**TRABAJO DE FIN DE GRADO:**

**DIGITALIZACIÓN DE PARTITURAS**

Universidad de Burgos



Autor:

**Martín González Saiz**

Tutor académico:

**José Manuel Galán Ordax**

**GRADO EN INGENIERÍA INFORMÁTICA**

Universidad de Burgos

Escuela Politécnica Superior

2024

# Sprint 1: 15/02/2024 - 01/03/2024

En la primera reunión para el primer sprint planeamos y acordamos preparar todos los repositorios en cada una de las plataformas, además de todos los programas, que son utilizados a lo largo del desarrollo del proyecto. Por tanto en estas semanas creé repositorios para GitHub y Trello.

GitHub para poder reflejar la evolución del trabajo a nivel de código y programación así como para tener un buen control de las versiones y cambios, mientras que Trello reflejaría el trabajo que hay detrás de todo el tiempo de investigación, desarrollo y estudio de diferentes aspectos del proyecto.

Además instalé y configuré Mendeley para poder registrar todas las referencias a las que haya accedido para posteriormente poder implementarlo en la memoria. También me serviría para poder acceder a paginas útiles siempre que quiera durante el desarrollo de la herramienta.

Otro aspecto importante fue en estudio del entorno en el que se utilizaría la aplicación. Hice una investigación para elegir entre una aplicación web o una de escritorio, valorando sus ventajas e inconvenientes.

Por último me decanté por desarrollar la memoria en Word por su simplicidad en comparación con Latex.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fecha | Tarea | Etiqueta | Estado | Tiempo estimado | Tiempo real |
| 16/02/2024 | Creación y preparación de la tabla de Trello | Configuration | En curso | 0.5 | 0.25 |
| 19/02/2024 | Creación del repositorio de GitHub | Configuration | Completado | 0.5 | 0.25 |
| 20/02/2024 | Instalación y configuración de Mendeley | Configuration | Completado | 1 | 1 |
| 21/02/2024 | Estudio de GitHub | Research | En curso | 1.5 | 2 |
| 22/02/2024 | Estudio de GitHub | Research | En curso | 1.5 | 1.5 |
| 23/02/2024 | Estudio de GitHub | Research | Completado | 1 | 1 |
| 26/02/2024 | Investigar sobre aplicaciones web | Research | Completado | 2 | 1.5 |
| 27/02/2024 | Investigar sobre aplicaciones escritorio | Research | Completado | 2 | 1.5 |
| 28/02/2024 | Investigación general de herramientas OMR | Research | Completado | 2 | 1.5 |
| 29/02/2024 | Decidir entorno en el que desarrollar la memoria | Research | Completado | 1 | 0.75 |

# Sprint 2: 01/03/2024 –17/03/2024

Este sprint se dedicó el trabajo a la investigación y el estudio de las herramientas a utilizar en el proyecto para levantar un servidor web. Entre ellas se valoró Flask o Django. Flask ofrece mucha simplicidad y flexibilidad dejando libertad para la implementación de otras herramientas y librerías, además, su aprendizaje es muy rápido permitiendo tener un mayor control en el desarrollo de la aplicación. Sin embargo, puede ser un problema si el proyecto crece lo suficiente con demasiadas funcionalidades, ya que habría que implementar muchas características. Por otro lado, Django es un framework que ofrece una amplia variedad de funcionalidades integradas de serie, también permite un desarrollo rápido de aplicaciones complejas gracias a estas características, así como escalabilidad para cualquier proyecto. Valorando todos estos aspectos y analizando la presente y futura situación he considerado que la mejor opción es usar Flask para mi proyecto.

También realicé una investigación de los paquetes OMR más sólidos y de código abierto en la actualidad. La elección más llamativa es Audiveris debido a su larga trayectoria y la comunidad que hay detrás ya que es muy activa y grande, además de por su amplia documentación, explicado cada punto con detalle. Otras tecnologías que valoré son SharpEye Music Reader, MuseScore.

