CFGS DESARROLLO DE APLICACIONES MULTIPLATAFORMA

GENERALITAT VALENCIANA
Conselleria d'Educació, Cultura i Esport

Florida Universitària

Proyecto Final de Ciclo

Dashboard-Musicos 2023-2024

Apellidos y nombre del autor/a: David Reinón García

Fecha de entrega: 24/05/2024

Contenido

1. RESUMEN DEL PROYECTO: Que se propone y de que está formado

Castellano:

Con este proyecto se propone crear una aplicación para toda la gestión interna sobre músicos y eventos (particiones) de una banda de música, en este caso la SJMA (Societat Juventut Musical d'Albal), ya que este proyecto va a ser en un entorno real y va a usarse una vez se finalice.

Podría considerarse un 'Dashboard', que es un tipo de aplicación con una funcionalidad y propósito claro. Podemos pensar en el dashboard como una especie de "resumen" que recopila datos de diferentes fuentes en un solo sitio y los presenta de manera digerible para que lo más importante salte a la vista (Ortiz, 2024).

Por esto mismo suele desarrollarse siempre con un diseño bastante marcado, donde predomina la vista de información por elementos, y funcionalidades muy simples y fáciles de usar.

Dentro de esta idea, el proyecto nace ya que normalmente las bandas, cuando van a algún acto contratados por alguna asociación, (no es el caso de un concierto organizado por la propia banda), como por ejemplo una falla en época de fallas, no se convocan a todos los músicos para que vayan, ya que son demasiados, por lo cual se crean particiones para seleccionar que músicos van a ir.

Estas particiones no dejan de ser una selección de los músicos que van a ir al acto junto más información relacionada con este. Y una de las funcionalidades principales de esta aplicación será poder crearlas de una forma fácil y lo más justo posible para los músicos, mediante un algoritmo que tiene en cuenta los ensayos acudidos y en cuantos actos has participado ya.

La aplicación la podremos ver que estará formado por las diferentes pantallas de despliegue de la información, dividido principalmente en: Home (Vista general), Músicos, Ensayos, Particiones, Instrumentos.

En el apartado de Particiones, aparte de poder ver todas las que se han creado hasta ahora, se van a poder crear más, mediante un formulario. Donde usando

el algoritmo mencionado anteriormente podremos ver la prioridad que tiene cada musico, a partir de ahí, el administrador puede decidir su elección.

Otra parte importante, es que tendremos otra pantalla más, con la opción para insertar nueva información a la BD. Esto se hará mediante un archivo CSV con mucha cantidad y columnas de información sobre músicos, ensayos y la respectiva asistencia a este, y en un formato especifico.

En el siguiente punto donde expongo los objetivos del proyecto y una justificación razonable explicaré porque se hace de esta forma tan concreta mediante CSV y no con formularios estándar.

Ingles:

This project proposes to create an application for the internal management of musicians and events (partitions) for a music band, in this case, the SJMA (Societat Juventut Musical d'Albal), as this project will be in a real environment and will be used once it is completed.

It could be considered a 'Dashboard', which is a type of app with a clear functionality and purpose. We can think of the dashboard as a kind of "summary" that compiles data from different sources into one place and presents it in a digestible manner so that the most important information stands out (Ortiz, 2024).

For this reason, it is usually developed with a very marked design, where the view of information by elements predominates, and the functionalities are very simple and easy to use.

Within this idea, the project arises because normally bands, when they are hired by an association for an event (not the case of a concert organized by the band itself), such as a "falla" during the "fallas" festival, do not summon all the musicians to go, since they are too many. Therefore, partitions are created to select which musicians will go.

These partitions are essentially a selection of the musicians who will attend the event along with more related information. One of the main functionalities of this application will be the ability to create these partitions easily and as fairly as possible for the musicians, using an algorithm that takes into account the rehearsals attended and the number of events they have already participated in.

The application will consist of different information display screens, mainly divided into: Home (General View), Musicians, Rehearsals, Partitions, and Instruments.

In the Partitions section, in addition to being able to see all those created so far, more can be created using a form. Using the aforementioned algorithm, we can see the priority of each musician, and from there, the administrator can make their selection.

