

2024

GUITARMATCH





Índice

1. Documento Descripción del Proyecto	4
1.1 Contexto del proyecto.....	4
1.1.1 Ámbito y entorno.....	4
1.1.2 Análisis de la realidad.....	5
1.1.3 Solución y justificación de la solución propuesta	8
1.1.4 Destinatarios.....	9
1.2 Objetivos	9
1.2.1 Versión Español.....	9
1.2.2 English Version.....	10
1.3 Marco Legal	10
2. Documento de Acuerdo del Proyecto	11
2.1 Historias de Usuario	11
2.1.1 Historias de Usuario para requisitos no funcionales:	12
2.2 Tareas.....	13
2.3 Metodología a seguir para la realización del proyecto.....	14
2.4 Planificación temporal de Tareas	15
2.5 Presupuesto	16
2.6 Análisis de Riesgos.....	18
2.7 Contrato.....	18
2.8 Pliego de condiciones	21
3. Documento de análisis y diseño	23
3.1. Análisis y diseño de la arquitectura de la aplicación	23
3.1.1 Tecnologías/Herramientas usadas y descripción de las mismas	23



3.1.2 Arquitectura de componentes de la aplicación.....	24
3.2 Modelado de datos	26
3.3 Análisis y diseño del sistema funcional	30
4.4 Análisis y diseño de la interfaz de usuario	33
4.Implementación	39
Bibliografía.....	42
Anexo 1.....	43
Anexo 2.....	43

Índice de Imágenes

Imagen 1 Guitar Center	6
Imagen 2 Sweetwater buscador	7
Imagen 3 GuitarLab	7
Imagen 4 findMyGuitar	8
Imagen 5 Extraído de GURU99	15
Imagen 6 Planificación de tareas Elaboración propia	16
Imagen 7 Planificación de tareas extendido Elaboración propia	16
Imagen 8 Visual Studio Logo.....	23
Imagen 9 SQL Server Logo	24
Imagen 10 Diagrama MVC Adaptado de “Qué es MVC” (Angel Álvarez, 2020)	25
Imagen 11 MVC Aplicación, Elaboración propia	26
Imagen 12 Modelado de datos	27
Imagen 13 Modelo E/R.....	28
Imagen 14 Casos de uso.....	29
Tabla 1 Historias de Usuario	12



Tabla 2 Historias de Usuario no funcionales.....	13
Tabla 3 Tareas	14
Tabla 4 Análisis de riesgos	18
Tabla 5 Presupuesto	18
Tabla 6 Implementación.....	39

1. Documento Descripción del Proyecto

Este apartado se va a centrar en explicar el porqué de la aplicación, el análisis de aplicaciones similares y los objetivos.

1.1 Contexto del proyecto

El contexto del proyecto va a desarrollar el ámbito y las aplicaciones contemporáneas y similares a la que se está creando.

1.1.1 Ámbito y entorno

El proyecto consiste en el desarrollo de una aplicación de escritorio que permitirá a los usuarios encontrar la guitarra ideal. Para ello, se utilizará una base de datos amplia de guitarras de diferentes marcas y modelos. Esta aplicación se dirigirá a usuarios que estén en busca de guitarras eléctricas y pretendan comparar los atributos de éstas. Además, se generará información con las búsquedas recopiladas para poder venderla a empresas de publicidad o fabricantes.



La aplicación contará con dos partes principales: la sección de búsqueda y la sección de comparaciones. En la sección de búsqueda, los usuarios podrán especificar sus preferencias en cuanto la marca o el modelo de la guitarra que buscan. La aplicación les mostrará una lista de guitarras que cumplan con estos criterios y les permitirá compararlas en detalle.

En la sección de comparaciones, la aplicación realizará una segunda búsqueda para mostrar la información tanto de la segunda como la primera guitarra para que el usuario simplemente compare las características. Además, la aplicación ofrecerá información detallada sobre cada guitarra, como reseñas, especificaciones técnicas y precios. Ambas secciones están en el mismo entorno, si se busca una sola guitarra no habrá comparación.

Además, contará con un programa CRUD de administrador para manejar la base de datos y otro de publicidad para hacer consultas a las búsquedas de las guitarras.

La aplicación real de este proyecto se enfocará en la venta del producto a tiendas físicas de instrumentos (en este caso guitarras) separando los tres programas de modo que el buscador de guitarras sea alojado en una tablet o dispositivo similar encontrado en la zona accesible para los posibles clientes. Asimismo, el administrador tendrá acceso a su programa en la oficina y el equipo de marketing tendrá acceso en su zona también.

1.1.2 Análisis de la realidad

La aplicación de búsqueda de guitarras partirá desde cero, pero tomará como referencia algunos otros buscadores que más adelante nombraré. Aunque existen en el mercado otras aplicaciones similares, nuestra empresa se esforzará por crear un producto novedoso que destaque tanto por sus características como por su diseño y contenido.

A diferencia de otras aplicaciones de búsqueda de guitarras existentes en el mercado, nuestra aplicación se enfocará en ofrecer una experiencia de usuario más amigable y sencilla. Además, la aplicación incluirá una amplia variedad de guitarras de diferentes marcas y modelos para que los usuarios puedan encontrar la guitarra ideal para ellos.

En la actualidad, hay otras aplicaciones de búsqueda de guitarras en el mercado, como:



- “Sweetwater”
- “findmyguitar”(s.f.)
- “Guitar Center”
- “Guitarlab”

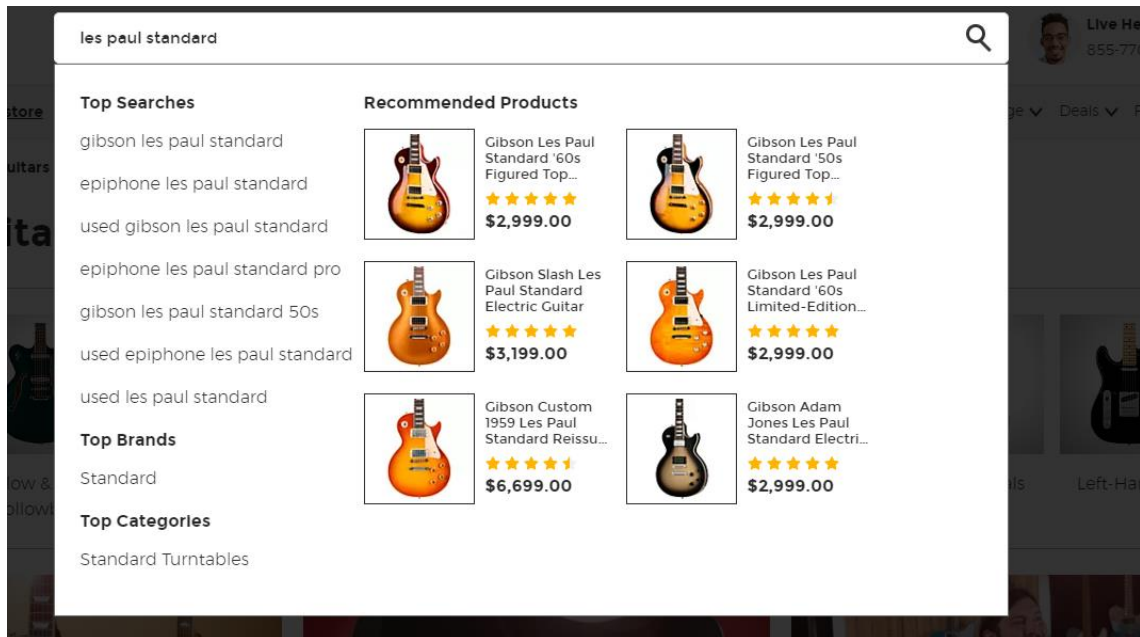


Imagen 1 Guitar Center

Pero éstas, a menudo carecen de una experiencia de usuario intuitiva y son parciales en cuanto a los instrumentos que se recomiendan.

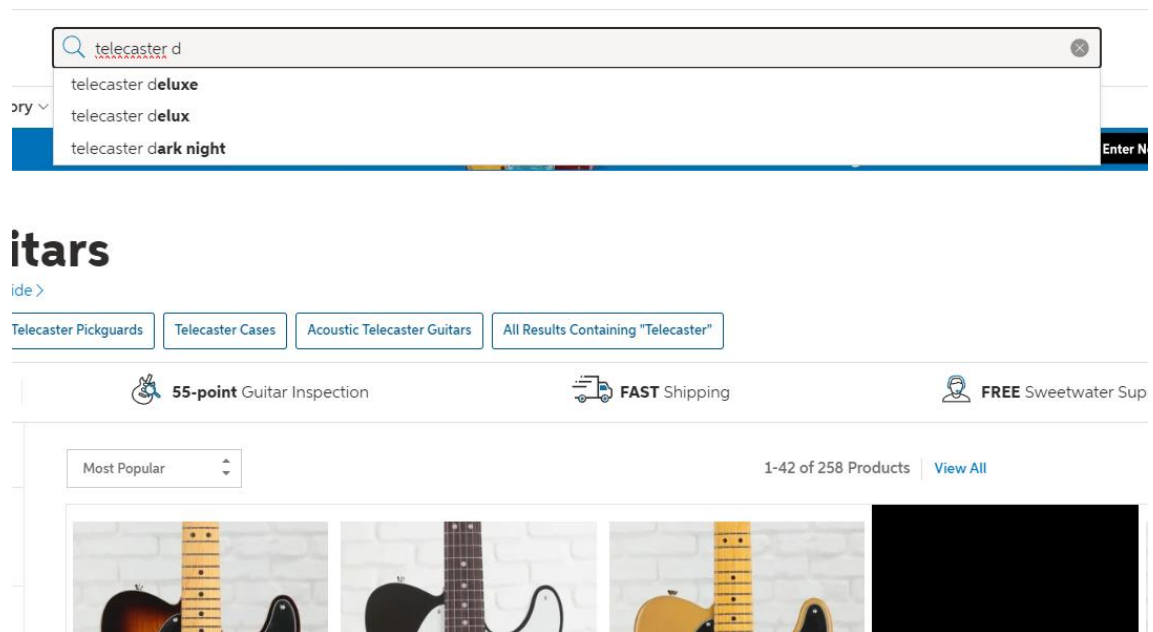


Imagen 2 Sweetwater buscador

SweetWater, GuitarLab (Thomann)y Guitar Center tienen una buena herramienta de búsqueda las tres, pero pecan de ser a la vez propietarias de una empresa de venta de guitarras por lo que es posible que lo que intenten vender no sea lo más adecuado para el usuario sino lo que más les conviene. Nuestra aplicación se esforzará por brindar una experiencia de usuario más limpia y libre de distracciones publicitarias innecesarias para que los usuarios puedan concentrarse en encontrar la guitarra ideal.

