

Práctica Módulo II



Universidad
Rey Juan Carlos

Sergio Ayuso Hernandez - 05955900G

Mohamad Sinan Omar Dallal - 51128196R

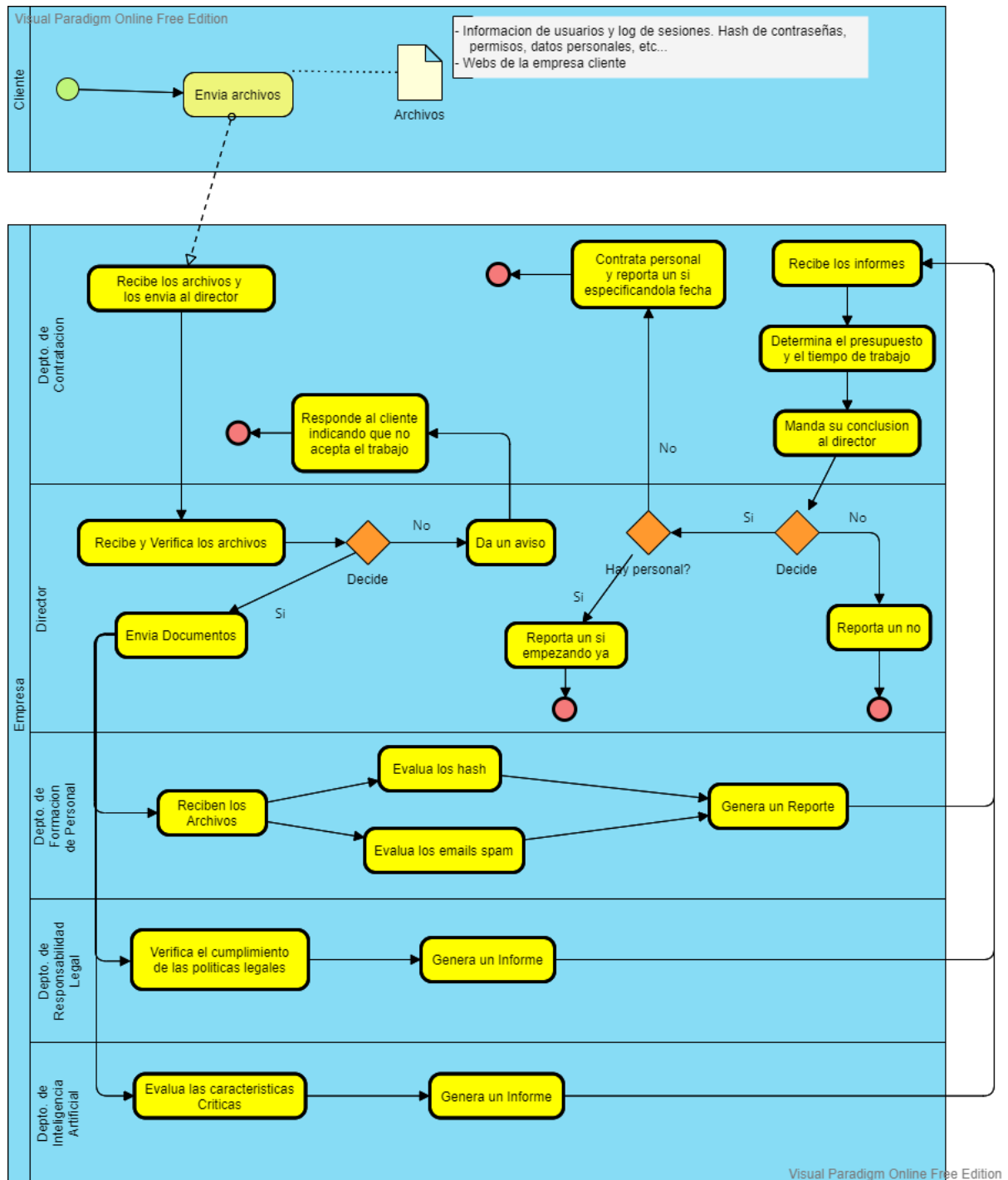
2021-2022

Sistemas de Información

Ingeniería de la Ciberseguridad

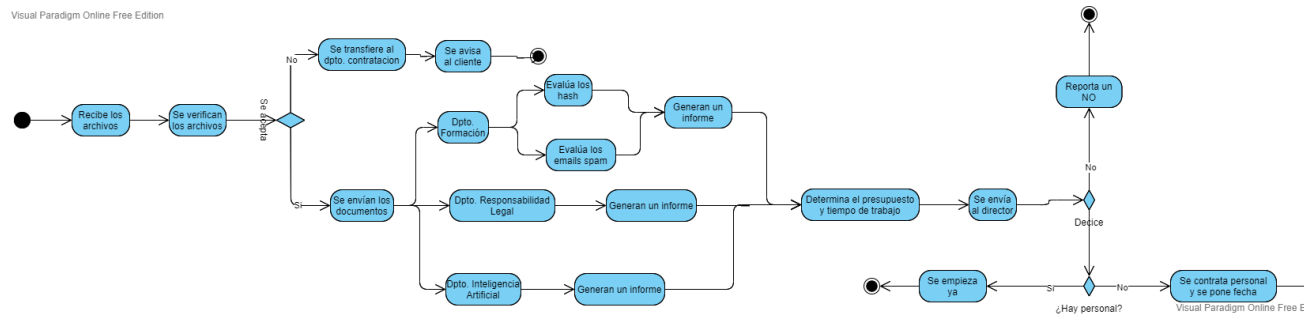
EJERCICIO 1

- BPMN



- UML

Visual Paradigm Online Free Edition



EJERCICIO 2

- Creamos las Tablas

```
CREATE TABLE LEGAL (  
url varchar(25) PRIMARY KEY,  
cookies boolean,  
aviso boolean,  
proteccionDatos boolean,  
creacion numeric(4,0)  
);
```

```
CREATE TABLE USERS (  
    nombre varchar(25) PRIMARY KEY,  
    telefono varchar(9),  
    contrasena varchar(32),  
    provincia varchar(20),  
    permisos boolean,  
    emailsTotal numeric,  
    emailsPhising numeric,  
    emailsCiclados numeric  
);
```

```
CREATE TABLE FECHAS_USER (  
    rtb_user varchar(25),  
    fecha date,  
    FOREIGN KEY (rtb_user) REFERENCES USERS (nombre)  
);
```

```
CREATE TABLE IPS_USER (  
    rtb_user varchar(25),  
    ip varchar(15),  
    FOREIGN KEY (rtb_user) REFERENCES USERS (nombre)  
);
```

- Introducimos los datos JSON en las tablas

Ver fichero main.py

- Creamos el DataFrame

Se usa en cada ejercicio (en cada archivo nuevo).

Véase ejercicio2.py

Se realizan distintos DataFrames en función de lo solicitado en el ejercicio

```
import sqlite3;
import pandas as pd;

con = sqlite3.connect('SISTINF.db');

query = con.execute("SELECT * From legal")
cols = [column[0] for column in query.description]
