


SISTEMA DE CONSULTAS



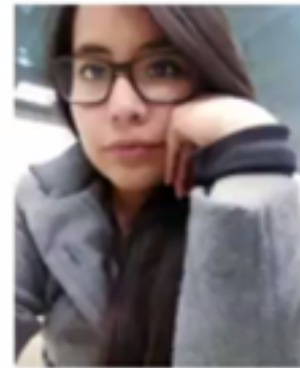
INTEGRANTES:

Equipo 6

Clic en Foto
para ver CV



Nombre: Alberto Pérez Rodríguez
Carrera: Ing. Industrial
Universidad: Instituto Tecnológico de Orizaba
Tecnologías: MySQL, Python



Nombre: Edith Alejandra Trejo Méndez
Carrera: Ing. En Informática
Universidad: IPN
Tecnologías: HTML5, CSS, Java



Nombre: José Luis Velázquez Rodríguez
Carrera: Ciencias de la Computación
Universidad: IPN
Tecnologías: HTML5, CSS, Java, Java Script, Python



Nombre: Karen del Carmen Cortés Morales
Carrera: Ing. Geofísica
Universidad: UNAM
Tecnologías: Python



Nombre: Luis Humberto Miranda Galindo
Carrera: Ing. Mecatrónica
Universidad: Instituto Tecnológico Superior
Tecnologías: HTML5, CSS, Java, Java Script, Python



Nombre: Orlando Bello Franquiz
Carrera: Ciencias de la Computación
Universidad: BUAP
Tecnologías: Java, Python



Nombre: Ricardo de Jesús Serra Silva
Carrera: Ing. En Mecatrónica
Universidad: TecMilenio
Tecnologías: HTML5, CSS, Java



Nombre: Jesús Alejandro Arzate Gómez
Carrera: Ing. Mecatrónica
Universidad: UNAM
Tecnologías: Java, Python



Requerido:

- Obtener informacion de un cliente por ID
- Solo Asesores registrados podran realizar consulta

