API - PYTHON

ADRIÁN GALLEGO, CARLOS CASTRO, ALEJANDRO ABAD

PROYECTO DE COMPUTACIÓN II

DESARROLLO DE LAS APIS EN PYTHON

ÍNDICE

Contenido

[INTRODUCCIÓN 1](#_Toc508747654)

[DESARROLLO 1](#_Toc508747655)

[Primera consulta 2](#_Toc508747656)

[Segunda consulta 2](#_Toc508747657)

[Tercera consulta 2](#_Toc508747658)

[Cuarta consulta 3](#_Toc508747659)

[Quinta consulta 3](#_Toc508747660)

[FORMATOS DEVUELTOS 3](#_Toc508747661)

# INTRODUCCIÓN

***OBJETIVO***

El objetivo es aprender a diseñar e implementar una API restful que permita hacer consultas y recuperar los datos de los casos de violencia de género recogidos con la herramienta de los pasos anteriores.

***ENUNCIADO***

Utilizando el módulo Bottle de Python, implementar una API [1] que permita hacer consultas y recuperar información almacenada en la BBDD de forma estructurada (CSV, Json, XML…).

Por ejemplo, si se hace una llamada a la dirección /provincia/Huelva devuelve todos los datos de caos que han ocurrido en Huelva, ... Para ello se deben realizar los siguientes pasos:

* Diseñar qué tipo de consultas queremos hacer públicas para nuestra API.
* Diseñar las consultas a la BBDD necesarias para devolver los datos asociados a las consultas definidas previamente.
* Implementar con Bottle la API.

Es interesante que la API devuelva los datos en varios formatos. Por ejemplo:

*/provincia/Huelva/csv --> la respuesta en csv*

*/provincia/Huelva/json --> la respuesta es un Json [2]*

**Links de referencia:**

[1] <https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones>

[2] <https://es.wikipedia.org/wiki/JSON>

# DESARROLLO

Para el desarrollo de esta entrega de PC2 hemos desarrollado una API en Python gracias a la librería “Bottle”, que nos ha permitido visualizar en un servidor en local la información que deseábamos. Para esta entrega hemos desarrollado 5 consultas posibles para el buscador una vez tenemos el programa en funcionamiento. Para acceder a estas consultas únicamente debemos escribir en el navegador lo siguiente:

Link: <http://localhost:8000>

Esta sería la primera parte, y a partir de aquí únicamente deberíamos concretar los datos que queremos obtener y su formato, pero esto lo mostraré paso a paso a continuación.

## Primera consulta

Esta primera consulta permite al usuario mostrar las víctimas de violencia de genero por id de comunidad autónoma

Link: [http://localhost:8000/CCAA/<ID\_CCAA>](http://localhost:8000/CCAA/%3cID_CCAA%3e)

Donde “<ID\_CCAA>” es un número entre 1-19 [3] en función de la comunidad autónoma.

Ej: <http://localhost:8000/CCAA/17>/csv

[3] "1, Andalucía”, “2, Aragón", "3, Asturias", "4, Islas Baleares", "5, Canarias", "6, Cantabria", "7, Castilla y León", "8, Castilla la Mancha", "9, Cataluña", "10, Comunidad Valenciana", "11, Extremadura", "12, Galicia", "13, Madrid", "14, Murcia", "15, Navarra", "16, País Vasco", "17, La Rioja", "18, Ceuta", "19, Melilla".

Segunda consulta

Esta segunda consulta permite al usuario mostrar las víctimas de violencia de genero por fecha

Link: [http://localhost:8000/Fecha/<dia>/<mes>/<ano>](http://localhost:8000/Fecha/%3cdia%3e/%3cmes%3e/%3cano%3e)

Donde “<dia>/<mes>/<ano>” hacen referencia al momento en que sucedió el caso y tienen un formato “dd/mm/aaaa”

EJ: <http://localhost:8000/Fecha/05/01/1999>/csv

## Tercera consulta

Al igual que en la segunda consulta, la tercera permite al usuario buscar datos en función de la fecha, pero esta vez cuenta con un formato más sencillo.

Link: [http://localhost:8000/Fecha/<fech>](http://localhost:8000/Fecha/%3cfech%3e%20)

Donde “<fech> “hace referencia al momento en que sucedió el caso y tienen un formato “dd-mm-aaaa”

EJ: <http://localhost:8000/Fecha/05-01-1999>/csv

## Cuarta consulta

La cuarta consulta permite al usuario mostrar las víctimas de violencia de genero por Provincias

Link: [http://localhost:8000/Prov/<Provincia>](http://localhost:8000/Prov/%3cProvincia%3e%20)

Donde “<Provincia> “es un número entre 1-52 [4] en función de la Provincia donde se diera el caso.

EJ: [http://localhost:8000/Prov/34/csv](http://localhost:8000/Prov/34)

[4] "1, Álava”, “2, Albacete”, “3, Alicante”, “4, Almería”, “5, Ávila”, “6, Badajoz”, “7, Islas Baleares”, “8, Barcelona”, “9, Burgos”, “10, Cáceres”, “11, Cádiz”, “12, Castellón”, “13, Ciudad Real”, “14, Córdoba”, “15, A Coruña”, “16, Cuenca”, “17, Girona”, “18, Granada”, “19, Guadalajara”, “20, Gipuzkoa”, “21, Huelva”, “22, Huesca”, “23, Jaén”, “24, León”, “25, Lleida”, “26, La Rioja”, “27, Lugo”, “28, Madrid”, “29, Málaga”, “30, Murcia”, “31, Navarra”, “32, Ourense”, “33, Asturias”, “34, Palencia”, “35, Las Palmas”, “36, Pontevedra”, “37, Salamanca”, “38, Santa Cruz de Tenerife”, “39, Cantabria”, “40, Segovia”, “41, Sevilla”, “42, Soria”, “43, Tarragona”, “44, Teruel”, “45, Toledo”, “46, Valencia”, “47, Valladolid”, “48, Bizkaia”, “49, Zamora”, “50, Zaragoza”, “51, Ceuta”, “52, Melilla"

## Quinta consulta

La última consulta permite al usuario mostrar las víctimas de violencia de genero por la relación que mantenían.

Link: [http://localhost:8000/RVA/<RelacionV\_A>](http://localhost:8000/RVA/%3cRelacionV_A%3e%20)

Donde “<RelacionV\_A> “puede ser una de las siguientes relaciones (Amante, Amigo, Cliente, Compañero, Conocido, Consuegro, Cuñado, Esposo, Familiar, Hermano, Hijastro, Hijo, Nieto, Padrastro, Padrastro, Pareja, Primo, Tío, Vecino, Yerno)

EJ:

<http://localhost:8000/RVA/Pareja/csv>

# FORMATOS DEVUELTOS

Finalmente añadimos una nueva sección “<formato>” en la que seleccionaremos el formato en que queremos obtener los datos. Las opciones con las que contamos son las siguientes:

***JSON***

Obtendremos el resultado en formato “JSON” introduciendo al final del link “/json”

***CSV***

Obtendremos el resultado en formato “CSV” introduciendo al final del link “/csv”