En este sprint también cambié el plan estructural de las tablas de Trello y la base de datos donde guardo las referencias de Mendeley a Zotero.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Tarea | Etiqueta | Estado | Tiempo estimado | Tiempo real |
| 06/03/2024 | Cambiar el plan estructural de Trello | Documentation | Completado | 0.5 | 0.3 |
| 07/03/2024 | Cambiar la base de datos de referencias de Mendeley a Zotero | Documentation | Completado | 0.25 | 0.25 |
| 10/03/2024 | Documentar Sprint 1 | Documentation | En curso | 1 | 2 |
| 11/03/2024 | Documentar Sprint 1 | Documentation | Completado | 1 | 1 |
| 12/03/2024 | Documentar Sprint 2 y avances de los paquetes OMR en la memoria | Documentation | En curso | 1 | 1.5 |
| 12/03/2024 | Investigar servidor web: Django vs Flask | Research | En curso | 1 | 1 |
| 13/03/2024 | Investigar servidor web: Django vs Flask | Research | En curso | 2 | 2 |
| 14/03/2024 | Investigar servidor web: Django vs Flask | Research | Completado | 1 | 2 |
| 16/03/2024 | Investigar paquetes OMR para mi proyecto y elegir el más ideal | Research | Completado | 2 | 2.5 |

# Sprint 3: 18/03/2024 – 15/04/2024

En esta etapa comencé con el desarrollo de la aplicación web, enfocándome en las funcionalidades más básicas de esta.

En primer lugar planteé una primera versión de los tipos de usuarios que habrá en mi aplicación, empezando por un usuario corriente y un usuario registrado. Además le asigné unas primeras funcionalidades básicas a estos usuarios, que son entrar a la ventana home, logearse y registrarse en el caso del usuario corriente y poder subir ficheros y hacer log out en el caso de un usuario registrado como funciones extra para este.

Tras ello creé un entorno virtual en el que trabajaré para el desarrollo de mi aplicación. Tomé esta decisión con el fin de evitar conflictos de dependencias de paquetes y versiones con otros proyectos.

Después creé una primera versión de mi proyecto siguiendo una estructura de ficheros común en este tipo de proyectos, luego instalé Flask. Gestionar el fichero de configuración en este punto fue clave para poder hacer las migraciones del modelo de las tablas a la base de datos.

Hice una primera implementación de un login y register haciendo uso de la librería de FlaskWTF para facilitarme hacer los forms. Como tuve problemas con el uso adecuado de esta herramienta cambié el código, retirando esta librería.

Para poder almacenar las entradas en base de datos los usuarios registrados instalé xampp y configuré la base de datos en phpadmin. Elegí este entorno debido a su facilidad de uso y su libre acceso a cualquier navegador web.

Para reflejar los modelos creados en código en base de datos, uso migraciones mediante comandos en la terminal para que las tablas se creen automáticamente. Estos pasos me dieron varios problemas debido a que la carpeta de migraciones almacenó más de un cambio y se solaparon entre ellos. Además los ficheros de la caché almacenaban datos antiguos apareciendo atributos fantasma. Para arreglarlo borré la carpeta y recreé la base de datos.

Por último terminé de vincular las ventanas y de estructurar la aplicación.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Tarea | Etiqueta | Estado | Tiempo estimado | Tiempo real |
| 19/03/2024 | Documentar Sprint 2 | Documentation | Completado | 1 | 1.5 |
| 20/03/2024 | Cambiar las gráficas de los sprints | Documentation | Completado | 1 | 1 |
| 21/03/2024 | Plantear los tipos de usuarios, sus roles y funcionalidades | Research | En curso | 4 | 0.5 |
| 21/03/2024 | Preparar la aplicación web | Development | En curso | 2 | 1.5 |
| 24/03/2024 | Desarrollo de ventana login y register | Development | En curso | 2 | 2 |
| 01/04/2024 | Desarrollo de ventana login y register | Development | En curso | 2 | 1.6 |
| 02/04/2024 | Desarrollo de ventana login y register | Development | En curso | 3 | 3 |
| 03/04/2024 | Instalación e implementación de xampp | Configuration | Completado | 1 | 1.5 |
| 03/04/2024 | Instalación e implementación de phpMyAdmin | Configuration | Completado | 2 | 2 |
| 04/04/2024 | Desarrollo de ventana login y register | Development | En curso | 2.5 | 2.5 |
| 07/04/2024 | Desarrollo de ventana login y register | Development | En curso | 2 | 2 |
| 08/04/2024 | Desarrollo de ventana login y register | Development | En curso | 3 | 3 |
| 09/04/2024 | Desarrollo de ventana login y register | Development | En curso | 3 | 3 |
| 10/04/2024 | Desarrollo de ventana login y register | Development | Completado | 1 | 1 |
| 10/04/2024 | Vinculación entre ventanas | Development | En curso | 1 | 1 |
| 10/04/2024 | Desarrollo de registro de partituras | Development | Completado | 1 | 1 |
| 11/04/2024 | Gestión de usuarios y sesiones | Development | En curso | 2 | 2 |
| 15/04/2024 | Documentación del sprint 3 | Documentation | Completado | 3 | 3 |

# Sprint 4: 15/04/2024 – 22/05/2024

Este sprint está enfocado en la importación de la tecnología OMR Audiveris en mi aplicación.