Another important part is that we will have another screen with the option to insert new information into the database. This will be done via a CSV file with a large amount of information and columns about musicians, rehearsals, and their respective attendance, in a specific format.

In the next section where I present the project objectives and a reasonable justification, I will explain why this specific method using CSV is employed instead of standard forms.

2. JUSTIFICACIÓN Y OBJETIVOS DEL PROYECTO: Para que se propone, a quien se dirige y que se pretende conseguir.

Como he comentado anteriormente, este proyecto se propone para satisfacer una necesidad real, para le gestión de una asociación, en este caso una banda de música. Pretende crear una solución principalmente para la mejora en la creación de las particiones, y a raíz de aquí, surge crear un Dashboard para la mejora de toda la gestión de información dentro de la banda y hacer una aplicación más completa para automatizar muchas tareas rutinarias o que se suelen repetir a lo largo del tiempo.

Cabe destacar, que ya existe una aplicación disponible globalmente que es muy famosa, llamada **Glissando**, dedicada a la gestión cultural específicamente de entidades musicales, como orquestas, bandas, charangas, etc. donde puedes ver todos los músicos que forman la banda, convocar ensayos, que los proprios músicos puedan confirmar si van a ir o no, y más funcionalidades muy útiles. De hecho, cuenta con muchos usuarios activos y la propia SJMA la utiliza.

Por lo cual, podríamos pensar, que para que esté desarrollando este proyecto, si ya hay una solución mucho más grande e integrada en este ámbito, y que además mi cliente la está utilizando.

Bien…, se trata de algo distinto, ya que, por una parte, Glissando no cuenta con un apartado de particiones ni nada por el estilo (ya que en muchas entidades musicales no se hacen) y es una de las principales características que se pretende implementar en este Dashboard.

Por lo cual se podría considerar, que mi desarrollo, por un lado, esta **complementado** a esta aplicación tan estandarizada, es más, por eso existe la sección de insertar información con un archivo CSV con un formato en concreto, porque se exportará la información ya disponible de Glissando, y este, ofrece esta forma de exportar información. Por otro lado, también sirve para una **propia gestión y análisis interno de la información** de la banda, basado en un desarrollo completo a medida.

En general, este proyecto va dirigido, tanto para **las personas que se encargan de la gestión de la banda (la junta),** para mejorar su proceso de decisión en las particiones y, además, para **los propios músicos**, porque de forma transparente, se les va a poder mostrar cómo funciona el proceso, y en que se basa la prioridad a la hora de elegir a unos o a otros.

Normalmente para decidir esto, se suelen basar, primero de todo, en los instrumentos que hacen falta para ese evento, y que músicos hay disponibles para cada uno de estos instrumentos, y segundo, la asistencia a los ensayos, ya que como es lógico, quien más asiste a los ensayos, y participa más en la banda, tiene prioridad.

El problema es que hacer esto a mano puede llegar a ser costoso, ya que tienes que analizar todo esto viendo los datos desde un Excel, o algún programa de este estilo para poder visualizar toda la información y realizar cálculos, y partir de ahí decidirlo.

Luego que se puede llegar a generar un mal ambiente en ciertos momentos, ya que algunos músicos pueden llegar a pensar en favoritismo cuando se convoca ‘a unos si, y a otros no’. Cuanto más transparente sea todo, mucho mejor.

La transparencia empresarial es más que una tendencia, es una necesidad en el mundo empresarial contemporáneo que beneficia tanto a las empresas como a la sociedad. Implementar prácticas transparentes mejora la reputación, fomenta una cultura de confianza y colaboración, y contribuye al desarrollo sostenible. (SAP españa, 2024)

3. DESARROLLO DEL PROYECTO

a. Metodologías utilizadas

SCRUM:

La principal metodología utilizada para todo el proceso de organización y durante el desarrollo del proyecto es **SCRUM**. Ya que así he podido definir unos objetivos claros, restructurados en etapas (sprints), y donde al final de cada etapa podía tener una retroalimentación, en este caso de nuestro tutor del PFC y grupo de alumnos.

