

# MEMORIA

**Integrantes del grupo:** Esteban Braganza Cajas y Ana Álvarez Sánchez

**Enlace al sitio web elegido:** <https://www.deviantart.com>

**Enlace al repositorio con el código de la práctica:**

<https://github.com/EstebanBraganza77/Web-Scraping-Practica1>

**Enlace al dataset publicado en Zenodo:** <https://zenodo.org/records/10974440>

**Enlace al vídeo de presentación de la práctica:**

Tipología y Ciclo de vida de los Datos - Práctica 1

---

## 1. Contexto

Según su propia descripción, [DeviantArt](#) es “la mayor comunidad de arte y galería en línea”. De ahí su interés para realizar un ejercicio de scraping de las imágenes que almacena y los datos que acompañan a cada una de ellas. Estos datos incluyen métricas interesantes para un posterior análisis como el número de vistas, favoritos, comentarios, etc, así como campos técnicos (resolución en píxeles o tamaño en MB de la imagen). Además, el buscador de la web permite realizar búsquedas en base a cualquier campo que se quiera indicar.

**Investigación previa:** Para la realización de esta práctica se han valorado diferentes webs (idealista.com, fotocasa.es, store.steampowered.com, plusvalia.com). Las dos primeras se descartaron después de comprobar que había numerosos ejemplos de scraping sobre ellas disponibles con una simple búsqueda en Google. Steam presentaba una navegación bastante “enrevesada”, con listados múltiples y diferentes visualizaciones de resultados (como lista, como tabla de imágenes,...) que no la hacían en una página muy “ limpia” para el ejercicio. Finalmente se decidió optar por deviantart.com por tener una gran base de datos de imágenes, una navegación que nos permitía utilizar herramientas más complejas como Selenium, y un conjunto de datos a partir de cada imagen bastante rico, con campos de texto y numéricos.

## 2. Título

Dataset de Imágenes y Métricas de DeviantArt: búsqueda por temas.

