A logo of a coat of arms

Description automatically generatedUNIVERSIDAD DE BURGOS

Escuela Politécnica SuperiorA logo of a machine

Description automatically generated

Gº en Ingeniería Informática

A black and white logo

Description automatically generated

**TFG Ingeniería Informática:**

**MelodyMatrix**

Presentado por Martín González Saiz

en Burgos el X julio de 2024

Tutores D. José Manuel Galán Ordax

**Introducción**

En el sector de la música, la preservación y el acceso al contenido antiguo es una necesidad crucial para garantizar que futuras generaciones puedan trabajar y estudiar estas obras. Este contenido puede corresponder a un amplio abanico de posibilidades, desde partituras y libros hasta instrumentos y grabaciones. Por ello la digitalización de este tipo de material es esencial no solo para evitar su deterioro por el paso del tiempo y garantizar su preservación, sino también para facilitar su acceso con plataformas digitales. Con ello se podría garantizar que toda esta riqueza cultural, artística y educativa que estas obras nos aportan pueda ser accesible con grata facilidad y eficiencia además de segura para todo el mundo.

En el caso de las partituras, el manejo, estudio, almacenamiento e investigación recae sobre muchas personas dedicadas a diferentes sectores dentro del mundo de la música, como musicólogos, músicos y educadores, los cuales se dedican y ganan la vida haciendo uso de este material. La transcripción de las partituras siempre se ha realizado manualmente, provocando que, muchas veces, se produzcan errores humanos y que el proceso sea lento e ineficiente, sobre todo con transcripciones de grandes colecciones o con partituras notoriamente complejas.

Con el gran avance de la tecnología, se han desarrollado varias herramientas y tecnologías que agilizan y optimizan este proceso. Una de las herramientas más utilizadas es el Reconocimiento Óptico de Música, la cual realiza la digitalización de partituras de manera automática. Una de las herramientas más populares y con mayor efectividad es Audiveris.

Este proyecto se enfoca en desarrollar una aplicación web que permita el almacenamiento, gestión y digitalización de las partituras de forma que sea simple y accesible de usar por el usuario. La funcionalidad principal es transcribir estas partituras desde cualquier formato hacia MusicXML, formato universal para el contenido musical digital. Tras ello se podría editar las notas a su gusto para luego descargar la nueva versión.

Otra finalidad de la herramienta es reducir la necesidad de intervención manual y generación de errores en la transcripción y edición. De esta manera, el patrimonio musical podría perdurar con mayor facilidad y el gremio de la música crecer nutriéndose de herramientas como esta.

**B. Objetivos del proyecto**

En primer lugar, se van a mostrar los objetivos principales del proyecto, abarcando desde los más teóricos hasta los más prácticos. También se van a abordar aspectos de desarrollo mas individuales de desarrollo personal.

El proyecto principalmente se basa en proporcionar a usuarios involucrados en el mundo de la música una plataforma donde pueda realizar una gestión cómoda de su material. Pudiendo estos realizar diferentes acciones ya sea de almacenaje, digitalización, preprocesamiento, borrado, etc.

1. **Objetivos generales**

A continuación, se van a listar todos los objetivos principales del proyecto:

* Incorporar una autenticación segura y efectiva con el uso de una encriptación manejada por bcrypt, así como una gestión de usuarios sólida.
* Dar un servicio de almacenamiento de partituras online para cualquier usuario con una interacción cómoda y rápida mediante las funcionalidades de subida y borrado de partituras, a través del uso de los servicios de Firebase
* Implementar un proceso de transcripción de partituras eficiente a través del uso del software de reconocimiento óptico de música (OMR), Audiveris. A su vez, contrastar los resultados a través de un análisis de los logs generados por el proceso de digitalización de Audiveris para informar al usuario.
* Proporcionar un sistema de almacenamiento independiente para las partituras que han sido digitalizadas.
* Implementar una funcionalidad de eliminación de partituras tanto subidas sin digitalizar, como partituras digitalizadas.
* Implementar una opción de preprocesamiento manual mediante el uso de tecnologías de procesamiento de imágenes, como desenfoque o umbralización adaptativa, con el fin de mejorar la calidad de la imagen subida y a digitalizar.
* Implementación de la posibilidad de visualización de las partituras digitalizadas mediante el uso de la librería Verovio.
* Utilización de servicios externos populares en el mundo del desarrollo web como las herramientas de Firebase, el Docker y sus contenedores o el despliegue con Heroku.