Para ello, primero, tuve que investigar y revisar bien la documentación que aportan los autores de la herramienta a los desarrolladores interesados en esta. Cloné la herramienta en mi sistema y la ejecuté.

Tuve problemas para ejecutarlo porque el nombre de los ficheros y archivos no se correspondían del todo con los ejemplos de la documentación.

Analicé e investigué los diferentes aspectos del programa para saber como se trabaja con esta, a la par que me documentaba con el pdf de ayuda para los desarrolladores.

Luego tuve que cambiar el formato en el que se aceptaba los archivos subidos a mi aplicación (solo se podía pdfs) para darle flexibilidad a la aplicación.

Creé un proyecto en Google Cloud para tener un servicio de almacenamiento para las partituras en la nube, me dio alguna dificultad adaptarlo a mi proyecto por lo que deseché la idea por no malgastar tiempo.

Intenté encapsular mi proyecto junto con audiveris para moverlo a un servicio de nube que desplegara mi proyecto públicamente para abrir su acceso a los usuarios, también aseguraría que el comportamiento es el mismo en un entorno de producción y localmente. Tuve que registrar todas las dependencias del proyecto y registrar las instalaciones del terminal del docker en un fichero que ejecutaría más tarde. Tuve problemas por la compatibilidad de los paquetes al construir el docker y, tras construirlo, el acceso a las rutas de las carpetas tuvo muchas ambigüedades, las cuales tuve que resolver. Otro inconveniente fue que en un inicio por cada cambio en el código del proyecto tuve que construir un nuevo contenedor e imagen, por lo que el nombre de estos se pisaban, ralentizaban el ordenador y colapsaron el almacenamiento. Dejé aparcada la idea debido a la ralentización del proceso y la incertidumbre de si llegaría a algo funcional.

Volví con la programación de una función que ejecutara Audiveris como un miniservicio de mi aplicación en segundo plano. Me documenté de las vías de ejecución que tiene Audiveris en su documentación para construir un comando que ejecute el servicio sin interfaz de usuario, haga el proceso de digitalización y exporte el fichero resultante. La construcción de este comando nuevamente dio problemas que llevaron un tiempo resolver. Esto se debe a que la ubicación de los ficheros de ejecución de java (.jar) no eran accesibles para la carpeta del proyecto. Cloné el repositorio de Audiveris a el directorio de la aplicación web para facilitar el acceso. También tuve que cambiar la versión de mi carpeta jdk de Java a la 17 y configurarlo en mis variables de entorno para asegurar un buen comportamiento, como indican las especificaciones de la documentación de Audiveris.

Mejoré las vistas de mi aplicación web con bootstrap para darle un mejor diseño a la web y un toque más profesional.