dLegal = pd.DataFrame.from_records(data = query.fetchall(), columns = cols)

query = con.execute("SELECT * From users")
cols = [column[0] for column in query.description]
dUsers = pd.DataFrame.from_records(data = query.fetchall(), columns = cols)

query = con.execute("select count(fecha) as numAccesos from fechas_user group by
rtb_user;")
cols = [column[0] for column in query.description]
dFechas_User = pd.DataFrame.from_records(data = query.fetchall(), columns = cols)

query = con.execute("SELECT count(ip) as numAccesos From ips_user group by rtb_user")
cols = [column[0] for column in query.description]
dIPS_User = pd.DataFrame.from_records(data = query.fetchall(), columns = cols)

con.close()
```

- **Número de Muestras**
dUsers['emailsPhising'].count()
- **Media y desviación estándar del total de fechas que se ha iniciado sesión**
Media: dFechas_User['numAccesos'].mean()
Desviación: dFechas_User['numAccesos'].std()
- **Media y desviación estándar del total de IPs que se han detectado**
Media: dIPS_User['numAccesos'].mean()
Desviación: dIPS_User['numAccesos'].std()
- **Media y desviación estándar del número de emails recibidos**
Media: dUsers['emailsPhising'].mean()
Desviación: dUsers['emailsPhising'].std()

- **Valor mínimo y valor máximo del total de fechas que se ha iniciado sesión**

Max: dFechas_User['numAccesos'].max()

Min: dFechas_User['numAccesos'].min()

- **Valor mínimo y valor máximo del número de emails recibidos**

Max: dUsers['emailsTotal'].max()

Min: dUsers['emailsTotal'].min()