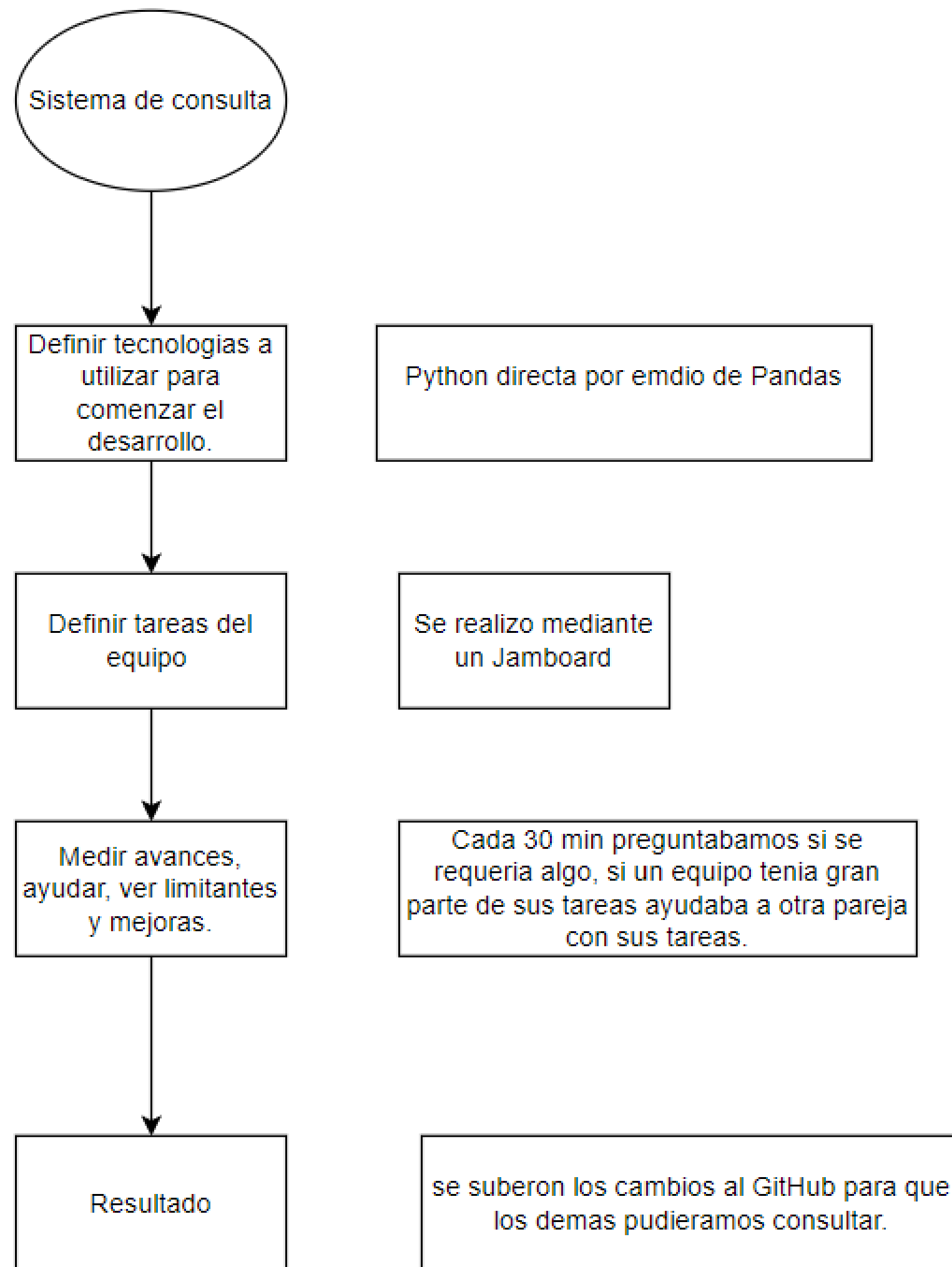
Definicion de herramientas

JamBoard (Online) Organizacion

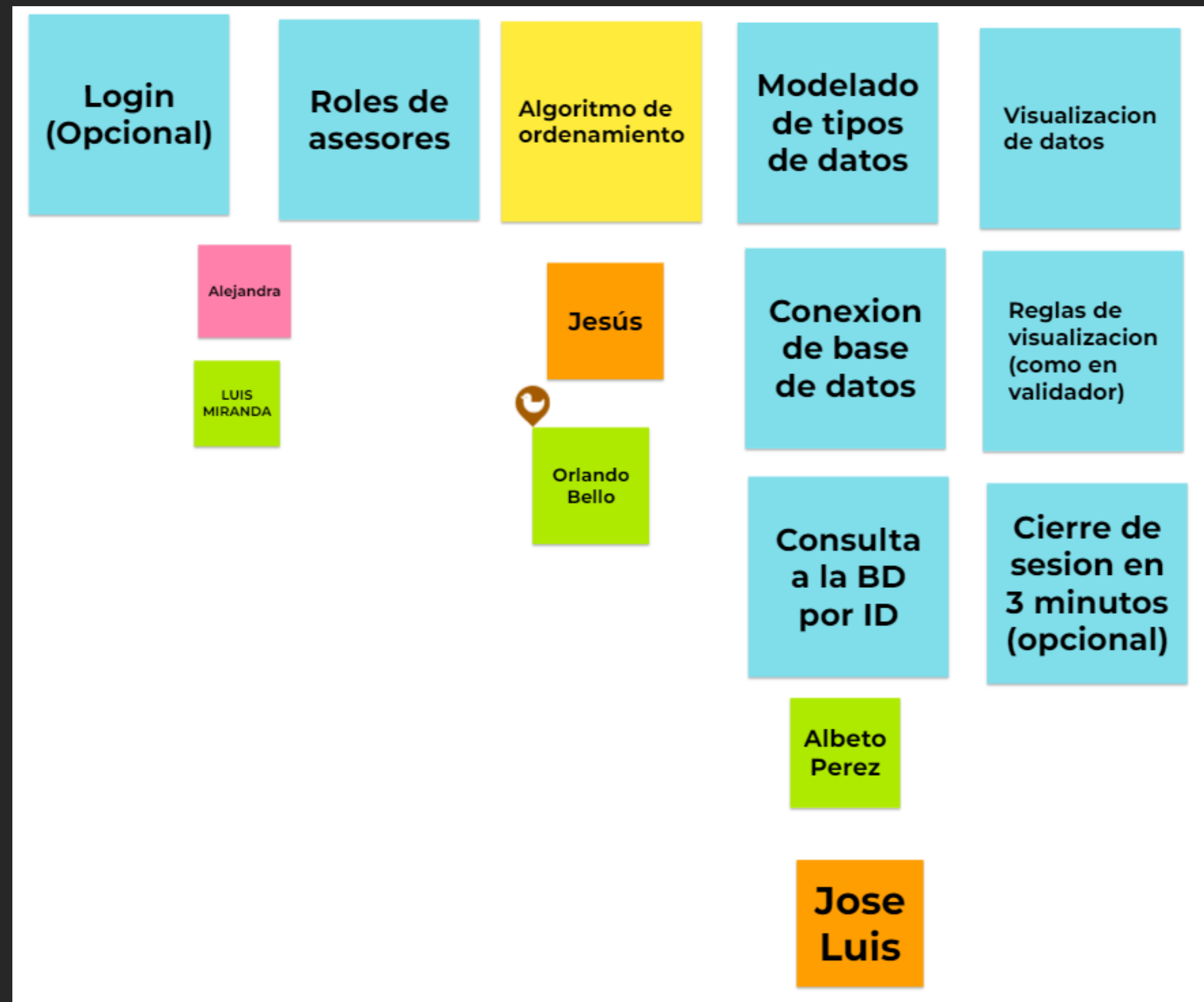
Lenguaje seleccionado Python (Django)
(Pandas)