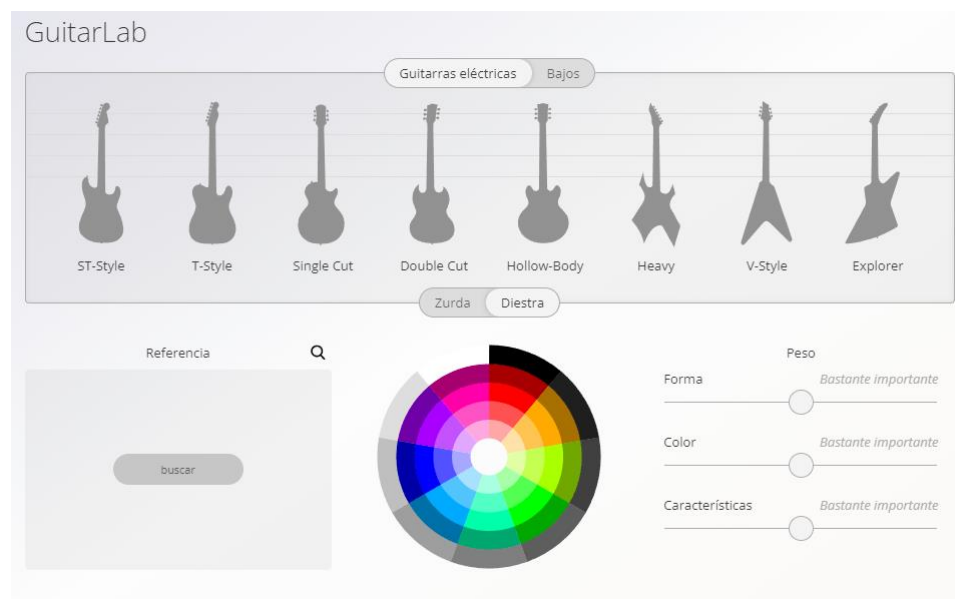


Imagen 3 GuitarLab



En el caso de findmyguitar, es sinceramente una aplicación web que me ha sorprendido enormemente. La idea del buscador de guitarras rondaba mi mente desde el año pasado para producirla con diferentes funcionalidades que al final se han ido desechando. Al buscar información para poder documentar el proyecto se encontró esta web que va a servir de punto de guía para realizar parte del proyecto, ya que es un buscador superior e integra bastantes de las propiedades que se buscan implementar.

Search Reviews

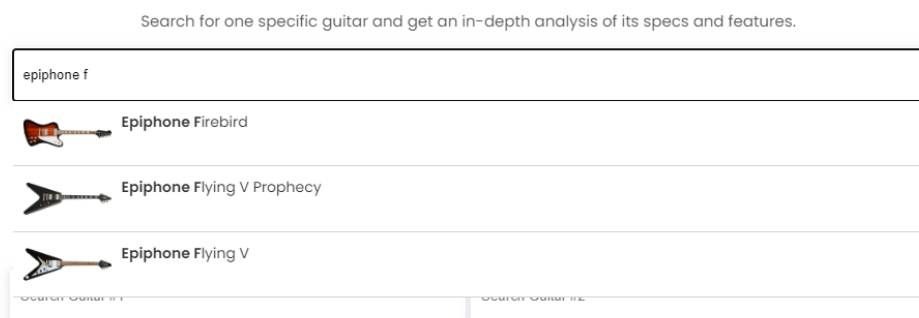


Imagen 4 findMyGuitar

Esta aplicación innovadora y atractiva será de gran interés para las empresas de publicidad que busquen asociarse con una marca que ofrece una experiencia de usuario superior y que se destaca de sus competidores. Además, nuestra aplicación ofrecerá un gran potencial para la recopilación de datos de los usuarios que se podrán utilizar para personalizar las campañas publicitarias y llegar a un público más específico y comprometido con la música.

1.1.3 Solución y justificación de la solución propuesta

La solución propuesta consiste en desarrollar una aplicación de escritorio que sirva como un buscador de guitarras altamente funcional. Esta aplicación permitirá a los usuarios realizar búsquedas personalizadas de guitarras según sus preferencias, tales como marca o modelo.

Una vez que los usuarios realicen una búsqueda, la aplicación les proporcionará una lista de opciones que cumplan con sus criterios. Además, se implementará una funcionalidad



de comparación que permitirá a los usuarios seleccionar múltiples guitarras y compararlas en detalle, facilitando así la toma de decisiones informadas.

La justificación de esta solución radica en la creciente demanda de herramientas especializadas para los aficionados a la música y la guitarra. Con esta aplicación, los usuarios podrán acceder fácilmente a una amplia variedad de guitarras y encontrar la opción que mejor se adapte a sus necesidades y preferencias. Además, la funcionalidad de comparación les permitirá evaluar las opciones de manera más eficiente, ahorrando tiempo y esfuerzo.

Esta solución ofrece una oportunidad para colaboraciones con empresas de publicidad interesadas en llegar a un público comprometido con la música y la guitarra.

1.1.4 Destinatarios

El destinatario final es el usuario que utilizará la herramienta para buscar la guitarra que más se adecue a sus gustos o necesidades, aunque la plataforma ha de buscar la manera de ser rentable mediante la venta de información. De esta manera se ofrecerá a marcas y tiendas la posibilidad de comprar esta información

1.2 Objetivos

En esta sección desarrollaré los objetivos del proyecto.

1.2.1 Versión Español

El objetivo del proyecto es desarrollar una aplicación de escritorio para ayudar a los usuarios a encontrar la guitarra ideal para ellos y para permitir a las empresas de publicidad promocionar sus productos. La aplicación permitirá a los usuarios buscar y comparar diferentes modelos de guitarras para que el usuario se pueda decidir por una guitarra, haciendo uso de videos, reseñas, la mencionada comparación y la lista de tiendas con sus precios.

El usuario creará una cuenta que almacenaremos en nuestra base de datos local y la misma base de datos, para así luego ofrecer como, segundo objetivo, a las empresas una información que podrán utilizar para generar publicidad.



1.2.2 English Version

The objective of the project is to develop a desktop application to assist users in finding their ideal guitar and to enable advertising companies to promote their products. The application will allow users to search for and compare different guitar models, aiding them in making an informed decision. Users can utilize videos, reviews, the aforementioned comparison, and a list of stores with their prices.

Additionally, users will create an account, which we will store in our local database. We will also retain records of their active searches in the same database. This information will be used to provide valuable insights to companies for targeted advertising purposes.

1.3 Marco Legal

Se informa que nuestra plataforma se rige por la Licencia Pública General de GNU (GPL), un marco legal que promueve la libertad de uso, estudio, modificación y distribución del software. Esta elección refleja nuestro compromiso con los principios del movimiento del software libre, facilitando un entorno de colaboración y acceso abierto. La adopción de la GPL permite a los usuarios beneficiarse de la transparencia y flexibilidad inherentes a esta licencia, promoviendo así la innovación y el intercambio de conocimientos en la comunidad. Localización de los términos en el Anexo 1.

En cumplimiento con la legislación española sobre protección de datos, se informa a los usuarios que todos los datos personales proporcionados serán tratados conforme a lo establecido en la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales, así como en el Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, conocido como GDPR. Se garantiza la confidencialidad y seguridad de los datos personales en todo momento, respetando su privacidad y derechos.



2. Documento de Acuerdo del Proyecto

En esta parte del documento voy a definir las historias de usuario y tareas sobre las que voy a trabajar, gracias a ellas definiré mejor el proyecto y los requisitos necesarios para llevarlo a cabo.

2.1 Historias de Usuario

Las historias de usuario van a comprender los requisitos básicos para la funcionalidad de la aplicación.

ID	Criterios de aceptación	Prioridad
HU-1	<p>La búsqueda debe devolver una lista de guitarras que cumplan con los criterios de búsqueda especificados.</p> <p>La lista de guitarras debe ser clara y fácil de entender para el usuario.</p> <p>La aplicación debe permitir al usuario comparar las guitarras de la lista para poder tomar una decisión informada.</p>	Alta
HU-2	<p>La aplicación debe permitir al usuario seleccionar dos guitarras para compararlas.</p> <p>La aplicación debe mostrar una tabla comparativa que incluya información relevante de las guitarras, como marca, modelo y especificaciones técnicas.</p> <p>La tabla comparativa debe ser fácil de leer y de entender para el usuario.</p> <p>La aplicación debe permitir al usuario agregar o quitar guitarras de la tabla comparativa según sus necesidades.</p>	Alta
HU-3	<p>La aplicación debe ser capaz de recopilar y almacenar la información de las búsquedas anteriores de cada usuario.</p> <p>La aplicación debe ser capaz de analizar la información recopilada y utilizarla para generar recomendaciones personalizadas.</p> <p>La aplicación debe ser capaz de mostrar información detallada sobre cada guitarra recomendada como especificaciones técnicas y precios.</p>	Baja
HU-4	<p>La base de datos debe permitir la actualización de la información de las guitarras</p>	Alta



	<p>de forma rápida y eficiente.</p> <p>Los usuarios deben tener acceso a la información actualizada de las guitarras tan pronto como sea posible.</p> <p>El proceso de actualización de la base de datos no debe afectar la funcionalidad de la aplicación, y los usuarios no deben experimentar interrupciones en el servicio mientras se actualiza la información.</p> <p>La actualización de la base de datos debe ser segura y no comprometer la seguridad de los datos de la aplicación o de los usuarios.</p>	
HU-5	<p>La aplicación debe permitir al usuario guardar sus búsquedas anteriores.</p> <p>Las búsquedas guardadas deben ser fácilmente accesibles desde la aplicación.</p> <p>Las búsquedas guardadas deben permanecer disponibles incluso después de cerrar y volver a abrir la aplicación.</p> <p>Las búsquedas guardadas deben actualizarse si el usuario realiza cambios en sus preferencias.</p>	Media
HU-6	<p>La aplicación generará una base de datos con las búsquedas para poder vender a las empresas de publicidad o marcas información relativa a estas búsquedas.</p>	Alta
HU-7	<p>La aplicación debe ser fácil de navegar y buscar para encontrar la información necesaria sobre las guitarras y su disponibilidad en las tiendas online.</p> <p>La información sobre la disponibilidad y los precios debe ser precisa y actualizada con frecuencia.</p> <p>Los usuarios deben poder comparar precios y disponibilidad entre diferentes tiendas para encontrar la mejor oferta posible.</p>	Baja

Tabla 1 Historias de Usuario Elaboración propia

2.1.1 Historias de Usuario para requisitos no funcionales:

Las historias de usuario no funcionales son una serie de puntos de mejora para que la aplicación se desarrolle de una manera más eficiente.