The screenshot shows a Python IDE with a project named 'pythonProject'. The file explorer on the left shows a directory structure with files like 'ejercicio2.py', 'ejercicio3.py', 'ejercicio4.py', 'main.py', 'legal.json', 'SISTINE.db', and 'users.json'. The editor displays the code for 'ejercicio2.py'.

```
25
26 con.close()
27
28
29 print("La media de correos recibidos son:", dUsers['emailsPhising'].mean())
30 print("La desviacion estándar de correos recibidos son:", dUsers['emailsPhising'].std())
31 print("El valor máximo de correos recibidos es:", dUsers['emailsTotal'].max())
32 print("La valor mínimo de correos recibidos es:", dUsers['emailsTotal'].min())
33
34
35 print("El valor máximo de las fechas que se ha iniciado sesión es:", dFechas_User['numAccesos'].max())
36 print("El valor mínimo de las fechas que se ha iniciado sesión es:", dFechas_User['numAccesos'].min())
37
38 print("La media de accesos por IP es de:", dIPS_User['numAccesos'].mean())
39 print("La desviación estándar de accesos por IP es de:", dIPS_User['numAccesos'].std())
```

The Run console at the bottom shows the output of the script:

```
C:\Users\Sinan\Documents\GitHub\PracticaSI1\pythonProject\venv\Scripts\python.exe C:/Users/Sinan/Documents/GitHub/PracticaSI1/pythonProject/ejercicio2.py
La media de correos recibidos son: 109.33333333333333
La desviacion estándar de correos recibidos son: 108.35425085413549
El valor máximo de correos recibidos es: 493
La valor mínimo de correos recibidos es: 20
El valor máximo de las fechas que se ha iniciado sesión es: 20
El valor mínimo de las fechas que se ha iniciado sesión es: 1
La media de accesos por IP es de: 9.733333333333333
La desviación estándar de accesos por IP es de: 6.141623582562274

Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8, line endings are CRLF, and it uses 4 spaces for indentation. It also shows the Python version as 3.10.

EJERCICIO 3

Se hace uso de las siguientes funciones:

Véase ejercicio3.py

```
import sqlite3;
import pandas as pd;

con = sqlite3.connect('SISTINF.db');

query = con.execute("SELECT * From users where permisos = 0")
cols = [column[0] for column in query.description]
dUsersUser = pd.DataFrame.from_records(data = query.fetchall(), columns = cols)

query = con.execute("SELECT * From users where permisos = 1")
cols = [column[0] for column in query.description]
dUsersAdmin = pd.DataFrame.from_records(data = query.fetchall(), columns = cols)

query = con.execute("SELECT * From users where emailsPhising >= 200")
cols = [column[0] for column in query.description]
dUsersMas200 = pd.DataFrame.from_records(data = query.fetchall(), columns = cols)

query = con.execute("SELECT * From users where emailsPhising < 200")
cols = [column[0] for column in query.description]
dUsersMenos200 = pd.DataFrame.from_records(data = query.fetchall(), columns = cols)

