no se hará la página de manera exacta.



- Observamos un navbar gris oscuro, letras blancas en él, un fondo blanco o whitesmoke y botones con curva redondeada amarillos y letras negras.
- Así se estuvieron las siguientes vistas.






Usuario

[Home](#)

Ingrese su documento de identificacion:

[Validate](#)




Generar Extracto

[Home](#) [Validar Usuario](#)

Identificacion válida.

[Generar Extracto](#)

- Al generar un archivo correctamente:



Usuario

[Home](#)

EXTRACTO generado en la carpeta Descargas o Downloads:
C:\Users\denil\Downloads\1004_extracto_5318.pdf.

RECUERDE: Su contraseña es su numero de identificacion:
1004

Ingrese su documento de identificacion:

[Validate](#)

Tener en cuenta:

- Considere la experiencia del usuario al desarrollar el formulario y las interacciones con la plataforma.
- Puede utilizar el lenguaje que considere más conveniente para las implementaciones.