Lenguaje seleccionado Python (Tkinter)

GitHub



Propuesta del equipo:



¡Logrado!



Login

```
Documents\BBVA\Equipob-BIT-jul2022\login.py
```

```
43:29
```

```
ingrese su usuario
```

```
█
```

```
ingrese su usuario
```

```
AndArt
```

```
ingrese su contraseña
```

```
qwerty█
```

```
Documents\BBVA\Equipob-BIT-jul2022\login.py
```

```
.5 Ingrese el id del cliente
```

```
.4 BF000002999
```

```
.3 idCliente: BF000002999
```

```
.2 nombre: Amelia
```

```
.1 apellidoPaterno: Rivero
```

```
.0 fechaNacimiento: 25/04/54
```

```
9 sexo: Femenino
```

```
8 segmento: 1.0
```

```
7 nacionalidad: Mexicana
```

```
6 rfc: RIC
```

```
5 tipoID: Lic
```

```
4 numeroID: Rpi
```

```
3 cuenta: BE12 7743 3050 0148 8965 5830
```

```
2 email: ame
```

```
1
```

```
[Process exited 0]█
```

```
1
```

```
2
```



```
1  usuarios = ['Luis', 'Alejandra']
2  path = ['1234', '5678']
3
4  print("ingrese su usuario")
5  usuario=(str(input()))
6  print("ingrese su contraseña")
7  auth=(str(input()))
8  print(usuario + " " + auth )
9  n=0
10 while 1==1:
11     if usuario != usuarios[n]:
12         n=n+1
13     else:
14         break
15
16
17
18
19
```

Login

Diseño



Inicio de sesion

— □ ×

SISTEMA DE CONSULTA

INICIO DE SESION

Usuario

Contraseña

Iniciar sesion

[Registrar usuario](#)