Exploré las posibilidades para el acceso a metadatos en base de datos en Firebase. La configuración y adaptación de este servicio a mi proyecto llevaron tiempo, debido a las dificultades que se me presentaron para la importación de paquetes así como adaptarme a la nueva estructura, sintaxis y funciones de este.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Tarea | Etiqueta | Estado | Tiempo estimado | Tiempo real |
| 16/04/2024 | Documentación del sprint 3 y cambios en las tablas y gráficos | Documentation | Completado | 0.5 | 0.5 |
| 23/04/2024 | Investigación para la implementación de la tecnología OMR | Research | En curso | 2 | 2 |
| 24/04/2024 | Investigación para la implementación de la tecnología OMR | Research | En curso | 2 | 2 |
| 25/04/2024 | Investigación para la implementación de la tecnología OMR | Research | En curso | 2 | 2 |
| 27/04/2024 | Subir archivos en todos los formatos | Development | Completado | 2 | 2 |
| 27/04/2024 | Investigación para la implementación de la tecnología OMR | Research | En curso | 3 | 3 |
| 30/04/2024 | Investigación para la implementación de la tecnología OMR | Research | En curso | 1.5 | 1 |
| 01/05/2024 | Investigación para la implementación de la tecnología OMR | Research | En curso | 4 | 4 |
| 03/05/2024 | Implementación de servicio de almacenamiento | Development | En curso | 3 | 2 |
| 04/05/2024 | Implementación de servicio de almacenamiento | Development | En curso | 5 | 5 |
| 05/05/2024 | Implementación de un docker para Audiveris | Development | En curso | 3 | 3 |
| 06/05/2024 | Implementación de un docker para Audiveris | Development | En curso | 5.5 | 5 |
| 07/05/2024 | Implementación de un docker para Audiveris | Development | En curso | 3 | 3 |
| 08/05/2024 | Implementación de un docker para Audiveris | Development | En curso | 2 | 2 |
| 09/05/2024 | Implementación de un docker para Audiveris | Development | En curso | 2.5 | 2.5 |
| 11/05/2024 | Implementación de un docker para Audiveris | Development | Completado | 1 | 1 |
| 11/05/2024 | Implementación de una funcionalidad que ejecute audiveris | Development | En curso | 2 | 2.5 |
| 12/05/2024 | Implementación de una funcionalidad que ejecute audiveris | Development | En curso | 7.5 | 8 |
| 13/05/2024 | Implementación de una funcionalidad que ejecute audiveris | Development | En curso | 5 | 5 |
| 14/05/2024 | Implementación de una funcionalidad que ejecute audiveris | Development | En curso | 3.5 | 3.5 |
| 15/05/2024 | Implementación de una funcionalidad que ejecute audiveris | Development | En curso | 4 | 4 |
| 17/05/2024 | Implementación de una funcionalidad que ejecute audiveris | Development | En curso | 3 | 3 |
| 18/05/2024 | Implementación de una funcionalidad que ejecute audiveris | Development | Completado | 3 | 4.5 |
| 18/05/2024 | Mejora de las vistas | Development | Completado | 2 | 2 |
| 18/05/2024 | Implementación de Firebase | Research | En curso | 2 | 2 |
| 19/05/2024 | Implementación de Firebase | Development | En curso | 2 | 2 |
| 19/05/2024 | Implementación de Firebase | Development | En curso | 3 | 3 |
| 20/05/2024 | Implementación de Firebase | Development | En curso | 1.5 | 2 |
| 20/05/2024 | Implementación de Firebase | Development | En curso | 3 | 5.5 |
| 21/05/2024 | Implementación de Firebase | Development | En curso | 5 | 5 |
| 22/05/2024 | Implementación de Firebase | Development | En curso | 5 | 7 |

# Sprint 5: 23/05/2024 – 20/06/2024

En este sprint valoré otras alternativas para la digitalización de partituras. Consideré hacer uso de Firebase ML que permite integrar aprendizaje automático en aplicaciones web. De esta forma podría implementarlo en mi web para que, en vez de usar Audiveris para transcribir las partituras, usar esta IA. Valoré la posibilidad de usar un modelo preentrenado con esta funcionalidad para facilitar el proceso, pero no obtuve resultados viables. Investigué como podría entrenar una por mi cuenta en base a mis preferencias, pero no lo llevé adelante debido a la falta de experiencia en este sector y el tiempo que me llevaría desarrollarlo y en base al tiempo que me quedaba.

También empecé la documentación de la memoria empezando por puntos esenciales y sencillos de implementar, como los objetivos de desarrollo sostenible o la introducción.

Retomé la funcionalidad de ejecutar Audiveris construyendo un comando que se ejecutaría como batch en segundo plano, además también replanteé la estructura de las funciones de cada archivo del directorio, ordenando las definiciones, evitando que Firebase se inicializara más veces de las necesarias.

El siguiente paso fue implementar una funcionalidad que permitiera visualizar las partituras que se consiguieran digitalizar. Para ello usé Verovio, instalándolo junto con sus dependencias en mi máquina. Tuve bastantes problemas en este punto ya que hubo muchos conflictos con la instalación de las dependencias y con las correcta configuración y establecimiento en mis variables de entorno. Incluí la funcionalidad de visualizar las partituras y Verovio en las vistas con bibliotecas, mostrándose las partituras mediante bloques y scripts.

Luego implementé una funcionalidad para que el usuario pudiera eliminar las partituras que desee en el listado de partituras registradas en la aplicación y Firebase. Aparqué la implementación debido a que me dio problemas con el acceso al bucket y al servicio de almacenamiento de Firebase.

Añadí alertas a las funcionalidades de la aplicación para que el usuario se mantenga informado acerca de los procesos fallidos al intentar ciertas interacciones.

Aprovechando esta última implementación, añadí una funcionalidad para que, en caso de que la digitalización de una partitura fallé, el usuario pueda alterar la imagen llegando a un proceso de preprocesamiento, facilitando a Audiveris poder procesar la imagen y llegar a una digitalización completa. Esta funcionalidad consiste en que el usuario inserta valores a funciones de preprocesamiento.