HUNF-1	Como usuario de la aplicación, quiero que la interfaz sea intuitiva y fácil de usar para que pueda navegar por la aplicación sin problemas.	<p>La navegación por la aplicación debe ser fluida y sin errores.</p> <p>Los botones y elementos de la interfaz deben ser fáciles de identificar y entender para el usuario.</p> <p>Los usuarios deben poder encontrar la información que buscan sin problemas, gracias a la organización y diseño de la aplicación.</p>	Alta
HUNF-2	Como usuario de la aplicación, quiero que la aplicación tenga un tiempo de respuesta rápido para que pueda realizar búsquedas y comparaciones de manera eficiente.	<p>La aplicación debe responder a todas las interacciones del usuario en un tiempo máximo de 2 segundos.</p> <p>El tiempo de carga de la aplicación no debe superar los 3 segundos en ningún momento.</p>	Media
HUNF-3	Como usuario de la aplicación, quiero que mis datos personales sean seguros y estén protegidos por medidas de seguridad adecuadas.	<p>La aplicación debe contar con una política clara de privacidad que detalle cómo se manejan los datos personales del usuario.</p> <p>La aplicación debe cumplir con las leyes y regulaciones vigentes en cuanto a la protección de datos personales.</p>	Alta

Tabla 2 Historias de Usuario no funcionales Elaboración propia



2.2 Tareas

Tareas	Fecha Inicio	Fecha Fin
Especificar Historias Usuario	11/03	27/03
Crear Historias de Usuario	11/03	27/03
Exposición de Tareas	11/03	27/03
Documentar análisis de riesgos	11/03	27/03
Descripción Proyecto	11/03	27/03
Documento de acuerdo Proyecto	11/03	27/03
Diseño Diagramas E/R	27/03	24/04
Diseño Diagramas casos de uso	27/03	24/04
Diseño y creación de la Base de datos	27/03	24/04
Desarrollo Final	24/04	15/05
Entrega		

Tabla 3 Tareas Elaboración propia

2.3 Metodología a seguir para la realización del proyecto

La metodología seleccionada para este proyecto es el Modelo Incremental, que se basa en una estrategia de desarrollo por fases en la que se van añadiendo funcionalidades de manera progresiva. Este modelo permite al cliente, si lo hubiera (en este caso no), participar en todo momento en el proceso de desarrollo y validar cada una de las fases antes de continuar con la siguiente.

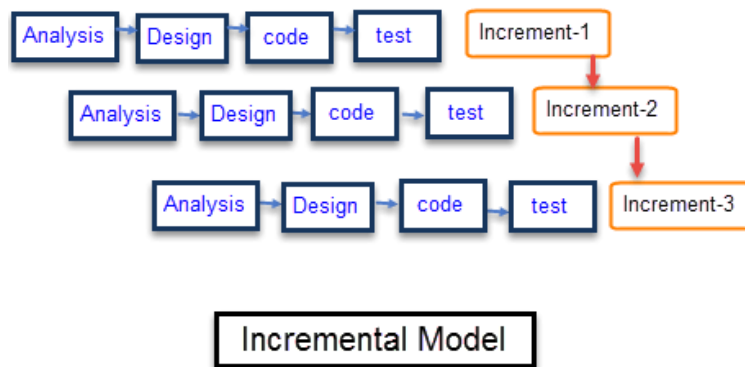


Imagen 5 Extraído de GURU99

El proyecto se divide en incrementos, cada uno con un conjunto de funcionalidades específicas que se deben desarrollar y validar antes de pasar al siguiente incremento. Esto permite que el cliente tenga un mayor control sobre el proyecto y se puedan realizar ajustes o cambios necesarios de manera temprana en el proceso.

Según Jeff Patton (2006) en su obra Incremental releases, el software ha de ser desarrollado como si de “una obra artística estuviéramos hablando” (p.22). De modo que a partir de un boceto se fueran aplicando mejoras y propiedades hasta el producto final, en este caso él lo cataloga como desarrollo iterativo dado que sobre el mismo producto se van aplicando actualizaciones.

Sin embargo, autores como Jeff Sutherland(2010) llaman a este proceso modelo incremental dado que lo visionan como “el crecimiento de una planta en vez del desarrollo de un producto completo a la vez”. Y en el caso de este proyecto la idea se acerca más a esto.

2.4 Planificación temporal de Tareas

La planificación de las tareas a completar se ha realizado teniendo en cuenta las fechas de las entregas necesarias para seguir la evaluación continua del módulo del proyecto. De esta manera se ha confeccionado la siguiente gráfica en la que entrelazan varias tareas a la vez trabajándose estas conjuntamente.



Imagen 6 Planificación de tareas Elaboración propia



Imagen 7 Planificación de tareas extendido Elaboración propia

2.5 Presupuesto

Para realizar un presupuesto del coste económico del desarrollo de esta aplicación se han tomado en cuenta el sueldo actual del alumno en la empresa en la que se están realizando las prácticas. Teniendo esto en mente y que estas corporaciones suelen iniciar la contratación intentando pagar el salario mínimo interprofesional o SMI, se ha aplicado una sencilla ecuación que ira evolucionando a lo largo del desarrollo del proyecto. El término coste de oportunidad se refiere al dinero que se podría estar ganando en vez de invertir las horas del individuo en proyectos propios como ser autónomo o éste en particular.

Así pues, dado que el proyecto inicial se presupone de 40 horas en total podemos multiplicar nueve euros la hora (un poco más del SMI) pero realmente serán muchas más por lo que multiplicaré por 80 para averiguar el coste de oportunidad teórico que serían 720 €. Aunque esto solo supone la inversión realizada por parte del alumno. Si se quisiera obtener un



beneficio económico de esta actividad habría como mínimo conseguir esta cifra para empezar a generar beneficios reales. Además, he añadido un par de apartados extras para tener en cuenta además de las horas del trabajo los dos requisitos previos a la realización de éste, siendo cobrados se lleve a cabo o no el proyecto.

Presupuesto 1	
Sergio Ramos C/ Paseo Grande, 1 NIF: A12345678 Teléfono: 612 456 789 Mail: sergio1993ramos@gmail.com	Guitarreos.com C/ Paseo mediano 1 NIF: D123456744 Teléfono: 612 409 789 Mail: guitarreos@guitarreos.com

Fecha del presupuesto	11/04/2024	Validez	30 días
------------------------------	------------	----------------	---------

DESCRIPCIÓN	UNIDADES	PRECIO	TOTAL
<i>Estudio del mercado</i>	1	50,00 €	50,00 €
<i>Diseño de la aplicación</i>	1	100,00 €	100,00 €
<i>Horas</i>	80	9,00 €	720,00 €

SUB-TOTAL	870,00 €
DESCUENTO	35,00 €
IVA %	21,00%
TOTAL PRESUPUESTADO	1.010,35 €



Firma

Firma del cliente

Tabla 4 Presupuesto Elaboración propia

2.6 Análisis de Riesgos

El análisis de riesgos va a repasar los problemas que pueden aparecer durante el desarrollo de la aplicación.

Riesgo	Gravedad	Probabilidad	Prevención
Desconocimiento de tecnología	Medio	Medio	Estudio e investigación
Errores en el código	Bajo	Alta	Usar herramientas que repasen el código. Escribir el código cuidadosamente.
Problema de Rendimiento	Bajo	Medio	La herramienta ha de estar optimizada para que el usuario no espere al realizar la consulta.
Fallos en conexiones entre bases de datos y aplicación	Alta	Medio	La conexión debe ser fuerte y revisada dado que es lo que proporciona la fuente de información.

Tabla 5 Análisis de riesgos Elaboración propia

2.7 Contrato



Contrato de Desarrollo de Aplicación

Entre:

Desarrollador: Sergio Ramos

Cliente: Guitarreos S.L.

Objeto del Contrato:

El presente contrato tiene como objetivo establecer los términos y condiciones para el desarrollo y entrega de una aplicación de búsqueda de guitarras por parte del Desarrollador al Cliente.

Alcance del Proyecto:

El proyecto consiste en desarrollar una aplicación de búsqueda de guitarras que permita a los usuarios buscar y visualizar información detallada sobre una amplia variedad de guitarras disponibles en el mercado.

Responsabilidades de cada parte:

Desarrollador:

Desarrollar y entregar la aplicación de búsqueda de guitarras según los requisitos acordados.

Proporcionar soporte técnico y mantenimiento durante un período acordado después de la entrega de la aplicación.

Garantizar que la aplicación cumpla con los estándares de calidad y seguridad establecidos.

Cliente:

Proporcionar los recursos necesarios, incluidos los datos y la retroalimentación necesarios, para el desarrollo oportuno y efectivo de la aplicación.



Realizar los pagos acordados según lo estipulado en este contrato.

Tiempo de Entrega y Condiciones:

La aplicación de búsqueda de guitarras se entregará en un plazo de 3 meses a partir de la fecha de firma del contrato, sujeto a posibles ajustes según lo acordado por ambas partes.

Pago:

El Cliente pagará al Desarrollador la suma total de [1010,35€] por los servicios de desarrollo de la aplicación. El pago se realizará de la siguiente manera:

Pago inicial: Estudio de mercado y diseño al inicio del proyecto.

Pago final: El saldo restante se pagará tras la entrega y aceptación satisfactoria de la aplicación por parte del Cliente.

Propiedad Intelectual:

El Cliente tendrá plena propiedad de la aplicación desarrollada, incluidos todos los derechos de propiedad intelectual asociados.

Confidencialidad:

Ambas partes acuerdan mantener la confidencialidad de toda la información confidencial compartida durante la ejecución de este contrato.

Terminación del Contrato:

Este contrato puede ser terminado por cualquiera de las partes con un aviso previo de [15] días por escrito. En caso de terminación, el Cliente compensará al Desarrollador por el trabajo completado hasta la fecha de terminación.

Ley Aplicable y Jurisdicción:



Este contrato se registrá e interpretará de acuerdo con las leyes de España. Cualquier disputa relacionada con este contrato será sometida a la jurisdicción exclusiva de los tribunales de Zaragoza.

Firma:

Firmado y aceptado por:

Desarrollador:

[Firma] Sergio Ramos [Fecha]

Cliente:

[Firma] Representante legal [Fecha]

2.8 Pliego de condiciones

Pliego de Condiciones

1. Descripción del Proyecto:

El proyecto consiste en el desarrollo de una aplicación de búsqueda de guitarras que permitirá a los usuarios buscar y visualizar información detallada sobre una amplia variedad de guitarras disponibles en el mercado.

2. Funcionalidades y Características:

La aplicación incluirá las siguientes funcionalidades principales:

Visualización detallada de información sobre cada guitarra, incluyendo imágenes, descripción, especificaciones técnicas, precios, etc.