1. **Objetivos técnicos**

* Utilizar la metodología Scrum para el desarrollo correcto del proyecto.
* Uso de la herramienta Trello para la gestión del las tareas y avances del proyecto.
* Manejo de GitHub como herramienta para gestionar los cambios y las versiones del proyecto.
* Utilizar GitHub Desktop para el adecuado control de los cambios de la aplicación y el acceso cómodo al historial de cambios.
* Uso de Flask como framework para el desarrollo web de mi aplicación de digitalización.
* Utilizar una herramienta de manejo y gestión de referencias para la documentación del proyecto.
* Utilizar librerías de Python para el desarrollo de funcionalidades de la aplicación.
* Implementación de código que permite utilizar softwares como miniservicios en segundo plano.

1. **Objetivos personales**

* Uso de Python como lenguaje de programación para el desarrollo
* Manejo de excepciones para el perfecto control del flujo de la aplicación y pruebas
* Manejo de la funcionalidad logging para debuguear y solventar errores en las funcionalidades de la aplicación.
* Utilizar herramientas y servicios externos relacionados con el mundo del desarrollo web.
* Desplegar la aplicación públicamente para posibilitar su uso para cualquier usuario.

1. **Enlaces adicionales**

En este apartado se van a adjuntar los enlaces correspondientes a las herramientas externas de control del proyecto utilizadas:

* **Repositorio de GitHub**:  
  <https://github.com/martingonzsaiz/Digitalizacion-Partituras>
* **Tablas de la gestión del proyecto de Trello**:
  + Sprint 1: <https://trello.com/b/s2npoqgv/tfg-digitalizacion-de-partituras-sprint-1-15-02-2024-01-03-2024>
  + Sprint 2: <https://trello.com/b/SEnQJM6h/tfg-digitalizacion-de-partituras-sprint-2-01-03-2024-17-03-2024>
  + Sprint 3: <https://trello.com/b/5qrTSteY/tfg-digitalizacion-de-partituras-sprint-3-18-03-2024-15-04-2024>
  + Sprint 4: <https://trello.com/b/W4oJQ7Ah/tfg-digitalizacion-de-partituras-sprint-4-15-04-2024>
  + Sprint 5: <https://trello.com/b/XWatG1p6/tfg-digitalizacion-de-partituras-sprint-5-23-05-2024-20-06-2024>
  + Sprint 6: <https://trello.com/b/OuDMWHQl/tfg-digitalizacion-de-partituras-sprint-6-21-06-2024>

**Objetivos de Desarrollo Sostenible**

**Objetivo 4: Educación de calidad**

El proyecto contribuye al Objetivo de Desarrollo Sostenible 4, ‘Educación de calidad’, al aportar en gran medida un recurso funcional para la educación musical. Siendo su acceso libre, permite la edición y manipulación de las partituras en formato digital, abriendo múltiples posibilidades a profesores, alumnos y personas autodidactas para gestionar diferente material del sector.

**Objetivo 9: Industria, innovación e infraestructura**

El proyecto contribuye al Objetivo de Desarrollo Sostenible 9, ‘Industria, innovación e infraestructura’, al implementar una tecnología para la digitalización y edición de partituras, mejorando la infraestructura tecnológica y fomentando la innovación en el sector.

