

HORSEBOOKING

Almudena Iparraguirre Castillo



8 DE ABRIL DE 2024

DESARROLLO DE APLICACIONES MULTIPLATAFORMA 2ªA

MARIA ANA SANZ

Agradecimientos

En primer lugar, agradecer a mi familia por todo el apoyo durante todos estos años, por su confianza y sus consejos. Gracias por empujarme a perseguir mis sueños.

A mis profesores por todo el apoyo y todos los contenidos enseñados. Por su dedicación y siempre estar ahí ante las dudas y los problemas surgidos.

A mis compañeros por hacer la vida en clase más amena, por hacerme reír siempre y por acogirme como una más en el grupo.

A mis amigas por su apoyo y por estar conmigo ahí siempre.

Índice

1. EXPLICACIÓN DE PROYECTO.....	4
1.1.INTRODUCCIÓN	4
1.2.ESTUDIO DE MERCADO	4
1.2.1. Definición de un objetivo	4
1.2.2. Preguntas de investigación	7
1.2.3. Recolección de datos.....	9
1.2.4. Análisis de resultados.....	10
1.2.5. Informe de resultados	10
2. OBJETIVOS	11
2.1.OBJETIVOS COMPROMETIDOS	11
2.2.OBJETIVOS NO COMPROMETIDOS	11
3. RECURSOS HARDWARE Y SOFTWARE.....	12
4. DESARROLLO DE LAS FASES	15
4.1. PLANIFICACIÓN	15
4.2. PLANIFICACIÓN DISEÑO	16
4.3. DISEÑO	19
4.4. DESARROLLO	29
4.5. PRUEBAS	49
4.5.1. Pruebas unitarias.....	49
4.5.2. Pruebas de integración.....	50
5. PRESUPUESTO PARA EL PROYECTO	52
6. LANZAMIENTO Y PUESTA EN MARCHA	53
7. CONCLUSIONES	55
8. BIBLIOGRAFÍA	56

1. EXPLICACIÓN DE PROYECTO

1.1. INTRODUCCIÓN

La aplicación desarrollada gestiona las **reservas, pagos o novedades** de una escuela de equitación. Esto es, facilita a los trabajadores la **organización** de las clases con los alumnos, permite organizar más fácilmente los caballos a asignar a los alumnos en cada clase, también tiene la opción de informar a través de esta las novedades, actualizaciones o noticias que haya en la hípica.

Asimismo, existe la posibilidad de que cada usuario sepa el nivel de experiencia que tiene en su perfil. Por último, tanto por parte de los trabajadores como por parte de los usuarios y alumnos de la escuela, se pueden controlar los pagos de cada clase, paseo o competición (si está o no pagado).

La idea del proyecto surgió gracias a que la experiencia obtenida en una hípica, ya que se sabe la cantidad de tiempo que lleva gestionar todas las clases, pagos, paseos o competiciones de manera manual.

Por ello surge **HorseBooking**, una aplicación que permite a los alumnos reservar sus clases sin tener que ponerse en contacto con los trabajadores, por lo que ahorra a estos el trabajo de apuntar y desapuntar todos los alumnos en cada clase.

Este proyecto podrá ser modificado y vendido a distintas escuelas, ya que se podría adaptar fácilmente a las distintas hípicas. Tan solo haría falta modificar la interfaz, ya que tanto el modo de funcionamiento como la lógica de la aplicación acabarían basándose en lo mismo.

1.2. ESTUDIO DE MERCADO

- Para evaluar la **necesidad y demanda** que va a obtener la aplicación HorseBooking se va a realizar un **estudio de mercado** que evalúe la cantidad de **beneficios** que puede tener el proyecto, para así seguir adelante con él o cancelar su continuación.
- Para realizar el estudio se van a seguir unos **pasos** que finalmente lleven a la conclusión de si esta aplicación es **viable o no viable**:

1.2.1. Definición de un objetivo

- **Naturaleza del mercado y sus tendencias:** El mercado de la hípica es un mercado constante, no está en declive y lleva años en funcionamiento, por lo que no está emergiendo, está en un momento de crecimiento constante.
- **Tamaño de mercado:** El mercado de la hípica y la equitación es enorme, es un mercado a nivel global. Tan solo en España, hay un total de 1.637 centros de hípica.

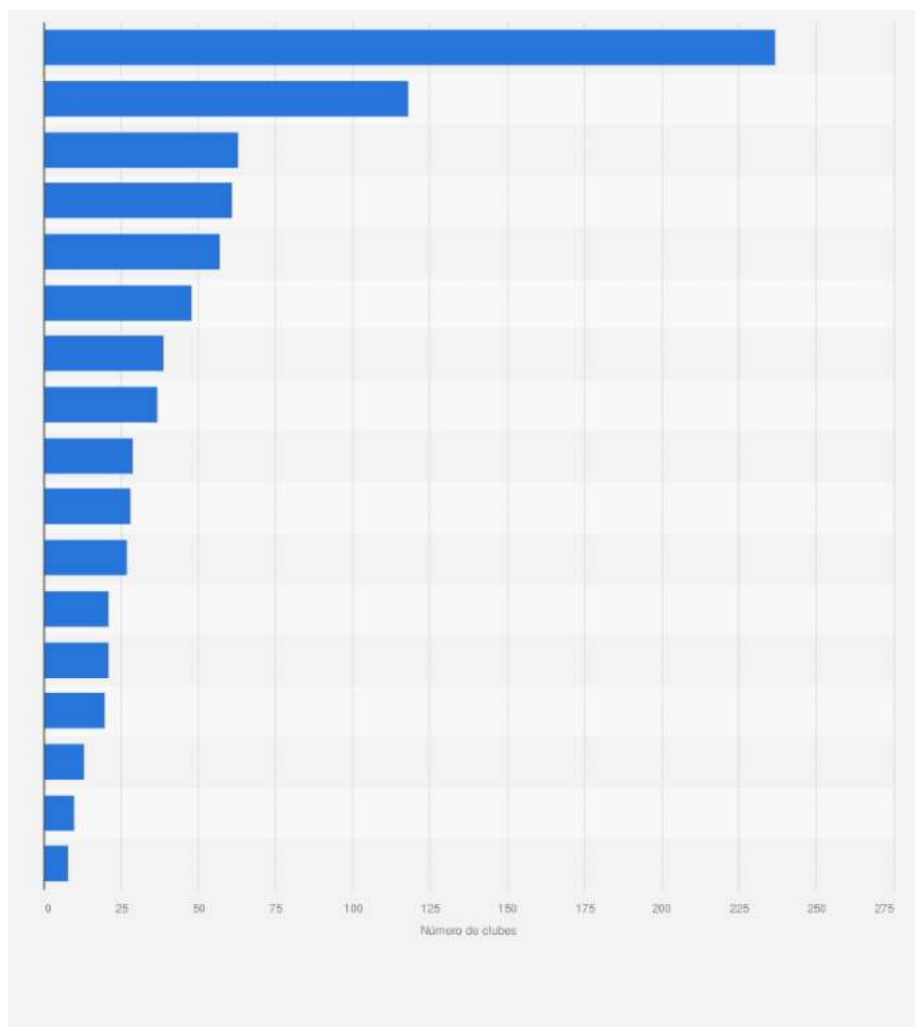


Gráfico de clubes federados de hípica por comunidad autónoma. 2019.

- En el mercado de la hípica **la tecnología** está involucrándose poco a poco, las aplicaciones de hípica, ya sea de reservas, de gestión, o de compras aumentan poco a poco para ayudar a estas escuelas a aumentar su fama y sus clientes y a facilitar las gestiones de sus negocios.
- **Clientes objetivo:** Los principales clientes de HorseBooking estarán directa o indirectamente relacionados con el centro de equitación que disponga de la aplicación. Principalmente los clientes a los que irá dirigida la app serán alumnos de la escuela, jinetes, amazonas, padres de alumnos, profesores de la hípica o público general al que le interese realizar alguna actividad en el centro.
- **Competencia:** Los posibles competidores actuales dentro del sector serán aplicaciones similares que ofrezcan funciones de gestión o reservas a las distintas hípicas. Para poder distinguirse de la competencia habrá que analizar las especificaciones de los distintos competidores. A continuación, se realizar un análisis rápido sobre tres posibles competidores:

a) **Gesthip**

- **Gesthip** es una web y aplicación de gestión para centros hípicos y que ayuda a controlar las asistencias, las horas de trabajo de los caballos, las facturas, asistencia de los trabajadores, el calendario...
- **Fortalezas:** Facilidad de uso, es multiplataforma, por lo que se puede utilizar en múltiples dispositivos y amplia gama de funciones de gestión.

b) **Equipe**

- **Equipe** es una aplicación móvil que permite a los usuarios seguir en tiempo real las competiciones de caballos que haya a nivel local, nacional e internacional.
- **Fortalezas:** Muy intuitiva para los usuarios, actualización en tiempo real de los eventos.

c) **Happie Horse – Gestión**

- **Happie Horse** es una aplicación de gestión que permite al usuario administrar los distintos aspectos de la vida del caballo, más concretamente la organización del establo, el plan de alimentación del animal, la gestión de su salud y el seguimiento de sus síntomas, así como realizar un seguimiento, una programación y un análisis del entrenamiento.
- **Fortalezas:** Recordatorio de tareas, aplicación móvil con interfaz intuitiva, con ejercicios y vídeos explicativos incluidos.



1.2.2. Preguntas de investigación

a) Elección de un tema general

- El primer paso en el apartado de preguntas de investigación es la elección de un tema general interesante para el proyecto.

Este punto ayudará en la investigación de áreas específicas que puedan ayudar en el desarrollo y lanzamiento de la aplicación. Dos áreas que podrían adecuarse en el desarrollo de la investigación podrían ser:

- A) **Eficacia de las reservas online para las escuelas de equitación:** Investigar la eficacia y ventajas que pueden ofrecer las reservas digitales en una hípica.
- B) **Gestión de las relaciones con los clientes:** Exploración de que qué manera puede la aplicación facilitar la comunicación y relación entre los entrenadores y los alumnos (desarrollo de mensajería, notificaciones, actualizaciones de noticias).

b) Investigación preliminar del tema en cuestión

- El paso previo a definir las preguntas de investigación es vital realizar la investigación preliminar con temas relacionados a la gestión de reservas en escuelas de equitación, aplicaciones de reservas...

Esto ayudará a entender las tendencias actuales en el mundo de la tecnología y del caballo, que permitirá realizar preguntas más definidas e informadas.