Interfaz de usuario intuitiva y atractiva para una experiencia de usuario óptima.

3. Requisitos Técnicos:



La aplicación será desarrollada utilizando Visual Studio Community como entorno de desarrollo integrado (IDE).

Se utilizará un servidor SQL Server para almacenar y proporcionar los datos de las guitarras.

La aplicación será compatible con los sistemas operativos Windows.

4. Diseño y Experiencia de Usuario:

El diseño de la interfaz de usuario será moderno, atractivo y fácil de usar.

Se seguirán las mejores prácticas de diseño de experiencia de usuario (UX) para garantizar una navegación fluida y una experiencia de usuario intuitiva.

5. Entrega y Aceptación:

La aplicación será entregada dentro de un plazo de 3 meses a partir de la fecha de inicio del proyecto.

El Cliente realizará pruebas exhaustivas de la aplicación y proporcionará retroalimentación para cualquier ajuste necesario.

La aceptación final de la aplicación se realizará después de que el Cliente esté satisfecho con la funcionalidad y el diseño.

6. Soporte y Mantenimiento:

El Desarrollador proporcionará soporte técnico y mantenimiento durante un período de 6 meses después de la entrega de la aplicación.

Cualquier problema técnico o actualización necesaria será abordado de manera oportuna por el Desarrollador.

7. Costos y Pagos:

El costo total del proyecto es de [720€].

El pago se realizará en dos partes: un pago inicial del [50]% al inicio del proyecto y el saldo restante tras la entrega y aceptación satisfactoria de la aplicación.

8. Cronograma:

El cronograma detallado del proyecto se proporcionará al inicio del proyecto, incluyendo fechas de inicio y finalización para cada fase importante.



9. Responsabilidades del Cliente:

El Cliente proporcionará acceso a los recursos necesarios, incluidos datos y retroalimentación, para el desarrollo oportuno de la aplicación.

El Cliente realizará pagos según lo estipulado en el contrato.

10. Requerimientos Legales y de Seguridad:

La aplicación cumplirá con todos los requisitos legales y de seguridad aplicables, incluyendo la protección de datos y la privacidad del usuario.

3. Documento de análisis y diseño

En este apartado se incluyen los diagramas y el diseño de la base de datos usada para el desarrollo de la aplicación.

3.1. Análisis y diseño de la arquitectura de la aplicación

A continuación, se va a explicar cómo se ha diseñado la arquitectura de este proyecto, haciendo uso de diferentes diagramas para visualizar mejor los conceptos.

3.1.1 Tecnologías/Herramientas usadas y descripción de las mismas

El entorno de desarrollo principal que se va a utilizar para la realización de este proyecto es Microsoft Visual Studio Community, la versión gratuita del software de Microsoft.



Imagen 8 Visual Studio Logo

Es un IDE que pertenece a la misma familia que Visual studio code, la diferencia es que Visual studio community está más enfocado hacia .NET y C# para Windows mientras que code



es multiplataforma y más personalizable. La elección de este entorno se ha debido a que el desarrollador (en este caso yo) está más familiarizado con esta tecnología para poder crear aplicaciones de escritorio utilizando Windows forms. Además se integra bastante bien con SQL y cuenta con extensiones propias que facilitan el uso de la base de datos.

Principalmente se va a utilizar esta herramienta para crear Windows forms con la intención de diseñar una interfaz intuitiva e interesante con la que el usuario interactuará y realizará las búsquedas sobre la base de datos proporcionada. Se creará si el tiempo lo permite además un programa aparte para gestionar la información de la base de datos de búsquedas proporcionada para las empresas de publicidad.



Imagen 9 SQL Server Logo

Además de Visual Studio Community se hará uso de la previamente mencionada base de datos SQL Server Management Studio. Se ha seleccionado esta herramienta por la facilidad a la hora de integrarse con Visual Studio community, además de la posibilidad de poder personalizar las tablas con diferentes tipos de datos.

3.1.2 Arquitectura de componentes de la aplicación

Para la realización de este proyecto se ha decidido implementar el uso el modelo vista-controlador también llamado (MVC). Es una metodología utilizada en el desarrollo de software que busca separar claramente los distintos componentes de una aplicación. "Su fundamento es la separación del código en tres capas diferentes" (Angel Álvarez,2020) estas capas son el



Modelo, la Vista y el Controlador. La separación de ellos permite una mejor organización del código y una mayor facilidad en el mantenimiento y la evolución del proyecto. De esta manera, el usuario interactuará con la interfaz del programa que mediante código interno ejercerá de controlador para acceder a los modelos y poder mostrar e interactuar con la información solicitada, ya sea de las guitarras como de los usuarios o búsquedas.

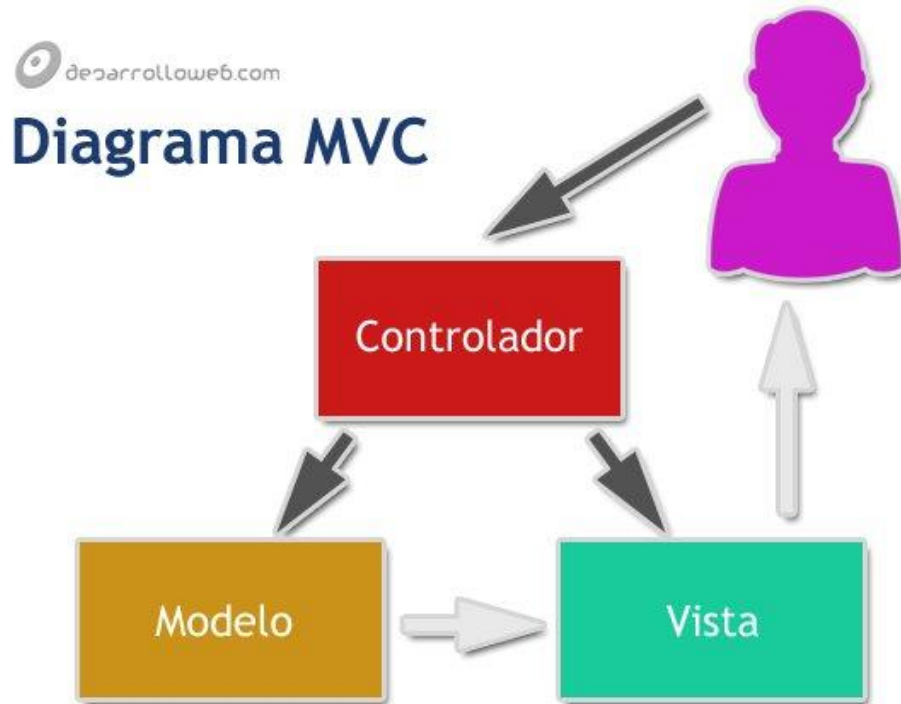


Imagen 10 Diagrama MVC Adaptado de “Qué es MVC” (Ángel Álvarez, 2020)

El principal objetivo del patrón MVC es promover la reutilización de código, la separación de responsabilidades y la facilidad de mantenimiento. De esta manera, la capa de modelo se centrará en almacenar la información de las guitarras y la de controlador se encargará de acceder a ésta para que luego la vista lo muestre o interactúe con el usuario.

Con este pequeño esquema se ejemplifica la implantación del MVC en el desarrollo del programa:

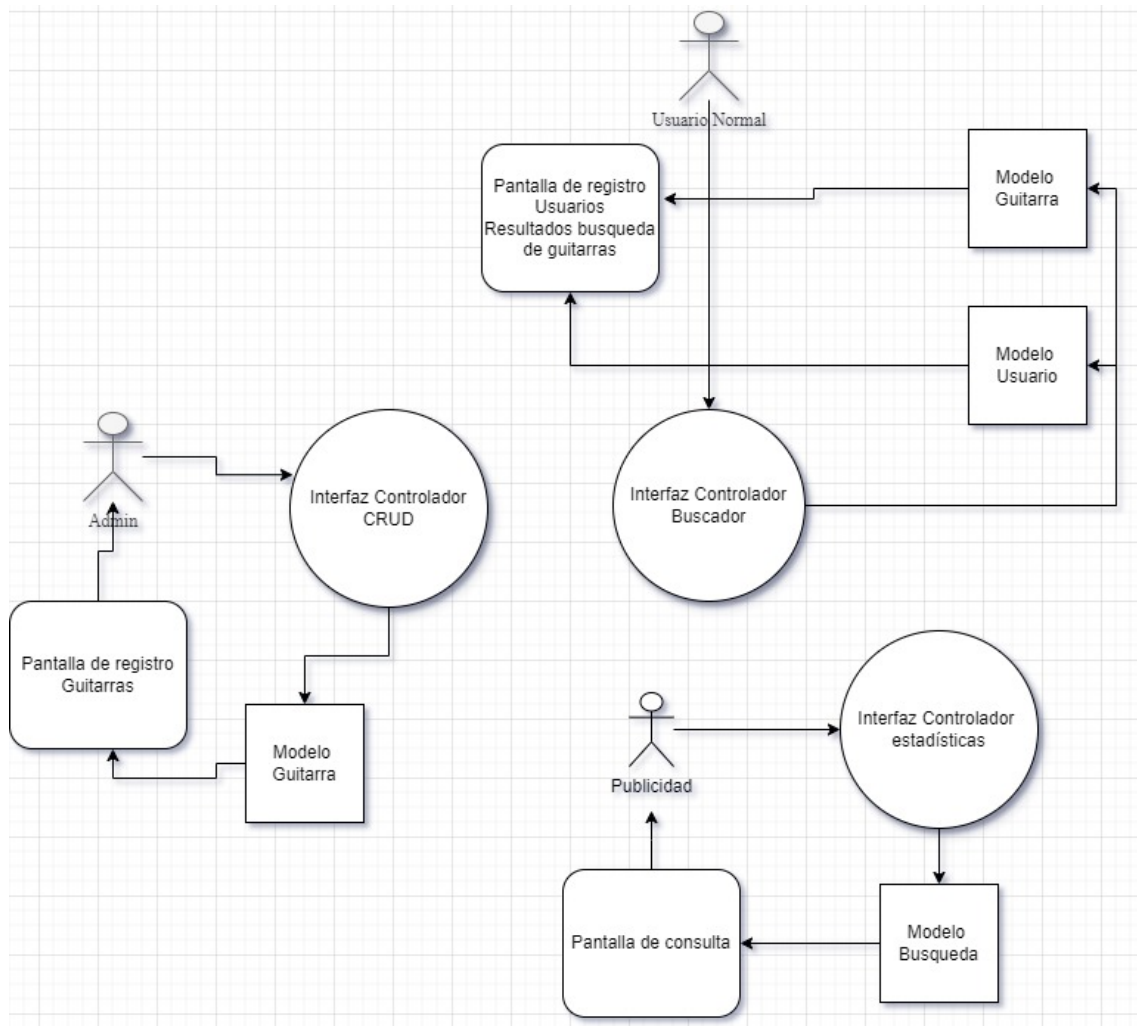


Imagen 11 MVC Aplicación, Elaboración propia

3.2 Modelado de datos

En este apartado voy a hablar sobre la creación de la base de datos y voy a hacer uso de varios esquemas para explicar cómo se relacionan las tablas, para que sirven y como los voy a usar. Para empezar, tengo que describir los datos/información que se van a manejar con esta aplicación a lo largo de su uso. Al iniciar la aplicación se le presenta al usuario la landing page en la cual ya se empiezan a manejar datos. Aquí trataremos con la información del login del usuario, si tiene una cuenta, este podrá ingresar sus datos y acceder a la aplicación mientras que si no tiene una cuenta puede clicar en registro con lo que generará un usuario nuevo. La



generación de un usuario implica que se van a insertar datos en la base de datos y se va a generar un nuevo directorio con el nombre del usuario en la carpeta local.