SCRUM es un marco de gestión de proyectos de metodología ágil, prácticamente la más conocida y utilizada desde hace muchos años en el ámbito de desarrollo de software, aunque se pude aplicar a todo tipo de trabajos.

Agile es una filosofía o enfoque para el desarrollo de software que se centra en la flexibilidad, la colaboración y la respuesta rápida a los cambios. En lugar de seguir un plan rígido, Agile promueve la adaptación continua y la entrega incremental de software funcional. (Soler, 2024)

Cuenta con unos objetivos/tareas marcadas desde un principio, con el llamado Backlog que es común dentro de esta metodología, en el cual no deja de ser, al empezar un sprint, redactar todas las tareas que se van a tener que realizar en este, teniendo en cuenta el tiempo que vas a tener y otros factores del equipo y proyecto.

Un sprint es la terminología clave y en lo que se basa principalmente SCRUM, lo que la hace tan flexible y amoldable en el tiempo, ya que en un Sprint que suele durar entre dos semanas y un mes (no es obligado siempre hacerlo entre este rango de tiempo) debes tener una versión del proyecto hecha.

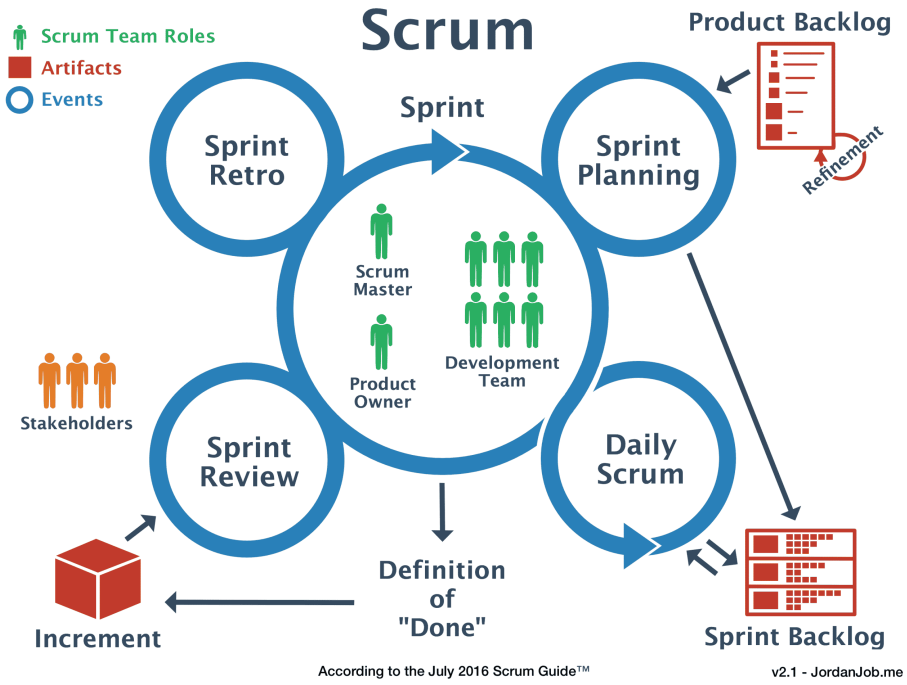
Por lo cual en un sprint has tenido que hacer todas las fases de gestión, diseño, desarrollo, pruebas, documentación, etc. de lo que tenías agendado en ese periodo de tiempo.

Lo que hace que una vez finalizado todo pueda haber retroalimentación de diferentes grupos de personas, desde el propio equipo de desarrollo, el Project Mánager e incluso el cliente, y poder ver mejoras y llegar a una conclusión. Y si por lo que sea no se está enfocando el proyecto por el camino correcto o hay que cambiar cosas, en el siguiente sprint se puede volver a cualquier fase de gestión y reestructurarla o mejorarla.

Evitando estipular un tiempo largo para desarrollar la aplicación completa, y luego al presentar todo, que no sea lo que el cliente quería o esperaba.