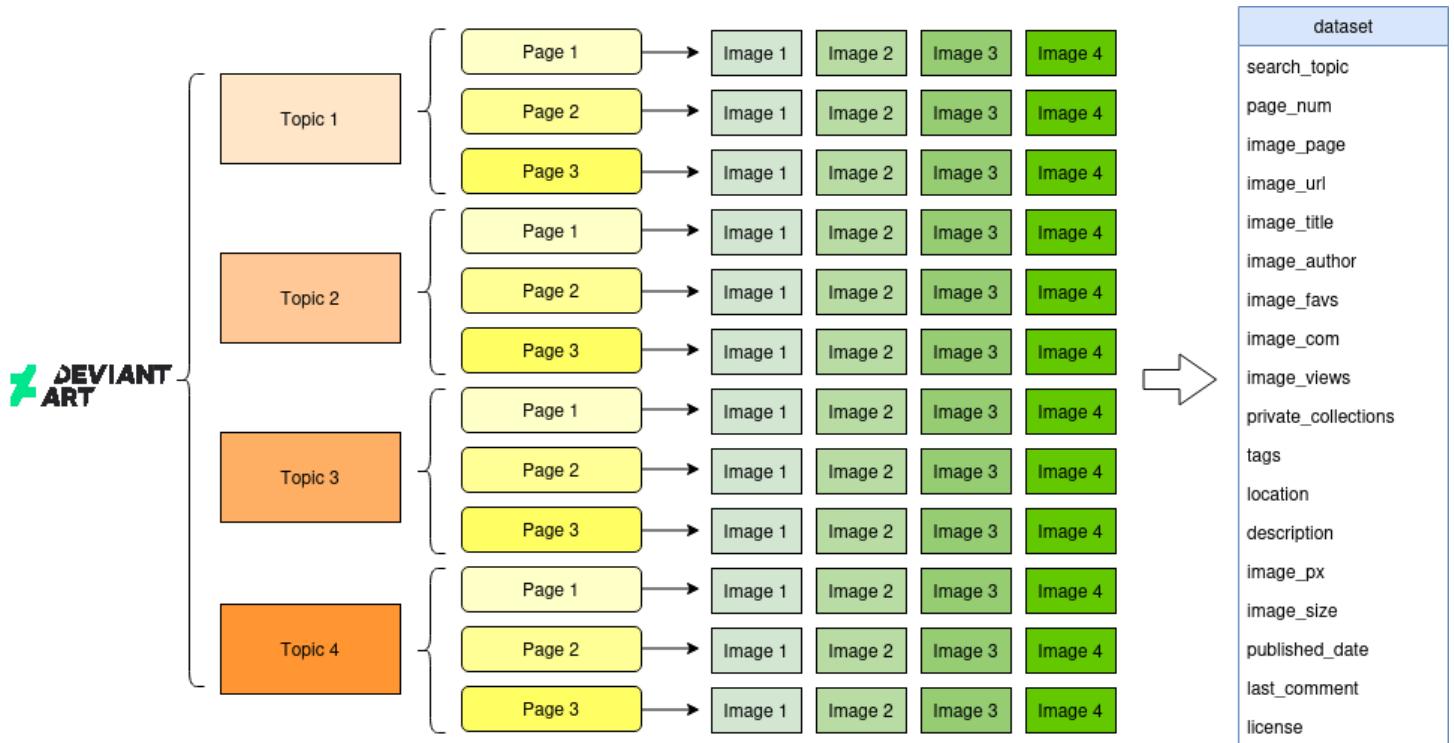
### **3. Descripción del dataset**

Conjunto de datos de las imágenes almacenadas por la web deviantart.com, como resultado de la búsqueda de 20 temas y de las 15 primeras páginas de cada tema, obtenido a fecha 2024-04-14 . Los temas buscados son:

- "Fantasy art",
- "Science fiction art",
- "Anime and manga art",
- "Fan art (for specific fandoms)",
- "Digital paintings",
- "Traditional drawings",
- "Character designs",
- "Creature concepts",
- "Landscape art",
- "Abstract art",
- "Surrealism",
- "Steampunk art",
- "Cyberpunk art",
- "Gothic art",
- "Horror art",
- "Cosplay photography",
- "Pixel art",
- "Concept art",
- "Comics and graphic novels",
- "Street art and graffiti",

El dataset contiene 7067 registros de imágenes relacionadas con estas temáticas.

### **4. Representación gráfica**



## 5. Contenido

Los campos que se incluyen en el dataset son:

- **Search\_topic:** la imagen es resultado de la búsqueda por este tema
- **Page\_num:** la imagen aparece en este número de página de la búsqueda
- **Image\_page:** enlace a la página con la información de la imagen
- **Image\_url:** enlace a la imagen
- **Image\_title:** título de la imagen
- **Image\_author:** autor/a de la imagen
- **Image\_favs:** número de veces que le han dado a “me gusta” en la imagen
- **Image\_com:** número de comentarios que tiene la imagen
- **Image\_views:** número de vistas a la imagen
- **Private\_collections:** número de veces que ha sido incluida en una colección privada
- **Tags:** etiquetas que se le han asignado a la imagen para facilitar su descubrimiento
- **Location:** país o localización geográfica, si el autor la quiere identificar
- **Description:** campo de texto abierto creado por el autor, que acompaña a la imagen. Puede incluir detalles técnicos o enlaces a las redes sociales del autor/a.
- **Image\_px:** dimensiones de la imagen, en pixeles
- **Image\_size:** peso de la imagen en MB.
- **Published\_date:** fecha de publicación de la imagen.
- **Last\_comment:** último comentario añadido a la imagen.

- **License:** licencia de la imagen

**Período de tiempo al que pertenecen los datos:** La extracción de los datos se realiza el día 14 de abril de 2024. Las imágenes que se incluyen dentro del dataset han sido publicadas en la web deviantart.com entre el 2003-03-11y 2024-02-18.

## 6. Propietario

Existen numerosos ejemplos de scrapers sobre las imágenes de deviantArt. Algunos de ellos son:

- [DeviantArt Scraper](#), de rje4242, que permite indicar una url de deviantart (por defecto es la home) y un número máximo de imágenes para descargar.
- [DeviantArt Scraper](#), de Kent-Lee, que utiliza multi-threads, y permite descargar imágenes por usuario y ranking.
- [Web Scraping Program](#), de JohnGarnerIII, que realiza una descarga de las imágenes a partir de la url de la página donde se incluyen.

También existen ejemplos de análisis de datos de deviantArt:

- [Analyzing Sonic Fan Art with data science](#): realiza un análisis de los tags que utilizan los artistas.
- [Explorative visualization and analysis of a social network for arts: The case of deviantART](#): incluye funcionalidades para extraer datos de usuarios y sus creaciones, usando network analysis para seleccionar usuarios clave.
- [Unleashing AI Art: Analyzing Popular Prompts with Stable Diffusion for Stunning Masterpieces](#): hace un análisis de alrededor de 80.000 prompts para descubrir aquellos más buscados (las combinaciones de palabras más utilizadas).