Dependiendo el programa que se utilice se procederá a tratar la información de manera diferente. Si el programa usado es la búsqueda, el usuario, verá información de guitarras y podrá guardar una lista de favoritos en su directorio, así como ver su historial de búsquedas. Si se inicia el administrador, tendrá acceso completo a la tabla de guitarras pudiendo hacer CRUD completo de ellas y delete de la de usuarios en caso de necesitarlo. Y por último la publicidad tiene acceso de lectura a la tabla de búsquedas.

En el anexo 2 de este documento se encuentra el script utilizado para generar esta base de datos. A continuación, se van a mostrar los modelos utilizados en la base de datos y los diagramas de E/R.

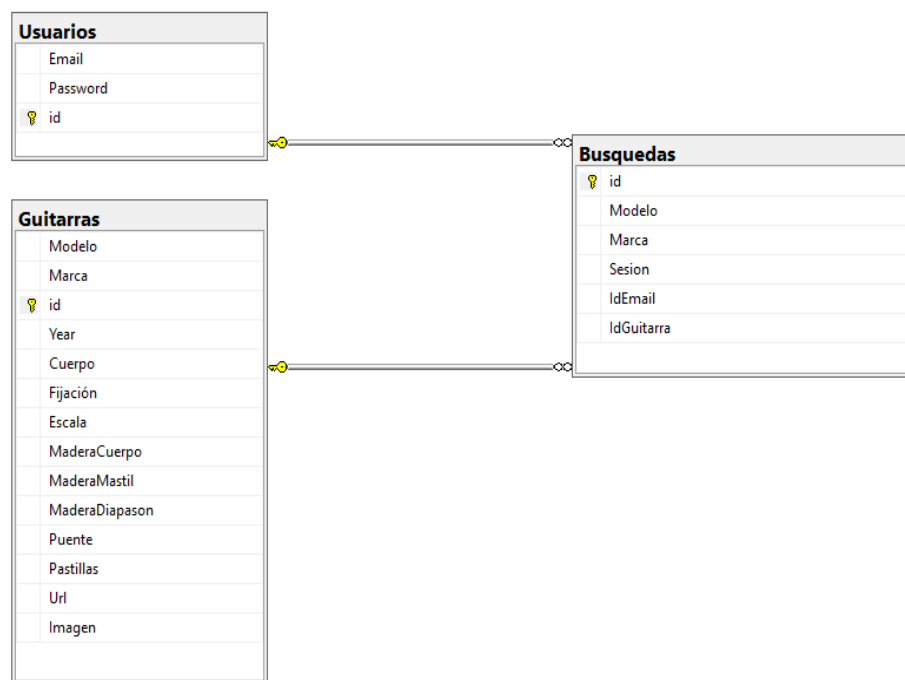


Imagen 12 Modelado de datos

Como se puede comprobar en la imagen, la aplicación (o en su defecto el conjunto de las tres aplicaciones) funciona con tres tablas en las que se van a almacenar diferentes datos. La tabla de usuarios maneja la información del Email (que servirá de nombre de usuario), y la contraseña cifrada, así como el id que será la primary key autogenerada e incremental (el resto de tablas tendrá una id creada de la misma manera).



La tabla de guitarras cuenta con el modelo, que será el atributo usado para buscar guitarras, la marca y una serie de características que serán presentadas al usuario por pantalla cuando realice la búsqueda. Además, se almacenará la imagen asociada con la guitarra con formato string y cada vez que se inserte o haga un select de la guitarra esta información será transportada e interpretada por el programa para poder mostrar dicha imagen. Por último, contará con una url que llevará al usuario a la tienda directamente cuando se clique el botón.

La tabla de búsquedas almacenará tanto el modelo como la marca de la guitarra buscada por el usuario, además manejaremos la sesión para saber cuántas veces se ha utilizado nuestra aplicación con éxito (con búsquedas realizadas y exitosas). En pos de guardar más información se almacenará también el id del usuario que ha buscado en dicha sesión y el id de la guitarra exacta por si se necesita una búsqueda más exhaustiva a posteriori.

El modelo E/R usando las mismas tablas explicadas quedaría pues de la siguiente manera:

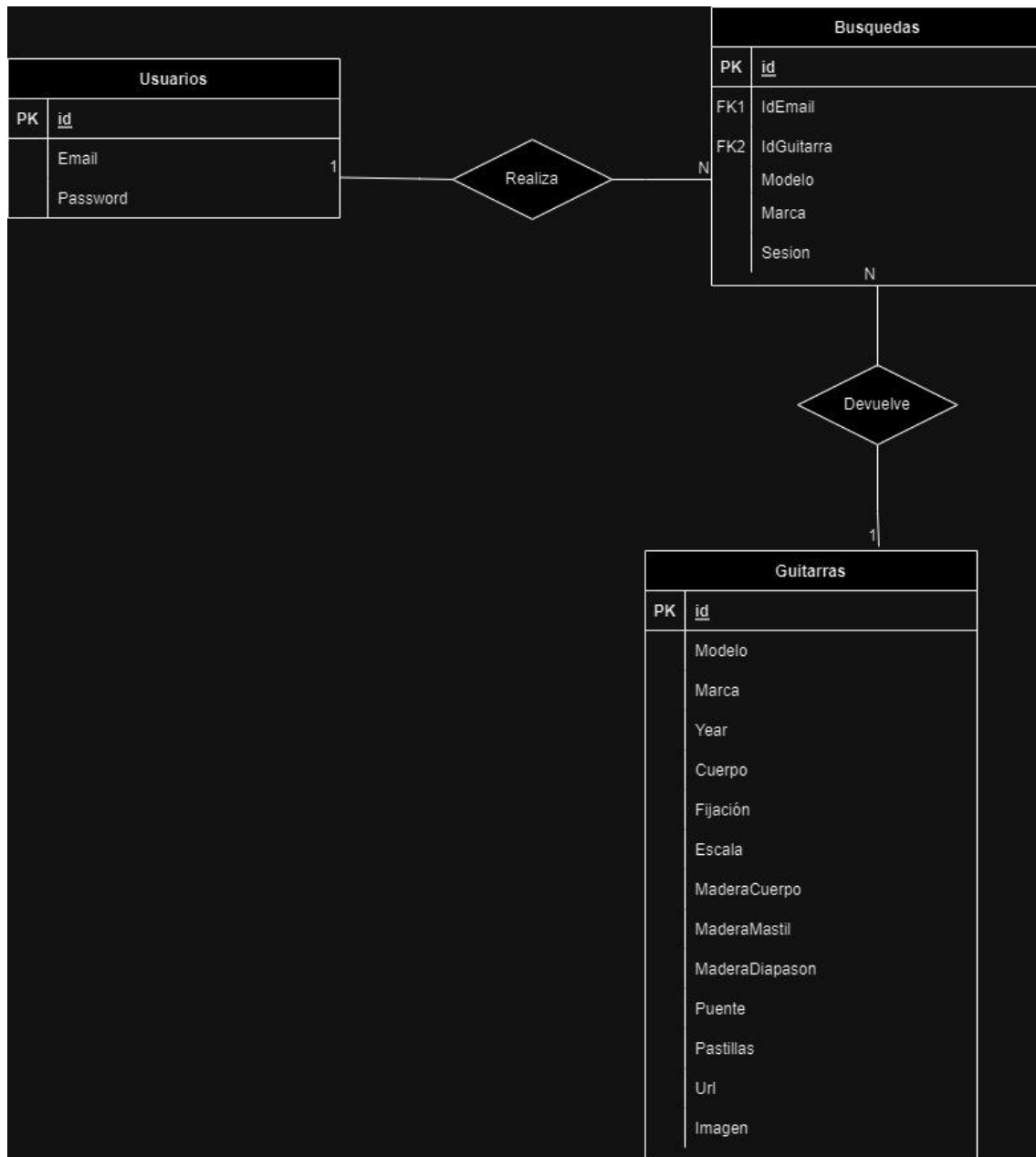


Imagen 13 Modelo E/R

Un usuario de la tabla puede realizar las búsquedas que quiera mientras que una búsqueda siempre va a pertenecer a un usuario únicamente. De esta manera todas las búsquedas que se almacenen en la base de datos quedaran ligadas por el id del usuario que las ha realizado. La búsqueda devolverá una guitarra, pero sin embargo la guitarra puede ser buscada por un numero n de búsquedas uniéndose ambas también por el id de la guitarra.

La tabla usuarios no diferencia roles por la sencilla razón de cómo está diseñada la aplicación. El usuario no va tener contacto físico alguno con los dos programas que consultan



las búsquedas y hacen el CRUD, dado que el usuario es el cliente que entra en la tienda y realiza búsquedas y comparaciones en la Tablet.