**Objetivo 12: Producción y consumo responsables**

El proyecto contribuye al Objetivo de Desarrollo Sostenible 12, ‘Producción y consumo responsables’, mediante el uso de la herramienta los usuarios reducirán el consumo de papel, al optar por un formato digital. Esta práctica fomenta prácticas más sostenibles que alienta por un recurso más eco-friendly y eficiente.

**Técnicas y Herramientas**

En esta sección de la memoria se va a listar y describir cada herramienta utilizada en el proyecto. También se van a citar aquellas que se hayan valorado usar para el desarrollo del proyecto o hayan estado presentes en alguna etapa del proyecto pero se hayan descartado, explicando su motivo.

1. **Metodología ágil – Scrum**

La metodología SCRUM se basa en el principio ágil de desarrollo iterativo e incremental.

Con esta metodología se ha llevado a cabo la planificación y gestión de tareas durante el desarrollo del proyecto. Esta metodología implica la descomposición del proyecto en sprints. Cada sprint se compone de tareas definidas en las reuniones, etiquetas para señalar la sección de trabajo de cada tarea, y bloques que indican el estado de cada tarea. Cada sprint comienza y termina con una reunión con el tutor, donde se discuten y tratan diferentes aspectos del proyecto.

1. **Herramienta de control de versiones**

* **Herramienta:** GitHub

GitHub es una herramienta de control de versiones utilizada para gestionar y controlar los cambios realizados en código de forma segura. A su vez permite a los desarrolladores coordinarse en un mismo proyecto contribuyendo a el código mediante la integración de cambios. También es la mayor red de código abierto donde multitud de proyectos están publicados.

Principalmente se ha utilizado esta herramienta por su popularidad y fácil uso, además de para tener un control seguro de los cambios que se han realizado.

1. **Herramienta de gestión del proyecto**

* **Herramienta:** Trello

En este proyecto se ha utilizado Trello para la planificación de las tareas, debido a su facilidad de uso y gratuidad.

1. **Herramienta para realización de la documentación**

* **Herramienta:** Microsoft Word

Se valoró entre el uso de LaTeX o Microsoft Word, pero finalmente se optó por este último dado su fácil uso y conocimiento de sus herramientas. LaTex requiere de un estudio y conocimiento especifico más enfocado en investigaciones científicas y matemáticas que para este proyecto no es del todo necesaria.

1. **Herramienta para gestionar el repositorio local-remoto**

* **Herramienta:** GitHub Desktop

Durante el desarrollo del proyecto se han registrado y gestionado los cambios en el repositorio del proyecto con GitHub Desktop. Esta herramienta permite la interacción rápida con las funcionalidades de GitHub mediante interfaz de usuario para realizar las acciones de Git, facilitando el proceso y haciendo el entorno más cómodo que con el uso de comandos en la línea de comandos. Además, ofrece una minisección con el historial de cambios realizados que ayuda al usuario a visualizar el orden del desarrollo del proyecto.

1. **Herramienta para la gestión de referencias bibliográficas**

* **Herramienta:** Zotero

Zotero es un gestor de referencias bibliográficas que ayuda a recoger y gestionar la documentación e información usada en una investigación o desarrollo.

Se ha utilizado en el proyecto ya que es libre y gratuito y permite insertar las referencias y objetos con gran facilidad en documentos de texto como Microsoft Word. (1)

**Alternativa**: En un inicio se empezó el proyecto contando con Mendeley como gestor de referencias, pero se migró a Zotero ya que el primero tiene un enfoque más comercial además de que limita sus funcionalidades a la versión gratuita, mientras que Zotero es gratis y de código abierto.