En cuanto a los **principios éticos y legales**, DeviantArt es, a menos que se indique lo contrario, el propietario de todos los derechos de autor y de datos sobre el Servicio y su contenido.

DeviantArt incluye una [política de copyright](#), sobre el uso de las imágenes, que no se estaría violando en el caso de este ejercicio, porque solo se agrupan datos de la página de cada imagen, datos disponibles públicamente, sin necesidad de cuenta en la web.

El campo de la licencia de cada imagen se ha incluido en el dataset, para que cualquier uso que se haga de dichos datos pueda tener en cuenta las diferentes licencias.

En los [términos del servicio](#) de DeviantArt existe una sección sobre “Data Scraping & Machine Learning Activities”, que hace referencia al uso de las imágenes para entrenar modelos de IA, con una política “NOAI”, que no permite este uso. En este ejercicio no se está entrenando ninguna IA, por lo que no se violan los términos del servicio.

DeviantArt will include a robots meta tag with the "noai" or "noimageai" directive in the head section of the HTML page associated with that Content on the Site, and will include an X-Robots-Tag HTTP response header with the "noai" directive when media files associated with that Content are downloaded from the Service.

A pesar de ello, DeviantArt entiende que, al publicar trabajos en su web, los datos e imágenes pueden ser scrapeados por terceros y utilizados en modelos de IA:

Users acknowledge that by uploading Content to DeviantArt, third-parties may scrape or otherwise use their works without permission. DeviantArt provides no guarantees that third parties will not include certain Content in external data sources, or otherwise use a creator's work for Artificial Intelligence Purposes, even when such directives are present.

La licencia específica de cada una de las imágenes incluidas en el dataset se recoge en el campo "license".

## 7. Inspiración

DeviantArt es una comunidad en línea que sirve como plataforma para que artistas de todo el mundo compartan y exhiban su trabajo creativo. DeviantArt ha crecido hasta convertirse en una de las comunidades de arte más grandes y diversas en la web. Además de ser una plataforma para mostrar el talento creativo, también ofrece características sociales que fomentan la interacción entre los miembros. Los usuarios pueden seguir a sus artistas favoritos, comentar en las obras de arte, participar en concursos y desafíos, unirse a grupos temáticos y compartir recursos y tutoriales.

Extraer datos de DeviantArt puede tener varias importancias y aplicaciones, tanto para usuarios individuales como para investigadores, artistas y empresas. A continuación algunas razones por las que la extracción de datos de DeviantArt puede ser importante:

- 1. Análisis de Tendencias Artísticas:** La extracción de datos puede ayudar a identificar tendencias emergentes en el mundo del arte como estilos populares, temas recurrentes o cambios en las preferencias de los usuarios. Este uso queda reflejado en el ejemplo citado en el apartado 6, [Unleashing AI Art: Analyzing Popular Prompts with Stable Diffusion for Stunning Masterpieces](#), donde se realiza análisis de prompts.
- 2. Descubrimiento de Nuevos Artistas:** Analizando los datos extraídos de DeviantArt es posible descubrir nuevos talentos y artistas emergentes basados en la interacción y gustos de los usuarios de la red. En el dataset tenemos campos que pueden ser analizados para esta tarea como número de favs o número de comentarios.
- 3. Investigación académica:** Los datos extraídos de DeviantArt pueden ser utilizados en investigaciones académicas sobre diversos temas, como sociología del arte, psicología del usuario, análisis de comunidades en línea, entre otros. Es el caso del paper

publicado con el título [Explorative visualization and analysis of a social network for arts: The case of deviantART](#) y mencionado anteriormente.

4. **Monitoreo de la industria del arte:** Las tendencias y la popularidad de las obras de arte de DeviantArt pueden servir como indicadores del estado de la industria del arte digital en general.
5. **Desarrollo de aplicaciones y herramientas:** El uso de los datos extraídos puede ayudar a la generación de aplicaciones relacionadas con el arte y sus tendencias.