3.3 Análisis y diseño del sistema funcional

El diagrama de casos de uso de la aplicación se ha realizado tomando en cuenta las historias de usuario explicadas previamente. Quedaría así:

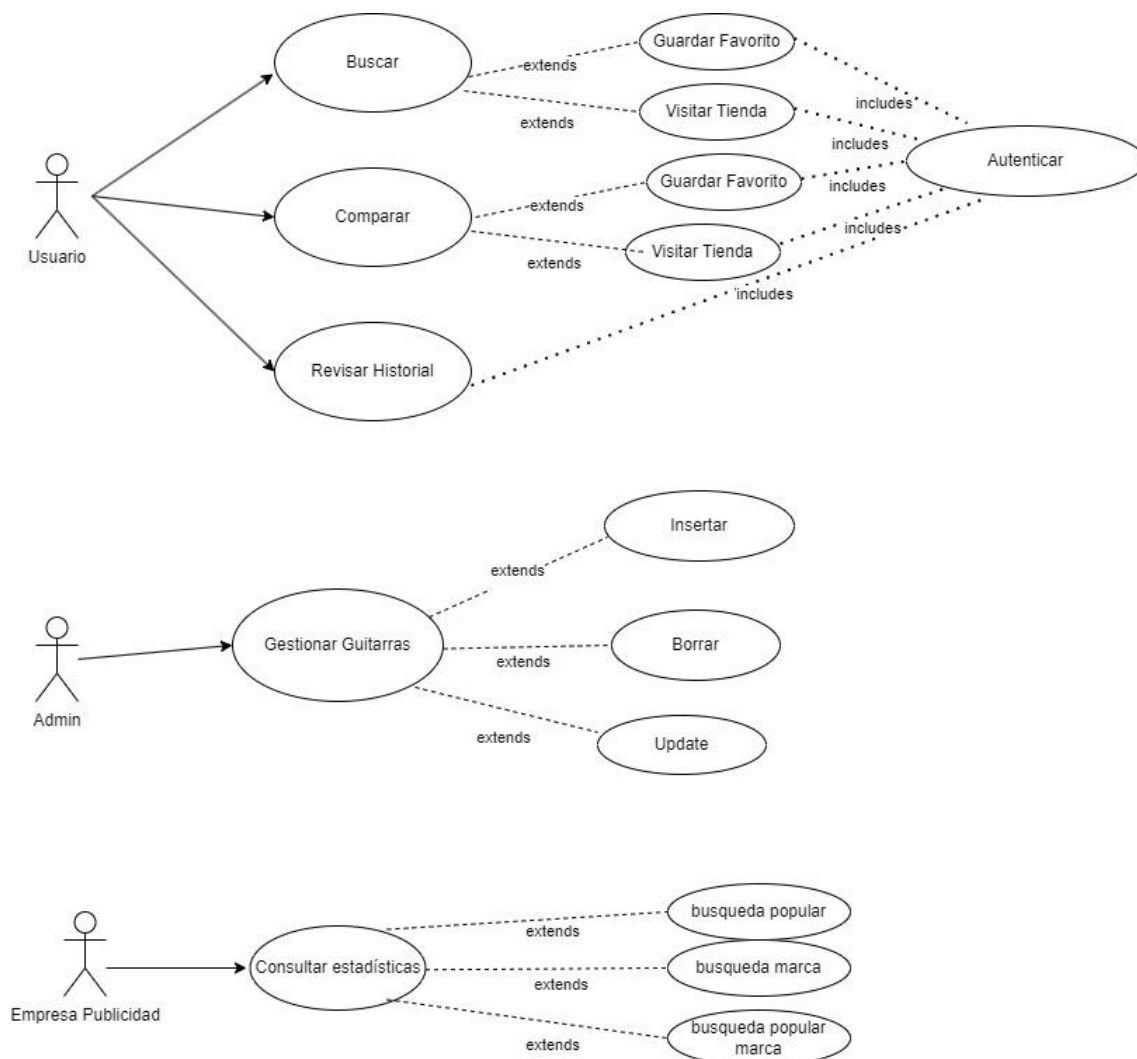


Imagen 14 Diagrama casos de uso



Como se puede apreciar en el diagrama los tres roles están bien diferenciados y no comparten ninguna función, de modo que el usuario es incapaz físicamente de interactuar directamente con la base de datos sino es realizando consultas, además en ningún momento será consciente de las peticiones que está realizando a la base de datos ya sea por selects de la tabla guitarras o los inserts en la tabla de búsquedas. De esta manera la capa de seguridad depende totalmente de la distribución de los dos otros programas, instalándose en la oficina de marketing y del administrador.

La seguridad de los datos del cliente depende pues del administrador de la aplicación que tenga instalado el programa y dado que la base de datos en un inicio será local, será mucho más fácil controlar el flujo de datos y la seguridad de estos. Ningún usuario de la base de datos va a tener acceso a ésta ya que al estar separado físicamente el programa no podrá interactuar con el CRUD ni la sección de publicidad. De la misma manera las personas de marketing no podrán interactuar con el CRUD ya que el administrador lo tendrá instalado en su dispositivo. Para facilitar la tarea del administrador se recomienda tener los tres programas instalados en su dispositivo, así, a tiempo real, podrá comprobar los cambios en el sistema y comprobar búsquedas.

El diagrama de clases del proyecto se parece bastante al diseño del modelado de las tablas de la base de datos, ya que a través de los modelos de las tablas vamos a crear los modelos de los objetos.

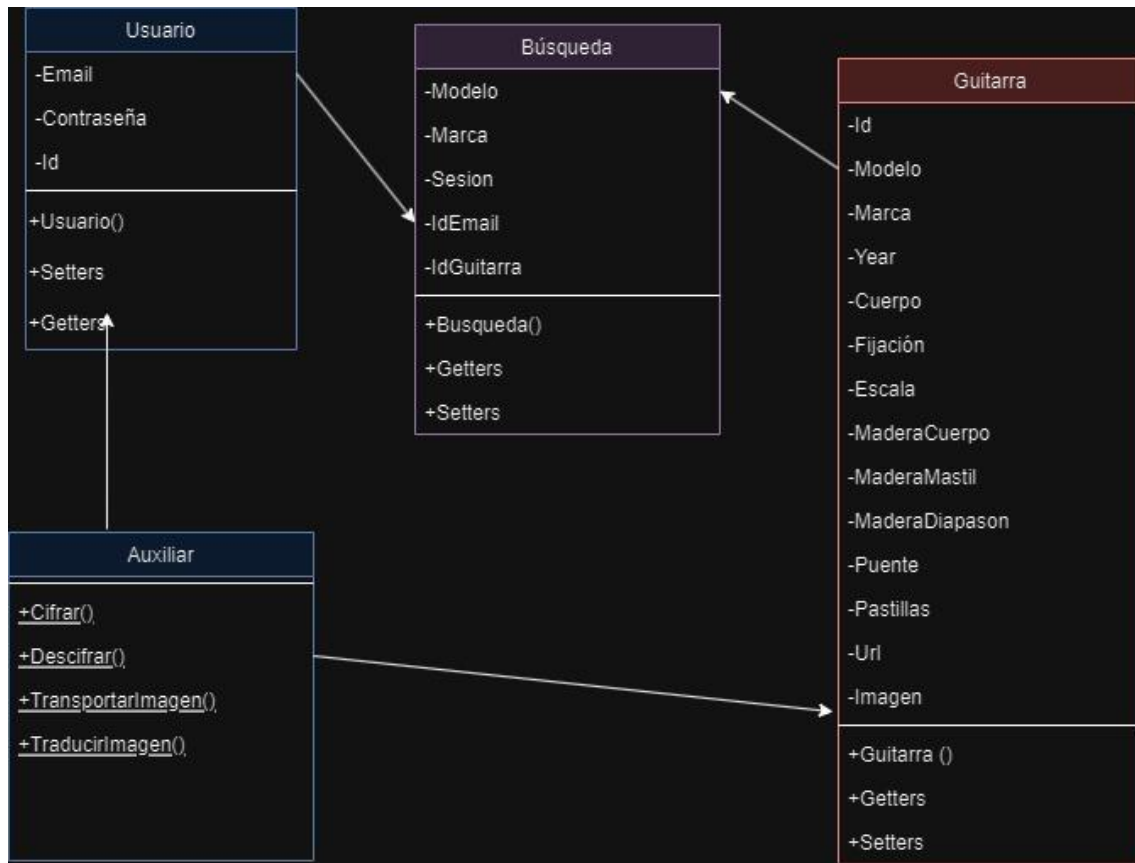


Imagen 15 Diagrama de clases

Con estas clases representaré los modelos de la base de datos y manejaré la información que contienen de la manera más que habitual de utilizar getters para recuperar los datos pertinentes en la situación. Por ejemplo, cuando el usuario realice una búsqueda de guitarras, almacenaremos la consulta en un objeto guitarra con el que luego mostraremos los datos y el los datos recogidos del usuario también los almacenaremos en un objeto usuario que pasaremos al siguiente form para manejar los directorios.

La clase que se desmarca en este caso es la auxiliar que contiene una serie de métodos estáticos que voy a utilizar a lo largo de al menos dos de los programas. Esta clase aportará los métodos de cifrado y descifrado para generar usuarios y almacenarlos en la base de datos. Y por otro lado también aporta la conversión de imagen a String para poder ser almacenada en base de datos y viceversa, la traducción de ese String a imagen de nuevo para poder mostrar la imagen cuando se le llame.

Estas son las clases que voy a utilizar pero métodos habrá más en cada form independiente para añadir funcionalidades a todo, véase los botones, los text fields etc.



4.4 Análisis y diseño de la interfaz de usuario

En esta sección voy a explicar ayudándome de los mockups como moverse por las aplicaciones. Este sería el programa para buscar y comparar guitarras:

Imagen 16 Landing Page

Esta es una sencilla landing page a la que el usuario accederá al iniciar el programa. Idealmente será la interfaz que esté por defecto en todas las tablets del cliente en su punto de compra, aunque también podemos simplemente mostrar el icono de la aplicación y que el usuario clique en ella para iniciarla. Esta opción daría pie a tener más aplicaciones disponibles en la tablet. Esta landing es un simple login que comprobará que el usuario esté registrado en la base de datos y si la contraseña que ha puesto es correcta. Si el usuario no tiene una cuenta puede clicar en registro e ir a la la form de registrar:

The image shows a web browser window titled "Guitarmatch". Inside the window is a registration form. The form consists of three text input fields stacked vertically. The first field is labeled "CorreoElectronico", the second is labeled "Password", and the third is labeled "Repita Password". Below these fields are two buttons: "Registro" on the left and "Volver" on the right. The browser window has a standard title bar with a close button (X) in the top right corner.