1. **Lenguaje de programación**

* **Herramienta:** Python

En este proyecto se ha utilizado Python debido a que este lenguaje simplifica mucho la gestión de dependencias y entornos gracias al gestor de paquetes pip. Además, es ideal para crear aplicaciones rápidamente haciendo cualquier tipo de prototipo. Otra característica que recalcar es que permite trabajar con el patrón MVC para diferenciar claramente estos tres conceptos en la estructura de los datos. (2)

1. **Interfaz gráfica**

* **Herramienta:** Flask

Flask es un micro Framework que ayuda a reducir el trabajo en el desarrollo y creación de aplicaciones web. Se les da este nombre porque desde un inicio tienen lo esencial para poder empezar a desarrollar una aplicación web. Además, su característica principal es que su funcionalidad se puede ampliar a gusto del usuario mediante plugins y librerías. (3)

**Alternativa:** Se ha preferido Flask frente a Django ya que la curva de aprendizaje del primero frente al segundo es notablemente menor, haciendo que el desarrollo sea más ágil. Además, en principio no se requerían todas las funcionalidades que incluye Django para el proyecto, pudiendo ir incluyendo las necesarias mediante librerías en Flask como solución. (4)

1. **Herramienta para la interfaz**

* **Herramienta:** BootStrap

Bootstrap es un framework de front-end muy popular que combina CSS con JavaScript para dar estilo a diferentes elementos de las vistas HTML de interfaces web. Esto ayuda a facilitar la interacción y comunicación con el usuario mediante menús de navegación, botones, formularios, tarjetas, modales y barras de navegación. (5)

Se ha decidido utilizar esta herramienta debido a que no se ha considerado relevante para el proyecto estilizar y maquetar el enfoque visual de la aplicación mediante la creación de fichero CSS y planificación de patrones de diseño.

1. **Herramienta de base de datos**

* **Herramienta:** Cloud Firestore

Cloud Firestore es una base de datos NoSQL dedicada al almacenaje y gestión de los datos de manera dinámica y fácil. Este servicio utiliza documentos y colecciones como estructura de datos jerarquizando estos elementos para relacionarlos y realizar las consultas con facilidad. Además, destaca por escalar automáticamente en función del tamaño del conjunto de resultados. (6)

**Alternativa:** Inicialmente, el proyecto utilizaba MySQL, gestionado a través de phpMyAdmin y el paquete XAMPP. Sin embargo, se decidió migrar a Firebase buscando sus ventajas, como el escalado automático y un fácil uso. Mientras que la primera opción requería migraciones cada vez que se modificaban los modelos, lo que hacía el proceso más tedioso y menos eficiente.

1. **Herramienta de almacenamiento de archivos**

* **Herramienta:** Google Cloud Storage

Esta herramienta es un servicio de almacenaje de objetos en la nube pudiendo estos ser de cualquier tipo y cantidad, siendo posible acceder a estos datos en un momento deseado. El almacenaje se lleva a cabo mediante buckets que son contenedores donde se almacena cada dato para luego ser identificado con facilidad. (7)

En mi proyecto se han utilizado los buckets para almacenar las partituras, tanto procesadas como sin procesar.

1. **Herramienta de encriptación de contraseñas**

* **Herramienta:** Bcrypt

Es una biblioteca que se utiliza para aumentar la seguridad de las aplicaciones haciendo uso de un hasheo intensivo de las credenciales evitando que usuarios malintencionados tengan éxito en sus ataques a la aplicación mediante diferentes estrategias.

1. **Codificación en ASCII**

* **Herramienta:** Base64

Esta herramienta es utilizada para convertir datos binarios en texto ASCII.

En este proyecto aparece en códigos que utilizan esta herramienta para codificar archivos JSON de servicios externos con el propósito de mejorar la seguridad e integración con el Docker de despliegue y el desarrollo de la aplicación.

1. **Herramienta de conversión de PDF a imagen**

* **Herramienta:** pdf2image

Esta herramienta es biblioteca de Python para convertir PDF a imagen. En la aplicación se ha utilizado para asegurarse de que el archivo es una imagen, formato compatible para el realizar el proceso de preprocesamiento.