## 8. Licencia

La licencia elegida para este caso sería [CC BY 4.0](#), que permite la redistribución y reutilización de una obra con la condición de que el creador reciba el crédito adecuado

Utilizamos esta licencia porque los datos incluidos en el dataset son públicos en deviantArt. Esta licencia afecta al conjunto de datos publicados. Cada uno de las imágenes incluidas en él tiene su propia licencia que debe ser respetada en cualquier utilización que se haga de ella.

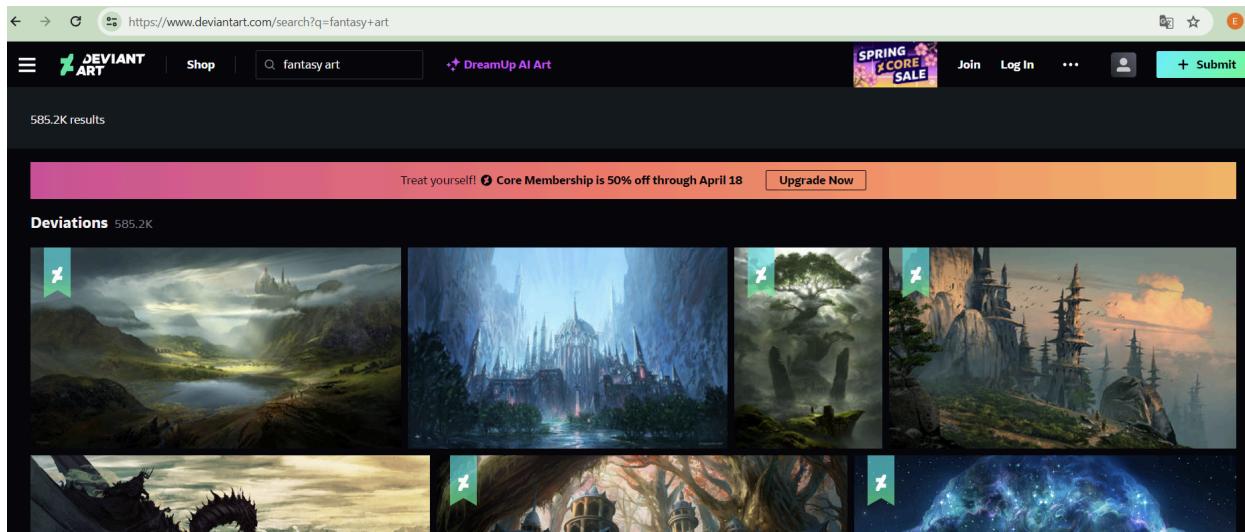
## 9. Código

El código que se ha desarrollado realiza una extracción de la información de imágenes basadas en la búsqueda de uno o varios temas de arte y un número de páginas establecido.

Cada vez que realizamos una búsqueda, la página web nos despliega una serie de páginas con imágenes que corresponden a la búsqueda realizada. El número de páginas puede ser muy grande, ya que cada búsqueda devuelve varios miles de imágenes distribuidas en 24 imágenes por página.

Para facilitar la búsqueda definimos el tema y el número máximo de páginas por las que navegará.

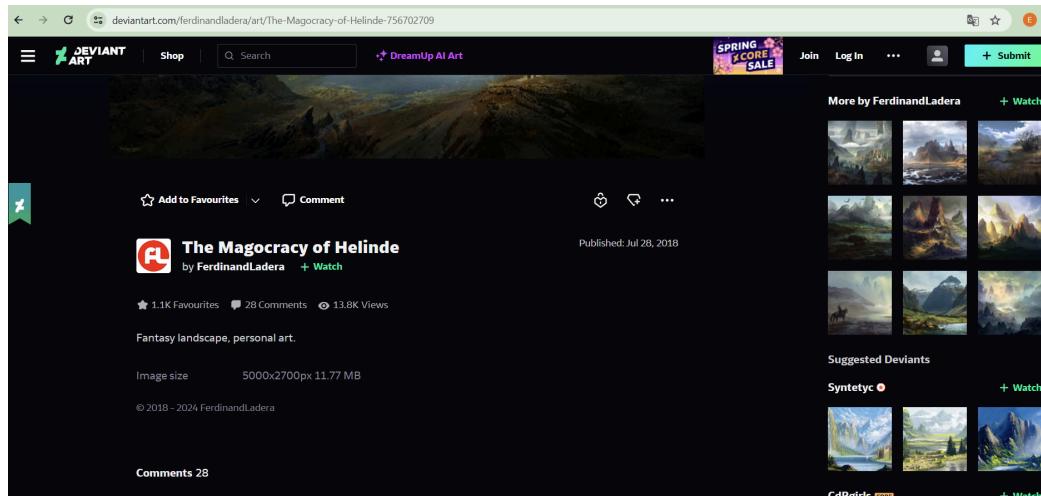
En el siguiente ejemplo, la búsqueda de “fantasy art” nos devuelve 582 K imágenes.