Ilustración 1:

Mapa visual del funcionamiento de SCRUM



Nota: Se muestra cómo funciona SCRUM con todas sus etapas. En azul tenemos ciclo/evento, en naranja los componentes con lo que se trabaja y en verde las personas involucradas. (Vroon, 2015)

KANBAN:

Kanban entra dentro del subconjunto del marco ágil, que he implementado junto a la metodología scrum, para la asignación y gestión de las tareas necesarias para completar un sprint.

Sirve para de una forma muy visual y lógica, puedas organizar y optimizar tu trabajo y tiempo, mediante tablas de **'To do' (Por hacer)**, **In Progress (En proceso)** y **Done (Hecho)**. Destaca sobre todo en equipos de varias personas, aunque esta vez la haya gastado individualmente, ya que cada tarea puede ser asignada a alguien distinto y ver en qué proceso esta.

Ilustración 2:

Ejemplo visual de tablero KANBAN

Nota: Observamos los diferentes elementos que representarían un tablero KANBAN, con diferentes tareas de ejemplo dentro, representadas con colores. (Crema, 2022)

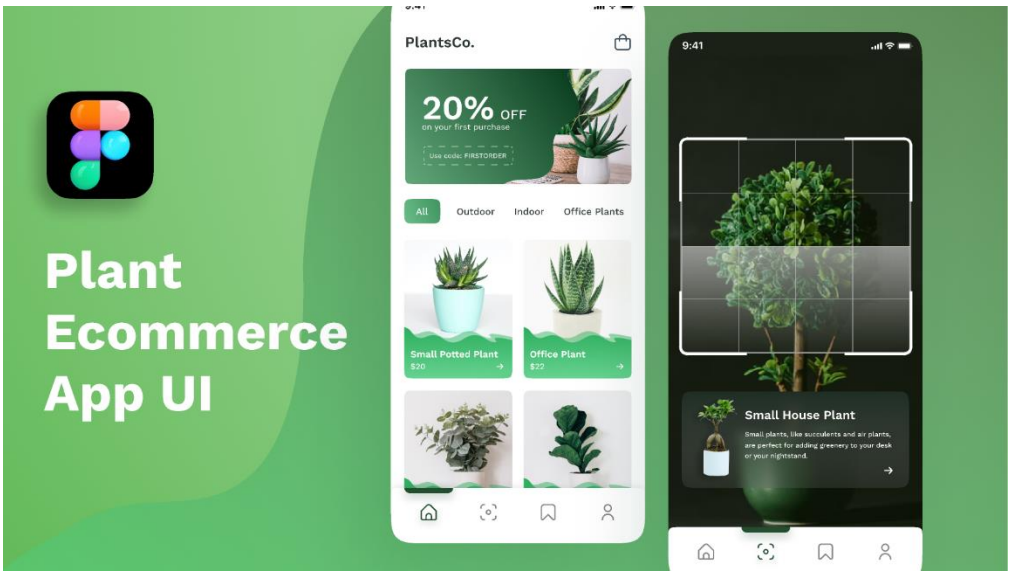
b. Descripción de los componentes de la aplicación: mockups, arquitectura, backend, tecnologías utilizadas, etc.

- Mockups:
  - ***Figma:***

    Con los Mockups he podido plasmar una idea principal del diseño de la aplicación en la realidad. Aunque durante el desarrollo ha sido sujeto a muchos tipos de cambios, siempre se recomienda poder tener un mockup o algún tipo de diseño como referencia, que es de gran ayuda sobre todo en la etapa inicial.

    Figma es una herramienta grafica muy popular pensada para hacer todo tipo de diseños de una forma fácil y accesible. Siendo muy simple a nivel de usabilidad. Dentro del desarrollo de software existen varias herramientas de este tipo, pero hace varios años que Figma suele ser la opción más popular.