Imagen 17 Registro

Esta form recoge los datos del usuario a crear y los cifra para luego insertarlos en la base de datos si se cumple la condición de que la contraseña se repita. Si se registra vuelve al formulario anterior y si no puede volver clicando al botón. De vuelta en el anterior formulario también hay un enlace a los términos y condiciones del uso de la aplicación y registro de datos que se mostrará así:

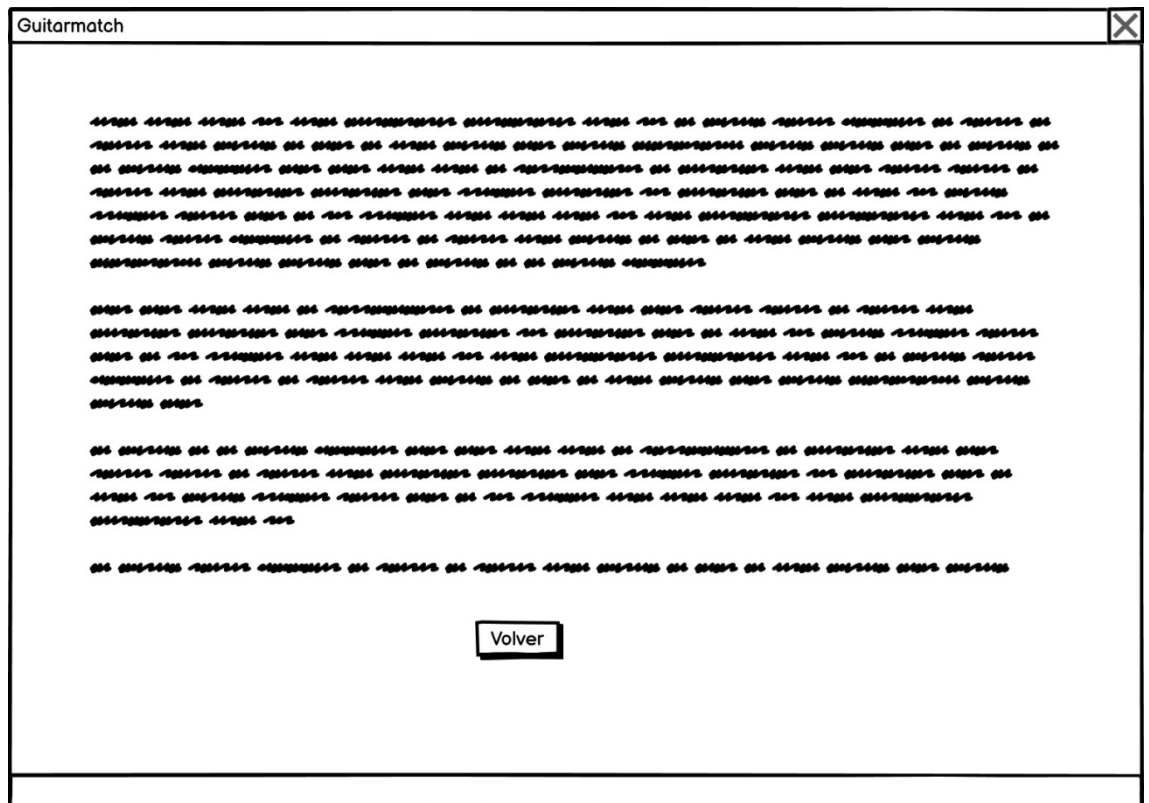


Imagen 18 Términos y condiciones

Simplemente se muestran las condiciones de uso de la aplicación y se informa al usuario donde puede ejercer sus derechos, con un botón para de nuevo volver a la landing page. Al iniciar sesión correctamente la ventana de búsquedas aparecerá, con esta vendrá transportada el objeto usuario que ha realizado el login para después utilizarlo tanto en la lista de búsquedas como en los directorios locales.

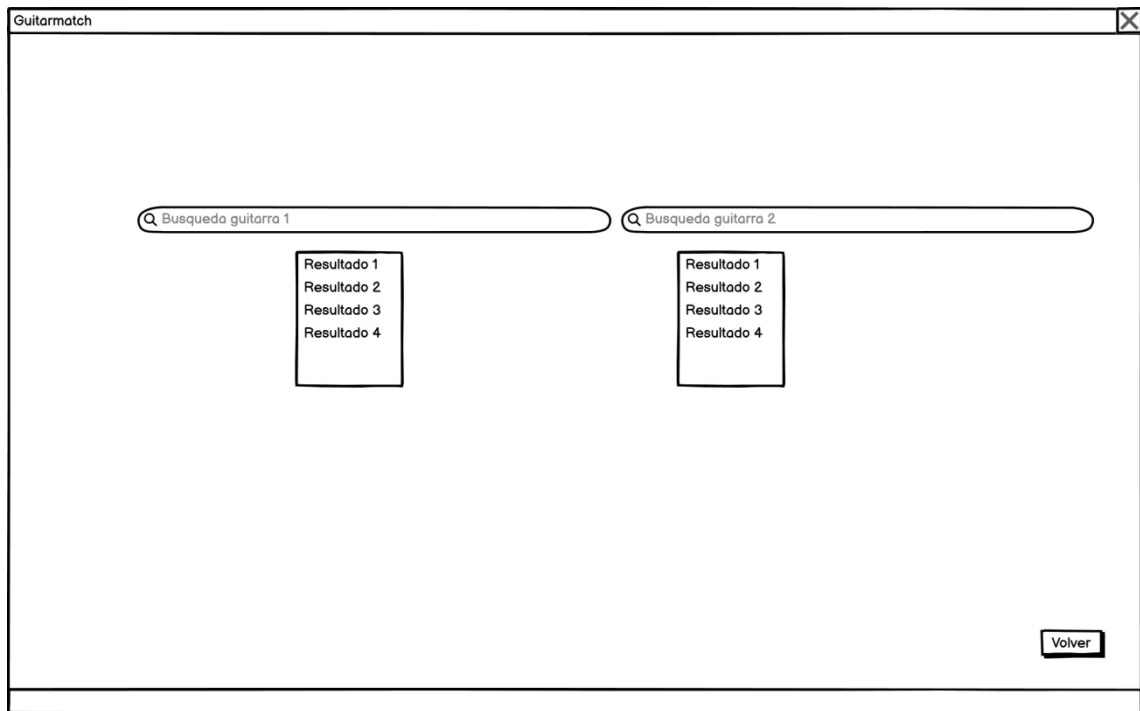


Imagen 19 Buscador

Esta es la interfaz del buscador de guitarras tras haber logueado, a partir de aquí la ventana va a ser la misma. Si se busca una guitarra se nos presentará la guitarra sola con sus características y si se buscan dos se mostrarán como el mockup de abajo para poder hacer una comparación. Habrá, además un botón para añadir a la lista de favoritos del usuario y otro para ver el historial. Por último, el usuario tiene un enlace a tienda que directamente le llevará a la web del fabricante para consultar el precio.

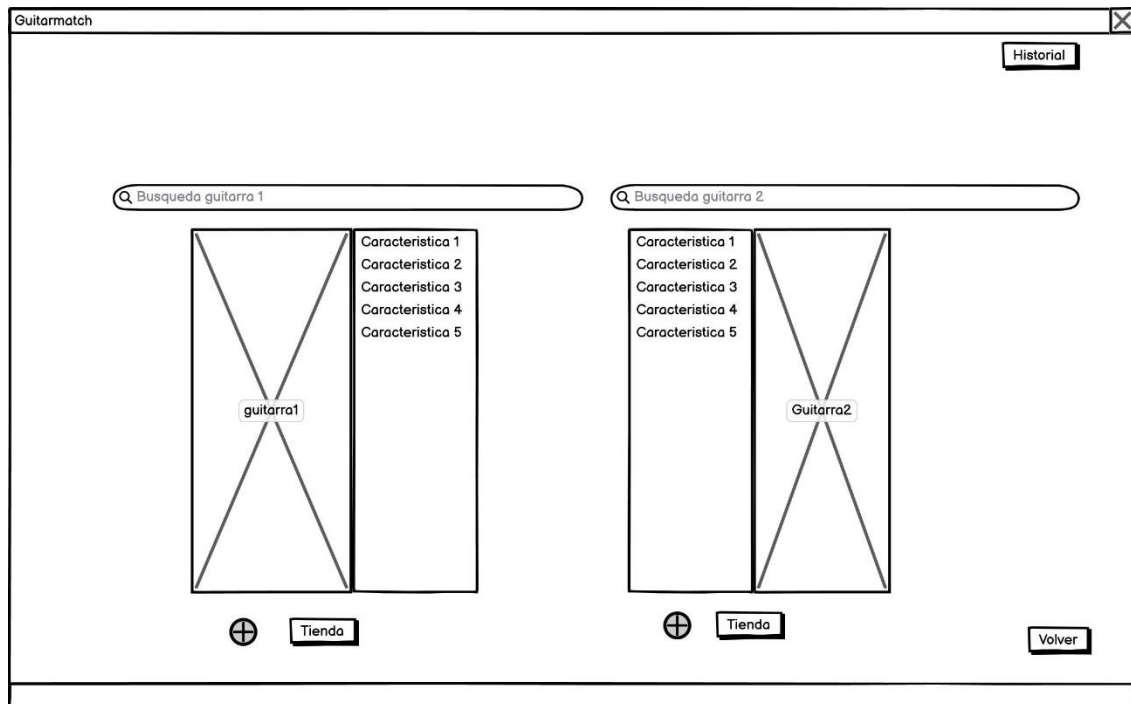


Imagen 20 Buscador con resultado

Como administrador tendremos acceso al formulario del CRUD:



Guitarmatch

Insertar

Borrar

Modelo

Cuerpo

MaderaMastil

MaderaCuerpo

MaderaDiapasón

Puente

Marca

Año

Escala

Fijación

Pastillas

URL

Cargar Imagen

RELLENE TODOS LOS CAMPOS

insertar

Imagen 21 Formulario administrador

En este formulario el administrador puede interactuar con la base de datos de guitarras y puede realizar todas las acciones del CRUD. En el botón de cargar imagen abrirá un explorador para elegir el archivo y cuando se haga clic en insertar se llamará al método de la clase auxiliar para transportarlo.

Por último, la ventana de marketing:

The image shows a wireframe of a web application window titled 'Guitarmatch'. Inside the window, there is a large rectangular area for content. At the top of this area, there are three tabs: 'Más popular', 'Más popular por marca', and 'Menos popular por marca'. The 'Más popular' tab is currently selected. The rest of the content area is empty, suggesting a list of items to be displayed.

Imagen 22 Formulario marketing

Es una ventana en la que el cliente podrá realizar varias consultas diferentes para poder extraer información relevante para generar leads y campañas de publicidad. Como se ha explicado anteriormente, al estar separado este programa de los demás no necesita de usuario porque la capa de seguridad es directamente física.

4.Implementación

Se adjunta el link al repositorio github:

<https://github.com/okronar/ProyectoFinalSergioRamos.git>

La parte de la implementación del código de la aplicación ha sido desarrollada de acorde a las historias de usuario. De esta manera se ha empezado el desarrollo con la creación



de la base de datos sql manager. Como se ha explicado en apartados anteriores la elección de sql como base de datos ha sido por la facilidad de implementación con Visual Studio y el amplio abanico de posibilidades que ofrece.