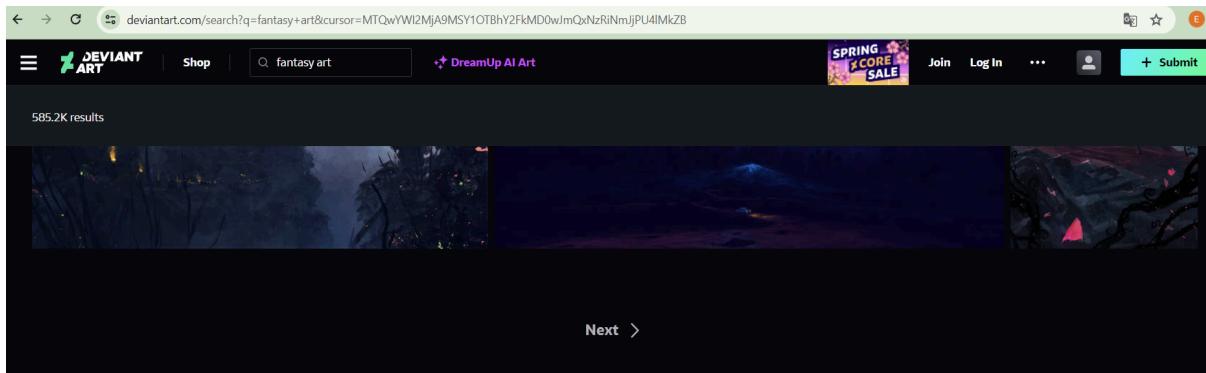
Las imágenes resultantes de la búsqueda no están ordenadas por fecha, ni por número de vistas, favs o comentarios.

Una vez realizamos la búsqueda tenemos la primera página con el set de 24 imágenes. De esta vista obtenemos el link de cada una de estas imágenes. Usando estos links extraemos la información necesaria para construir el data set.

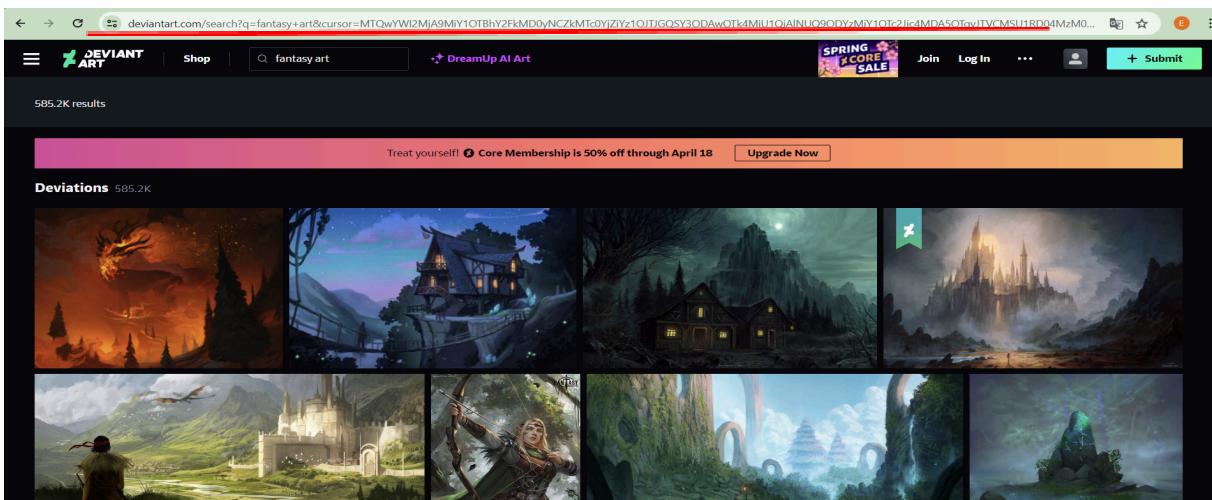
Cada link nos lleva a la siguiente vista con los detalles de la imagen, de donde obtenemos la información definida anteriormente.