Ilustración 3: Ejemplo de la interfaz de una APP diseñada en Figma



Nota: Observamos el diseño de dos pantallas para una aplicación móvil sobre e-commerce de plantas. (Urvashi, 2022)

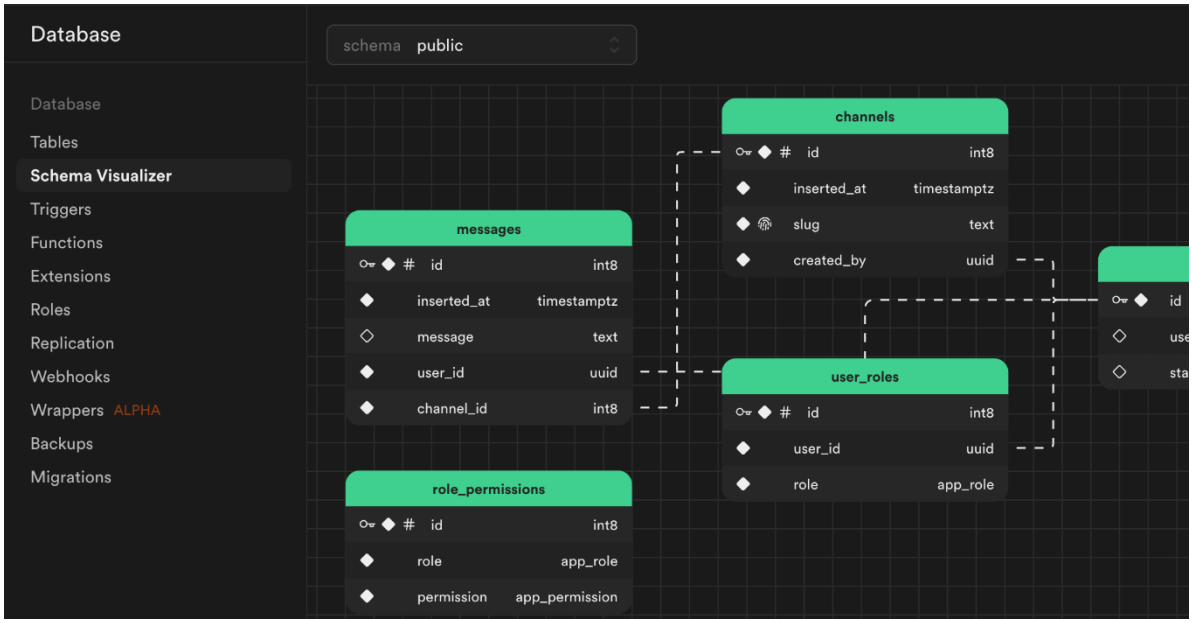
- Arquitectura backend:
    - ***Supabase:***

      El backend aunque no se aprecie a nivel usuario, es una de las partes más importantes dentro de los proyectos de desarrollo, ya que se encarga de toda la gestión de información, que dependiendo del caso puede llegar a ser muy valiosa. Por eso es importante estudiar e investigar bien que nos interesa utilizar cada caso en concreto.

      La arquitectura backend que he elegido me ha servido para manejar toda la información de la base de datos y consultas a esta, de una forma fácil y bien estructurada.

      Mi elección para el Dashboard de músicos ha sido **Supabase**, una alternativa Open Source a Firebase (Google), muy potente y con mucho soporte de la propia empresa y de la comunidad de desarrolladores.

Ilustración 4: Esquema visual de una BD en Supabase



Nota: Podemos apreciar dentro de Supabase un ejemplo de todas las tablas de una BD con las respectivas relaciones entre ellas.

He podido implementar tanto la creación y uso de una BD, con querys de la propia API de supabase para manipular los datos, como la autenticación de los usuarios que al iniciar tu proyecto añadiendo directamente con la opción Supabase, que te viene casi implementado, incluyendo la parte visual.

Supabase ofrece todos los servicios y herramientas de backend necesarias para crear una aplicación escalable y segura: gestión de base de datos, autenticación, almacenamiento de archivos, generación automática de APIs y actualizaciones en tiempo real, entre otros. (APLYCA, 2023)

- Desarrollo frontend:

  - ***Next.js (TypeScript):***

    Para el desarollo frontend podemos llegar a encontrar todo tipo de opciones: Lenguajes, Frameworks, Librerías… y una vez más las más acertada será la que mejor se adapte al desarrollo de la aplicación, y también con la que tú te sientas más cómodo/a y puedas llegar a dominar.

    En el caso de mi aplicación Dashboard, he elegido el framework Next.js dentro del lenguaje TypeScript.