con.close()
```

- **Número de observaciones**

User: dUsersUser ['emailsPhising'].count()

Admin: dUsersAdmin ['emailsPhising'].count()

Mas200: dUsersMas200 ['emailsPhising'].count()

Menos200: dUsersMenos200 ['emailsPhising'].count()

- **Número de valoraciones ausentes**

- **Mediana**

User: dUsersUser ['emailsPhising'].median()

Admin: dUsersAdmin ['emailsPhising'].median()

Mas200: dUsersMas200 ['emailsPhising'].median()

Menos200: dUsersMenos200 ['emailsPhising'].median()

- **Media**

User: dUsersUser ['emailsPhising'].mean()

Admin: dUsersAdmin ['emailsPhising'].mean()

Mas200: dUsersMas200 ['emailsPhising'].mean()

Menos200: dUsersMenos200 ['emailsPhising'].mean()

- **Varianza**

User: dUsersUser ['emailsPhising'].var()

Admin: dUsersAdmin ['emailsPhising'].var()

Mas200: dUsersMas200 ['emailsPhising'].dUsersMenos200()

Menos200: dUsersMenos200 ['emailsPhising'].dUsersMenos200()

- **Valores máximo y mínimo**

MaxUser: dUsersUser['emailsPhising'].max()

MinUser: dUsersUser['emailsPhising'].min()

MaxAdmin: dUsersAdmin['emailsPhising'].max()

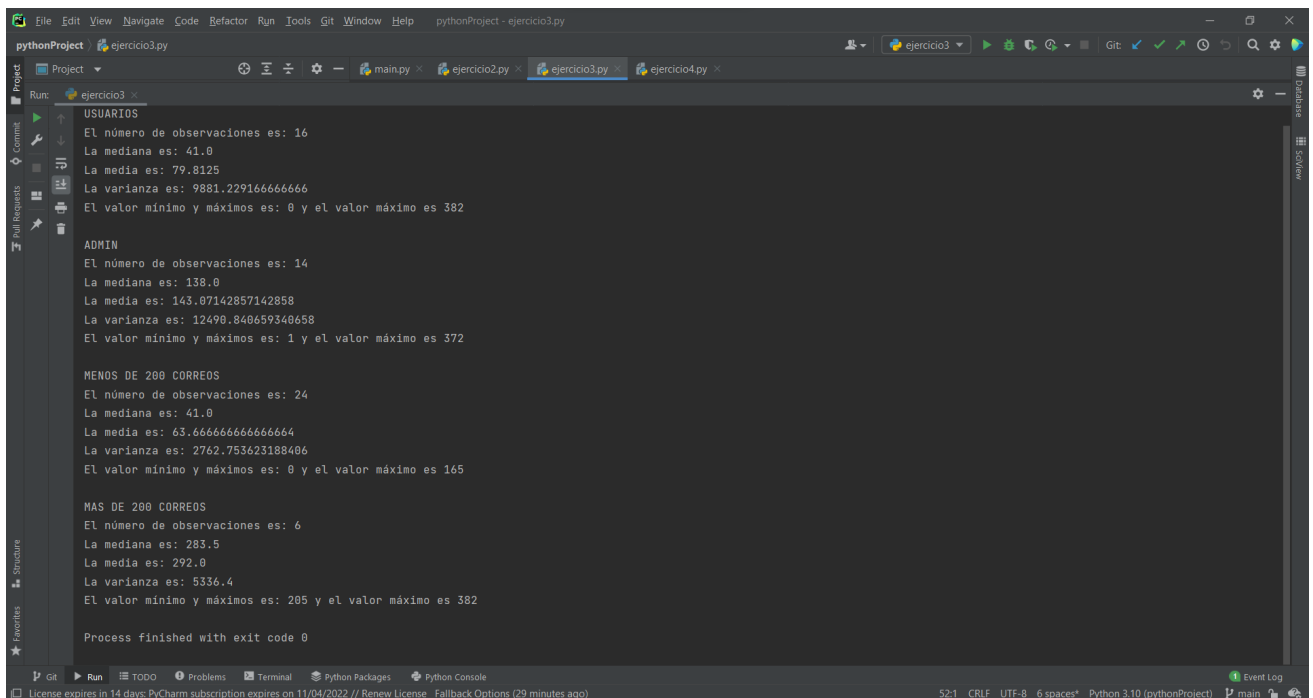
MinAdmin: dUsersAdmin['emailsPhising'].min()

MaxMas200: dUsersMas200 ['emailsPhising'].max()

MinMas200: dUsersMas200 ['emailsPhising'].min()

MaxMenos200: dUsersMenos200 ['emailsPhising'].max()

MinMenos200: dUsersMenos200 ['emailsPhising'].min()1



```
pythonProject - ejercicio3.py
pythonProject - ejercicio3.py
ejercicio3
Run
ejercicio3
USUARIOS
El número de observaciones es: 16
La mediana es: 41.0
La media es: 79.8125
La varianza es: 9881.229166666666
El valor mínimo y máximos es: 0 y el valor máximo es 382

ADMIN
El número de observaciones es: 14
La mediana es: 138.0
La media es: 143.07142857142858
La varianza es: 12490.840659340658
El valor mínimo y máximos es: 1 y el valor máximo es 372

MENOS DE 200 CORREOS
El número de observaciones es: 24
La mediana es: 41.0
La media es: 63.666666666666664
La varianza es: 2762.753623188406
El valor mínimo y máximos es: 0 y el valor máximo es 165

MAS DE 200 CORREOS
El número de observaciones es: 6
La mediana es: 283.5
La media es: 292.0
La varianza es: 5336.4
El valor mínimo y máximos es: 205 y el valor máximo es 382

Process finished with exit code 0
```


Ejercicio 4

Véase ejercicio4.py

- **Mostrar los 10 usuarios más críticos (un usuario crítico es aquel usuario que tiene la contraseña débil y además tiene mayor probabilidad de pulsar en un correo de spam), representadas en un gráfico de barras.**
- **Mostrar las 5 páginas web que tienen más políticas (cookies, protección de datos o aviso legal) desactualizadas, representadas en un gráfico de barras según las políticas.**
- **Mostrar la media de conexiones de usuarios con contraseña vulnerable, frente a los que no son vulnerables.**
- **Mostrar según el año de creación las webs que cumplen todas las políticas de privacidad, frente a las que no cumplen la política de privacidad.**
- **Mostrar el número de contraseñas comprometidas y contraseñas no comprometidas.**

GitHub

El código del proyecto ha sido subido a un repositorio público en Github al que se puede acceder mediante el siguiente enlace:

<https://github.com/sinanod/PracticaSI1>