4.1 Implementación

La mayoría de historias de usuario contemplan relación con la base de datos y en el caso de este proyecto ha sido completar la totalidad de ellas. En la base de datos se han creado tres tablas que nos van a permitir trabajar con la información completamente mediante los diez procedimientos que se han creado de propio para este proyecto. La mayoría de ellos corresponden al ejecutable del administrador en procesos relacionados con el CRUD de la aplicación ya sea el de la tabla guitarras o el de la de usuarios.

HU	Implementación y prueba	Resultado
HU-3 y HU-5 Relacionadas con la información del usuario y los datos recopilados	Registro de Usuario con Email único e irrepetible. Varios filtros de texto Y creación de directorios particulares para cada usuario	El usuario es único por id en la base de datos y no puede repetir Email, se cumplen una serie de condiciones como longitud de textos y al registrar al usuario se crea un directorio con dos archivos diferentes en la carpeta local de la aplicación.
HU-3 y HU-5 Relacionadas con la información del usuario y los datos recopilados	Inserción de búsquedas en la Base de datos teniendo en cuenta los datos del usuario, manejo de la información.	El usuario se guarda cuando se hace login y la aplicación es capaz de utilizar esta información a posteriori. Se introducen los datos en la tabla de búsquedas con el usuario y se escribe un historial en el directorio local.
HU-1 y HU-7 Búsqueda	Búsqueda y presentación de la información	La aplicación realiza búsquedas concretas o parciales y devuelve la información por pantalla de manera concisa y clara.
HU-2 Comparativa	La búsqueda proporcionada puede ser comparada con otra guitarra	La aplicación permite una comparativa de las dos búsquedas presentando las guitarras para poder distinguir entre sus características.
HU-6	Se recopila información para la creación de una aplicación de marketing	La información se almacena correctamente en tablas y permite realizar las consultas requeridas en la aplicación creada para el equipo de marketing.

Tabla 6 Implementación

Tras crear la base de datos se han creado las clases que recogen dichos datos de las tablas para una mayor legibilidad del código además de una clase extra para tratar con diferentes informaciones y eventos llamada auxiliar. El primer programa creado ha sido el de Administración para implementar el CRUD y tener una manera de realizar cambios y pruebas con la base de datos. Se ha creado usando un solo formulario de visual studio al que se le han añadido dos tab pages, uno para guitarras y el otro para usuarios, y dentro del de guitarras otro tab page más para cambiar entre insertar, update y delete.

Una vez creada esta app se creó la del buscador empezando por el login y el registro, comprobando que se insertaran y crearan los datos de los usuarios tanto en la base de datos



como en el directorio local. Comprobados estos puntos, se pasó a la creación del formulario principal que contiene el buscador y el comparador. Finalmente se hizo un formulario para contener las búsquedas de marketing.

4.2 Pruebas

Para la realización de las pruebas se ha pedido a diferentes personas que interactúen con la aplicación. Se ha pedido que se prueben todos los botones de los tres programas para revisar que las funcionalidades cumplan con lo establecido en las pruebas de implementación, y se ha pedido un registro completo de usuarios, así como una serie de búsquedas para cerciorar de que tanto la base de datos como los directorios locales funcionan según lo previsto.



Bibliografía

Sutterland, J. (13 de Enero de 2010). "Iterative vs. Incremental Development."
Recuperado de: <https://www.scruminc.com/iterative-vs-incremental-development/>

Patton, J. (2006). "Incremental Releases" [Archivo PDF]. Recuperado de:
<https://www.jpattonassociates.com/wp-content/uploads/2015/01/JPA-Incremental-Releases-handout-supplement.pdf>

Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.

Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/46/CE (Reglamento general de protección de datos).

Findmyguitar TheGuitarDataBase (s.f.). Recuperado en Marzo de 2024 de:
<https://findmyguitar.com/>

Angel Álvarez, M. (20 de Septiembre de 2023). "Qué es MVC" Recuperado de:
<https://desarrolloweb.com/articulos/que-es-mvc.html>

Martín, M. (30 de Diciembre de 2023). "Modelo incremental en SDLC: uso, ventajas y desventajas" Recuperado de: <https://www.guru99.com/es/what-is-incremental-model-in-sdlc-advantages-disadvantages.html>



Anexo 1

Fuente y términos de la licencia: <https://www.gnu.org/licenses/licenses.es.html>

Anexo 2

Script de creación de la base de datos SQL:

[ProyectoSergio]

```
USE [master]
GO
```

```
/****** Object: Database [ProyectoSergio] Script Date: 17/04/2024 16:37:16 *****/
CREATE DATABASE [ProyectoSergio]
CONTAINMENT = NONE
ON PRIMARY
( NAME = N'ProyectoSergio', FILENAME = N'D:\SqlServer\MSSQL16.MSSQLSERVER\MSSQL\DATA\ProyectoSergio.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )
LOG ON
( NAME = N'ProyectoSergio_log', FILENAME = N'D:\SqlServer\MSSQL16.MSSQLSERVER\MSSQL\DATA\ProyectoSergio_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )
WITH CATALOG_COLLATION = DATABASE_DEFAULT, LEDGER = OFF
GO
ALTER DATABASE [ProyectoSergio] SET COMPATIBILITY_LEVEL = 160
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [ProyectoSergio].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [ProyectoSergio] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [ProyectoSergio] SET ANSI_NULLS OFF
GO
ALTER DATABASE [ProyectoSergio] SET ANSI_PADDING OFF
GO
ALTER DATABASE [ProyectoSergio] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [ProyectoSergio] SET ARITHABORT OFF
GO
ALTER DATABASE [ProyectoSergio] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [ProyectoSergio] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [ProyectoSergio] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [ProyectoSergio] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
```

```
ALTER DATABASE [ProyectoSergio] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [ProyectoSergio] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [ProyectoSergio] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [ProyectoSergio] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [ProyectoSergio] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [ProyectoSergio] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [ProyectoSergio] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [ProyectoSergio] SET DISABLE_BROKER
GO
ALTER DATABASE [ProyectoSergio] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [ProyectoSergio] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [ProyectoSergio] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [ProyectoSergio] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [ProyectoSergio] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [ProyectoSergio] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [ProyectoSergio] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [ProyectoSergio] SET RECOVERY FULL
GO
ALTER DATABASE [ProyectoSergio] SET MULTI_USER
GO
ALTER DATABASE [ProyectoSergio] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [ProyectoSergio] SET DB_CHAINING OFF
GO
ALTER DATABASE [ProyectoSergio] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [ProyectoSergio] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
ALTER DATABASE [ProyectoSergio] SET DELAYED_DURABILITY = DISABLED
GO
ALTER DATABASE [ProyectoSergio] SET ACCELERATED_DATABASE_RECOVERY = OFF
GO
EXEC sys.sp_db_vardecimal_storage_format N'ProyectoSergio', N'ON'
GO
ALTER DATABASE [ProyectoSergio] SET QUERY_STORE = ON
GO
ALTER DATABASE [ProyectoSergio] SET QUERY_STORE (OPERATION_MODE = READ_WRITE, CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = 30), DATA_FLUSH_INTERVAL_SECONDS = 900, INTERVAL_LENGTH_MINUTES = 60, MAX_STORAGE_SIZE_MB = 1000, QUERY_CAPTURE_MODE = AUTO, SIZE_BASE
```



[dbo].[Busquedas]

```
USE [ProyectoSergio]
```

```
GO
```

SQL

```
/****** Object: Table [dbo].[Busquedas] Script Date: 17/04/2024 16:37:17 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[Busquedas](  
    [Id] [int] IDENTITY(1,1) NOT NULL,  
    [Modelo] [varchar](50) NOT NULL,  
    [Marca] [varchar](50) NOT NULL,  
    [Sesion] [int] NOT NULL,  
    [IdEmail] [int] NOT NULL,  
    [IdGuitarra] [int] NOT NULL,  
    CONSTRAINT [PK_Busquedas] PRIMARY KEY CLUSTERED  
(  
    [Id] ASC  
    )
```

```
WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]  
) ON [PRIMARY]  
GO
```

SQL

[dbo].[Guitarras]

```
/****** Object: Table [dbo].[Guitarras] Script Date: 17/04/2024 16:37:17 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[Guitarras](  
    [Modelo] [varchar](50) NOT NULL,  
    [Marca] [varchar](50) NOT NULL,  
    [Id] [int] IDENTITY(1,1) NOT NULL,  
    [Year] [varchar](50) NOT NULL,  
    [Cuerpo] [varchar](50) NOT NULL,  
    [Fijación] [varchar](50) NOT NULL,  
    [Escala] [varchar](50) NOT NULL,  
    [MaderaCuerpo] [varchar](50) NOT NULL,  
    [MaderaMastil] [varchar](50) NOT NULL,  
    [MaderaPlapason] [varchar](50) NOT NULL,  
    [Puente] [varchar](50) NOT NULL,  
    [Pastillas] [varchar](50) NOT NULL,  
    [Url] [varchar](50) NOT NULL,  
    [Imagen] [varchar](max) NOT NULL,  
    CONSTRAINT [PK_Guitarras] PRIMARY KEY CLUSTERED  
(  
    [Id] ASC  
    )
```

```
WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]  
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]  
GO
```

SQL

[dbo].[Usuarios]

```
/****** Object: Table [dbo].[Usuarios] Script Date: 17/04/2024 16:37:17 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[Usuarios](  
    [Email] [nvarchar](50) NOT NULL,  
    [Password] [nvarchar](50) NOT NULL,  
    [Id] [int] IDENTITY(1,1) NOT NULL,  
    CONSTRAINT [PK_Usuarios] PRIMARY KEY CLUSTERED  
(  
    [Id] ASC  
    )
```

```
WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]  
) ON [PRIMARY]  
GO
```

SQL



[FK_Busquedas_Guitarras]

```
ALTER TABLE [dbo].[Busquedas] WITH CHECK ADD CONSTRAINT [FK_Busquedas_Guitarras] FOREIGN KEY([IdGuitarra])
REFERENCES [dbo].[Guitarras] ([Id])
GO
ALTER TABLE [dbo].[Busquedas] CHECK CONSTRAINT [FK_Busquedas_Guitarras]
GO
```

SQL

[FK_Busquedas_Usuarios]

```
ALTER TABLE [dbo].[Busquedas] WITH CHECK ADD CONSTRAINT [FK_Busquedas_Usuarios] FOREIGN KEY([IdEmail])
REFERENCES [dbo].[Usuarios] ([Id])
GO
ALTER TABLE [dbo].[Busquedas] CHECK CONSTRAINT [FK_Busquedas_Usuarios]
GO
```

SQL

[ProyectoSergio]

```
USE [master]
GO
```

SQL

```
ALTER DATABASE [ProyectoSergio] SET READ_WRITE
GO
```

SQL