    Next.js no deja de ser React.js (uno de los frameworks más famosos en el desarrollo web y con mucho apoyo y mantenimiento de la comunidad) con algunas funcionalidades extras para mejorar los procesos en ciertos casos, con el renderizado de ciertas partes de la web mediante el servidor y con una navegación entre pantallas ya integrada.

    Por otra parte, TypeScript es la implementación de JavaScript con tipos. JavaScript es uno de los lenguajes más utilizados y la opción más asequible entre las personas que empieza a programar, una de las razones es porque el tipado es dinámico y no lo tienes que definir tú.

En proyectos más grandes esto puede llegar ser poco beneficioso, ya que puede haber errores de ejecución que no controlas por la asignación errónea de tipos y muchas veces puede llegar a ver código poco legible ya que es difícil saber con qué se está trabajando.

Para solucionar este problema se creó TypeScript, que sigue creciendo hoy en día, de hecho, es el lenguaje predeterminado cuando inicias cualquier Proyecto de React.js, a no ser que de forma manual le especificas lo contrario.

- Librerias:

  Las librerías son tanto, elementos visuales, como funcionalidades extra que puedes ir añadiendo a tu aplicación, y no viene de forma predeterminada al instalar tu SDK (Software Development Kit) o entorno de desarrollo. Existen tanto desarrolladas para y por un mismo lenguaje de programación, como de terceros, que son independientes y sirven para muchos lenguajes. Aunque sí que tienes que comprobar que sea compatible con lo que estas gastando.

  En este caso, Next.js/React.js no cuenta con muchos componentes base y por eso siempre se suelen gastar librerías, sobre todo para los componentes visuales que forman parte del diseño de la aplicación.

  Por eso he optado por utilizar las siguientes librerías pensadas para React. *En componentes de diseño:* **NextUI y Shadcn**. Ya que están muy asentadas y ofrecen con una usabilidad clara, diseños y funcionalidades que hacen que tu aplicación pueda incrementar el nivel visualmente.

  En funcionalidades, una de las características destacables es que puedes importar toda la información necesaria mediante un CSV, y por eso use un gestor de datos CSV llamado: **CsvToJson**, para poder transformar toda la información a tipos de datos en mi código.

- Control de versiones:
    - ***Git:***

      Es fundamental que, en cualquier proyecto de desarrollo, este presente un control de versiones sobre tu código. Por eso, aunque mi Dashboard de Músicos era de forma individual he implementado este sistema.

      Hoy en día, Git es, con diferencia, el sistema de control de versiones moderno más utilizado del mundo. Git es un proyecto de código abierto maduro y con un mantenimiento activo que desarrolló originalmente Linus Torvalds, el famoso creador del kernel del sistema operativo Linux, en 2005. (Atlassian, 2022)

      Te ofrece la opción de guardar todos los estados de tu proyecto desde que lo inicias, lo que te da una perspectiva de lo que se ha ido haciendo, y partir de aquí te da muchas opciones para poder gestionar la buena calidad del código. Con distintas herramientas puedes por ejemplo volver atrás en el tiempo, ver quien hizo x cambios y porqué, etc.…

      La mayoría de los proyectos a lo largo de la carrera de un programados son en equipo, si no existiera esta herramienta sería imposible poder organizarse bien, te ofrece ver el trabajo de tus compañeros, y poder compararlos e integrarlos.

- ***GitHub (repo en la nube)***:

  GitHub no deja de ser un servicio en la nube pensado para poder alojar tus versiones en los llamados repositorios y tener guardado su historial mediante el uso de Git. Te da la opción de todo lo que tienes con Git en local, poder tenerlo disponible 'on-line', e ir actualizando los cambios desde ahí. También ofrece una sería de herramienta que lo hacen muy potente.

  Es necesario para un proyecto en equipo, ya que todos trabajáis en base a este repo en la nube, subiendo cambios desde vuestra versión en loca, no podemos acceder a la versión local de los demás en nuestro ordenador.

c. Problemas/dificultades encontradas en el desarrollo del PFC, y que soluciones se han buscado para su solución.