1. **Herramientas de procesamiento de imágenes**

* **Herramientas:** OpenCV y NumPy

OpenCV es una biblioteca de código abierto que tiene una amplia gama de aplicaciones las cuales pueden abarcar un amplio abanico que pueden ir desde la manipulación de imágenes hasta el reconocimiento de patrones homogéneos en imágenes y el aprendizaje automático. (8)

Numerical Python es una biblioteca utilizada para realizar operaciones matemáticas y estadísticas avanzadas. En el proyecto se ha utilizado para procesar las imágenes en las funciones que manejan la conversión y el procesamiento, alterando las matrices de las imágenes para su manipulación. (9)

1. **Herramienta de digitalización de partituras**

* **Herramienta:** Audiveris

Audiveris es un sotftware de reconocimiento óptico de música de código abierto. Este es utilizado para convertir partituras escaneadas en formato digital. El proceso que es muy parecido al realizado para el reconocimiento de caracteres en textos, pero adaptándolo a elementos musicales como pentagramas, notas, tempos, etc. Normalmente tras el proceso de digitalización el archivo pasa a un formato MusicXML, formato universal y compatible para su interacción con la mayoría de los programas. (10)

**Alternativa:** Al elegir un software a integrar en la aplicación para poder transcribir las partituras se valoró usar, además de Audiveris, SharpEye, Music Reader o MuseScore. Al final se optó por Audiveris porque esta herramienta ofrece resultados de transcripción superiores al resto, lo que aportaba una solución más sólida y adaptable a los intereses de los usuarios que se focalizan en la transcripción. Además Audiveris cuenta con una comunidad más amplia, lo cual ayuda a acceder a una gran variedad de recursos para cualquier tipo de desarrollo con esta herramienta, ya sean foros, repositorios, códigos, etc.

1. **Herramienta de visualización de partituras**

* **Herramienta:** Verovio

Es una biblioteca de software de código abierto que permite la renderización de la música. Permite un gran abanico de tipos de notación que van desde notación musical estándar hasta históricas y modernas. (11)

Se ha utilizado esta herramienta para la visualización de las partituras que han obtenido un resultado aplicable para poder ser visualizadas. Ha entrado en consideración ya que ofrece un buen servicio de representación de notaciones complejas.

1. **Herramienta de despliegue con contenedores**

* **Herramienta:** Docker

Es una plataforma que permite el encapsular maquinas virtuales de diferentes tipos para ser transportadas, copiadas o implementadas, ideal para el despliegue de aplicaciones en la nube. Permite encapsular aplicaciones con todas sus dependencias y librerías en este Docker lo que abre posibilidades de mejorar entornos colaborativos y de testeo gracias a esta portabilidad. Otro punto que recalcar es que asegura que la aplicación se ejecutará igual en producción que localmente al contar con este punto de aislamiento. (12)

Esta herramienta ha sido ideal para la aplicación dada la complejidad nacida de el gran número de dependencias, así como de la utilización de softwares como mini servicios haciendo que la migración y gestión de la aplicación a la capa publica o la nube sea más compleja de gestionar. Por ello los contenedores han ofrecido una solución óptima al permitir aislar y encapsular cada componente con todas sus dependencias.

1. **Herramienta para albergar la aplicación**

* **Herramienta**: Heroku

Heroku es una herramienta que se encarga de la infraestructura de despliegue de una aplicación. Mediante el uso de contenedores virtuales, que son totalmente escalables, almacena y ejecuta las aplicaciones. Además, soporta los principales lenguajes de programación del gremio, pudiéndose instalar mediante buildpacks. (13)

Se valoró desplegar la aplicación con Heroku dado su plan gratuito y su fácil adaptación a los contenedores Docker (se explica más adelante). Al final se revertió la implementación de este servicio dado que el contenedor sobrepasó el limite de memoria que permite el plan inicial.

**Aspectos relevantes de desarrollo del proyecto**

A continuación, se va a citar y explicar cada punto importante en el desarrollo de la aplicación, abordando tanto las decisiones tomadas con sus razonamientos, como los bloqueos y problemas que han surgido en el proceso.