Roles y asesores

```
18 if s.iloc[0] == auth:
17     print("Bienvenido " + u.nombreCompleto)
16 else:
15     print("Usuario o contraseña incorrectos")
14 os.system("cls")
13 print("Ingrese el id del cliente")
12 idCliente=(str(input()))
11 c = conexion.getCliente(idCliente)
10 if u.perfil.astype("string").iloc[0] == "Manager":
9     print(c)
8 elif u.perfil.astype("string").iloc[0] == "Validador":
7     print("idCliente: " + c.idCliente.astype("string").iloc[0])
6     print("nombre: " + c.nombre.astype("string").iloc[0])
5     print("apellidoPaterno: " + c.apellidoPaterno.astype("string").iloc[0][0:3] + "***")
4     print("apellidoMaterno: " + c.apellidoMaterno.astype("string").iloc[0][0:3] + "***")
3     print("fechaNacimiento: " + c.fechaNacimiento.astype("string").iloc[0][0:3] + "***")
2     print("sexo: " + c.sexo.astype("string").iloc[0])
1     print("segmento: " + c.segmento.astype("string").iloc[0])
1     print("nacionalidad: " + c.nacionalidad.astype("string").iloc[0][0:3] + "***")
1     print("rfc: " + c.rfc.astype("string").iloc[0][0:3] + "***")
2     print("tipoID: " + c.tipoID.astype("string").iloc[0][0:3] + "***")
3     print("numeroID: " + c.numeroID.astype("string").iloc[0][0:3] + "***")
4     print("cuenta: " + c.cuenta.astype("string").iloc[0])
5     print("email: " + c.email.astype("string").iloc[0][0:3] + "***")
6     print("TDD: " + c.TDD.astype("string").iloc[0][0:3] + "***")
7
```

Algoritmo de ordenamiento

ordenamiento_pandas.py > ...

```
1 import pandas as pandasForSortingCSV
2
3 csvData = pandasForSortingCSV.read_csv("baseClientesHackaton2022.csv")
4
5 csvData.sort_values(["idCliente"],
6                     axis=0,
7                     ascending=[True],
8                     inplace=True)
9
10 print(csvData)
```

Modelado de tipos de datos

bbva > clientesAsesores > migrations > 0001_initial.py > ...

```
1  # Generated by Django 4.0.6 on 2022-07-14 14:20
2
3  from django.db import migrations, models
4
5
6  class Migration(migrations.Migration):
7
8      initial = True
9
10     dependencies = [
11     ]
12
13     operations = [
14         migrations.CreateModel(
15             name='Asesores',
16             fields=[
17                 ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
18                 ('usuario', models.CharField(max_length=50)),
19                 ('auth', models.CharField(max_length=50)),
20                 ('nombre', models.CharField(max_length=100)),
21                 ('area', models.CharField(max_length=50)),
22                 ('segmento', models.IntegerField()),
23                 ('perfil', models.CharField(max_length=50)),
24             ],
25         ),
26         migrations.CreateModel(
27             name='Clientes',
28             fields=[
29                 ('idCliente', models.CharField(max_length=11, primary_key=True, serialize=False)),
30                 ('nombre', models.CharField(max_length=50)),
31                 ('apellidoPaterno', models.CharField(max_length=50)),
32                 ('apellidoMaterno', models.CharField(max_length=50)),
33                 ('fechaNacimiento', models.DateField()),
34                 ('sexo', models.CharField(max_length=20)),
35                 ('segmento', models.IntegerField()),
36                 ('nacionalidad', models.CharField(max_length=20)),
37                 ('rfc', models.CharField(max_length=13)),
38                 ('tipoID', models.CharField(max_length=50)),
39                 ('numeroID', models.CharField(max_length=50)),
40                 ('cuenta', models.CharField(max_length=50)),
41                 ('email', models.CharField(max_length=50)),
42                 ('TDD', models.CharField(max_length=50)),
43             ],
44         ),
45     ]
46
```

Conexion base de datos

```
0 import pandas as pd
9
8 def getCliente(idCliente):
7     clientes = pd.read_csv('baseClientesHackaton2022.csv')
6     return clientes[clientes.idCliente==idCliente]  Cannot access member "idCliente" for t
5
4 def getUsuario(usuario):
3     usuarios = pd.read_csv('baseUsuarios.csv')
2     return usuarios[usuarios.usuario==usuario]  Cannot access member "usuario" for type "T"
1
print(getUsuario("pacoG"))
1 print(getCliente("BF000002999"))
```

Consulta de BD

Ingrese el id del cliente

BF000002999

idCliente: BF000002999

nombre: Amelia

apellidoPaterno: Riv***

apellidoMaterno: Cal***

fechaNacimiento: 25/***

sexo: Femenino

segmento: 1.0

nacionalidad: Mex***

rfc: RIC***

tipoID: Lic***

numeroID: Rpi***

cuenta: BE12 7743 3050 0148 8965 5830

email: ame***

 pwsh ~\Documents\BBVA\Equipo6-BIT-jul2022  main  36s 583ms

 14,11:40