Durante todo el desarrollo del Proyecto Final de Ciclo (PFC) han ido surgiendo diferentes tipos de problemas, los cuales se han tenido que poner foco y buscar la opción más asequible para poder continuar de la mejor forma posible. Han surgido en partes distintas del proyecto que son las siguientes:

- ***Organización:***

  Al empezar a trabajar en el PFC, y tener todas las practicas por delante, te da la sensación de que tienes mucho tiempo, y realmente no tenemos tanto, ya que este proyecto se tiene que desarrollar después de estar parte del día trabajando con tareas de la empresa de las practicas.

  Entonces nada mas empezar no le di mucha importancia a seguir un planing y una organización, y simplemente iba haciendo tareas de vez en cuando, algunas de desarrollo, otras de gestión de BD, etc.

  Pero al poco tiempo me di cuenta de que no estaba optimizando bien mi tiempo ni organizando ni priorizando las cosas, que eso es algo principal en cualquier proyecto, estaba pasándome mucho tiempo en tareas que no eran tan importantes, y adelantándome a funcionalidades que necesitaban tener antes unas bases desarrolladas.

  Por lo cual si quería hacer un buen proyecto necesitaba solucionar este problema, y decidí implementar un tablero Kanban con Trello para organizar las distintas tareas que tenía que hacer y poder centrarme en las de mayor prioridad, haciendo primero todo el backlog (pila de tareas del proyecto).

  A partir de aquí entraba al tablero casi todos los días para organizar las distintas tarjetas (vista de las tareas), ver cuales había acabado, cuales tenía en proceso y en que me quería centrar ese día.

Aun así, en proyectos individuales no es tan importante como en los de equipo, donde esta organización es indispensable para poder avanzar y tener un buen ámbito de trabajo. Hoy en día no se concibe un proyecto en equipo sin estos requisitos mínimos.

- ***Aprendizaje y Desarrollo:***

  El proceso de desarrollo siempre suele ser una etapa divertida, pero donde mas problemas y errores te surgen del propio código que desarrollas, tienes que emplear herramientas y buscar foros para solucionar esto. Hasta aquí todo bien, ya que forma parte del trabajo y sabes que como programador gran parte del tiempo es ese.

  La cosa es que en este proyecto me he abarcado a usar nuevas tecnologías y herramientas con las cuales no tenía mucha experiencia, sí que es verdad que no empiezas de cero, ya que la base la tenía y había trabajado con recursos parecidos, pero siguen siendo cosas nuevas y hay pequeños detalles y funcionalidades que cambian.

  Por lo cual me he pasado mas tiempo de lo normal arreglando errores, y aprendiendo nuevas características y utilidades, lo cual era en parte necesario si quería hacerlo de una forma correcta, pero llegaba un momento que me frustraba ya que quería avanzar rápido y no podía.

  Por eso creo que es bueno mentalizarse de que, al abarcarse en cosas nuevas que en las que no tienes mucha experiencia es normal que tengas que pasar horas aprendiendo y nutriéndote de información, porque si no puedes sentir que no avanzas y que no salen las cosas, y perder la motivación en el proyecto. Lo importante es que todo esto sirve de experiencia para situaciones futuras.

d. Resultados obtenidos

Los resultados finales creo que han sido muy satisfactorios, tanto por mi parte como desarrollado como para el cliente (SJMA) para el que va la aplicación.

Está claro que queda muchas cosas por pulir y para que realmente quede una aplicación profesional a la altura, pero para la disposición de horas y recursos que teníamos, el proyecto y como se ha envuelto todo ha sido muy positivo.

Un diseño simple pero bonito, funcional y profesional. Ha sido implementada con éxito la vista de todos los datos y la opción de crear particiones finalmente funciona correctamente, controlando todas las excepciones que pueden ocurrir con un usuario base.

Ilustración 5: Captura de la pantalla Home de mi aplicación

Nota. Imagen creada por el autor.

4. CONCLUSIONES: Conclusiones obtenidas después de la realización del PFC, tanto a nivel profesional como a nivel personal.

Las conclusiones obtenidas después de la realización del PFC son las siguientes…

A nivel profesional, ha sido una gran experiencia para saber gestionar la creación desde cero un proyecto real y medianamente grande para el nivel actual que tenemos.