1. **Inicio del proyecto**

La idea de este proyecto nació de la idea, explicada anteriormente, de la necesidad de muchas entidades del mundo de la música por transcribir fragmentos de música, obras, partituras en peligro de deterioro o perdida por sus condiciones deplorables por el paso de tiempo. En concreto muchas universidades e institutos de musicología necesitan de recursos para la enseñanza, Por ello debe existir la posibilidad de poder preservar este material y convertirlo en un formato que sea accesible para alumnos, profesores e investigadores. Dado que hoy en día el formato mas funcional de transmitir datos es digitalmente es necesario el desarrollo de tecnologías que permitan realizar esta conversión de manera efectiva.

Consulté información acerca de este proyecto a José Manuel, para valorar su alcance y posibilidades. Tras la primera reunión, habiendo explicado y aclarado los objetivos, dudas y puntos a alcanzar consideré el proyecto. Motivado por la propuesta y pasión por este arte y cultura en el mundo, acepté el trabajo y empezamos a planear los primeros pasos y preparativos del proyecto.

1. **Metodologías**

Durante el desarrollo del proyecto se ha seguido una metodología ágil de gestión y desarrollo de proyectos Scrum. Se decidió aplicar esta metodología con el fin de concentrarse en la iteración y la mejora continua del proyecto proyectos. Dado que en proyecto ha sido totalmente individualista no se ha sacado provecho del aspecto colaborativo que esta metodología aporta, pero sí en lo que a ciclos iterativos respecta. (14) Las características clave y más notorias en el desarrollo del proyecto son las siguientes:

* Reuniones en cada sprint, conectando cada sprint con ellas con el fin de hablar los avances, cambios y problemas que se han presentado en el sprint anterior y tratando los objetivos, pautas y pasos a seguir en el siguiente.
* Se ha llevado un enfoque incremental.
* Se han utilizado etiquetas para clasificar la categoría de cada tarea, dividiéndolas en: Development (Desarrollo), Documentation (Documentación), Configuration (Configuración), Investigación (Research), Discusión (Discuss) y Completado (Completed).
* Se han utilizado bloques para marcar el estado de cada tarea habiendo: Reuniones donde se especifica el momento en el que se han realizado reuniones, tareas por hacer, tareas en curso, tareas bloqueadas y tareas completadas.
* Se ha utilizado Trello para la gestión de todas estas características.
  + Cada tarea ha sido estimada por día, fijando un tiempo estimado que debe llevar o en el que se debe trabajar en esa tarea y el tiempo real que ha llevado. Se ha llevado este procedimiento para tener una visión clara del progreso y gestión del tiempo, así como para detectar la desviación de tiempos debido a errores o bloqueos.

1. **Formación**

Algunas de las herramientas utilizadas en el proyecto han requerido de una investigación y un estudio previo para poder desarrollar un trabajo con su uso. Este estudio ha podido ser desde tutoriales o listas de reproducción de youtube con guías, hasta revisar la documentación oficial de la herramienta en cuestión.

**GitHub**

A pesar de que esta herramienta es crucial y básica en el mundo de la programación, fue necesario hacer un miniestudio en el que revisar sus funcionalidades principales empezando por Git y su adaptación en la interfaz de GitHub. Finalmente, para disminuir el costo de manejar comandos por cada cambio se decidió integrar el uso de GitHub Desktop que ofrece una interfaz de usuario cómoda e intuitiva.