Finalmente avanzamos a la siguiente página y para esto utilizamos el botón "next" que se encuentra en la parte inferior de la página de la búsqueda.



Al hacer clic sobre él avanzamos a la siguiente página.



Para la navegación entre las páginas de la búsqueda se hizo necesario el uso de Selenium, porque, al cambiar de página, la ruta a la siguiente página contiene una combinación de códigos que son difíciles de predecir y automatizar.

Para solventar esta dificultad y obtener los mejores resultados al hacer web scraping hemos decidido dividir nuestro programa en cuatro partes principales.

- **Navegación:** Esta parte del código contiene la lógica de navegación y nos ayudamos con Selenium para esta tarea. Básicamente podemos navegar a través de los resultados de la búsquedas de ciertos temas predefinidos.
  - Realiza una búsqueda del tema elegido
  - Obtiene los links de cada una de las imágenes.
  - Una vez extraídos los datos de cada imagen los guarda.
  - Avanza a la siguiente página.

- Cuando llega al máximo de páginas definidas realiza la búsqueda del siguiente tema.
  - Repite el proceso para el resultado de búsqueda del siguiente tema.
- **Extracción:** Esta parte mantiene la lógica de extracción de los campos específicos de la página de cada imagen.
  - Para esto utilizamos la librería Request y BeautifulSoup.
  - A partir de la url de cada imagen, realizamos un request de la página y extraemos los campos de interés.
  - Establecemos un timeout de 5 segundos, para asegurarnos de que no continúa haciendo peticiones a la página si no recibe respuesta en este tiempo, como medida para evitar sobrecargas a la web en caso de que esta no responda.
  - También se hace gestión de errores en caso de que la página devuelva None o un status code distinto a 200.
  - Se manejan distintos casos y se envían excepciones si algún campo no existe, devolviendo None si alguna imagen no cuenta con el campo requerido.
  - Se limpian los campos en la medida de lo posible para esta primera fase:
    - Igualar la representación de todas las métricas a unidades (convirtiendo valores expresados como miles o millones)
    - Limpieza de espacios en strings
    - Dejar valores numéricos únicamente siempre que sea posible
  - Los campos con la información de cada imagen los guardamos como diccionario.
  - Este proceso se realiza simultáneamente con la navegación.
- **Guardado:** Despues de obtener la información de interés, utilizamos el diccionario final para generar un dataframe de pandas y guardarlo en formato csv.
- **Ejecución:** Esta parte contiene el archivo main.py, que debe ser ejecutado para iniciar el proceso. En él se pueden definir los temas de búsqueda y el máximo de páginas por cada tema.

Para realizar este trabajo.

Hemos creado la clase **DeviantArt Scraper** y para cada etapa del proceso tenemos métodos que se enfocan en cada parte.

- **Navegación:**
  - Método: run\_scraper
- **Extracción:**
  - navigate\_image\_links:
 

Itera a través de los links de cada imagen de la página. En este método tenemos un tiempo de espera de 2 segundos entre cada imagen y así evitamos sobrecargar el servidor de DevianArt.

- `get_info_from_url`:  
Obtiene los datos realizando un session request y beautiful soup para extraer la información.
- **Guardado**
  - `generate_df`
  - `save_to_csv`
  - `save_to_json`
  - `download_image`:  
Proporcionamos un método para descargar una imagen a partir de la url, aunque no es necesario para la creación del dataset)
- **Ejecución** main.py y función main

#### **Presentación del código:**

Tanto la clase como los métodos incluidos dentro de `scraper.py` se han comentado debidamente. También se han incluido docstrings y typehints, siguiendo las recomendaciones de uso de la librería typing para las clases List y Dict.

Fuente: <https://realpython.com/python-type-checking/>

## **10. Dataset**

Publicación en Zenodo: <https://zenodo.org/records/10974440>

DOI: <https://doi.org/10.5281/zenodo.10974440>

## **11. Vídeo**

Enlace del vídeo:

<https://drive.google.com/file/d/16LKJLT5PpwydhV8sF9VdB5LZIR332ogm/view?usp=sharing>

Contribuciones	Firma
Investigación previa	E.B.C., A.A.S.
Redacción de las respuestas	E.B.C., A.A.S.
Desarrollo del código	E.B.C., A.A.S.
Participación en el vídeo	E.B.C., A.A.S.