Para seguir las distintas metodologías de trabajo y poder sacar el máximo partido a tu tiempo, el aprendizaje de nuevas tecnologías y la resolución de errores e inconvenientes. Cada día ves como dominas más ciertos aspectos y vas dando pasitos hacia delante en este mundo del desarrollo.

A nivel personal, me llevo un sabor agridulce, ya que estoy muy orgulloso del trabajo que he hecho y de todo lo conseguido, pero ha sido difícil gestionarlo.

Después de estar trabajando alrededor de 8 horas, ponerte a programar de nuevo en un proyecto importante tu solo, se hace pesado. Tienes muchas cosas que hacer y que aprender, y darle mucho cariño si realmente quieres que salga algo presentable.

Esto te quita tiempo también para salir a la calle y hacer cosas que te gusten, que quieras que no es un punto muy importante. En el caso que trabajes en remoto, que, en mi caso, la mitad de las horas de la semana las hacia desde casa, surgen días de que te pasas 10-11 horas delante del ordenador y sin que te de la luz de sol. Es un esfuerzo que hay que asumir en un periodo corto de tiempo, pero hay que estar mentalizado que conlleva mucho trabajo, el adaptarte a un entorno real de trabajar en una empresa y además hacer después el PFC.

5. LÍNEAS FUTURAS DE TRABAJO

Las líneas futuras de trabajo sobre este proyecto están claras, ya que como he explicado en el punto [e. Resultados obtenidos] se ha conseguido un buen resultado, pero queda mucho por hacer.

Por ejemplo, por tema de tiempos se ha asignado la funcionalidad de inserción de datos con el CSV para el siguiente sprint. Que es una pena porque tenia casi toda la lógica implementada, pero había que darles prioridad a otras funcionalidades que el usuario puede ver. Por lo cual esta sería uno de los principales focos a futuro.

Además, se puede pulir mucho mas lo que ya hay, ideas como: agregarle funcionalidades a las tablas que simplemente muestran información, hacer una interfaz con los márgenes de todos los elementos bien alineados, etc.…

Sin contar que el código se intenta hacer lo mas limpio posible a la hora de desarrollar, pero seguro que hay un montón de funcionalidades, servicios, lógica y componentes que se pueden refactorizar para hacerlos óptimos y entendibles.

En conclusión, aunque finalice la etapa del PFC, seguiré por mi propia cuenta haciendo todas las mejoras posibles, que me sirva también para seguir formándome en época donde las clases se han terminado.

6. BIBLIOGRAFÍA

APLYCA. (28 de 02 de 2023). *Supabase: una alternativa ágil de código abierto*. Obtenido de https://www.aplyca.com/blog/blog-supabase-una-alternativa-agil-de-codigo-abierto

Atlassian. (2022). *Qué es Git*. Obtenido de https://www.atlassian.com/es/git/tutorials/what-is-git

Crema. (09 de 11 de 2022). *What's the Difference Between Kanban and Scrum?* Obtenido de https://www.crema.us/blog/whats-the-difference-between-kanban-and-scrum

Ortiz, D. (2024). *¿Qué es un dashboard y para qué se usa?* Obtenido de https://www.cyberclick.es/numerical-blog/que-es-un-dashboard

SAP españa. (21 de 03 de 2024). *¿Qué es la transparencia empresarial y por qué es importante?* Obtenido de https://news.sap.com/spain/2024/03/la-transparencia-empresarial/#:~:text=La%20transparencia%20empresarial%20es%20m%C3%A1s%20que%20una%20tendencia%2C,confianza%20y%20colaboraci%C3%B3n%2C%20y%20contribuye%20al%20desarrollo%20sostenible.

Soler, D. (10 de 01 de 2024). *¿Qué es un sprint en Agile?* Obtenido de https://keepcoding.io/blog/que-es-un-sprint-en-agile/

Urvashi, K. (2022). *Plant Ecommerce App UI*. Obtenido de https://www.figma.com/community/file/1103279171641876086

Vroon, M. (2015). *Agile Scrum*. Obtenido de https://marcelvroon.nl/agile-scrum/