**Bootstrap**

Con el fin de no profundizar en tareas de diseño y maquetación, se decidió utilizar la librería Bootstrap, implementándola en todas las vistas con un diseño estándar. De esta forma se ahorró tiempo y recursos y a su vez se consiguió una interfaz de usuario válida. Para conocer la implementación necesaria en mis vistas se accedió a la documentación oficial de Bootstrap:

<https://getbootstrap.com/docs/4.1/getting-started/introduction/> (15)

**Flask**

Este framework ha sido uno de los pilares más importantes del proyecto y mas detalladamente de la aplicación. Si bien durante la carrera ya se había trabajado con Python en varias asignaturas, fue necesario conocer los aspectos y estructura del entorno para poder realizar el desarrollo correcto. Por ello se investigó en varios cursos y tutoriales para dar con el contenido mas completo. Finalmente

<https://www.youtube.com/watch?v=QG-RW4XpqTg&list=PLsvvBhdpMqBzPV-kZ_GfsdP_SqQcQQ68C>

**Bibliografía**

1. Delgado-Vázquez, Ángel. «Guías BibUpo: Zotero: Qué es Zotero». Accedido 18 de junio de 2024. <https://guiasbib.upo.es/zotero/que_es_zotero>.
2. «Tool recommendations - Python Packaging User Guide», 18 de junio de 2024..
3. OpenWebinars.net. «Qué es Flask y ventajas que ofrece | OpenWebinars». Accedido 18 de junio de 2024. <https://openwebinars.net/blog/que-es-flask/>.
4. Kinsta®. «Flask vs Django: Let’s Choose Your Next Python Framework», 5 de junio de 2023. <https://kinsta.com/blog/flask-vs-django/>.

The JetBrains Blog. «Django vs Flask: Which Is the Best Python Web Framework? | The PyCharm Blog», 10 de noviembre de 2023. <https://blog.jetbrains.com/pycharm/2023/11/django-vs-flask-which-is-the-best-python-web-framework/>

Django Project. «Django». Accedido 13 de marzo de 2024. <https://www.djangoproject.com/>.

1. Rock Content - ES. «Bootstrap: ¿qué es, para qué sirve y cómo instalarlo?», 12 de abril de 2020. <https://rockcontent.com/es/blog/bootstrap/>.
2. Firebase. «Cloud Firestore | Almacena y sincroniza los datos de tu app a escala global». Accedido 18 de junio de 2024. <https://firebase.google.com/products/firestore?hl=es-419>.

contributors, phpMyAdmin. «phpMyAdmin». phpMyAdmin. Accedido 3 de abril de 2024. <https://www.phpmyadmin.net/>.

«MySQL». Accedido 3 de abril de 2024. <https://www.mysql.com/>.

1. Google Cloud. «Cloud Storage». Accedido 19 de junio de 2024. <https://cloud.google.com/storage>.
2. «Procesamiento de imágenes con OpenCV en Python». Accedido 19 de junio de 2024. <https://imaginaformacion.com/tutoriales/opencv-en-python>.
3. Formación en ciencia de datos | DataScientest.com. «NumPy : La biblioteca de Python más utilizada en Data Science», 18 de enero de 2023. <https://datascientest.com/es/numpy-la-biblioteca-python>.
4. «Audiveris/audiveris». Java. 2016. Reprint, Audiveris, 15 de marzo de 2024. https://github.com/Audiveris/audiveris.
5. Reference book for Verovio. «About This Book», 16 de mayo de 2024. <https://book.verovio.org/introduction/>.

«Optical Music Recognition». En Wikipedia, 12 de febrero de 2024. <https://en.wikipedia.org/w/index.php?title=Optical_music_recognition&oldid=1206438514>.

1. «¿Qué es Docker y cómo funciona? Ventajas de los contenedores Docker». Accedido 19 de junio de 2024. <https://www.redhat.com/es/topics/containers/what-is-docker>.
2. «¿Qué es Heroku? ¿Para qué sirve? ¿Con Salesforce cómo se integra? | SEIDOR». Accedido 18 de junio de 2024. <https://www.seidor.com/blog/heroku-que-es>.
3. Atlassian. «¿Qué es scrum? [+ Cómo empezar]». Atlassian. Accedido 21 de junio de 2024. <https://www.atlassian.com/es/agile/scrum>.
4. contributors, Mark Otto, Jacob Thornton, and Bootstrap. «Introduction». Accedido 21 de junio de 2024. https://getbootstrap.com/docs/4.1/getting-started/introduction/.