- a) **Identificación de los temas clave:** Mencionados en el punto anterior *Elección de un tema general*, los temas clave son la eficacia de las reservas online para las escuelas de equitación y la gestión de las relaciones con los clientes.
- b) **Búsqueda de fuentes relevantes:** Se debe realizar una búsqueda profunda de fuentes, artículos, noticias o publicaciones directamente relacionadas con los temas anteriores.
- c) **Revisión de lo existente:** Se analiza y lee la literatura existente sobre los temas seleccionados anteriormente. Esto ayudará a comprender los hallazgos de otros investigadores en el propio tema en cuestión.
- d) **Extracción de la información relevante:** Toma de notas detalladas sobre lo más relevante encontrado.

c) Acotación

- Tras realizar una exhaustiva investigación preliminar sobre la gestión de escuelas de equitación, las aplicaciones de reserva en línea y otros aspectos relevantes para el proyecto HorseBooking, se ha procedido a acotar el tema de investigación. En este sentido, se ha identificado la necesidad de centrarse en un aspecto específico de la gestión de las reservas, con el objetivo de profundizar en un área particular y obtener resultados significativos para el desarrollo de HorseBooking.
- El aspecto seleccionado para acotar el tema de investigación se centra en la experiencia del usuario en la aplicación de reserva de clases de HorseBooking. Esta decisión se basa en la importancia de garantizar una experiencia de usuario óptima para los clientes de la escuela de equitación, lo cual puede influir directamente en la efectividad y la satisfacción del servicio ofrecido.
- Al enfocarse en la experiencia del usuario en la aplicación de reserva de clases, se busca analizar diversos aspectos, como la usabilidad de la interfaz, la claridad de la información proporcionada, la facilidad de navegación y la eficacia del proceso de reserva. Esto permitirá identificar posibles áreas de mejora y optimización en la aplicación, con el fin de ofrecer a los usuarios una experiencia intuitiva y satisfactoria que fomente la fidelización y la recomendación.

d) Redacción y evaluación de la pregunta

- Tras la formulación de la pregunta de investigación centrada en la experiencia del usuario en la aplicación de reserva de clases de HorseBooking, es fundamental realizar una evaluación exhaustiva para garantizar su claridad, especificidad y relevancia para los objetivos del proyecto.
- La pregunta de investigación planteada es la siguiente: **"¿Cómo afecta la experiencia del usuario en la aplicación de reserva de clases de HorseBooking a la efectividad y satisfacción del servicio ofrecido por la escuela de equitación?"**

Clara y específica: La pregunta es clara y específica en su enfoque en la experiencia del usuario en la aplicación de reserva de clases de HorseBooking. Define claramente el aspecto que se va a investigar y proporciona un marco claro para la investigación.

- Relevante para los objetivos del proyecto: La pregunta es altamente relevante para los objetivos del proyecto HorseBooking, ya que se centra en un aspecto crítico para la efectividad y la satisfacción del servicio ofrecido por la escuela de equitación. Aborda directamente el tema de interés y tiene el potencial de generar información valiosa para mejorar la aplicación y la experiencia del usuario.
- Abordable y de interés para la audiencia: La pregunta de investigación es abordable y de interés tanto para la audiencia del proyecto como para los posibles usuarios de la aplicación HorseBooking. Ofrece una perspectiva relevante y práctica que puede generar un interés genuino en aquellos involucrados en el desarrollo y la utilización de la aplicación.
- En general, la pregunta de investigación cumple con los criterios de claridad, especificidad y relevancia para los objetivos del proyecto HorseBooking. No obstante, se estará atento a cualquier ajuste necesario durante el proceso de investigación para garantizar su efectividad y pertinencia continua.

1.2.3. Recolección de datos

- La recolección de datos busca obtener datos en tiempo real y se suele realizar a través de encuestas online.

En este caso se va a realizar una encuesta online a través de [QuestionPro](#), una web que permite crear encuestas de manera fácil y rápida.



- En la encuesta se van a realizar distintas preguntas que van a ser de distintas categorías, esto es, va a haber preguntas directas, indirectas, abiertas, cerradas, reflexivas...
- [Aquí](#) se muestra la encuesta realizada para obtener información sobre HorseBooking.

1.2.4. Análisis de resultados

- Para realizar un análisis de los resultados hay que seguir unos pasos definidos a continuación:

1) **Revisión de datos**

- Se realiza una revisión de los datos recopilados en la encuesta

2) **Identificación de tendencias**

- Búsqueda de las similitudes y las diferencias entre las respuestas de las personas encuestadas.

3) **Análisis de respuestas abiertas**

- En caso de la presencia de preguntas abiertas en la encuesta, analizar las distintas respuestas para obtener información adicional y las distintas perspectivas sobre una misma pregunta.

4) **Segmentación de datos**

- Se segmentan los datos según las distintas categorías demográficas de los usuarios (edad, género, ciudad, nacionalidad, nivel de experiencia en la equitación).

5) **Identificación de “insights”**

- Búsqueda de insights (descubrimientos), que surgen en el análisis de datos. Esto puede proporcionar información relevante sobre las necesidades y las preferencias de los usuarios.

6) **Comparación con objetivos**

- Comparación de los resultados de la encuesta con los objetivos iniciales.

7) **Preparación de informe**

- Preparación de un informe detallado que resuma los hallazgos del análisis de los datos.

8) **Conclusiones**

- Elaboración de conclusiones basadas en los descubrimientos obtenidos.

1.2.5. Informe de resultados

- Una vez obtenidos los resultados y las estadísticas de la encuesta desarrollada en “QuestionPro”, se puede llegar a la conclusión de que es un proyecto completamente viable y puede tener beneficios, tanto para los desarrolladores como para la escuela de equitación.

2. OBJETIVOS

- Previamente al transcurso del desarrollo de la aplicación, se han establecido unos objetivos **comprometidos** (objetivos principales) y unos objetivos **no comprometidos** (objetivos secundarios). Seguidamente se muestran los distintos objetivos establecidos:

2.1. OBJETIVOS COMPROMETIDOS

1. Registrar a un alumno y un trabajador con éxito en la base de datos.
2. Poder iniciar sesión con éxito en la aplicación tanto de móvil como de escritorio.
3. Que los alumnos puedan reservar una clase y que los trabajadores puedan controlar qué alumnos y clientes están apuntados para asignar caballos y actividades.
4. Que los trabajadores puedan actualizar novedades y noticias de la hípica en la aplicación de escritorio para que a su vez se actualicen en la aplicación móvil.
5. Mostrar a los alumnos sus datos, novedades o avisos en el perfil, así como el nivel de experiencia de este.

2.2. OBJETIVOS NO COMPROMETIDOS

1. Chat de comunicación entre profesores y alumnos tanto en la aplicación móvil como en la aplicación de escritorio. Si un alumno escribiera a un profesor este podría observar la notificación tanto en el móvil como en la aplicación de escritorio.
2. Prueba de nivel teórico-práctico para verificar el nivel del alumno en la app.
3. Ventana para la gestión de competiciones e integración con redes sociales.

3. RECURSOS HARDWARE Y SOFTWARE

- Para desarrollar el proyecto presente, se va a hacer uso de recursos tanto hardware como software. A continuación, se especifican los recursos necesarios para el desarrollo de esta aplicación:

a) Recursos hardware

1. Un portátil o un ordenador de sobremesa con la capacidad suficiente para soportar las distintas ejecuciones de prueba.

En este caso se hará uso de un **portátil hp** con una RAM de 16gb, Windows 11 y un sistema operativo de 64 bits.

2. De manera opcional en un portátil, un **ratón** para más comodidad de manejo (de manera obligatoria en un ordenador de sobremesa).

En este caso se agregará un ratón externo al portátil para hacer más cómodo su uso y manejo, el teclado usado será el del portátil.

3. **Dispositivo móvil** para las pruebas en Android Studio: Es opcional, pero bastante cómodo contar con uno o más dispositivos físicos para realizar las pruebas pertinentes de la aplicación móvil.

En el proyecto presente se utilizará un Samsung para realizar los tests, ejecutar e instalar la aplicación móvil desde Android Studio.

b) Recursos software

1. **Android studio**: Este entorno de desarrollo es necesario para desarrollar la aplicación del móvil, concretamente en Android Studio con el lenguaje de programación Kotlin.



2. **Visual studio (2019):** Este entorno hace posible el desarrollo de la aplicación de escritorio para los trabajadores, donde podrán gestionar las reservas, los pagos y otras funciones de la hípica.



3. **Word:** Informe de los objetivos, recursos, desarrollo de las fases y conclusiones del avance y desarrollo del proyecto.



4. **Github:** Esta herramienta de desarrollo permite la gestión más sencilla del proyecto, de tal manera que se puedan utilizar distintos ordenadores para avanzar con el proyecto donde se dejó.

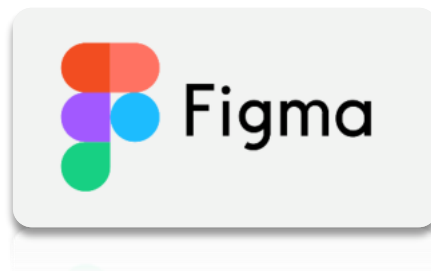


5. **Firebase:** Este sistema gestor de bases de datos almacena toda la información necesaria para que HorseBooking funcione de manera correcta.



6. **Emuladores para pruebas:** En el caso de no contar con un dispositivo físico para las pruebas de Android Studio, se pueden utilizar los distintos emuladores que ofrece Android para realizar las distintas pruebas.

7. **Figma:** Figma es una herramienta de diseño en la que se pueden realizar distintos prototipos de la aplicación y así poder hacer más sencilla la decisión del diseño definitivo.

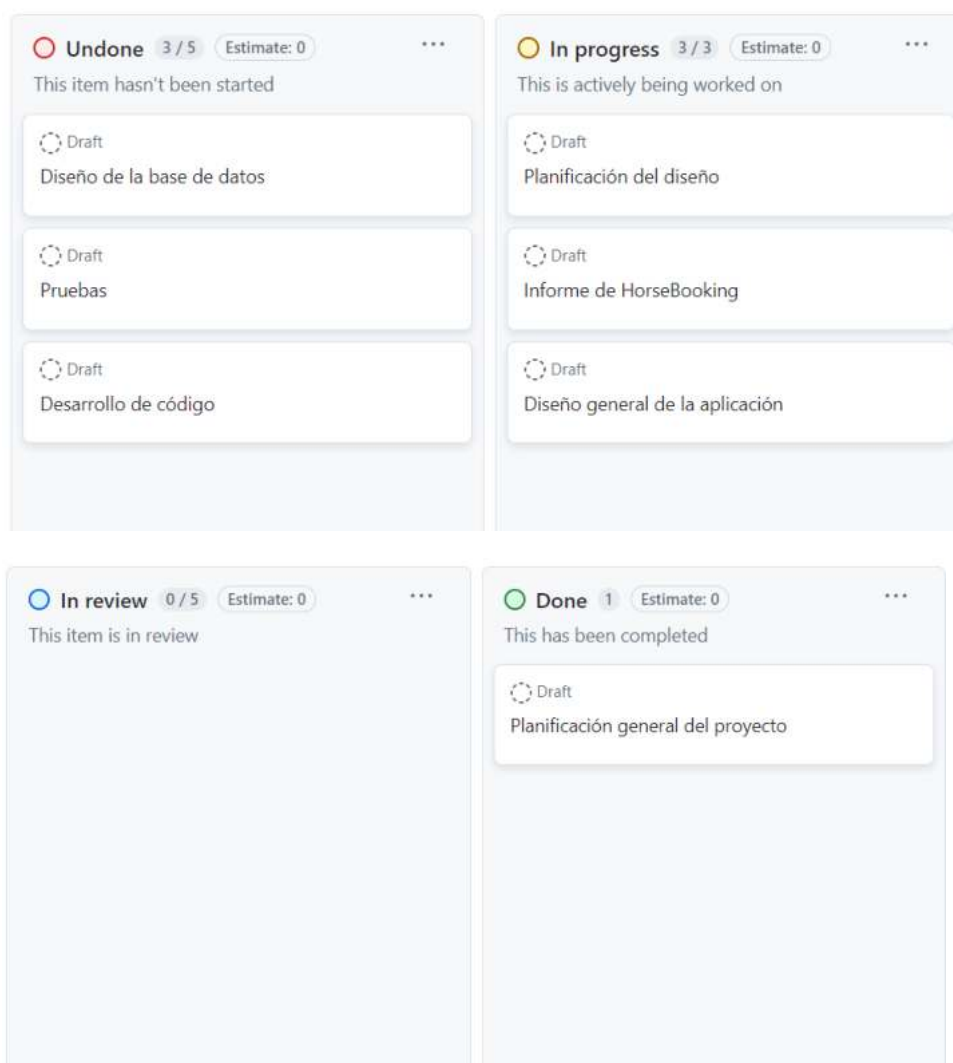


4. DESARROLLO DE LAS FASES

4.1. PLANIFICACIÓN

- Para el desarrollo de una aplicación es necesaria una **planificación inicial**, un mapa mental de en qué va a consistir, cómo va a ser, en qué plataformas se va a lanzar...

A continuación, se va a realizar la explicación de la planificación general del proyecto “HorseBooking”. En esta se va a profundizar en la **estimación de finalización** de las distintas fases y se va a organizar el tiempo y las tareas con el apartado “Projects” de Github.



Resumen de las etapas de las fases en Github.

- Después de una vista general de en lo que va a consistir la aplicación y cómo va a funcionar se inicia la planificación del diseño de la interfaz, del diseño de la base de datos, así como de la lógica con la que va a funcionar la aplicación.

- ¿Qué **pantallas** va a tener la aplicación? ¿Cuántas? El siguiente paso a realizar es visualizar un mapa mental de la cantidad de pantallas, elementos, tablas de base de datos, etc. que va a tener esta.
- La aplicación móvil tendrá un mínimo de 5 o 6 pantallas, entre ellas el registro e inicio de sesión. En cuanto a la aplicación de gestión constará, al igual que la de móvil, de un registro e inicio de sesión, además de las pantallas de inserción, y actualización de datos.

4.2. PLANIFICACIÓN DISEÑO

g

- En la planificación del diseño se realizará un borrador de las distintas ventanas de HorseBooking para ayudar a la elección final de los colores, los botones, formas de los elementos y finalmente el conjunto de todos los elementos para la pantalla final.

a) En primer lugar, se va a mostrar la elección de colores de HorseBooking.

Para la elección de colores se va a utilizar la herramienta “[coolors](#)”, una web que ofrece un generador de paletas de colores, donde se puede presionar la tecla espacio y que te genere cinco colores aleatorios que puedan combinarse entre sí. Ofrece la posibilidad de bloquear un color y generar otros que puedan combinar con este. Asimismo, en cada color ofrece su código hexadecimal, esto es, un sistema de notación numérica utilizado para representar los distintos colores en las páginas web o aplicaciones, como en el caso actual.



En HorseBooking se utiliza una combinación de colores fríos, en este caso verdes y azules. En concreto el fondo de la aplicación es el azul “#7BB8B1”, un azul claro combinado a su vez con blanco (#FFFFFF) y para los botones un color verde-azul (#19535F).

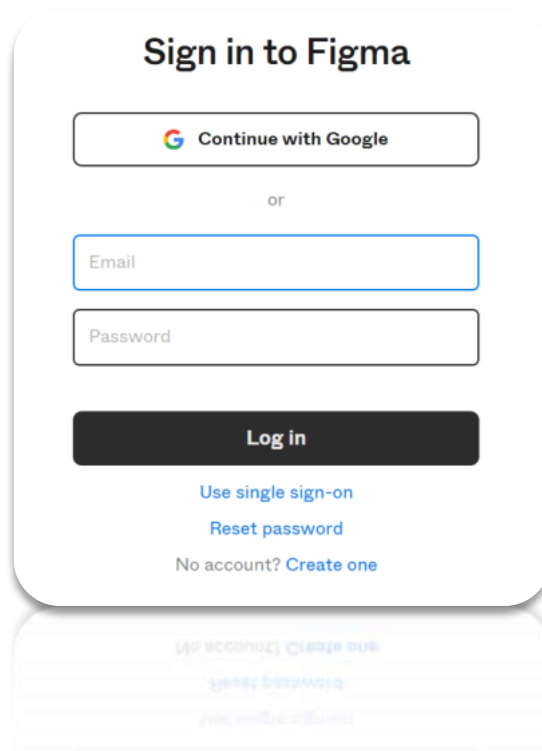
b) En el punto actual, se va a mostrar **el logo y el eslogan** diseñados para HorseBooking.

Ambos se han diseñado con otra web llamada [Looka](#), un diseñador de logos gratuito en el que introduces el nombre de tu aplicación o empresa junto con su eslogan y genera logos variados, con imágenes, distintas formas, etc. y ofrece la posibilidad de escoger uno y editar sus colores o formas e imágenes.



- c) El siguiente paso a la planificación inicial es el planteamiento del diseño de la interfaz y las pantallas. Para ello se utilizará una herramienta de diseño, en este caso **Figma**.

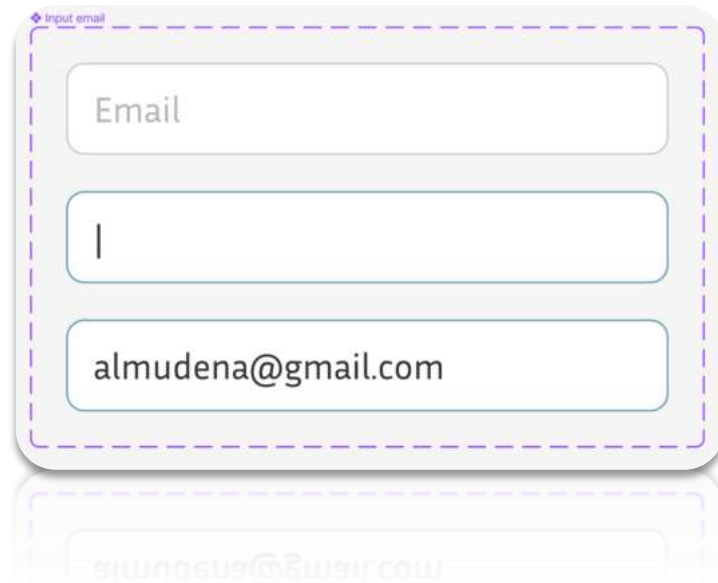
Figma es una herramienta que permite crear borradores para la interfaz de una aplicación. En ella se pueden probar los distintos colores, estilos de letra, formas de los elementos o distribución de pantallas, pudiendo ejecutar y probar la interfaz creada.



Pantalla de inicio de sesión y registro de Figma.

En este punto, se crean los llamados “**elementos**” en Figma. Estos son los distintos campos, botones, flechas, imágenes que van a tener las pantallas y los cuales pueden cambiar de estado. Por **ejemplo**: Un campo en el que rellenar el email o la contraseña que cambia de color, forma o el texto según el usuario le dé clic o escriba en él.

En la imagen siguiente se muestran los **tres estados** posibles para la entrada del **email de un usuario**. En este caso se muestra en primer lugar la entrada por defecto, el segundo elemento es el estado al darle clic al elemento para introducir el email y finalmente el estado al introducir completamente el email:



Estados del campo de un email en Figma.

- d) Estos elementos se introducen luego en los llamados “frames”. Esto son las ventanas o pantallas que simulan la forma de un móvil, Tablet u ordenador, según donde quieras desarrollar la aplicación.



Diseño de pantalla de registro en Figma.

4.3. DISEÑO

- Se realizarán dos diseños en la aplicación de HorseBooking: Diseño para **móvil** y diseño para **aplicación de escritorio**. Ambos diseños buscarán ser intuitivos para que el usuario no tenga problemas a la hora de darle uso y pueda utilizarlo de manera rápida y sencilla.

a) Diseño aplicación móvil

- Como se ha mencionado anteriormente, el borrador del diseño de la interfaz móvil se realiza en Figma. Aquí se decide la cantidad de **pantallas** que va a contener la aplicación.
- En HorseBooking, la aplicación móvil constará de más de cinco ventanas, como pueden ser el inicio de sesión, el registro del usuario, el olvido de contraseña, el perfil del usuario, las novedades de la hípica, sus precios, o la ventana de reservas.

A) Pantalla de registro: La pantalla de registro de HorseBooking pide al usuario distintos datos como el email, el nombre, los apellidos, el número de teléfono, la dirección, la fecha de nacimiento y la contraseña. Esto creará una nueva cuenta desde la que el usuario podrá realizar las reservas y otras acciones.



Diseño de pantalla de registro en Figma.

- B) Pantalla de inicio de sesión:** Esta pantalla permitirá al usuario acceder a una cuenta ya creada introduciendo el email y la contraseña y así realizar reservas en su perfil.

Desde esta ventana también tendrá la posibilidad de acceder al registro, si es que no tiene una cuenta ya creada con anterioridad, además de poder restablecer la contraseña, en caso de que haya olvidado la contraseña.



Diseño de pantalla de inicio de sesión en Figma.

- C) **Pantalla de olvido de contraseña:** El olvido de contraseña concede al usuario la oportunidad de restablecer la contraseña de la cuenta si no puede iniciar sesión por el olvido de la contraseña.

En este caso se envía un correo a la cuenta de email que haya introducido el usuario en el input, en el que aparecerá un link en el que podrá resetear su contraseña.



The image shows a mobile app screen for password reset. At the top, there is a white horse head logo above the text 'HORSEBOOKING' and 'TU PRÓXIMO PASO A UN CLIC'. Below this, the title 'RESTABLECE LA CONTRASEÑA' is displayed in large, bold, white capital letters. Under the title, a subtitle in smaller white text reads: 'Escribe el email y te enviaremos un correo electrónico para restablecer la contraseña'. There is a light blue rounded rectangular input field with the placeholder text 'Email'. Below the input field is a dark teal rounded rectangular button with the text 'Confirmar' in white. At the bottom of the screen, there is a link that says 'Volver' in white text. The entire screen has a teal background and is shown with a reflection effect at the bottom.

Diseño de pantalla de restablecimiento de contraseña en Android Studio.

- D) Pantalla de perfil de usuario:** Esta ventana muestra al usuario el perfil con sus datos personales y la foto de perfil que decida escoger.

En caso de ser una cuenta recién creada y no haber elegido foto de perfil, se establece por defecto como perfil el logo y el eslogan de la aplicación.



Diseño de pantalla de perfil de usuario en Android Studio.

- E) **Pantalla de novedades de la hípica:** Las novedades de la hípica se muestran en la primera pantalla después del inicio de sesión, con las distintas actualizaciones que pueda tener esta y que se pondrán desde la aplicación de escritorio.



Diseño de pantalla de novedades en Android Studio.

- F) **Pantalla de clases de la hípica:** Pantalla en la que se muestran todas las clases disponibles con sus distintos tipos (doma, salto...) , precios y duraciones.



Diseño de pantalla de clases en Android Studio.

- G) Pantalla de reservas:** Pantalla en la que el usuario puede ver todas las clases reservadas por él mismo. Da la opción de desinscribirse de estas.

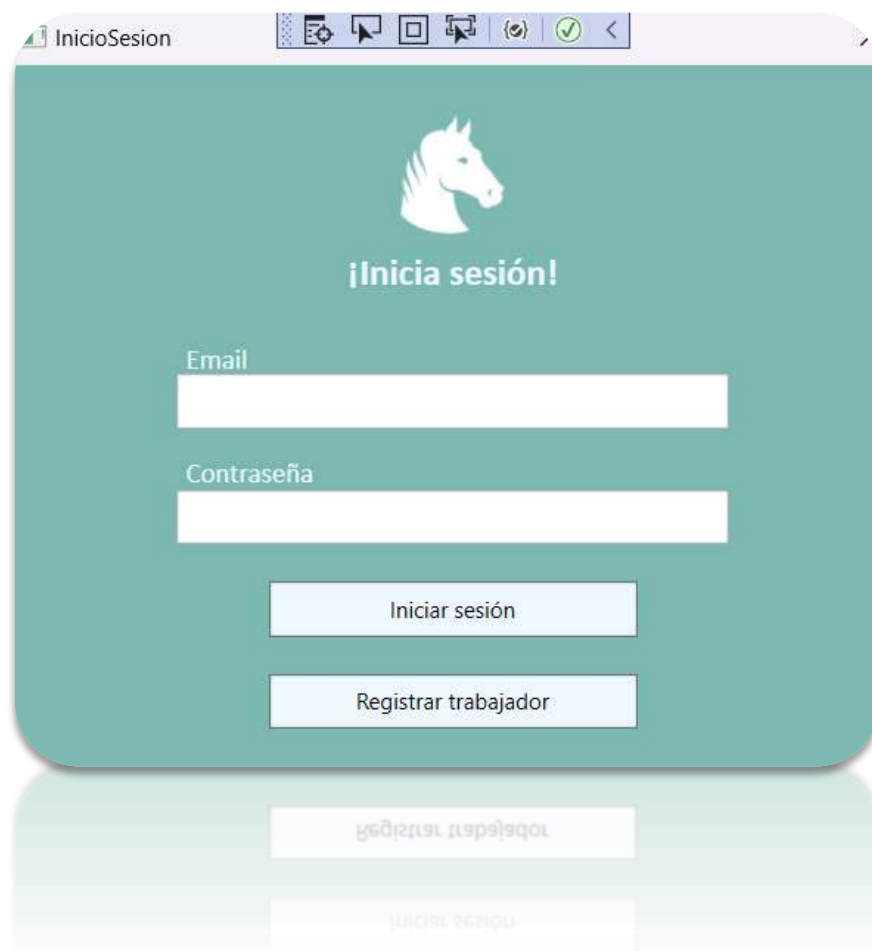


Diseño de pantalla de reservas en Android Studio

b) Diseño aplicación escritorio

- Al igual que en la aplicación móvil, el borrador de la aplicación escritorio se realiza en Figma. A continuación, se explican las distintas ventanas de la aplicación de escritorio:

A) Pantalla de inicio de sesión: Esta pantalla permitirá, al igual que en la aplicación móvil, iniciar sesión al usuario con una cuenta ya existente.



Pantalla de inicio de sesión en escritorio. Visual Studio 2019.

- B) Pantalla de registro:** El registro permitirá crear una nueva cuenta, en este caso, a un trabajador, y se asignará un rol.



Pantalla de registro de un trabajador en escritorio. Visual Studio 2019.

- C) Pantalla inicial:** Esta pantalla, una vez iniciada la sesión, da la bienvenida a la plataforma con un mensaje y el logo.



Pantalla de bienvenida en escritorio. Visual Studio 2019.

- D) Pantalla de resumen de clases:** El resumen de clases mostrará al trabajador, en este punto en concreto, profesor, donde podrá ver un esquema de las clases con sus alumnos.

Crear clase

Titulo de la clase

Descripción de la clase

Fecha de inicio
 Seleccione una fecha

Fecha final
 Seleccione una fecha

Hora

Precio

Cargar imagen

Profesor

Disciplina de la clase

Titulo	Descripción	Fecha de Inicio	Fecha Final	Hora	Minutos	Precio	Disciplina	Profesor

Pantalla de creación y resumen de clases en escritorio. Visual Studio 2019.

- E) **Pantalla de novedades:** En esta pantalla estará disponible la opción de crear novedades y actualizaciones sobre la escuela. El trabajador o jefe podrá introducir el título y la descripción de las novedades.

Novedades

Novedades de la hípica

Título de la novedad

Descripción de la novedad

Crear noticia

Ver noticias creadas

Cargar imagen

Cargar imagen

Crear noticia

Ver noticias creadas

Título	Descripción	Fecha de Inicio

Cargar imagen

Pantalla de creación de novedades en escritorio. Visual Studio 2019.

4.4. DESARROLLO

- La fase de desarrollo es la fase más delicada y densa de la evolución de una aplicación. En este punto se lleva a cabo toda la lógica de HorseBooking, esto incluye relacionar los elementos a la base de datos, navegación entre ventanas, notificaciones...

a) **Análisis de tecnologías:** En este primer punto del desarrollo realizaremos un análisis de las tecnologías usadas para el desarrollo de la aplicación, analizando también así sus ventajas y desventajas.

a. **Tecnologías de la aplicación móvil**

- **Kotlin:** Kotlin es un lenguaje de programación de código abierto utilizado principalmente para el desarrollo de aplicaciones **Android**. Fue creado en 2010 por **JetBrains**.
 - **Principales ventajas:** Interoperable con Java, desarrollo **multiplataforma**, **reducción del tiempo** de programación...
 - **Principales desventajas:** Soporte a medias y es menos conocido que Java, por lo que hay menos recursos disponibles.



b. **Tecnologías de la aplicación de escritorio**

- **C#:** C# es un lenguaje de programación multiparadigma desarrollado y estandarizado por la empresa Microsoft como parte de su plataforma .NET.
 - **Principales ventajas:** Seguridad, desarrollo rápido, soporte multiplataforma y comunidad sólida.
 - **Principales desventajas:** Limitado a plataformas de Microsoft, rendimiento inferior a lenguajes nativos, menos adecuado para sistemas de baja potencia.



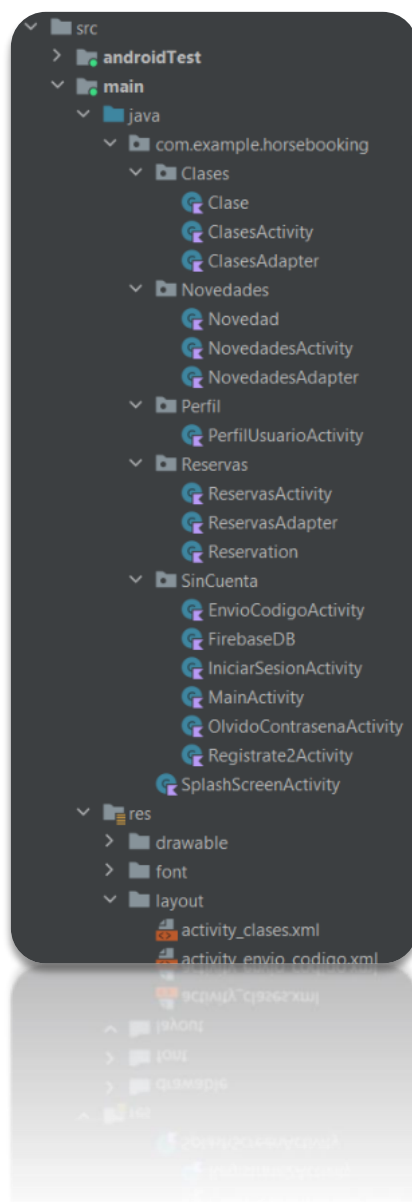
c. **Tecnologías para el servidor**

- **Firestore:** Firestore es una plataforma para el desarrollo de aplicaciones web y móviles con año de lanzamiento en 2011 y adquirida por Google en 2014.
- **Principales ventajas:** Integración con Google, almacenamiento en tiempo real, autenticación de usuarios sencilla...
- **Principales desventajas:** Coste, limitaciones en la gestión de datos o dificultades en las migraciones.



b) Aspectos esenciales de código – Aplicación móvil

- En este punto se va a realizar una revisión del código que puede ser más relevante para la aplicación, así como el código que más problemas ha causado.
- **Estructura de carpetas:** A continuación, se va a mostrar la estructura de directorios y archivos que se conforma en Android Studio, para la organización de la aplicación móvil.



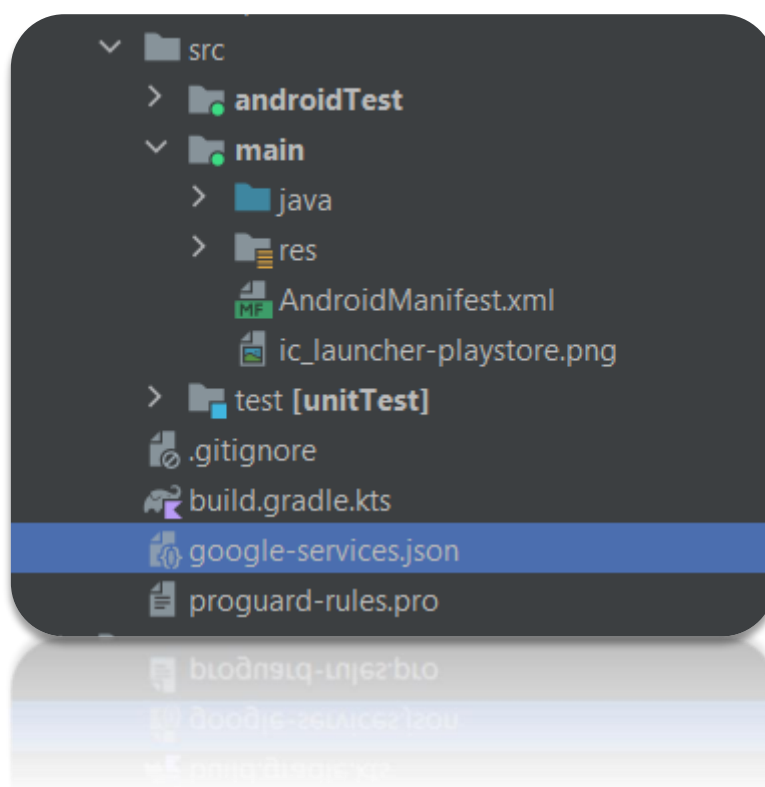
Estructura de directorios de Android Studio.

- En primer lugar, se puede ver la carpeta “src”, que será la que contenga todos los archivos y carpetas en las que se desarrolle el código para el proyecto.

Dentro de esta, a su vez, las carpetas más relevantes son la carpeta “main”, que contendrá los archivos con toda la lógica de la aplicación, donde también se realizará la conexión a la base de datos, así como la carpeta “res”, donde se centrarán principalmente todos los archivos que ayuden a conformar la interfaz.

- **Uso y conexión de la base de datos Firebase**

Seguidamente, se va a mostrar el código que ha sido requerido para conectar la aplicación al servidor (en este caso a Firebase). En primer lugar, se requiere del archivo “google-services.json”, situado dentro de la carpeta “src”, mencionada anteriormente.



Archivo google-services.json situado en la carpeta “src”.

- Este archivo lo proporciona Firebase durante la creación del proyecto en la plataforma. Permite conectar el proyecto de Android Studio creado al proyecto creado en Firebase.

- En cuanto al código desarrollado, para permitir al usuario iniciar sesión en Firebase se utiliza una función que proporciona este, en la que se pasa como parámetros el email y contraseñas solicitados al usuario.

```
private fun signIn(email: String, contraseña: String) {
    firebaseAuth.signInWithEmailAndPassword(email, contraseña)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                // Inicio de sesión exitoso
                val user = firebaseAuth.currentUser
                Toast.makeText(baseContext, text: "Autenticación exitosa", Toast.LENGTH_SHORT).show()
                // Crear un Intent para ir a MainActivity
                val intent = Intent(packageContext: this, NovedadesActivity::class.java)
                // Pasar el correo electrónico como extra al Intent
                intent.putExtra(name: "email", email)
                // Iniciar la actividad
                startActivity(intent)
                finish()
                // Aquí irías a la segunda actividad
                Log.d(tag: "IniciarSesion", msg: "Inicio de sesión exitoso")
            } else {
                // Error al iniciar sesión
                Toast.makeText(baseContext, text: "Error de email o contraseña", Toast.LENGTH_SHORT).show()
                Log.e(tag: "IniciarSesion", msg: "Error de autenticación: ${task.exception?.message}")
            }
        }
}
```

Función que comprueba credenciales del usuario en Android Studio.

- A su vez se ha creado una función que verifica cada vez que se inicia la aplicación en cuestión si el usuario ha introducido sus credenciales anteriormente en el dispositivo, de tal manera que si ya ha iniciado sesión antes permite acceder directamente a la cuenta.

```
fun comprobarSesion(firebaseAuth: FirebaseAuth) {
    val firebaseUser = FirebaseAuth.getInstance().currentUser

    if (firebaseUser == null && this !is IniciarSesionActivity) {
        // redirigir a la pantalla de inicio de sesión
    } else if (firebaseUser != null) {
        // Hay un usuario autenticado, redirigir a la pantalla principal
        val intent = Intent(packageContext: this, NovedadesActivity::class.java)
        startActivity(intent)
        finishAffinity() // Cierra todas las actividades anteriores
    }
}
```

- Asimismo, se ha creado una función para el **registro del usuario**, donde se asignan todos los campos solicitados al usuario en Firebase, implementando el correo de este como **identificador único**. Los datos recogidos se almacenan en “Realtime Database”, una base de datos no sql alojada en la nube que permite almacenar y sincronizar datos en tiempo real.

```

fun registrarUsuarioEnFirebase() {
    val email = intent.getStringExtra( name: "email")
    firebaseAuth.createUserWithEmailAndPassword(
        email.toString(),
        inputRegistroContraseña.text.toString()
    )
    .addOnCompleteListener(this) { task ->
        if (task.isSuccessful) {
            val user: FirebaseUser? = firebaseAuth.currentUser
            val emailUsuario = intent.getStringExtra( name: "email")
            val nombre = intent.getStringExtra( name: "nombre")
            val apellidos = intent.getStringExtra( name: "apellidos")
            val telefono = intent.getStringExtra( name: "telefono")

            val userId = emailUsuario?.replace( oldValue: ".", newValue: ",")
            userId?.let { it: String
                val userData = HashMap<String, Any>()
                userData["email"] = emailUsuario.toString()
                userData["nombre"] = nombre.toString()
                userData["apellidos"] = apellidos.toString()
                userData["fechaNacimiento"] = inputRegistroFechaNacimiento.text.toString()
                userData["direccion"] = inputRegistroDireccion.text.toString()
                userData["telefono"] = telefono.toString()
                databaseReference.child( pathString: "usuarios").child(userId).setValue(userData)
                .addOnSuccessListener { it: Void!
                    Toast.makeText(
                        context: this,
                        text: "Usuario registrado correctamente en Firebase",
                        Toast.LENGTH_SHORT
                    )
                }
            }
        }
    }
}

```

Función que permite el registro de un usuario en la aplicación. Firebase.

- Por último, haciendo referencia al **restablecimiento de contraseña**, se ha hecho uso de una función proporcionada por Firebase, que permite enviar un correo al email que haya proporcionado el usuario para poder introducir una nueva contraseña para la cuenta.

```

boton.setOnClickListener { it:View?
    val emailText = email.text.toString().trim()

    if (emailText.isEmpty()) {
        textoAdvertencia.visibility = View.VISIBLE
        Log.d( tag: "RestableceContraseña", msg: "No se puede enviar el correo porque el email está vacío")
    } else if (!isValidEmail(emailText)) {
        textoAdvertencia.visibility = View.VISIBLE
        Log.d( tag: "RestableceContraseña", msg: "Formato de correo electrónico no válido")
    } else {
        textoAdvertencia.visibility = View.GONE

        // Asegúrate de haber inicializado FirebaseAuth antes de usarlo
        auth.sendPasswordResetEmail(emailText).addOnSuccessListener { it:Void?
            Toast.makeText( context: this, text: "Por favor revisar el email", Toast.LENGTH_SHORT).show()
        }.addOnFailureListener { it:Exception
            Toast.makeText( context: this, it.toString(), Toast.LENGTH_SHORT).show()
        }
    }
}

```

Función que envía email al usuario para restablecer la contraseña. Firebase.

- **Gestión de las reservas mediante código**

Para la gestión de reservas se han creado distintas clases que permiten la actualización de los “recycler views”, de tal manera que aparezcan las clases introducidas a través de la aplicación de escritorio en Firebase.

A continuación, se muestra el código desarrollado para mostrar las clases existentes en la hípica, mostrando el nombre, la fecha en la que se imparten, el precio y la disciplina impartida.

```
@SuppressWarnings("SetTextI18n")
override fun onBindViewHolder(holder: ClaseViewHolder, position: Int) {
    val clase = clasesList[position]
    holder.tituloTextView.text = clase.titulo
    holder.tipoClaseTextView.text = "Disciplina: " + clase.tipo
    holder.fechaInicioTextView.text = "Fecha de inicio: " + clase.fecha_inicio
    holder.fechaFinTextView.text = "Fecha de fin: " + clase.fecha_fin
    holder.precioTextView.text = clase.precio.toString() + "€"

    if (clase.booked) {
        holder.btnInscribirse.isEnabled = false
        holder.btnInscribirse.text = "Inscrito"
    }
}
```

Función que conecta los datos a la vista del recycler view en Android Studio.

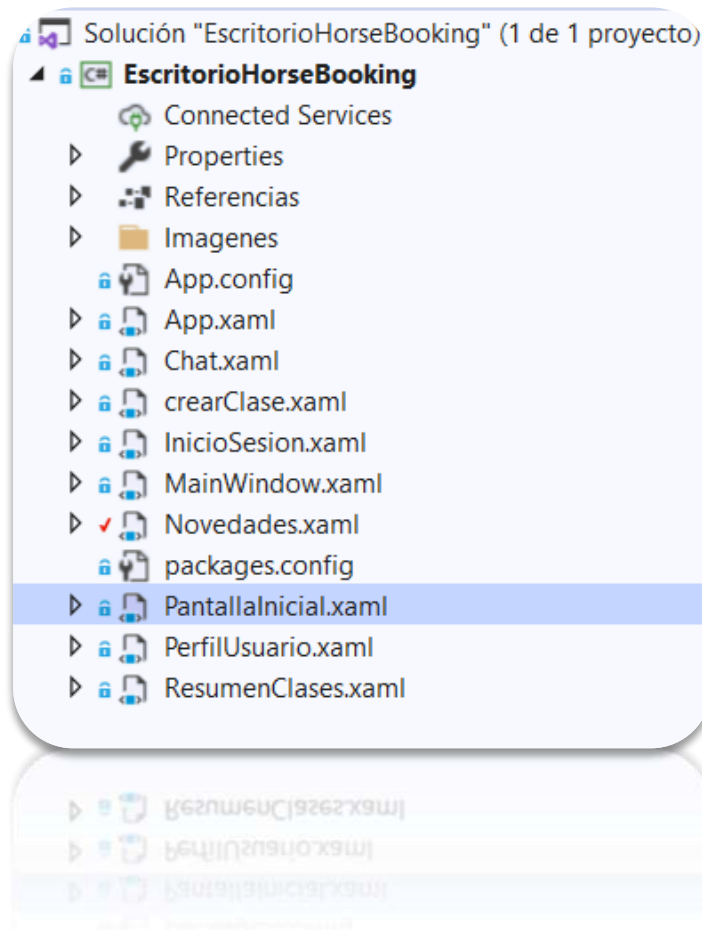
```
database.addValueEventListener(object : ValueEventListener {
    override fun onDataChange(snapshot: DataSnapshot) {
        clasesList.clear()
        val userId = FirebaseAuth.getInstance().currentUser?.uid
        for (claseSnapshot in snapshot.children) {
            val clase = claseSnapshot.getValue(Classe::class.java)?.apply { this.Clase() }
            codigo = claseSnapshot.key ?: ""
            // Asignar el estado de reserva basado en otra consulta a las reservas del usuario
            booked = snapshot.child("usuarios/$userId/reservas/$codigo/booked").getValue(Boolean::class.java) ?: false
        }
        if (clase != null) {
            clasesList.add(clase)
        }
        clasesAdapter.notifyDataSetChanged()
    }
})

override fun onCancelled(error: DatabaseError) {
    Toast.makeText(context, "Error al cargar las clases: ${error.message}", Toast.LENGTH_LONG).show()
}
```

Función que lee los datos alojados en Firebase.

c) Aspectos esenciales de código – Aplicación de escritorio

- Al igual que en el punto anterior, se va a realizar una revisión del código relevante desarrollado en este caso en la aplicación de escritorio.
- **Estructura de carpetas:** En este punto se va a mostrar la estructura de ficheros del que consta la aplicación de gestión en Visual Studio 2019:



Estructura de directorios de HorseBooking en Visual Studio.

- La estructura de directorios en visual studio está compuesta únicamente por las ventanas wpf que conforman el proyecto y la carpeta imágenes en la que se almacena la imagen del logo.

- **Conexión y uso de la base de datos**

Para realizar la conexión a Firebase a través de WPF es necesario introducir el mismo código en distintas ventanas, para poder realizar comprobaciones, actualizaciones o eliminaciones en el servidor. A continuación, se muestra el código para conectar con Firebase:

```
IFirebaseConfig config = new FirebaseConfig
{
    AuthSecret = "b4EKkNKwSvFpmAdcRdMRWPv90myyYgIirOv6QULs",
    BasePath = "https://horsebooking-54bbe-default-rtdb.europe-west1.firebaseio.com";
};
IFirebaseClient client;
```

Variables para preparar la conexión a Firebase.

```
public Novedades()
{
    InitializeComponent();

    client = new FireSharp.FirebaseClient(config);
    if (client == null)
    {
        MessageBox.Show("Error en la conexión a Firebase");
    }
}
```

Inicialización variable "client" para la conexión a Firebase. Visual studio 2019.

- Para realizar actualizaciones o inserciones en la base de datos se hace uso de un método, en el que se asigna un ID automático al objeto insertado en la base de datos y se le pasa como parámetro todos los campos requeridos para ese objeto en la tabla, así como el nombre de la tabla.

```
FirestoreResponse response = await client.PushAsync("novedades/", new { titulo = tituloNovedad,
descripcion = descripcion, fecha = DateTime.Now });
```

- Inserción o actualización de datos

- En el registro de trabajadores se insertan nuevos usuarios en esta tabla, con un email y una contraseña. El código a continuación permite la implementación de un usuario en la base de datos. Este código asigna como id el correo electrónico de usuario.

```
// Añadir usuario a la base de datos en tiempo real
FirebaseResponse response = await client.SetAsync("trabajadores/" + emailTexto.Replace('.', ' '),..
    new { Email = emailTexto, Password = contrasenaTexto });
```

- En la aplicación de escritorio de HorseBooking se realizan inserciones de datos de novedades y las clases que se van a impartir. Para ello se solicitan los datos necesarios en la interfaz y una vez introducidos se actualiza la base de datos. Seguidamente se muestra código implementado para esto:

```
private async void crearClase_Click(object sender, RoutedEventArgs e)
{
    string tituloClase = textBoxTituloClase.Text;
    string descripcion = textBoxDescripcionClase.Text;
    string fecha_fin = fechaFinDatePicker.SelectedDate?.ToString("yyyy-MM-dd") ?? "";
    string fecha_inicio = fechaInicioDatePicker.SelectedDate?.ToString("yyyy-MM-dd") ?? "";

    if (!(comboBoxHora.SelectedItem is ComboBoxItem horaItem) ||
        !int.TryParse(horaItem.Content.ToString(), out int hora))
    {
        MessageBox.Show("Selecciona una hora válida.");
        return;
    }

    if (!(comboBoxMinutos.SelectedItem is ComboBoxItem minutoItem) ||
        !int.TryParse(minutoItem.Content.ToString(), out int minuto))
    {
        MessageBox.Show("Selecciona un minuto válido.");
        return;
    }

    if (!int.TryParse(precioTextBox.Text, out int precio))
    {
        MessageBox.Show("Ingresa un precio válido.");
        return;
    }

    // ... (código para insertar en la base de datos) ...
}
```

Función crearClase (I) en Visual Studio 2019.


```

string tipo = (comboBoxDisciplina.SelectedItem as ComboBoxItem)?.Content.ToString();
string profesor = (comboBoxProfesor.SelectedItem as ComboBoxItem)?.Content.ToString();

if (string.IsNullOrEmpty(tipo) || string.IsNullOrEmpty(profesor))
{
    MessageBox.Show("Selecciona un tipo de disciplina y un profesor.");
    return;
}

try
{
    var clase = new
    {
        titulo = tituloClase,
        descripcion = descripcion,
        fecha_inicio,
        fecha_fin,
        hora,
        minuto,
        precio,
        tipo,
        profesor
    };

    FirebaseResponse response = await client.PushAsync("clases/", clase);

    // ... (código adicional) ...
}

```

Función crearClase (II) en Visual Studio 2019.

- En las imágenes mostradas anteriormente se recogen los datos introducidos y escogidos por el usuario para poder introducir una clase de equitación en la tabla de clases. Se hacen conversiones de tipos para poder subir la clase con sus campos de forma correcta.

- En el inicio de sesión se realiza la consulta de si los datos introducidos son ya existentes en la tabla trabajadores. La siguiente imagen muestra el código para ello:

```
private async void buttonIniciarSesion_Click(object sender, RoutedEventArgs e)
{
    string emailText = email.Text.ToString();
    string contraseñaText = contrasena.Password;

    if (string.IsNullOrEmpty(emailText) || string.IsNullOrEmpty(contraseñaText))
    {
        MessageBox.Show("Por favor, introduce tu email y contraseña.");
        return;
    }

    // Accede a la base de datos en tiempo real
    IFirebaseClient client = new FireSharp.FirebaseClient(config);
    try
    {
        FirebaseResponse response = await client.GetAsync("trabajadores/" + emailText.Replace('.', ''));
    }
}
```

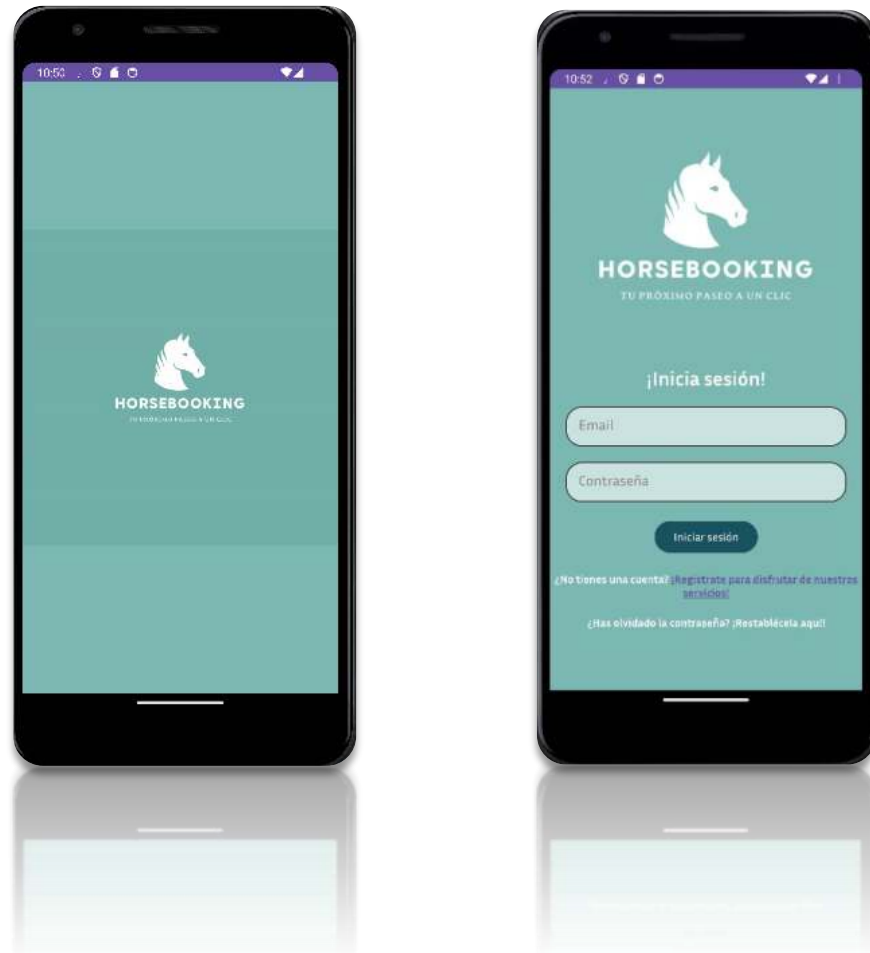
Comprobación de datos (I) para el inicio de sesión. Visual Studio 2019.

```
// Comprueba si el usuario existe en la base de datos
if (response.Body != "null")
{
    var trabajador = response.ResultAs<Dictionary<string, string>>();
    if (trabajador["Password"] == contraseñaText)
    {
        // Dispatch UI updates back to the UI thread
        Dispatcher.Invoke(() =>
        {
            MessageBox.Show("Inicio de sesión exitoso.");
            PantallaInicial pantallaInicial = new PantallaInicial();
            pantallaInicial.Show();
            this.Close();
        });
    }
    else
    {
        MessageBox.Show("La contraseña es incorrecta.");
    }
}
else
{
    MessageBox.Show("El usuario no existe.");
}
}
catch (Exception ex)
{
}
```

Comprobación de datos (II) para el inicio de sesión. Visual Studio 2019.

d) Demostración de ejecución – Aplicación móvil

- Al iniciar la aplicación se inicia la “SpashScreen”, que dura unos segundos, para después proceder a la ejecución de la pantalla de inicio de sesión.



Demo pantallas Splash Screen e inicio de sesión.



Inicio de sesión exitoso en Android Studio

- En caso de que el usuario ya tenga una cuenta creada, puede introducir el correo electrónico y la contraseña e iniciar sesión para disfrutar de los servicios ofertados por HorseBooking. En caso contrario el usuario podrá darle a registrarse y rellenar todos los campos para crearse una nueva cuenta.

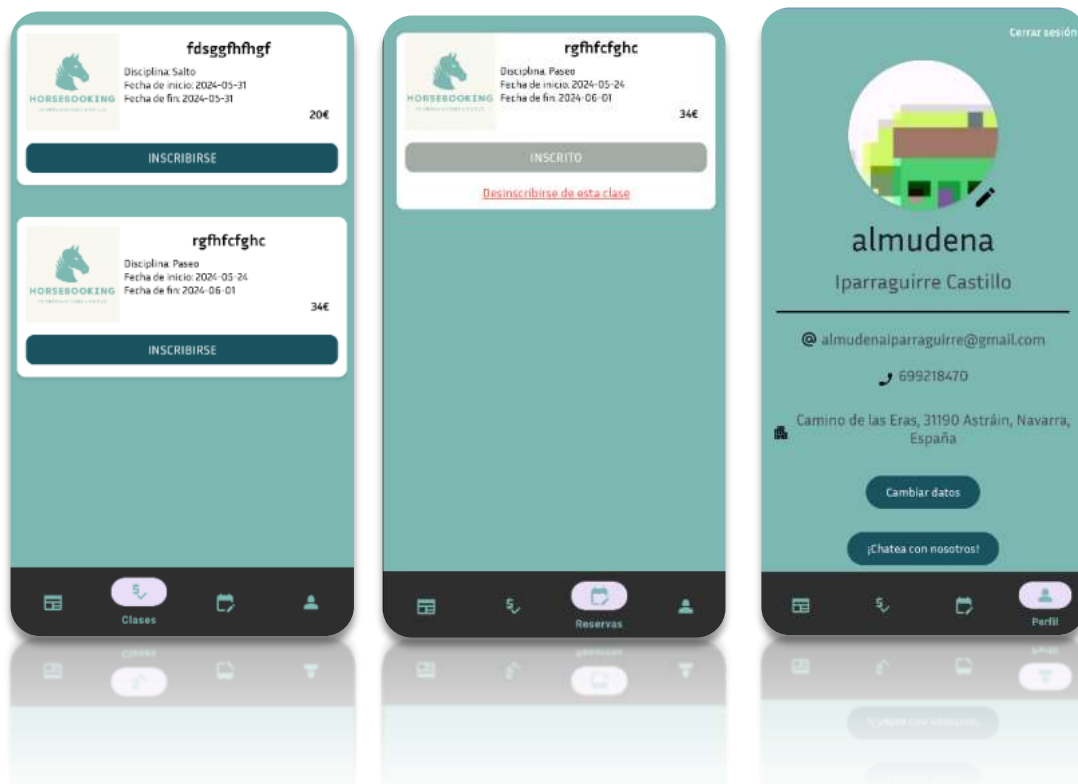
Si el usuario consta de una cuenta, pero ha olvidado su contraseña, podrá darle a resetear contraseña en la parte inferior de la pantalla, que le llevará a la pantalla de restablecimiento de contraseña.

- Una vez creada la cuenta o iniciada la sesión, la aplicación redirigirá al usuario a la pantalla de novedades, donde podrá ver todas las actualizaciones colgadas por la hípica desde la aplicación de gestión.



Demostración de pantalla de novedades en Android Studio.

- A través del menú navegación inferior, el usuario podrá navegar a la pantalla de clases, de reservas hechas por el usuario y el perfil de este último.



Demostración de ventanas de clases, reservas y perfil.

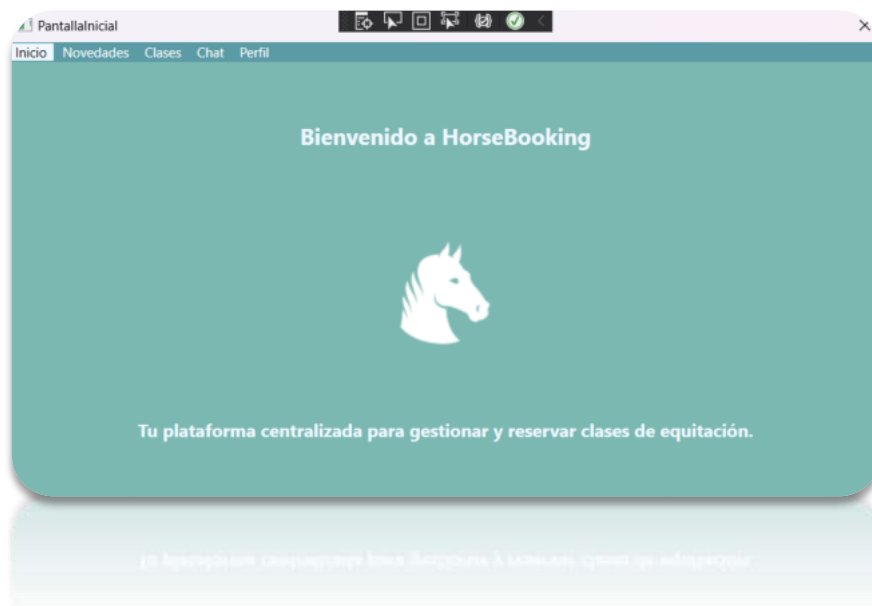
e) Demostración de ejecución – Aplicación escritorio

- Al arrancar la aplicación de gestión, la primera ventana será la de inicio de sesión, donde los trabajadores deberán acceder si ya tienen una cuenta creada, y en caso de no tenerla, podrán registrarse a través del botón inferior “Registrar trabajador”.



Demostración de inicio de sesión exitoso en Visual Studio.

- Una vez el trabajador ha accedido a su cuenta, aparecerá una pantalla de bienvenida de “HorseBooking”. De ahí podrá navegar a las ventanas de clases (donde podrá crear las clases y ver las ya creadas) así como a novedades, donde podrá crear nuevas noticias.



Demostración de pantalla de bienvenida en Visual Studio.



Demostración de novedades en Visual Studio.

Inicio Novedades Clases Chat Perfil

Crear clase

Título de la clase
Clase de equitación

Descripción de la clase
Clase avanzada para adultos

Fecha de inicio
19/05/2024

Fecha final
19/05/2024

Hora
12 10

Precio
12

Disciplina de la clase
Doma

Profesor
Julia

Crear clase

×

Clase creada exitosamente.

Aceptar

Demostración de creación de clases en Visual Studio.

4.5. PRUEBAS

- Las pruebas es la parte final del desarrollo de una aplicación. En este punto se llevan a cabos los llamados “tests”, que permiten verificar que todo el código está desarrollado de manera correcta y que todo va a la perfección.

4.5.1. Pruebas unitarias

- Las **pruebas unitarias** son aquellas que verifican el correcto funcionamiento de una función o un método.
 - a) **Prueba unitaria en olvido de contraseña:** En este test se comprueba si la verificación del email se realiza correctamente.

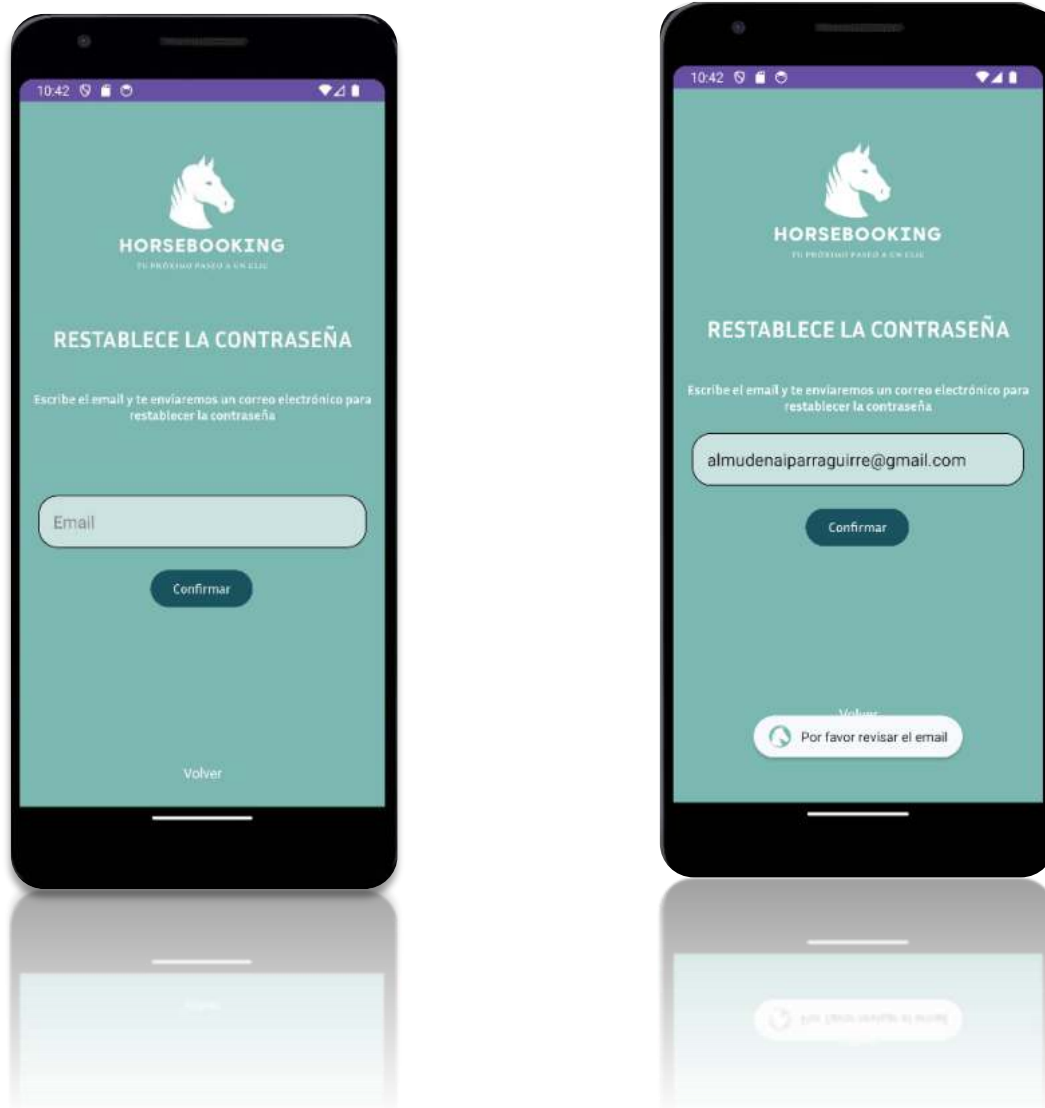


```
@Test
fun `verificacion email`() {
    val email = "almudenaiparraguirre@gmail.com"
    val result = OlvidoContrasenaActivity.isValidEmail(email)
    assertThat(result).isTrue()
}
```

Test que verifica el funcionamiento del método “isValidEmail”. Android Studio.

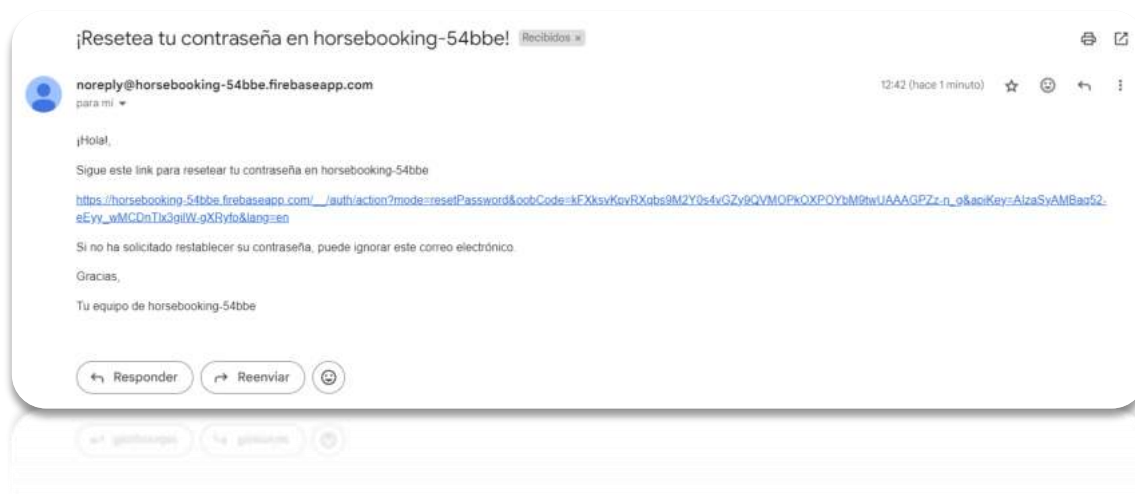
4.5.2. Pruebas de integración

- Las **pruebas de integración** buscan verificar que los distintos módulos en conjunto funcionan correctamente en la aplicación.
- a) **Prueba de funcionamiento de olvido de contraseña:** En este apartado se va a realizar un test sobre el funcionamiento del envío de un correo al usuario para restablecer su contraseña. Se introduce el correo de un usuario que no esté autenticado en la base de datos.



Prueba de funcionamiento de pantalla de olvido de contraseña. Android Studio.

- El correo de restablecimiento de contraseña llega correctamente al email que ha introducido el usuario en cuadro de texto.



Correo de restablecimiento de contraseña.

Reset your password

for almudenaiparraguirre@gmail.com

New password

SAVE

Página de reseteo de contraseña.

5. PRESUPUESTO PARA EL PROYECTO

- En este apartado se va a realizar una estimación del presupuesto de HorseBooking. Para ello se debe tener en cuenta la cantidad de personas contribuyentes del proyecto, la cantidad de tiempo dedicado, así como los recursos necesitados a lo largo del desarrollo de este.
- En referencia a la cantidad de contribuidores del proyecto, va a haber en total 1 persona encargada del total desarrollo de HorseBooking (Almudena Iparraguirre Castillo).
- La estimación de tiempo total dedicado es de unas 80 horas, de las cuales 50 estarían dedicadas a la aplicación móvil.
- **Salario del desarrollador:** El costo por hora del desarrollador será de 11 euros. Asumiendo este coste, el costo total en salario sería de $80 \times 11 = 880$ euros totales.
- En cuanto al precio de las **tecnologías**, todas las herramientas usadas durante el desarrollo de este proyecto han sido totalmente gratuitas, por lo que el coste total sería de **0 euros**.
- Haciendo referencia al hardware, se ha requerido de un portátil de 800 euros, así como de un ratón de 20 euros. Por último, una base refrigeradora para el portátil con un coste de 15 euros. Con todo esto el coste total del hardware sería de **835 euros**.
- Con estos datos se llega a la conclusión de que el proyecto habría supuesto un coste total de **1715 euros**.

6. LANZAMIENTO Y PUESTA EN MARCHA

- **Aplicación móvil:** Para la puesta en marcha de la aplicación móvil se seguirán los siguientes pasos:

- **Configuración de la aplicación para su lanzamiento:** Se realizan las pruebas pertinentes para la versión de lanzamiento en al menos un teléfono.

Se comprueba a su vez que los recursos necesarios para la aplicación, estén actualizados e incluidos.

En el caso de los servidores como Firebase, asegurarse y comprobar que está todo listo para su uso.

- **Lanzamiento de la aplicación a los usuarios:** En este caso se va a utilizar Google Play como plataforma de lanzamiento.

Esta es el mercado principal para las aplicaciones de Android y esta herramienta proporciona a los desarrolladores la oportunidad de analizar sus ventas y controlar a quién están distribuyendo sus aplicaciones.



- En el caso de publicar el producto en Google Play, al crear la cuenta de desarrollador en la plataforma, el coste total para ello será de 25\$.
- Una vez creada la cuenta de desarrollador y pagada la tasa de Google Play, se hace la preparación de materiales necesarios para la aplicación.

Los materiales necesarios son:

- **El APK:** Es el paquete que contiene el código y los recursos de la app.
- **Detalles:** Datos usados para mostrar la aplicación en la tienda y ayudar a los usuarios a encontrarla (título, descripción, icono).
- **Ajustes de distribución y monetización:** Países en los que está disponible, precio, anuncios...
- **Subir aplicación a Google Play Console.** Una vez hechos los puntos anteriores, toca **crear la nueva aplicación**. Luego se debe subir el archivo APK mencionado anteriormente y rellenar los detalles y ajustes.

- **Revisión y publicación.** Revisión de toda la información introducida para asegurarse de la ausencia de errores. Una vez listo, se debe enviar la aplicación realizada al equipo de Google Play. Si la revisión es aprobada por el equipo, será lanzada automáticamente a la tienda y estará disponible para todos los usuarios.

a) **Aplicación escritorio**

- En cuanto a la aplicación de escritorio, para su puesta en marcha tan solo se requiere de un portátil o un ordenador de sobremesa con la capacidad suficiente para lanzar la aplicación en cuestión.

Esta aplicación será directamente vendida a una escuela de equitación, a la que se proporcionará directamente el ejecutable para que realicen la instalación en los equipos necesarios.

7. CONCLUSIONES

- A modo de cierre, la aplicación “HorseBooking” es un proyecto completamente viable para sacarlo adelante.

Quiero remarcar que el proyecto tiene muchos aspectos mejorables, que, por motivos académicos, más concretamente de selectividad, y motivos de tiempo, es decir, de trabajo y ocio, no me han sido posibles de finalizar, o en el caso del chat, de iniciar.

Durante el desarrollo de esta aplicación me he encontrado con muchas dificultades, de las cuales muchas han podido ser resueltas y sacadas adelante, sin embargo, muchas otras no han sido posible por tema de tiempo y algunas ideas han tenido que ser descartadas.

Echando la vista atrás, este proyecto tenía unos objetivos comprometidos, de los cuales se han cumplido la mayoría. Tan solo un punto no ha sido posible de cumplir, que ha sido mostrar a los alumnos sus avisos en el perfil, así como el nivel de experiencia de este.

Como conclusión final, este ha sido un proyecto en el que he aprendido muchas cosas nuevas, entre ellas la importancia de la organización del tiempo, además que puede ser un proyecto viable para un futuro si decido sacar adelante mis propios proyectos personales.

8. BIBLIOGRAFÍA

<https://firebase.google.com/docs?hl=es>

<https://help.figma.com/hc/en-us>

<https://www.questionpro.com/blog/es/como-realizar-un-estudio-de-mercado/>

<https://colors.co/>

<https://looka.com/explore-ai>

<https://docs.github.com/en/desktop/overview/about-github-desktop>

<https://talently.tech/blog/que-es-kotlin/#:~:text=Kotlin%20es%20un%20lenguaje%20de,fundamentalmente%20para%20desarrollar%20aplicaciones%20Android.>

https://es.wikipedia.org/wiki/C_Sharp

<https://www.mytaskpanel.com/introduccion-a-csharp/>

<https://www.atlassian.com/es/continuous-delivery/software-testing/types-of-software-testing>

<https://developer.android.com/studio/publish/preparing>

<https://grupoelede.com/elementor-1235/>