Por último, empecé el desarrollo para desplegar la aplicación públicamente. Para ello tomé dos caminos, el primero fue desplegar la aplicación haciendo uso de un archivo Procfile y buildpacks en Heroku. Esta opción me dio problemas debido a que Heroku no permite la ejecución de scripts en tiempo de compilación sin configurar sus buildpacks, y ello conllevaba realizar las instalaciones necesarias de Audiveris en estos. Además los archivos que se generaban con los scripts no persistían por alguna razón. La opción que dio resultados fue utilizar el Docker creado para tener un mejor control del entorno de ejecución y usar Heroku para configurar la aplicación y desplegarla. También dio problemas por no estar bien adaptada al contexto la inicialización de la aplicación en código y los errores en la interpretaciones del puerto en el Dockerfile al desplegar la aplicación con Heroku. El despliegue esta terminado pero la funcionalidad de digitalizar las partituras no se puede completar en la versión desplegada de Heroku ya que sobrepasa la memoria permitida en mi plan gratuito de mi cuenta.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Tarea | Etiqueta | Estado | Tiempo estimado | Tiempo real |
| 23/05/2024 | Documentar sprint 4 | Documentation | Completado | 2 | 3 |
| 24/05/2024 | Investigar herramientas de procesamiento de imágenes para Firebase | Research | En curso | 0.4 | 0.4 |
| 24/05/2024 | Investigar herramientas de procesamiento de imágenes para Firebase | Research | En curso | 1 | 1 |
| 25/05/2024 | Documentación de la memoria | Documentation | En curso | 1 | 1.5 |
| 26/05/2024 | Reimplementación de un docker | Development | Completado | 1 | 2 |
| 26/05/2024 | Documentación de la memoria | Documentation | En curso | 1 | 1 |
| 27/05/2024 | Reimplementación de audiveris | Development | En curso | 2 | 2 |
| 28/05/2024 | Reimplementación de audiveris | Development | Completado | 2 | 3 |
| 28/05/2024 | Reestructuración para Firebase | Development | Completado | 2 | 3.5 |
| 31/05/2024 | Documentar la memoria | Documentation | En curso | 1 | 2.5 |
| 01/06/2024 | Implementación de Verovio | Development | En curso | 3 | 3 |
| 02/06/2024 | Implementación de Verovio | Development | En curso | 2 | 2 |
| 03/06/2024 | Implementación de Verovio | Development | En curso | 2 | 2 |
| 06/06/2024 | Implementación de Verovio | Development | En curso | 6 | 6 |
| 07/06/2024 | Implementación de Verovio | Development | En curso | 7 | 7 |
| 08/06/2024 | Implementación de Verovio | Development | En curso | 8 | 10 |
| 09/06/2024 | Implementación de Verovio | Development | Completado | 9 | 5 |
| 09/06/2024 | Funcionalidad para eliminar partituras | Development | Bloqueado | 2 | 2 |
| 10/06/2024 | Arreglos en la aplicación | Development | Completado | 3 | 6.5 |
| 11/06/2024 | Alertas en las vistas | Development | Completado | 4 | 6 |
| 12/06/2024 | Función de preprocesamiento | Development | Completado | 4 | 5.5 |
| 13/06/2024 | Despliegue de la aplicación | Development | En curso | 5 | 8 |
| 14/06/2024 | Despliegue de la aplicación | Development | En curso | 4 | 8 |
| 15/06/2024 | Despliegue de la aplicación | Development | En curso | 5 | 10 |
| 16/06/2024 | Despliegue de la aplicación | Development | Bloqueado | 5 | 7.5 |
| 17/06/2024 | Documentación del sprint 5 | Documentation | Completado | 3 | 5 |
| 18/06/2024 | Documentación de la memoria | Documentation | En curso | 5 | 7 |
| 19/06/2024 | Documentación de la memoria | Documentation | En curso | 7.5 | 7.5 |
| 20/06/2024 | Funcionalidad para eliminar partituras | Development | Completado | 3 | 3 |
| 20/06/2024 | Documentación de la memoria | Documentation | En curso | 4 | 4 |

# Sprint 6: 23/05/2024 – 17/06/2024

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Tarea | Etiqueta | Estado | Tiempo estimado | Tiempo real |
| 21/06/2024 | Documentación de la memoria | Documentation | En curso | 8 | 8 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |