Universitat Oberta de Catalunya

uoc.edu

Herramienta de ticketing para comunidades

Alberto Antonio Rubio Sánchez
Grado en ingeniería informática
Desarrollo multiplataforma de aplicaciones móviles

Carles Sànchez Rosa y Jordi Almirall López
Carles Garrigues Olivella

Junio de 2023

Alberto Antonio Rubio Sánchez
74824024X

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Herramienta de ticketing para comunidades*
Nombre del autor:	*Alberto Antonio Rubio Sánchez*
Nombre del consultor/a:	*Carles Sànchez Rosa y Jordi Almirall López*
Nombre del PRA:	*Carles Garrigues Olivella*
Fecha de entrega (mm/aaaa):	06/2023
Titulación:	*Grado en ingeniería informática*
Área del Trabajo Final:	*Desarrollo multiplataforma de aplicaciones móviles*
Idioma del trabajo:	*Castellano*
Palabras clave	*Ticketing, Comunidad, Frontend*

Resumen del trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

El objetivo de este trabajo fin de grado es el de proveer de una aplicación de ticketing sencilla, similar a la que se usa para la gestión de incidencias o tareas, orientada a comunidades, como puedan ser comunidades de vecinos.

A nivel técnico se trata de desarrollar una plataforma compatible con entornos de escritorio, Windows y MacOS, y móviles, Android y iOS, haciendo uso de las últimas herramientas disponibles para ello, como es el desarrollo de Progressive Web Apps (PWA).

El núcleo de la aplicación consistirá en ventanas codificadas en Javascript bajo Vue.js con herramientas Vite, y se desplegará mediante Firebase, haciendo uso de de sus utilidades de autenticación y gestión de información.

Alberto Antonio Rubio Sánchez
74824024X

Abstract (in English, 250 words or less):

The objective of this final degree project is to provide a simple ticketing application, similar to the one used for incident or task management, oriented towards communities, such as residential communities.

At a technical level, the goal is to develop a platform compatible with desktop environments, Windows and MacOS, and mobile devices, Android and iOS, making use of the latest available tools such as Progressive Web App (PWA) development.

The core of the application will consist of windows coded in Javascript using Vue.js with Vite tools, and will be deployed using Firebase, making use of its authentication and information management utilities.

Alberto Antonio Rubio Sánchez
74824024X

Índice

1. Introducción
    1.1 Contexto y justificación del Trabajo
    1.2 Objetivos del trabajo
    1.3 Enfoque y método seguido
    1.4 Planificación del Trabajo
    1.5 Breve sumario de productos obtenidos
    1.6 Breve descripción de los otros capítulos de la memoria
2. Despliegue básico de la aplicación
    2.1 Código de la aplicación
    2.2 Funcionamiento de la aplicación: “Hola mundo”
3. Diseño centrado en el usuario
    3.1 Usuarios y contexto de uso
    3.2 Diseño conceptual
    3.3 Prototipado
    3.4 Evaluación
4. Diseño técnico
    4.1 Definición de casos de uso
    4.2 Diseño de la arquitectura
5. Desarrollo
    5.1 Descripción de las herramientas utilizadas
    5.2 Análisis del estado del proyecto
6. Pruebas
    6.1 Descripción de cómo está previsto probar la aplicación
    6.2 Pruebas unitarias realizadas
7. Producto
    7.1 Código fuente
    7.2 Manual de usuario
    7.3 Instrucciones compilación y ejecución
8. Presentación
9. Conclusiones
    9.1 Descripción de las conclusiones
    9.2 Reflexión crítica sobre el logro de los objetivos
    9.3 Análisis crítico del seguimiento de la planificación y la metodología
    9.4 Líneas de trabajo futuro
10. Glosario
11. Bibliografía
12. Anexos

1. Introducción

1.1 Contexto y justificación del Trabajo

En el ámbito de las comunidades de vecinos, especialmente cuando son comunidades muy grandes, resulta muy difícil mantener un seguimiento de las incidencias y propuestas que se hacen dentro y fuera de las juntas respecto a la infraestructura y convivencia. Es muy habitual que un vecino realice una reclamación, y, aunque se tome nota en algún acta, esta quede aparcada indefinidamente y finalmente olvidada si no se trata de un tema crítico.

Entre una junta de vecinos y la siguiente puede pasar hasta un año, y un problema recurrente es el de no saber y tener que preguntar de manera repetida sobre el estado de cada asunto pendiente. Esta problemática da lugar a redundancia, esfuerzo innecesario por parte de la administración y frustración por parte de los vecinos.

Esta aplicación tratará de acercar la experiencia de una herramienta de ticketing a usuarios ajenos a un entorno profesional con gestión de incidencias. Para ello la herramienta a desarrollar deberá ser sencilla e intuitiva, además de abierta en el sentido funcional, evitando restringir o marcar su uso en exceso.

De esta manera, se permitirá la gestión de asuntos de entornos como comunidades de vecinos, por usuarios sin una experiencia previa en este tipo de aplicaciones, o conocimientos técnicos en general. Esto ofrecerá la posibilidad de contar con un registro de cada petición realizada a la administración, con una traza de su flujo de trabajo, estimación de la resolución y comentarios. Así, por ejemplo, un usuario tendrá mucho más fácil entrar en la aplicación y ver qué ha pasado con su demanda que llamar al presidente o administrador para saber del tema, ahorrando tiempo y esfuerzo a todos los implicados.

1.2 Objetivos del trabajo

El objetivo principal es el desarrollo de una aplicación web codificada en Javascript pero que sea igualmente instalable en un entorno Android o iOS como Progressive Web App (PWA).

La aplicación cubrirá unos mínimos como herramienta de gestión de tickets, permitiendo los cambios de estado y flujos de trabajo entre usuarios con diferentes roles (usuarios y administradores). De manera esquemática, los objetivos funcionales prioritarios a implementar para obtener un producto mínimo serían las siguientes:

- Control de usuarios simple
- Creación de comunidades
- Roles de administrador y usuario básico
- Creación de tickets

- Visualización de detalle de tickets
- Cambio de estado de los tickets por parte del usuario responsable
- Añadir comentario a un ticket
- Visualización de listado de tickets pendientes de gestionar según comunidad y rol del usuario
- Visualización de listado de tickets abiertos

Objetivos funcionales menos prioritarios respecto a características que podrían incluirse según se completen las funcionalidades mínimas:

- Autenticación de usuarios de Google
- Visualización de listado histórico de tickets resueltos
- Visualización de traza completa desde la apertura al cierre de un ticket
- Múltiples comunidades por usuario
- Múltiples administradores en una comunidad

Otras funcionalidades deseables que quedan, a priori, fuera de alcance:

- Personalización de perfil de usuario
- Roles y permisos avanzados, delegación de tareas
- Listados con filtros personalizados
- Herramientas de moderación en los comentarios
- Herramientas de gestión de usuarios para el administrador de la comunidad
- Búsqueda de tickets por palabras clave y filtros
- Detección automática de abuso de la aplicación por parte del usuario

De manera similar, algunos objetivos no funcionales a cubrir serían los siguientes:

- La aplicación debe ser lo más autoexplicativa posible en todo momento, sin necesidad de un manual exhaustivo o textos demasiado largos, al estar dirigida a un público general.
- La aplicación debe ser accesible desde dispositivos Android, Windows, macOS o iOS.

A medida que se cubren los objetivos técnicos se cubrirán también objetivos formativos, como es el aprendizaje tanto de las herramientas de codificación como de la metodología para el despliegue y puesta en marcha de la aplicación.

1.3 Enfoque y método seguido

Con la idea de cubrir los objetivos mencionados de la mejor manera posible, y al no haber tampoco otro punto de partida a considerar, se desarrollará un producto completamente nuevo, pero haciendo uso de todas las herramientas contemporáneas posibles para enfocar el esfuerzo a los componentes más relevantes a nivel formativo.

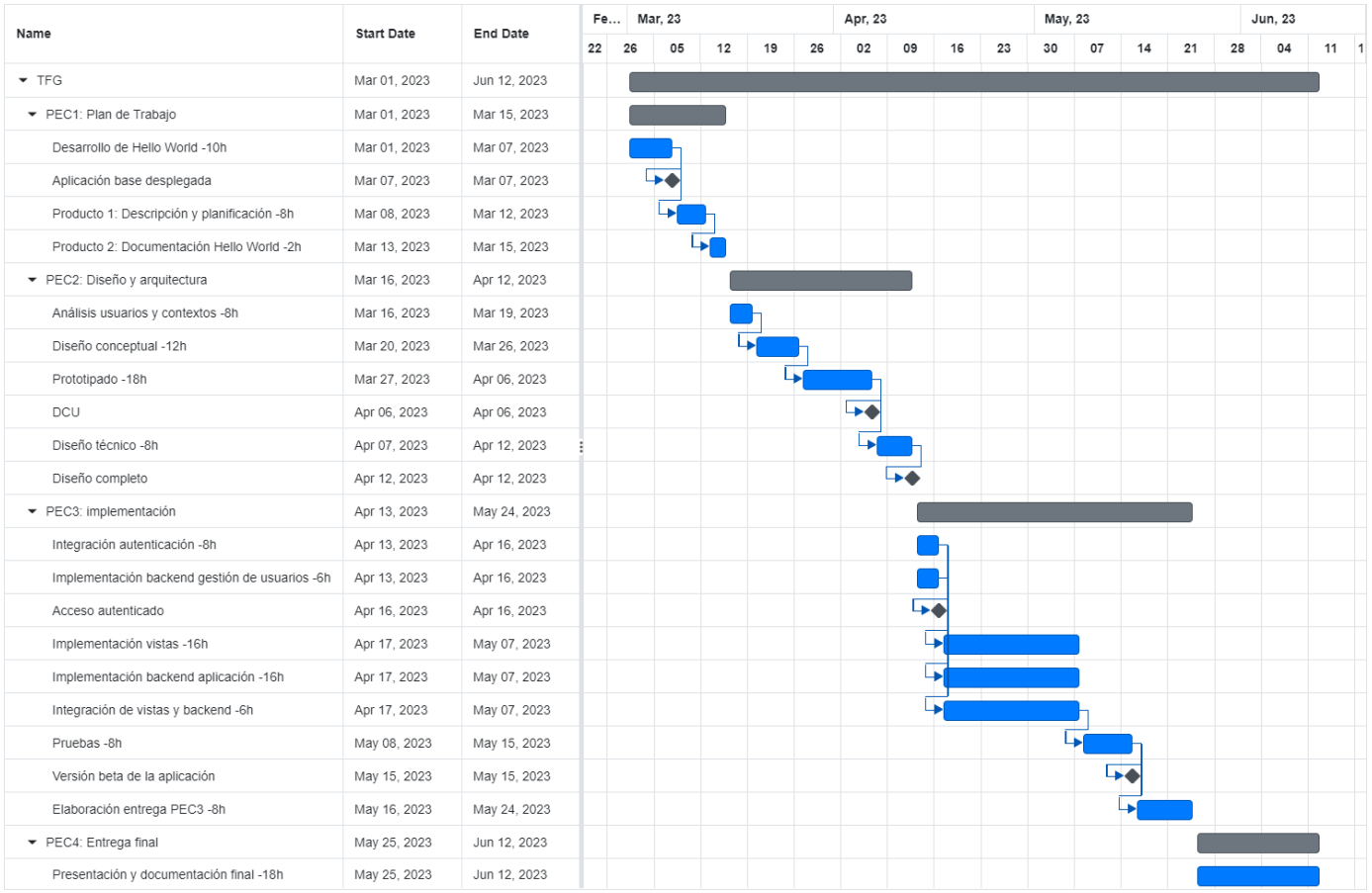
La metodología que se utilizará para el desarrollo será derivada de la metodología en cascada, tratando de compartimentar las fases del desarrollo secuencialmente, especialmente la planificación y análisis. Este planteamiento estaría alineado a la planificación de la asignatura mediante pruebas de evaluación continua y al hecho de que el desarrollo lo realiza una única persona.

Llegados a la fase de implementación, se hará el uso posible de técnicas de desarrollo e integración continuos para ir probando en el entorno productivo cada pequeño avance a medida que se va progresando en el desarrollo de la aplicación.

1.4 Planificación del Trabajo

Se cuenta para la realización de estas tareas con una capacidad de trabajo aproximada con un mínimo de 10-12 horas semanales (idealmente pueden ser: 0-2 de lunes a jueves, 4 los viernes y 6-8 los sábados y domingos, pero dependen de otra asignatura y tareas). Estas horas pueden ser ampliables a 20 o más según necesidad e imprevistos. A partir de esta estimación y las dependencias entre tareas se ha establecido la longitud en días de cada una y su estimación individual en horas (mostrado junto al nombre de la tarea en la imagen).

Se ha definido el siguiente listado de tareas a realizar junto con los hitos principales a alcanzar:



Name	Start Date	End Date
TFG	Mar 01, 2023	Jun 12, 2023
PEC1: Plan de Trabajo	Mar 01, 2023	Mar 15, 2023
Desarrollo de Hello World -10h	Mar 01, 2023	Mar 07, 2023
Aplicación base desplegada	Mar 07, 2023	Mar 07, 2023
Producto 1: Descripción y planificación -8h	Mar 08, 2023	Mar 12, 2023
Producto 2: Documentación Hello World -2h	Mar 13, 2023	Mar 15, 2023
PEC2: Diseño y arquitectura	Mar 16, 2023	Apr 12, 2023
Análisis usuarios y contextos -8h	Mar 16, 2023	Mar 19, 2023
Diseño conceptual -12h	Mar 20, 2023	Mar 26, 2023
Prototipado -18h	Mar 27, 2023	Apr 06, 2023
DCU	Apr 06, 2023	Apr 06, 2023
Diseño técnico -8h	Apr 07, 2023	Apr 12, 2023
Diseño completo	Apr 12, 2023	Apr 12, 2023
PEC3: implementación	Apr 13, 2023	May 24, 2023
Integración autenticación -8h	Apr 13, 2023	Apr 16, 2023
Implementación backend gestión de usuarios -6h	Apr 13, 2023	Apr 16, 2023
Acceso autenticado	Apr 16, 2023	Apr 16, 2023
Implementación vistas -16h	Apr 17, 2023	May 07, 2023
Implementación backend aplicación -16h	Apr 17, 2023	May 07, 2023
Integración de vistas y backend -6h	Apr 17, 2023	May 07, 2023
Pruebas -8h	May 08, 2023	May 15, 2023
Versión beta de la aplicación	May 15, 2023	May 15, 2023
Elaboración entrega PEC3 -8h	May 16, 2023	May 24, 2023
PEC4: Entrega final	May 25, 2023	Jun 12, 2023
Presentación y documentación final -18h	May 25, 2023	Jun 12, 2023

1.5 Breve sumario de productos obtenidos

- Documentación de diseño
- Prototipo de la aplicación
- Aplicación accesible en funcionamiento
- Documentación de desarrollo
- Documentación de pruebas
- Código fuente
- Manual de usuario e instrucciones de compilación y ejecución
- Presentación en vídeo de la aplicación

1.6 Breve descripción de los otros capítulos de la memoria

1.6.1 PEC 1

- *2. Despliegue básico de la aplicación*

  Código y capturas de una aplicación base “Hola mundo” desplegada.

1.6.2 PEC 2

- *3. Diseño centrado en el usuario*

  Características de los usuarios objetivo (perfiles de usuario) y contextos de uso, diseño conceptual, prototipado y evaluación del prototipo.

- *4. Diseño técnico*

  Definición de casos de uso y diseño de la arquitectura: diseño de base de datos, de entidades y clases y diagrama de arquitectura del sistema.

1.6.3 PEC 3

- *5. Desarrollo*

  Descripción de las herramientas utilizadas en el desarrollo y justificación de decisiones tomadas al respecto, enlace al código fuente de la aplicación (o su ubicación) y análisis del estado del proyecto.

- *6. Pruebas*

  Descripción de cómo se han realizado las pruebas de la aplicación y pruebas unitarias.

1.6.4 PEC 4

- *7. Producto*

  Enlace al código fuente o ubicación del fichero ZIP con el código de la aplicación, manual de usuario e instrucciones de compilación y ejecución.

- *8. Presentación*

  Enlace a la presentación en video de la aplicación.

1.6.5 Capítulos transversales

- *9. Conclusiones*

  Descripción de las conclusiones, reflexión crítica del cumplimiento de los logros planteados inicialmente, análisis del seguimiento de la planificación y líneas de trabajo futuro.

- *10. Glosario*

  Definición de los términos y acrónimos más relevantes utilizados dentro de la Memoria.

- *11. Bibliografía*

  Lista numerada de las referencias bibliográficas utilizadas dentro de la memoria.

- *12. Anexos*

  Listado de apartados que son demasiado extensos para incluir dentro de la memoria y tienen un carácter autocontenido.

2. Despliegue básico de la aplicación

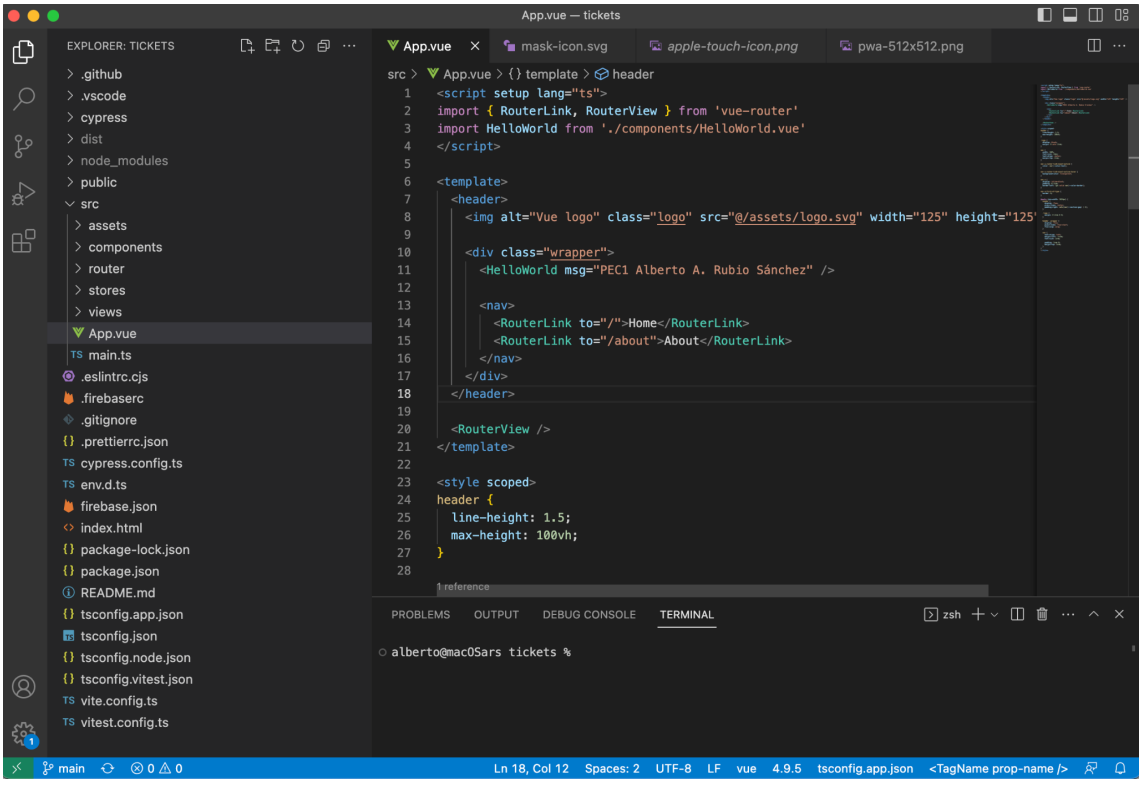
Para el desarrollo de la aplicación “Hola mundo” se ha aplicado la configuración necesaria para relacionar Firebase y GitHub de manera que el despliegue se haga de manera automática al ejecutar un push al repositorio desde local.

También se ha instalado el plugin y aplicado la configuración necesaria para pasar las restricciones que permiten a la aplicación ser una PWA (se han creado iconos provisionales de todos los tipos requeridos), lo que a su vez permite que la aplicación sea instalable en los distintos dispositivos.

2.1 Código de la aplicación

Para el desarrollo se está usando el IDE Visual Studio Code. Se ha instalado Node.js y los paquetes requeridos para Vue.js, Vite y uso de Firebase.

Se ha modificado el mensaje proporcionado en la vista por defecto para que muestre “PEC1 Alberto A. Rubio Sánchez” en lugar de “Hello world”.

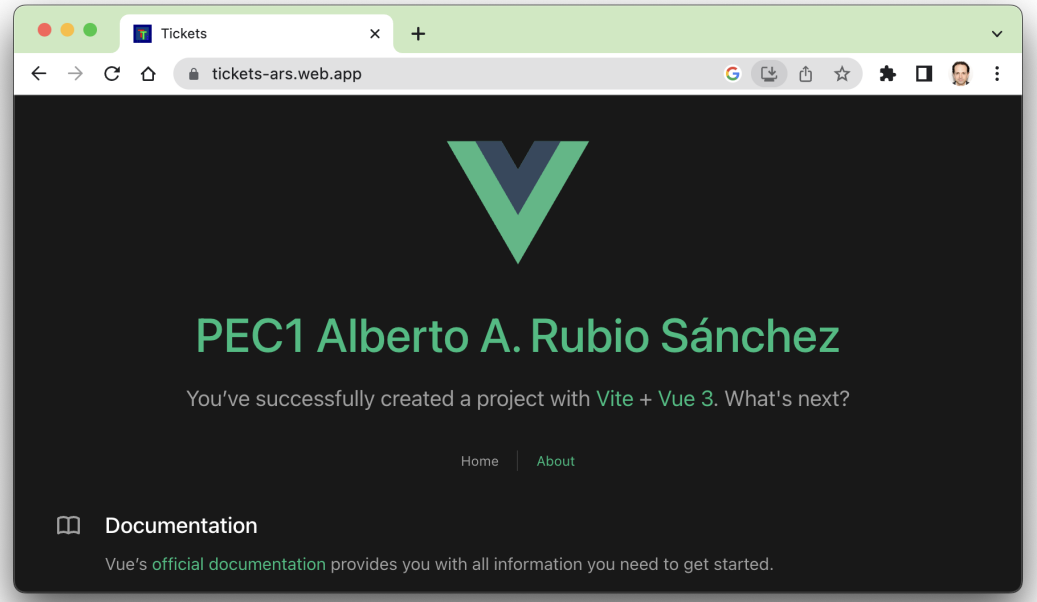


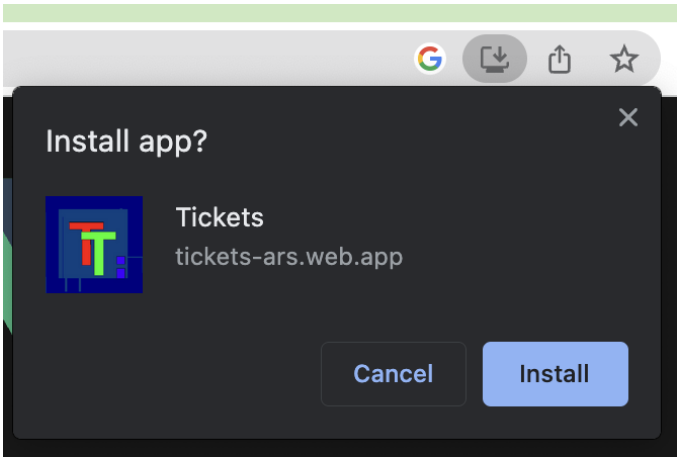
2.2 Funcionamiento de la aplicación: “Hola mundo”

La aplicación se encuentra actualmente desplegada mediante Firebase y accesible desde la URL https://tickets-ars.web.app/ .

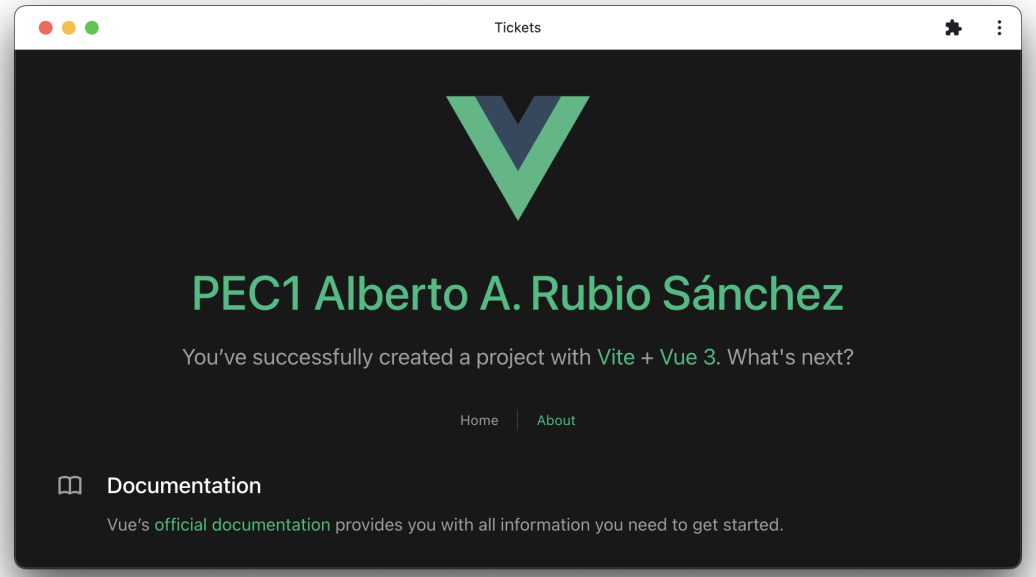
2.2.1 MacOS

El hecho de ser una PWA hace que la mayoría de navegadores ofrezcan una opción para instalar la aplicación. Puede verse esta opción en Google Chrome sobre MacOS.



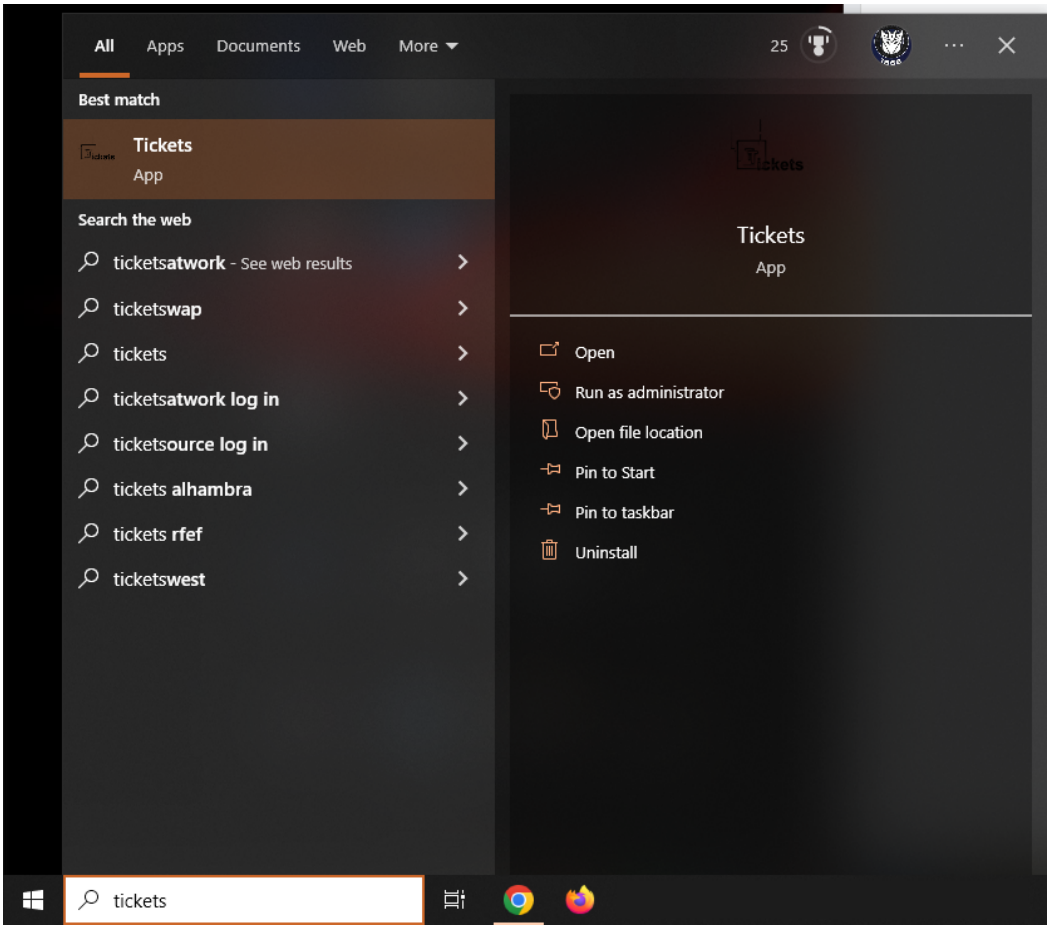


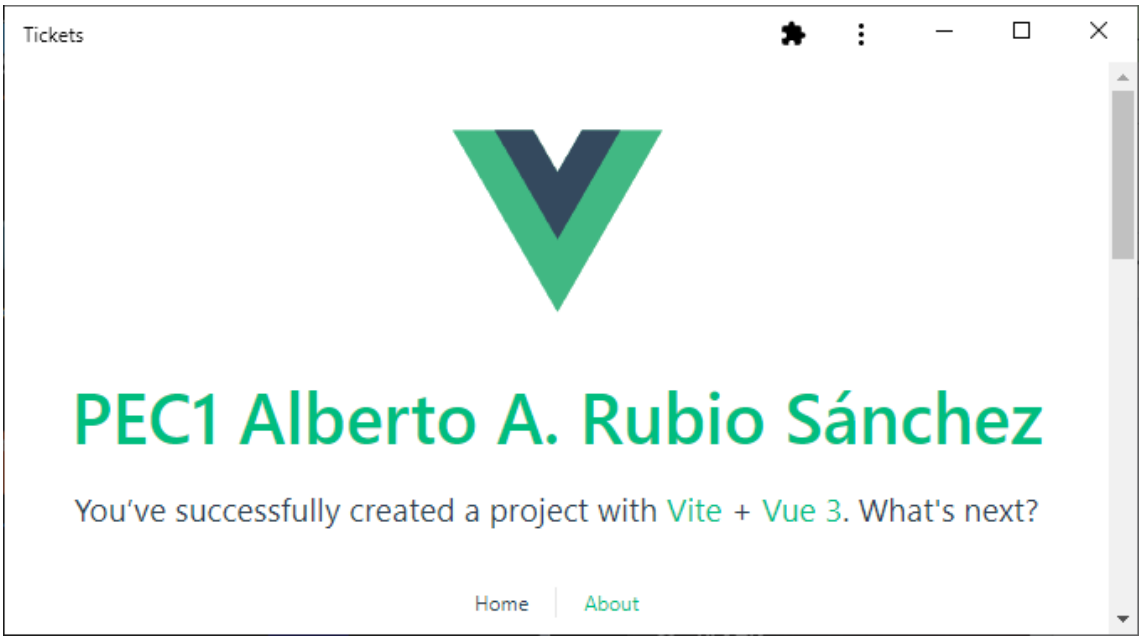
Una vez instalada la aplicación, y aunque esta no deja de ser una aplicación web, se muestra en una ventana sin barra de navegación ni pestañas.


Tickets
PEC1 Alberto A. Rubio Sánchez
You've successfully created a project with Vite + Vue 3. What's next?
Home
About
Documentation
Vue's official documentation provides you with all information you need to get started.

2.2.2 Windows

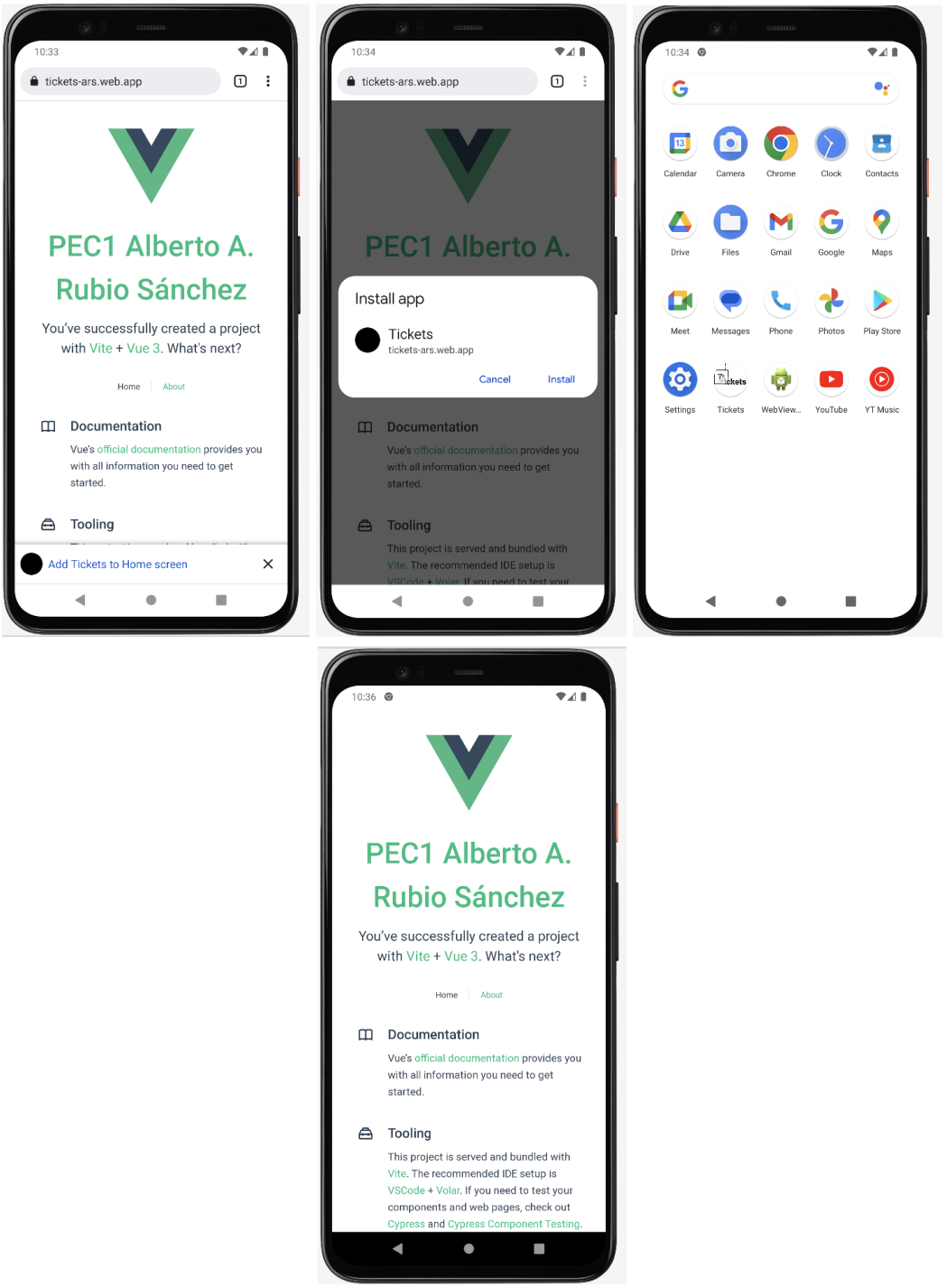
Instalada de una manera similar al entorno MacOS, puede comprobarse que la app también queda accesible como una aplicación nativa en Windows.





2.2.3 Android

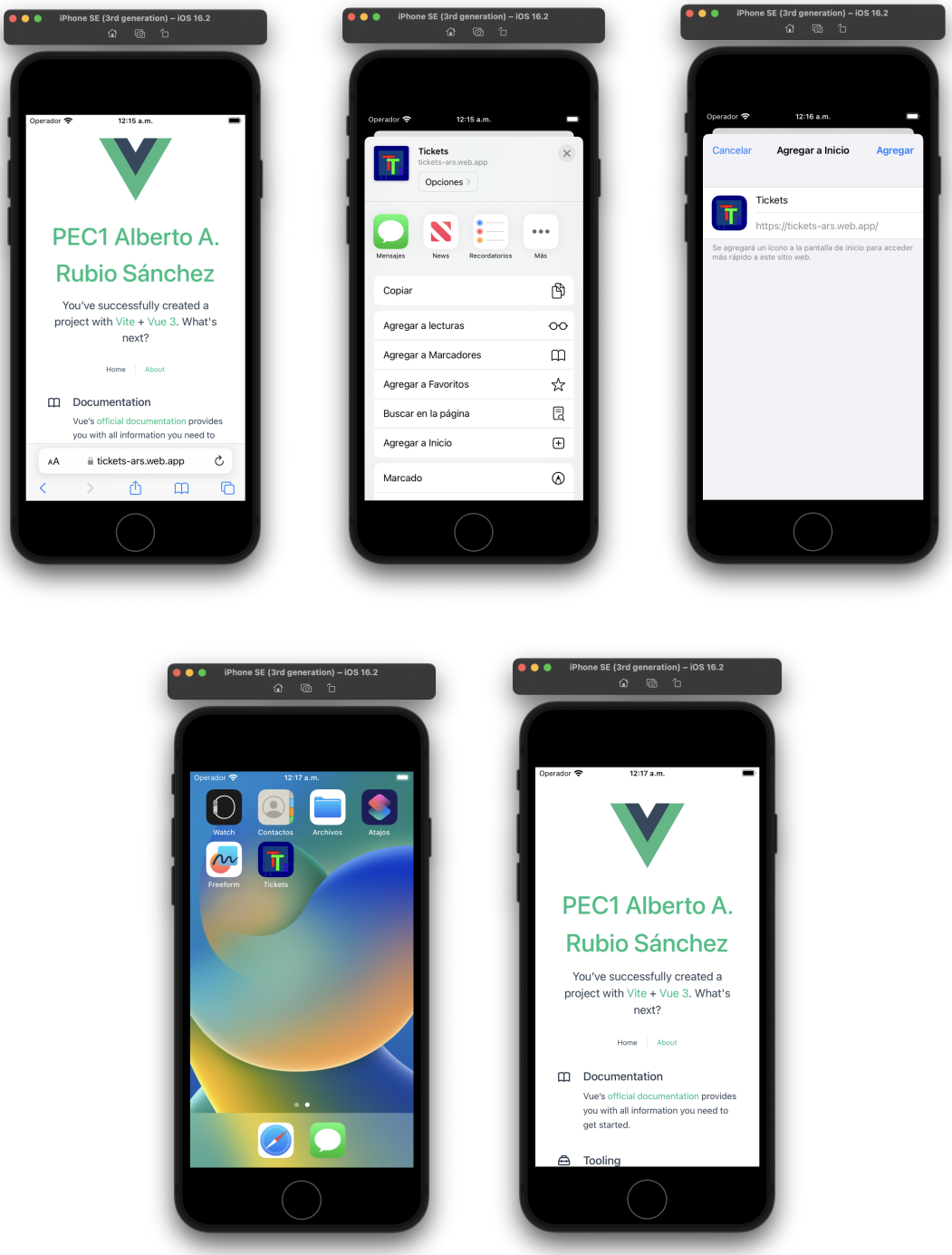
Instalación y prueba sobre emulador de Android 11 en Google Pixel 5.



En la primera captura puede verse la aplicación mostrada por el navegador, en la segunda la instalación, en la tercera el icono de la app en entre el resto de aplicaciones y en la cuarta la ejecución directa de la app.

2.2.4 iOS

Instalación y prueba sobre emulador de iOS 16.2 en iPhone SE (3ª generación)



En la primera captura puede verse la aplicación mostrada por el navegador, en la segunda y tercera el proceso de instalación, en la cuarta el icono de la app en entre el resto de aplicaciones y en la quinta la ejecución directa de la app.

3. Diseño centrado en el usuario

3.1 Usuarios y contexto de uso

3.1.1 Métodos de indagación: justificación y aplicación

3.1.1.1 Elección y justificación

Se utilizará el método de indagación “Entrevistas en profundidad”. Este método permite obtener conclusiones válidas a partir de entrevistas con un guión poco estructurado y abierto.

Este desarrollo tiene como punto de partida la experiencia personal en el ámbito al que apunta la propuesta. Es a partir de las vivencias y observación en el contexto de una comunidad de vecinos que surge la idea de la necesidad a cubrir, y de la experiencia profesional trabajando con distintas aplicaciones de gestión de tareas e incidencias que se plantea el cómo hacerlo.

Considerando lo anterior, una indagación en torno a entrevistas abiertas permite depurar y enfocar los planteamientos que ya están establecidos, así como discutir posibles funcionalidades no troncales y priorizarlas.

3.1.1.2 Planteamiento y desarrollo

Las entrevistas, tal y como plantea la teoría, tendrán una estructura mínima y permitirán que los usuarios potenciales puedan hablar en torno al planteamiento de la aplicación a desarrollar. Por lo tanto, el guión consistirá en explicar a cada usuario el planteamiento básico ya definido de la aplicación y, a partir de ahí, se les preguntará por su opinión general de la propuesta, qué les parece atractivo de ella y qué consideran que se puede añadir.

Este proceso se aplicará a 3 usuarios pertenecientes a diferentes demografías, y se realizará de manera presencial en conversaciones de unos 20-30 minutos.

Los perfiles de usuario se detallan en el apartado correspondiente más adelante, y se han buscado de manera que abarquen distintas edades, ámbitos profesionales, distintas experiencias con tecnologías móviles y que pertenezcan a distintos tipos de comunidades de vecinos. También, de manera menos documentable se ha valorado que haya caracteres más introvertidos y más extrovertidos.

3.1.1.3 Resultados y conclusiones

En las 3 conversaciones se ha podido validar el interés por una aplicación de este tipo, las facilidades que supondría para el usuario en su día a día y la necesidad de las distintas características ya planteadas de base.

A partir de las entrevistas también se han obtenido y evaluado distintas características que se detallan en el apartado correspondiente a los perfiles de usuario.

Alberto Antonio Rubio Sánchez
74824024X

Se extraen varios puntos comunes a los que prestar la mayor atención posible: la necesidad de que la aplicación aporte al usuario la sensación de que su demanda es relevante y no cae en saco roto, la necesidad de poder ver un estado de los asuntos de la comunidad de manera inmediata en cualquier momento y la necesidad de notificaciones.

Se da más detalle de las conclusiones a continuación en el desarrollo de los perfiles de usuario.

3.1.2 Perfiles de usuario

Perfil 1

Características del perfil

Hombre de 68 años, psicólogo jubilado.

Vive en un bloque de pisos antiguo (más de 30 años) en una comunidad grande (más de 40 viviendas).

Tiene cierta experiencia en el uso de aplicaciones móviles en un entorno Android, aunque limitada a aplicaciones básicas de comunicación, cámara de fotos y banca personal.

Contextos de uso

Este usuario potencial haría uso de la aplicación en una comunidad asentada, con una edad media de los vecinos bastante elevada y que, en general tienen pocas incidencias.

Análisis de tareas

Considerando que los objetivos de uso serían los de un vecino que necesitase puntualmente emitir algún aviso a la administración de la comunidad, las tareas más relevantes serían las de darse de alta como usuario, unirse a una comunidad y abrir y consultar tickets.

Características a considerar

Se establecen como características deseables en la aplicación la catalogación de tickets según tipología (cortes de suministro, morosos, limpieza, presupuestación, etc.) y prioridad. También aparece la posibilidad de establecer interesados/suscriptores a cada ticket, de manera que luego puedan aplicarse filtros o notificaciones dirigidas.

En resumen:

- Categorización de tickets
- Priorización de tickets
- Sistema de suscriptores

En principio todas las propuestas parecen asumibles, aunque el sistema de suscriptores puede tener cierta complejidad.

Perfil 2

Características

Hombre de 36 años, ingeniero software.

Vive en una comunidad de chalés adosados creada recientemente (menos de 5 años) de tamaño relativamente pequeño (menos de 20 viviendas).

Tiene una experiencia muy amplia tanto en el uso de todo tipo de aplicaciones entornos iOS y Android. También conocimientos de diseño y programación de aplicaciones.

Contextos de uso

Este usuario haría uso de la aplicación en una comunidad de vecinos de reciente creación, relativamente pequeña, con una media de edad joven y con cierta probabilidad tendría que administrarla como presidente de la comunidad. Aunque su comunidad no es particularmente conflictiva, el hecho de no llevar mucho tiempo en marcha puede implicar un flujo constante de demandas por parte de los vecinos.

Análisis de tareas

Dado el contexto, los objetivos de este perfil para el uso de la aplicación serían darse de alta, crear una comunidad, gestionar y cerrar tickets y añadir comentarios a un ticket.

Características a considerar

Se establece la necesidad de que la interfaz transmita de manera rápida un informe de situación, de manera que, si ha sucedido un evento a considerar, este aparezca como un asunto más y pueda ser revisado en cualquier momento. Se comenta también la posibilidad de elaborar un sistema que permita evitar redundancia en los tickets más allá de la buena voluntad de que quien vaya a abrir uno revise los anteriores.

En resumen:

- Necesidad de poder observar la situación de un vistazo
- Sistema para evitar duplicidad de tickets

Aunque la primera característica se puede asumir a la hora de diseñar una vista resumen, la segunda puede ser excesivamente compleja para un primer acercamiento técnico a la aplicación.

Perfil 3

Características

Mujer de 36 años, bailarina y administrativa.

Vive en un bloque de pisos antiguo (más de 30 años) de tamaño mediano (más de 20 viviendas, pero menos de 40).

Tiene amplia experiencia en el uso de aplicaciones, principalmente en entorno iOS.

Contextos de uso

Este perfil haría uso de la aplicación en una comunidad de vecinos instaurada hace mucho tiempo con una edad media en torno a los 50 años. Aunque la comunidad tenga cierta antigüedad, se prevé que los vecinos jóvenes puedan adoptar y dar mucho uso a la solución aportada por esta aplicación.

Análisis de tareas

Los objetivos de este usuario respecto a la aplicación consistirán en el alta como usuario, unirse a una comunidad, abrir tickets, añadir comentarios, y hacer seguimiento y consulta de ellos.

Características a considerar

Además de otras características ya mencionadas por los anteriores usuarios potenciales, se indica la posibilidad de que la aplicación envíe distintos tipos de notificaciones. Se comenta también la posibilidad de integrar otras herramientas para contactar directamente desde la aplicación mediante un chat de texto que permitiese la comunicación uno a uno.

En resumen:

- Sistema de notificaciones por cambios en tickets abiertos por el usuario
- Contacto directo por chat entre usuarios

Aunque la gestión de notificaciones parece viable (y posiblemente más usuarios la echarían de menos de no implementarla), la inclusión de otras herramientas complejas dentro de la aplicación no es factible en el desarrollo inicial de ésta.

3.2 Diseño conceptual

3.2.1 *Personas*

Se establecen tres personajes añadiendo datos ficticios a los perfiles de los usuarios potenciales del apartado “Usuarios y contexto de uso”.

Estos tres personajes representan a tres posibles usuarios de la aplicación a desarrollar y se utilizarán para elaborar los casos de uso.

Persona 1

A partir del *Perfil 1*

“William Riker”	**Comportamiento**
	- Es pragmático y organizado - Le gustan la series americanas y el cine de ciencia ficción - Vive en un bloque de pisos - Siente mucha curiosidad por cualquier avance tecnológico - Hace un uso moderado de aplicaciones en entorno Android - Los cambios drásticos e imprevistos le pueden incomodar en gran medida - Tiene ciertos recelos a realizar operaciones por Internet
Demografía	**Necesidades y objetivos**
- Hombre - 68 años - Psicólogo jubilado	- Disponer de tiempo para desarrollar sus aficiones - Evitar disrupciones en sus rutinas diarias - Mantenerse al día con el panorama tecnológico

Persona 2

A partir del *Perfil 2*

"Montgomery Scott"	**Comportamiento**
	- Toca la batería en una banda de éxito - Aficionado a los deportes de motor - Disfruta su trabajo - Tiene un número elevado de gatos en casa - Vive en una urbanización de chalés adosados - Trabaja por cuenta ajena - Hace un uso intensivo de aplicaciones móviles en entorno Android, y tiene conocimientos sobre su desarrollo - Está muy pendiente de nuevos desarrollos tecnológicos
Demografía	**Necesidades y objetivos**
- Hombre - 36 años - Ingeniero software	- Edificarse una casa por módulos en un terreno lejos de la ciudad - Conseguir la mejor cabina de simulación de conducción posible - Evitar trámites burocráticos innecesarios

Persona 3

A partir del *Perfil 3*

"Beverly Crusher"	**Comportamiento**
	- Le gusta viajar, hacer vida social, la fotografía y los videojuegos competitivos - Practica ciclismo de competición - Es muy activa en redes sociales - Trabaja como autónoma - Vive en un bloque de pisos con su perro Mollete - Hace uso intensivo de aplicaciones móviles en entorno iOS
Demografía	**Necesidades y objetivos**
- Mujer - 36 años - Bailarina	- Alcanzar una capacidad económica que le permita hacer varios viajes internacionales al año - Montar su propio negocio - Mudarse a una ciudad más grande

3.2.2 Escenarios de uso

Escenario 1

Perfil, contexto y objetivos

A partir de la Persona 1

William Riker (psicólogo jubilado de 68 años) vive con su pareja desde hace más de 20 años en el primer piso de un bloque en una zona histórica de la ciudad. Siente mucha curiosidad por cualquier novedad tecnológica, y está cómodo usando su móvil *Android* para comunicarse por *WhatsApp*, mirar su cuenta del banco o navegar por Internet; aunque también se siente muy cansado cuando las cosas no funcionan exactamente como él está acostumbrado y espera.

Esta mañana ha vuelto a notar un olor extraño en el aparcamiento, zona que al parecer no se está teniendo demasiado en cuenta últimamente en la limpieza del bloque. Parece que los anteriores avisos a la administración del bloque han pasado desapercibidos, y aún quedan meses para la siguiente junta de vecinos.

Descripción de la tarea, necesidades de información y de funcionalidades

Will piensa que la mejor manera de solucionar su problema es que quede constancia de su aviso y que, además, sus vecinos puedan también verlo y apoyarlo si lo ven conveniente.

Lo ideal sería contar con algo inmediato como puedan ser los grupos de *WhatsApp* pero estructurado de manera que pueda realizarse un seguimiento de la atención que preste la administración a la incidencia.

Sabe que en su comunidad de vecinos han empezado a usar una nueva aplicación que cubriría esta necesidad y está dispuesto a probarla.

Desarrollo de las tareas

Entrando en la página web asociada, *Will* podrá descargar la aplicación en su móvil. Una vez autenticado con su usuario de Google e introduciendo el identificador de su comunidad, verá la lista de *asuntos pendientes*, a los que podrá añadir su incidencia y seguir su desarrollo durante los próximos días.

Escenario 2

Perfil, contexto y objetivos

A partir de la Persona 2

Montgomery Scott (ingeniero software de 36 años) ha comprado un chalé adosado de nueva construcción en una urbanización a las afueras de la ciudad hace menos de un año, y ahora se enfrenta a los primeros pasos de una comunidad de vecinos recién establecida.

Su día a día en su trabajo y en su casa de puertas para adentro es bastante cómodo, pero la participación en la gestión de su comunidad supone un quebradero de cabeza detrás de otro. Con la intención de mejorar la situación, propia y de sus vecinos, ha decidido dar un paso adelante y presentarse a presidente de la comunidad en la última junta de vecinos.

Descripción de la tarea, necesidades de información y de funcionalidades

Con su nuevo cargo recién estrenado, *Scotty* desea un marco de trabajo que pueda compartir con sus vecinos, donde cualquiera de ellos pueda estar informado del estado de los temas de la comunidad de manera rápida y fiable.

Aunque no le importa echar una mano donde haga falta, le gustaría poder tratar las demandas de sus vecinos de una manera ordenada, secuencial y sobre todo, de manera asíncrona y por escrito y evitar así que sus vecinos le entretengan más tiempo del necesario cuando tiene otras cosas que hacer.

Desarrollo de las tareas

A través del navegador de su portátil, *Scotty* podrá abrir la aplicación, autenticarse y crear una nueva comunidad. Una vez creada, podrá compartir con sus vecinos el identificador de ésta y pedirles tratar por este medio cualquier petición.

Scotty podrá abrir tickets con las gestiones que vaya a realizar de las que quiera mantener informados a sus vecinos, y también

Escenario 3

Perfil, contexto y objetivos

A partir de la Persona 3

Beverly Crusher, (administrativa, que también ejerce como bailarina en eventos culturales, de 36 años) vive con su perro en un piso heredado en una zona residencial. Su vivienda pertenece a un bloque bastante antiguo y, aunque le gustaría mudarse, no está en disposición de hacerlo en este momento o en un futuro cercano.

Durante las últimas semanas parece que las grietas de la fachada han empezado a provocar humedades en algunas zonas comunes, y teme que este problema termine afectando además a su vivienda. Aunque ha notificado el problema varias veces, no tiene constancia de que se esté trabajando en él, o si el resto de sus vecinos son conscientes de la problemática.

Descripción de la tarea, necesidades de información y de funcionalidades

A *Beverly* le gustaría poder quedarse tranquila sabiendo que se está haciendo algo con el problema y contar, además, con el apoyo de sus vecinos antes de que este asunto vaya a más.

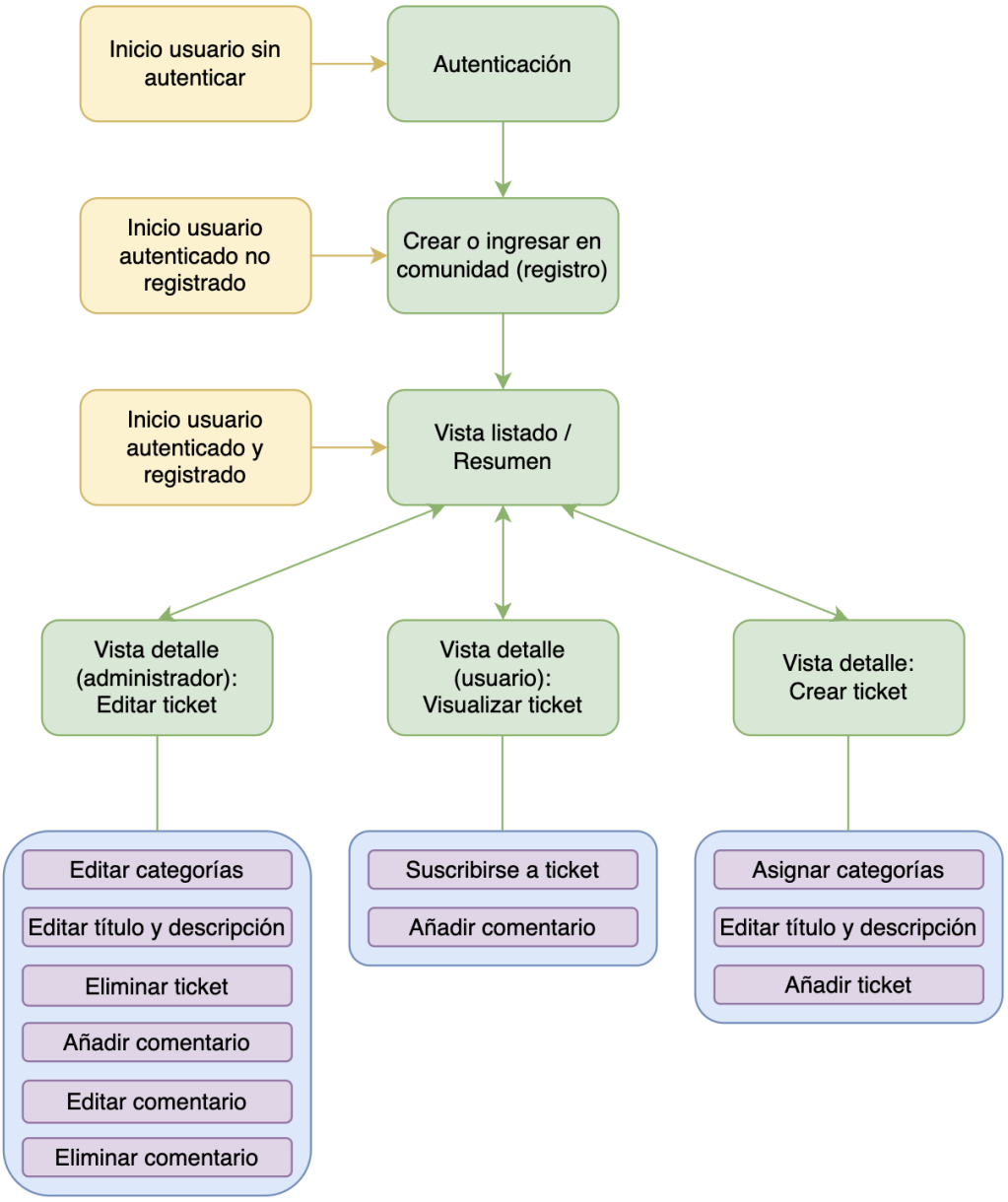
Dado el caso, lo que más le urge es poder recibir noticias, lo más inmediatamente posible sobre cómo se está afrontando la incidencia y saber que el resto de sus vecinos están al tanto.

Desarrollo de las tareas

Una vez constituida la comunidad en la aplicación, *Beverly* podrá instalarla en su móvil, autenticarse y acceder a su comunidad mediante el identificador correspondiente.

A partir de la lista de incidencias, podrá comprobar si existe ya una con el problema detectado, y asignarse como suscriptora para recibir notificaciones al respecto, o bien abrir una incidencia nueva describiendo la situación, a la que, como creadora, quedará suscrita por defecto.

3.2.3 Flujos de interacción



3.3 Prototipado

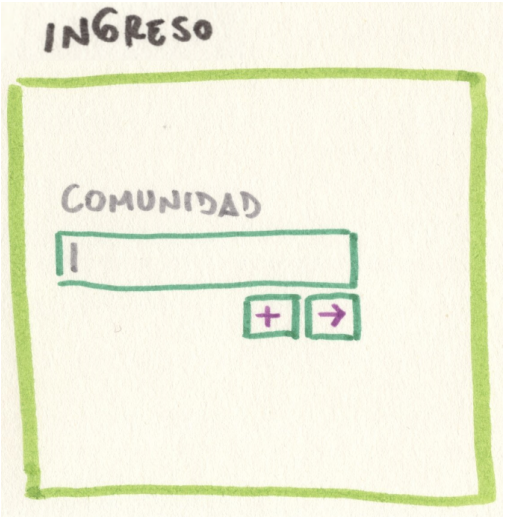
3.3.1 Bocetos

Autenticación

La autenticación debe ser lo más simple posible. La intención es que sólo se permita el acceso mediante una cuenta de Google, trabajando así de la manera más estandarizada posible a la hora de gestionar usuarios únicos válidos.

Esta vista mostrará el logotipo, un enlace para autenticarse a través de una cuenta de Google y una breve descripción del objetivo de la aplicación.

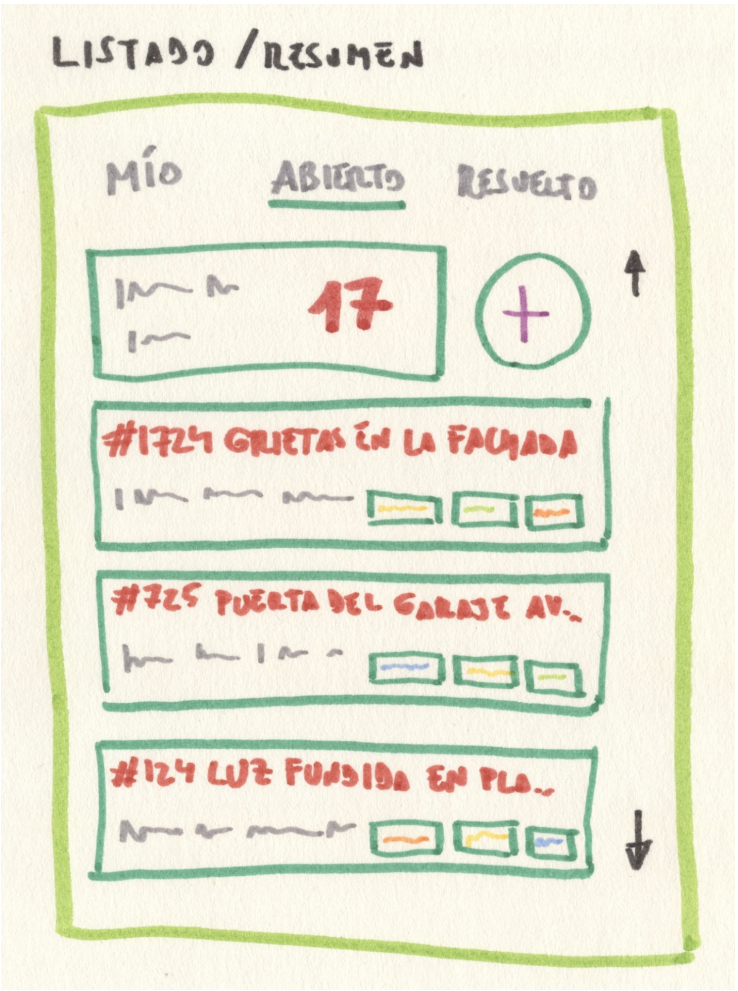
Ingreso



Una vez autenticado el usuario, este deberá asignarse a una comunidad. La misión de esta vista es que el usuario pueda ingresar en una comunidad ya creada o bien crear una nueva a partir de un identificador dado.

Alberto Antonio Rubio Sánchez
74824024X

Vista listado / resumen

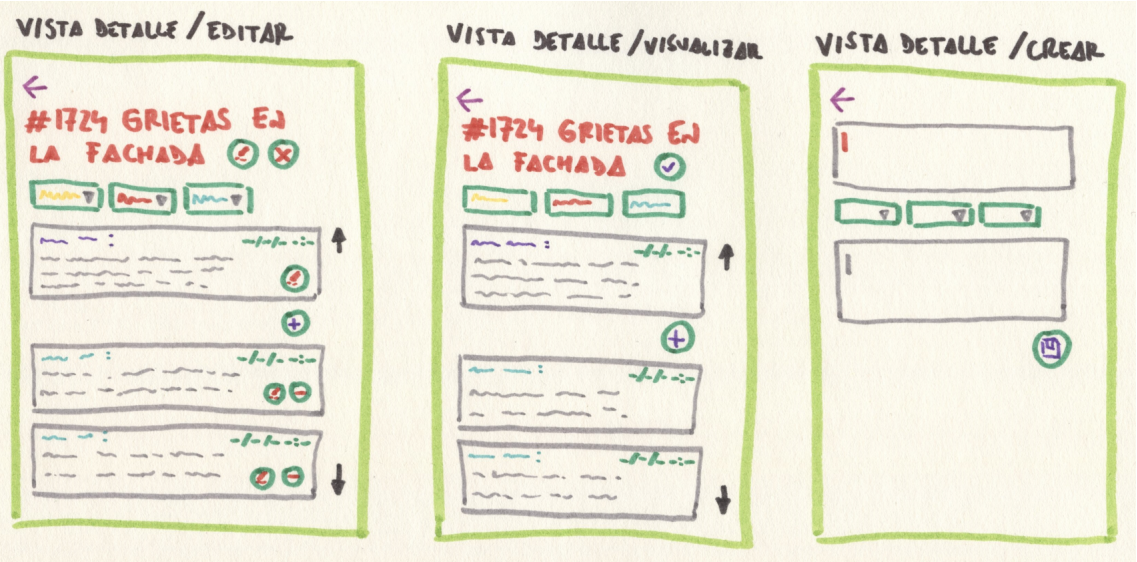


La vista resumen hará la función de *dashboard*, mostrando una lista de elementos según se seleccione entre: los elementos a los que está suscrito o ha creado el usuario, los elementos globales abiertos, o los elementos globales ya resueltos.

Habrá una zona de cabecera con un resumen con la cantidad de elementos mostrados en cada caso y un botón para generar un nuevo elemento. Debajo de esta se mostrará un listado de elementos, con un resumen de cada uno. Dicho resumen contendrá el secuencial identificador, el título, el inicio del texto de la descripción, estado, clase y prioridad.

La vista permitirá desplazamiento vertical, de manera que se pueda recorrer todo el listado.

Vista detalle



La vista detalle se mostrará de tres maneras diferentes: edición, visualización y creación. Las tres contarán con el espacio para el título, estado, clase, prioridad y descripción. Las vistas de edición y visualización contarán también con espacio de comentarios y posibilidad de añadirlos.

La vista detalle de edición sólo será accesible para el creador de la comunidad, y se mostrará cuando éste entre desde la vista resumen a cualquier elemento. Esta vista permitirá eliminar el ticket y modificar su título, resumen, estado, clase o prioridad. También permitirá editar o eliminar cualquiera de los comentarios asociados.

La vista detalle de visualización se mostrará al resto de usuarios cuando estos seleccionen un elemento de la vista resumen, y permitirá suscribirse al ticket (para recibir notificaciones de cambios de estado o nuevos comentarios) y añadir nuevos comentarios.

La vista detalle de creación permitirá generar un nuevo ticket, indicando título, estado, clase, prioridad y resumen. Contará con un botón de guardar para finalizar los cambios.

En los tres casos habrá un botón para volver a la vista resumen.

3.3.2 Prototipo de alta fidelidad

El prototipo se ha desarrollado generando las vistas necesarias en ficheros *SFC .vue* sin secciones de *script* (Javascript) o *style* (CSS), sólo la parte de *template* (HTML), haciendo uso de clases de *Bootstrap* y adaptando alguno de sus *snippets*.

Este desarrollo se aprovechará como esqueleto para la implementación de la versión final de la aplicación.

La versión web interactiva del prototipo queda disponible en la siguiente dirección y no se modificará durante el desarrollo de la versión final:

https://tickets-ars.web.app/prototipo

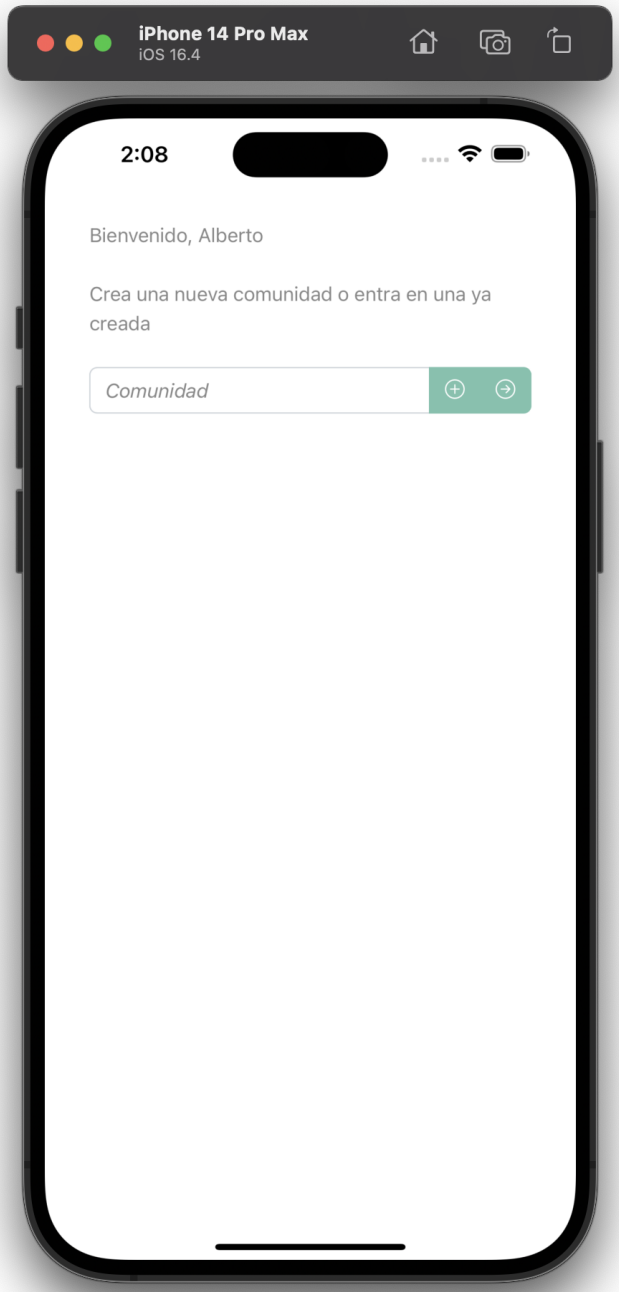
Las capturas se toman de la *PWA*, alojada en el momento de tomar las capturas en la dirección final que tendrá la aplicación.

https://tickets-ars.web.app/

Autenticación

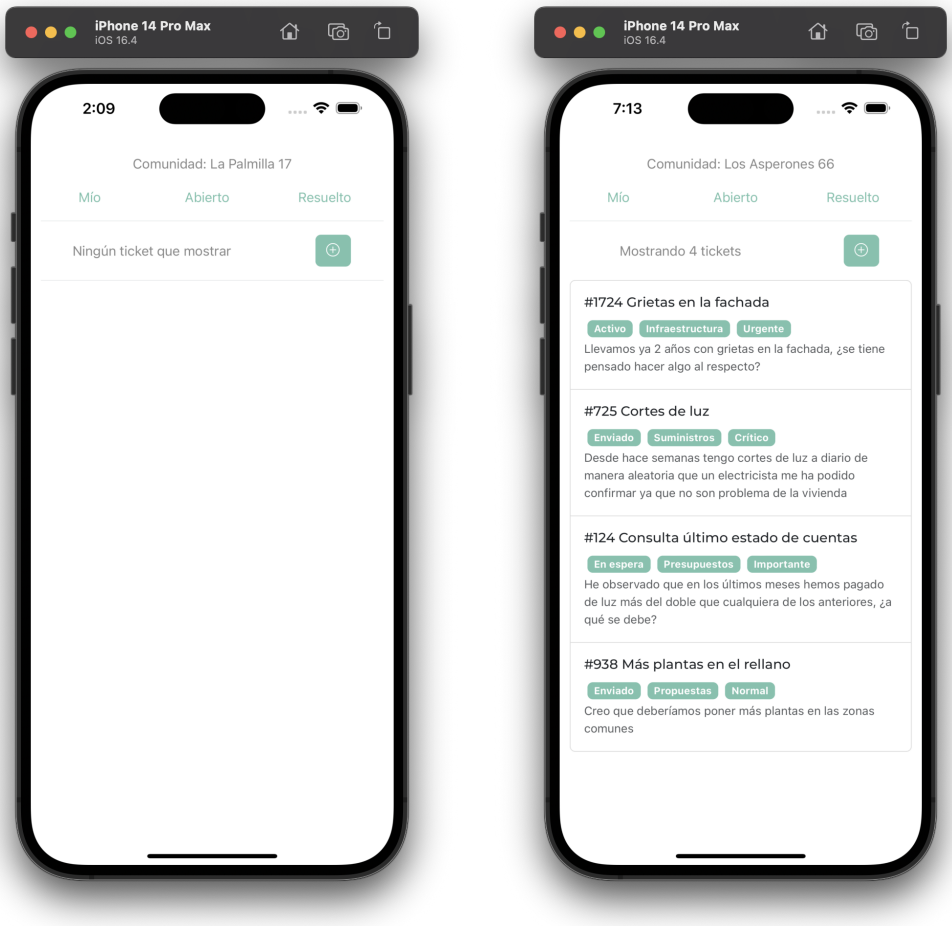
Esta vista cuenta con un solo botón, que en la aplicación final deberá trasladar al usuario al formulario estándar de autenticación de Google, y permitir pasar a la siguiente vista con un token de usuario y un nombre que almacenar en la base de datos de la aplicación.

Ingreso



En esta vista se solicitará al usuario que introduzca un identificador de comunidad, ofreciendo usarlo para crear una comunidad o unirse a una ya existente, con la idea de que se muestren mensajes de error si se trata de crear una comunidad con un identificador que ya exista en base de datos o unirse a una comunidad mediante un identificador que no exista.

Vista listado / resumen



Estas vistas son, respectivamente, las correspondientes a crear una comunidad nueva, donde se muestra una lista vacía, y a unirse a una comunidad existente, donde puede verse una lista de tickets ya creados.

Esta vista contará con una barra de navegación que permitirá acceder a listados de tickets con distintos filtros: tickets creados o suscritos por el usuario, tickets en tratamiento y tickets cerrados.

El listado de ítems contendrá de cada uno su secuencial, título, categorías (estado, clase y prioridad) y un extracto de su descripción.

Desde aquí el usuario podrá crear tickets nuevos, pulsando el botón correspondiente, o acceder a los ya existentes, pulsando sobre un elemento de la lista.

En el prototipo se puede acceder a la vista para crear ticket pulsando el botón, a la vista de detalle para usuario pulsando en el primer ítem o a la vista de detalle para administrador pulsando el segundo ítem.

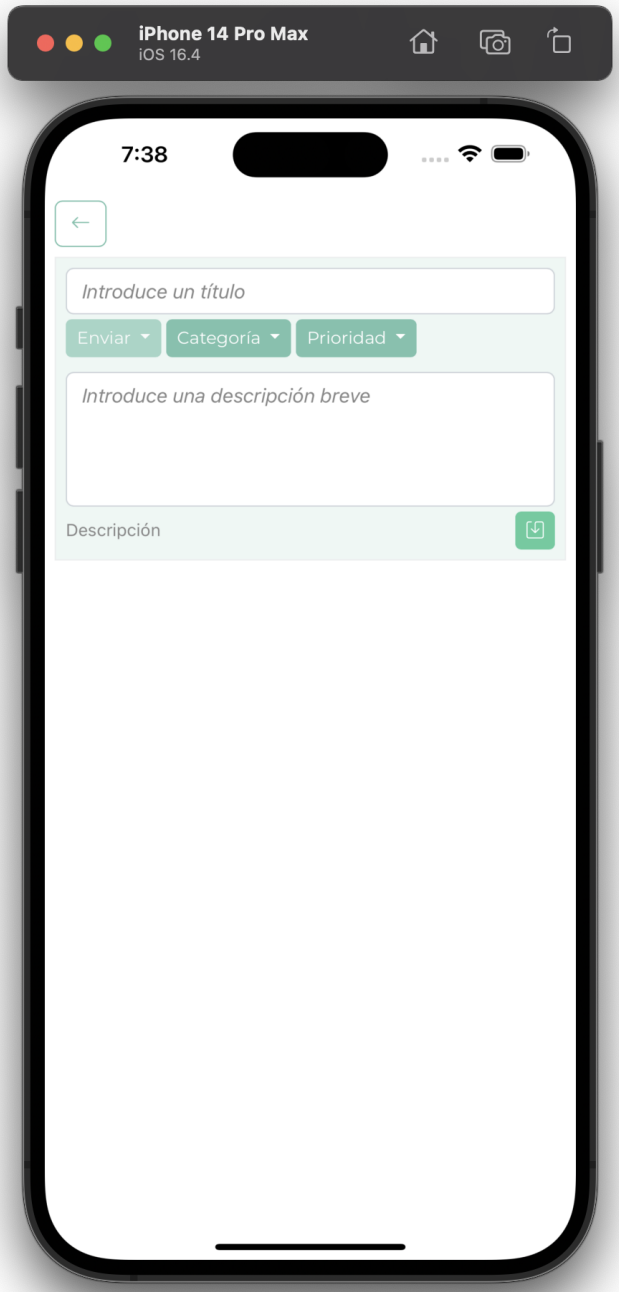
Vista detalle / administrador (editar)

Esta es la vista que verá el administrador seleccionar cualquier ítem del listado de tickets y le permitiría modificar o eliminar el ticket o cualquiera de sus comentarios, además de poder añadir también sus propios comentarios.

Vista detalle / usuario (visualizar)

Esta vista permite al usuario visualizar el contenido de un ticket, suscribirse (para recibir notificaciones cuando haya cambios) y añadir comentarios.

Vista detalle / crear



Esta vista permitirá al usuario crear un nuevo ticket indicando un título y una descripción.

3.4 Evaluación

El formato de la evaluación será presencial e individualizado y se capturará la pantalla mientras el usuario realiza las tareas, pidiéndole que verbalice cualquier reflexión que considere interesante y tomando todas las anotaciones posibles.

Para dar comienzo a la sesión se hará una breve presentación indicando brevemente en qué consistirá, explicando que habrá que realizar una serie de tareas relacionadas con una aplicación de gestión de incidencias para comunidades de vecinos.

Una vez presentada la aplicación y el formato de la sesión se pasará un cuestionario con preguntas para identificar la demografía del usuario y ubicarlo específicamente como usuario de la aplicación.

Seguidamente se indicarán las tareas a realizar, explicando que deben hacerse con la naturalidad esperada de un usuario normal y se deben evitar las preguntas, aunque se valoran muy positivamente las reflexiones en voz alta durante el proceso.

Después de las tareas, se pasará un segundo cuestionario a los usuarios con preguntas abiertas sobre las que podría iniciarse un diálogo, del que obtener todas las ideas posibles, y otras cerradas, del que obtener resultados cualitativos.

Tras finalizar el proceso con el usuario se realizará el análisis de resultados a partir de toda la información recopilada. Este análisis debe llevar a realizar una lista de propuestas de mejora en el prototipo y a una nueva iteración de evaluación. Este proceso se llevará a cabo cuantas veces sea necesario hasta obtener unos resultados que se consideren suficientemente buenos para proceder a empezar el desarrollo.

3.4.1 Recopilación de preguntas previas a las tareas

Preguntas para identificar la demografía general del usuario:

- Edad
- Género
- Estado civil
- Profesión
- Nivel educativo
- Nivel de experiencia con tecnologías móviles
- Necesidades especiales
- Intereses

Preguntas específicas sobre su situación relativa al uso de la aplicación:

- Sistema operativo de su dispositivo móvil
- Tipo de vivienda
- Número de vecinos en su comunidad
- Nivel socioeconómico general de su comunidad

- Expectativas sobre la aplicación (en cuanto a características y al tipo de uso que pueda darle)

3.4.2 Tareas para los usuarios

El prototipo sólo cuenta con funcionalidades de navegación, las cuales permiten validar su ergonomía, pero no otras funcionalidades. Igualmente puede validarse esta característica con tareas para los usuarios y evaluar si la navegación y uso de botones que requiere cada una resultan poco intuitivos en algún caso. Para ello deberá tomarse nota de los tiempos de respuesta de cada usuario para cada tarea y anotar cualquier reflexión que éste pueda realizar durante el proceso.

Tareas a realizar como administrador de la aplicación:

- Autenticación
- Creación de una nueva comunidad
- Creación de un nuevo ticket
    - Añadir título y descripción
    - Asignar categorías
    - Guardar ticket
    - Volver a la ventana de listado (alternativa a guardar el ticket)
- Acceso a la vista de edición de un ticket existente
    - Edición de título, categorías y descripción
    - Añadir comentario
    - Editar comentario existente
    - Eliminar comentario existente
    - Eliminar ticket
    - Volver a la ventana de listado (alternativa a eliminar el ticket)

Tareas a realizar como usuario (no administrador) de la aplicación:

- Autenticación
- Unirse a una comunidad existente
- Creación de un nuevo ticket
    - Añadir título y descripción
    - Asignar categorías
    - Guardar ticket
    - Volver a la ventana de listado (alternativa a guardar el ticket)
- Visualización de un ticket existente
    - Suscribirse
    - Añadir comentario

3.4.3 Preguntas sobre las tareas

Preguntas abiertas sobre las tareas realizadas:

- ¿Qué dificultades han encontrado a la hora de realizar las tareas?
- ¿Qué tareas consideras que podrían ser más sencillas? ¿Por qué?
- ¿Qué funcionalidades echas de menos en la aplicación?

- ¿Qué funcionalidades existentes piensas que pueden ser más útiles?
- ¿Qué funcionalidades existentes piensas que pueden ser menos útiles?

Preguntas cerradas sobre las tareas realizadas (valoración 1-10 de nada de acuerdo a completamente de acuerdo):

- ¿Consideras que la aplicación tiene un diseño intuitivo?
- ¿Consideras que la aplicación tiene un diseño estéticamente agradable?
- ¿Consideras que la aplicación puede ser útil en tu día a día?
- ¿Recomendarías esta aplicación a tus amigos y familiares?

3.4.4 Análisis de resultados

Para evaluar las tareas será necesario tener a un colectivo de usuarios lo más grande posible siempre que se disponga del tiempo para dedicar a cada uno de ellos, considerando que las preguntas abiertas deben dar lugar a un diálogo en el que permitir al usuario explayarse de manera que se puedan sacar todas las conclusiones posibles.

Toda la documentación recopilada en los pasos anteriores se estudiará minuciosamente para obtener todas las propuestas de mejora posibles y realizar un nuevo prototipo en base a ellas.

Cualquier respuesta negativa en cuanto a la ergonomía debe ponderarse según el número de usuarios que puedan tener una opinión similar y proponer una mejora que mitigue el problema. Una respuesta masivamente negativa a las preguntas cualitativas sobre la utilidad de la aplicación debería hacer reconsiderar el desarrollo en su totalidad.

Las preguntas abiertas sobre la utilidad de las funcionalidades pueden llevar también a retoques en el diseño, que den más protagonismo a las funcionalidades que se consideren más prioritarias, estableciendo también la criticidad de cara al desarrollo inicial de la aplicación.

4. Diseño técnico

4.1 Definición de casos de uso

Identificador	**CU-001**
Nombre	Autenticación / alta en la plataforma
Prioridad	Alta
Descripción	Paso inicial para poder hacer uso de la aplicación.
Precondiciones	Ninguna
Actor / Iniciado por	Cualquier tipo de usuario
Flujo	El usuario entra en la aplicación y pulsa en el botón de autenticación
Postcondiciones	- El usuario queda registrado en la plataforma

Identificador	**CU-002**
Nombre	Creación de comunidad
Prioridad	Alta
Descripción	Alta de una nueva comunidad
Precondiciones	- El usuario debe estar autenticado - El usuario no puede pertenecer ya a una comunidad - El identificador de la comunidad no puede estar ya en uso
Actor / Iniciado por	Cualquier tipo de usuario
Flujo	Un usuario autenticado que aún no pertenece a una comunidad inicia la aplicación. Aparecerá la ventana para crear o ingresar en una comunidad. El usuario indica un nombre de comunidad que aún no exista y pulsa el botón para crearla.
Postcondiciones	- Se crea una comunidad - El usuario queda registrado en la comunidad - El usuario recibe rol de administrador en la comunidad

Identificador	**CU-003**
Nombre	Ingreso en una comunidad existente
Prioridad	Alta
Descripción	Un usuario ingresa en una comunidad ya existente
Precondiciones	- El usuario debe estar autenticado - El usuario no puede pertenecer ya a una comunidad - La comunidad debe haberse creado
Actor / Iniciado por	Cualquier tipo de usuario
Flujo	Un usuario autenticado que aún no pertenece a una comunidad inicia la aplicación. Aparecerá la ventana para crear o ingresar en una comunidad. El usuario indica un nombre de comunidad existente y pulsa el botón para ingresar en ella.
Postcondiciones	- El usuario queda registrado en la comunidad

Identificador	**CU-004**
Nombre	Creación de ticket
Prioridad	Alta
Descripción	Un usuario genera un nuevo ticket
Precondiciones	- El usuario debe estar autenticado - El usuario debe pertenecer a una comunidad
Actor / Iniciado por	Cualquier tipo de usuario
Flujo	Un usuario autenticado que pertenece a una comunidad inicia la aplicación. Aparecerá la ventana resumen con el listado de tickets. El usuario pulsa el botón para crear uno nuevo, navegando a la ventana de detalle, introduce todos los detalles requeridos y pulsa el botón de guardar.
Postcondiciones	- Se genera un nuevo ticket, visible desde la ventana resumen

Identificador	**CU-005**
Nombre	Edición de ticket
Prioridad	Media
Descripción	Un usuario administrador edita un ticket existente
Precondiciones	- El usuario debe estar autenticado - El usuario debe pertenecer a una comunidad - El usuario debe ser administrador - Debe existir el ticket a editar
Actor / Iniciado por	Usuario administrador
Flujo	Un usuario administrador autenticado que pertenece a una comunidad inicia la aplicación. Aparecerá la ventana resumen con el listado de tickets. El usuario pulsa sobre uno de los tickets, navegando a la ventana de detalle, pulsa sobre el botón de editar en cualquiera de los campos de texto o selecciona nuevos valores de los desplegables, finalmente pulsa sobre el botón de guardar.
Postcondiciones	- El ticket editado queda con los nuevos valores establecidos

Identificador	**CU-006**
Nombre	Creación de comentario
Prioridad	Media
Descripción	Un usuario crea un comentario sobre un ticket existente
Precondiciones	- El usuario debe estar autenticado - El usuario debe pertenecer a una comunidad - Debe existir el ticket al que crearle un comentario
Actor / Iniciado por	Cualquier tipo de usuario
Flujo	Un usuario autenticado que pertenece a una comunidad inicia la aplicación. Aparecerá la ventana resumen con el listado de tickets. El usuario pulsa sobre uno de los tickets, navegando a la ventana de detalle, pulsa sobre el botón de crear un comentario, lo escribe y guarda.
Postcondiciones	- Un comentario queda registrado en el ticket con el usuario que lo ha escrito, fecha y hora.

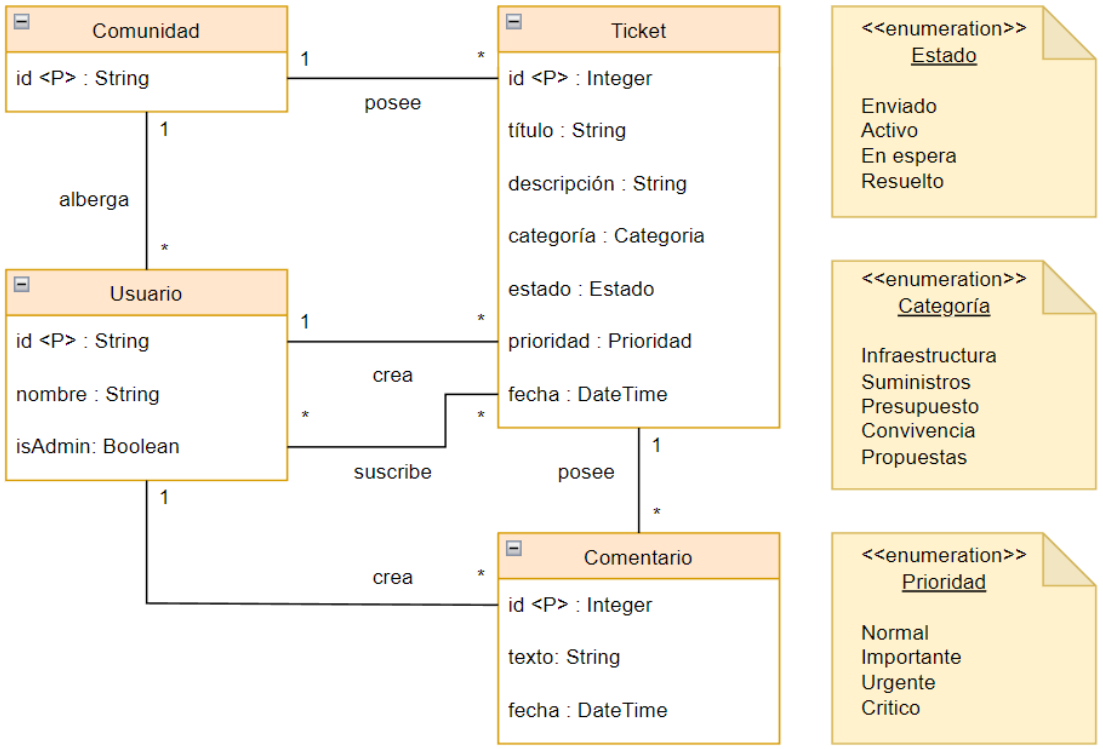
Identificador	**CU-007**
Nombre	Suscripción a ticket
Prioridad	Baja
Descripción	Un usuario crea un comentario sobre un ticket existente
Precondiciones	- El usuario debe estar autenticado - El usuario debe pertenecer a una comunidad - El usuario no puede ser administrador - Debe existir el ticket al que suscribirse
Actor / Iniciado por	Usuario no administrador
Flujo	Un usuario autenticado que pertenece a una comunidad inicia la aplicación. Aparecerá la ventana resumen con el listado de tickets. El usuario pulsa sobre uno de los tickets, navegando a la ventana de detalle, pulsa sobre el botón de suscripción.
Postcondiciones	- El usuario queda registrado como suscrito al ticket, permitiendo que se le notifiquen los cambios.

Identificador	**CU-008**
Nombre	Eliminación de ticket
Prioridad	Media
Descripción	Un usuario administrador elimina un ticket existente
Precondiciones	- El usuario debe estar autenticado - El usuario debe pertenecer a una comunidad - El usuario debe ser administrador - Debe existir el ticket a eliminar
Actor / Iniciado por	Usuario administrador
Flujo	Un usuario administrador autenticado que pertenece a una comunidad inicia la aplicación. Aparecerá la ventana resumen con el listado de tickets. El usuario pulsa sobre uno de los tickets, navegando a la ventana de detalle y pulsa sobre el botón de eliminar ticket.
Postcondiciones	- El ticket eliminado y todos sus comentarios dejan de ser visibles para cualquier tipo de usuario

Identificador	**CU-009**
Nombre	Eliminación de comentario
Prioridad	Media
Descripción	Un usuario administrador elimina un comentario existente de un ticket
Precondiciones	- El usuario debe estar autenticado - El usuario debe pertenecer a una comunidad - El usuario debe ser administrador - Deben existir el ticket y su comentario asociado a eliminar
Actor / Iniciado por	Usuario administrador
Flujo	Un usuario administrador autenticado que pertenece a una comunidad inicia la aplicación. Aparecerá la ventana resumen con el listado de tickets. El usuario pulsa sobre uno de los tickets, navegando a la ventana de detalle y pulsa sobre el botón de eliminar comentario de alguno de los comentarios.
Postcondiciones	- El comentario eliminado deja ser visible para cualquier tipo de usuario

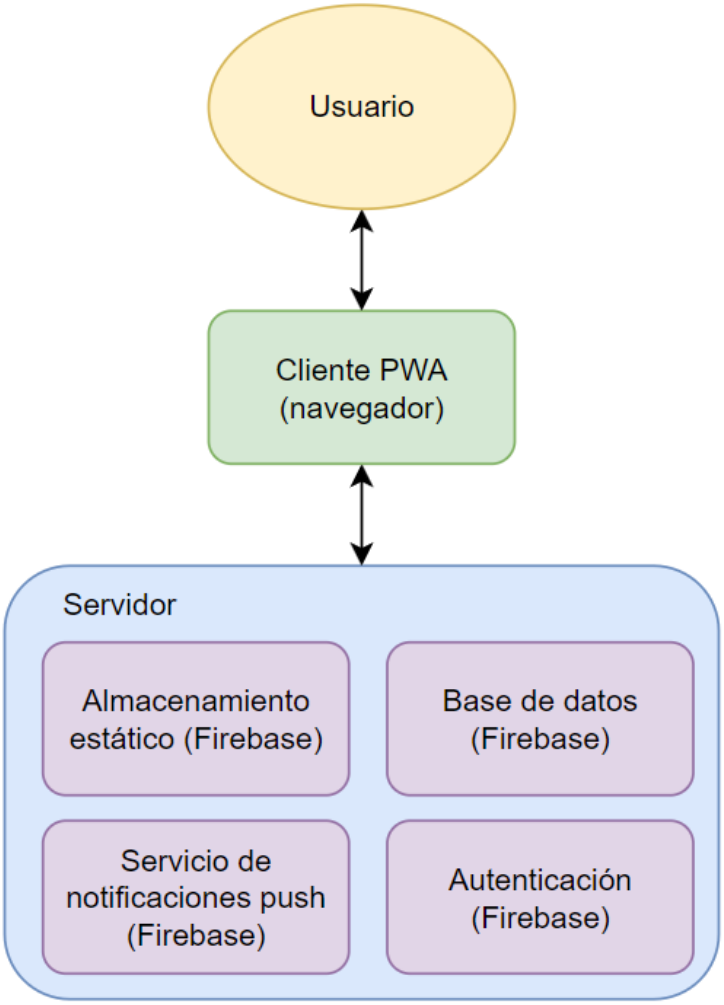
4.2 Diseño de la arquitectura

4.2.1 Diagrama de la base de datos



Alberto Antonio Rubio Sánchez
74824024X

4.2.2 Diagrama explicativo de la arquitectura del sistema



La arquitectura se corresponde con un modelo cliente/servidor, donde el cliente es la interfaz de usuario accesible a través de navegador, y el servidor son todos los servicios de Firebase a los que tiene que acceder la aplicación de manera independiente.

5. Desarrollo

5.1 Descripción de las herramientas utilizadas

Visual Studio Code

Editor de código generalista que limita sus funcionalidades de IDE para resultar ligero y cómodo de usar. En la elaboración de la aplicación se ha utilizado para toda la tarea de codificación y pruebas, además de para la instalación de paquetes mediante npm y gestión del repositorio mediante git.

Node.js

Entorno de ejecución de JavaScript orientado a eventos asíncronos. Requerido por Vite.

Vite

Herramienta de compilación para proyectos web. Destaca por su velocidad de compilación, enfoque de desarrollo con módulos y optimización de la carga en el navegador. Creada principalmente para su uso con Vue 3, aunque admite otros frameworks.

Vue.js (Vue 3)

Framework Javascript progresivo, para el desarrollo de interfaces web. Es el framework para el que se ha desarrollado toda la aplicación, modularizando el contenido en ficheros para vistas independientes con su contenido propio de código Javascript y HTML.

Pinia

Herramienta para almacenamiento en memoria de cliente de información, pensada para su uso con Vue 3. En la aplicación se usa para intercambiar entre vistas la información de autenticación del usuario y la información de la comunidad a la que está asignado este.

Firebase

Hosting

La aplicación web se encuentra alojada en el servicio de hosting de Firebase. El workflow asociado al merge de git realiza el despliegue en Firebase sin más intervención que la aprobación de la acción.

Authentication

Firebase permite integrar la autenticación de distintos proveedores en la aplicación que se esté desarrollando. En el caso de esta aplicación se utiliza exclusivamente la autenticación mediante Google, para lo que la aplicación muestra un pop-up de login de Google desde el que se modifica la instancia del objeto de autorización accesible desde la aplicación, permitiendo identificar unívocamente al usuario y almacenar los datos necesarios en base de datos.

Cloud Firestore

Esta utilidad permite hacer uso de una base de datos en la nube con escalamiento automático que, además, permite consultar la información y mostrarla en tiempo real. Esto permite a esta aplicación tener un backend mínimo, y limitado a la gestión de las notificaciones.

Cloud Functions

Firebase Cloud Functions permite subir al servidor de Firebase el código que se requiera ejecutar en el servidor. Se usa principalmente para la gestión de eventos disparados por Firestore. Para esta aplicación se ha utilizado para implementar observadores sobre la base de datos que permitan la emisión de notificaciones al actualizar tickets o generar comentarios.

Cloud Messaging

Es un servicio de mensajería de Firebase para la implementación de mensajería push a dispositivos móviles. Permite el uso de mensajería en la aplicación, tanto el envío al dispararse desde cambios en la base de datos como la recepción en la aplicación cliente.

Rules Test Environment

Esta es una utilidad pensada para hacer pruebas locales sobre las reglas de seguridad de Cloud Firestore, pero también tiene funcionalidades para realizar cualquier tipo de pruebas unitarias relacionadas con Firebase. En el desarrollo de esta aplicación se ha utilizado para instanciar un emulador de Firestore que

permita hacer pruebas unitarias sobre el tipo de operaciones que se realizan durante la ejecución.

Vitest

Framework de pruebas para Vite. Permite escribir y ejecutar pruebas unitarias y de integración. Para el desarrollo de esta aplicación se ha usado para el desarrollo de las pruebas unitarias.

Github

El repositorio y control de versiones de la aplicación se encuentra en Github. Durante el desarrollo de la aplicación, a pesar de sólo haber un programador implicado, se han mantenido las distintas ramas y subido el contenido productivo de manera progresiva al repositorio y el hosting de producción.

También se ha hecho uso de Github Actions para la aplicación de los workflows generados por Firebase y otro workflow para la ejecución de los tests unitarios al hacer un *pull request*.

5.2 Análisis del estado del proyecto

Se han completado todos los objetivos marcados en las fases previas y se han añadido algunas pequeñas mejoras de usabilidad que no estaban originalmente previstas en el diseño.

De manera esquemática, se han cubierto estos objetivos:

Aplicación instalable en móviles (PWA)

Se han estudiado y cubierto los requisitos para que la aplicación sea instalable y se muestre al usuario como una aplicación de móvil corriente.

Autenticación y gestión de usuarios

La aplicación permite identificar, mediante Firebase Authentication, al usuario que accede a la misma, dándolo de alta y guardando la información relevante en la base de datos.

Gestión de base de datos

Se ha establecido una base de datos en Firebase Cloud Firestore para almacenar ahí la información relativa a usuarios, comunidades, tickets y comentarios. Más adelante se han añadido también aquí la gestión de favoritos y los tokens de dispositivos para la gestión de notificaciones push.

Desarrollo de vistas

Se han desarrollado todas las vistas descritas en el diseño y mostradas en el prototipo y estas contienen las funcionalidades requeridas: autenticación, acceso a comunidad, listado de tickets, creación de ticket, detalle de ticket para usuario no administrador y detalle de ticket para administrador.

Gestión de comentarios y actualización de estado de tickets

En cada ticket cualquier usuario puede añadir comentarios, como una funcionalidad dentro de las vistas de detalle de ticket. El usuario administrador puede, además, modificar el estado del ticket.

Se ha añadido la posibilidad de que el administrador pueda entrar en las dos vistas de detalle del ticket, de manera que en una pueda acceder al detalle del ticket como un usuario normal (aunque igualmente pueda modificar el estado del ticket, a diferencia del resto de usuarios), y también a la vista donde puede modificar cualquier valor contenido en el ticket, incluyendo los comentarios.

Notificaciones push

Se ha desarrollado la funcionalidad que dispara notificaciones push a usuarios suscritos a tickets en el momento en el que el estado de estos se actualiza o se añade un comentario. Esta funcionalidad hace uso de Firebase Cloud Messaging para el envío de notificaciones desde Firebase y la recepción desde la aplicación cliente. Para el envío de las notificaciones, la lógica se ha subido a la nube mediante Firebase Cloud Functions.

Test unitarios

Se han desarrollado tests unitarios mediante Vitest para cubrir las funcionalidades más relevantes de la aplicación, relativas al manejo de la base de datos. Para ello se ha contado con el entorno de test de reglas de seguridad de Firebase, el cual permite el uso de un emulador de Firestore para realizar transacciones contra una base de datos ficticia cuyos registros desaparecen después de la ejecución de los tests.

Workflow para tests para integración continua

Se ha implementado una pequeña funcionalidad para la integración continua. Además de los workflows de despliegue que ya genera Firebase, se ha añadido otro workflow que dispara los tests unitarios al realizar un *pull request* sobre el repositorio.

6. Pruebas

6.1 Descripción de cómo está previsto probar la aplicación

En general, al haber desarrollado la aplicación siguiendo un patrón de integración continua, se ha ido probando cada funcionalidad, primero compilando en local y luego en el entorno productivo de la aplicación. Esto es tanto en navegador web de escritorio como en la aplicación instalada como PWA en dispositivos móviles Android e iOS.

Además de estas pruebas, que pueden considerarse end-to-end, se han realizado pruebas unitarias automatizadas mediante el framework Vitest y el uso de un emulador de Firebase. Estas pruebas cubren la mayor parte de funcionalidades de interacción con la base de datos Firestore.

6.2 Pruebas unitarias realizadas

Se han desarrollado tres bloques de pruebas unitarias diferenciadas, centradas en el uso de la base de datos Firestore para la gestión de usuarios, comunidades y tickets, incluyendo este último la gestión de comentarios.

Estas pruebas pueden encontrarse en la ruta **/src/test** del código de la aplicación.

Usuarios / usuarios.test.ts

Incluye pruebas unitarias de la creación de usuarios en base de datos, la actualización de los mismos y su borrado.

Comunidades / comunidades.test.ts

Incluye pruebas unitarias de la creación, actualización y borrado de comunidades.

Tickets / tickets.test.ts

Incluye pruebas unitarias de la creación, actualización y borrado de tickets, así como la creación y borrado de comentarios.

7. Producto

7.1 Código fuente

El código fuente puede descargarse del siguiente enlace:

https://drive.google.com/file/d/1GKtoRh9RAWiSUYUebVbLVtat2a6yqp53/view?usp=share_link

La aplicación requiere del proyecto definido en Firebase para su ejecución. Por lo que el código no es ejecutable por sí mismo.

7.1.1 Ubicaciones de especial interés en el código fuente

A continuación se describen las ubicaciones del directorio del código fuente donde se encuentran los puntos más relevantes para el desarrollo.

/

En la raíz del árbol de directorios se encuentran la mayoría de ficheros de configuración de la aplicación. Entre otros, los de Node, Vite, Vitest y Firebase.

/functions/

En este directorio se encuentran los ficheros necesarios para el uso de Firebase Cloud Functions. Concretamente el fichero **index.js** tiene el código desarrollado para la aplicación para disparar las notificaciones a los suscriptores cuando se crea un nuevo comentario en un ticket o se actualiza el estado del mismo, una función para cada tipo de notificación.

/public/

Aquí se encuentran tanto los iconos necesarios para que la aplicación pueda ser tratada como una PWA (incluyendo una imagen vectorial) como el fichero que necesita Firebase Cloud Messaging tener expuesto para su funcionamiento.

/src/

Aquí se encuentran los directorios con los distintos componentes de código, descritos más adelante, y, también los ficheros **main.ts**, el cual inicializa la aplicación, el enrutador para la navegación entre vistas y Pinia, y el fichero App.vue, que se corresponde el punto de entrada para el usuario de la aplicación.

/src/firebase/index.ts

Este fichero contiene la generación de las instancias de la base de datos y el servicio de mensajería de Firebase. También obtiene el token inicial del dispositivo del usuario para poder enviarle notificaciones push.

/src/router/index.ts

Este fichero identifica las vistas Vue creadas con rutas navegables, y es el que utilizan las instrucciones de navegación ligadas a los botones de cada vista.

/src/stores/

Contiene la definición de los objetos para el almacenamiento local de la información de usuario y comunidad, creados para su uso con Pinia.

/src/test/

Contiene el código de los tests unitarios.

/src/views/

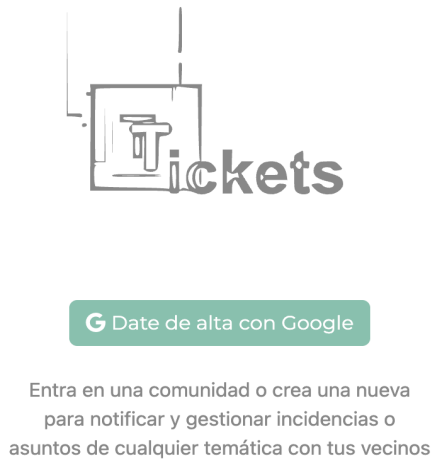
Contiene el código de todas las vistas que usa la aplicación, exceptuando la vista principal, fichero **App.vue**, que se encuentra en **/src/**.

7.2 Manual de usuario

La aplicación pretende ser bastante autoexplicativa y sencilla, y cuenta con tooltips en todos los botones.

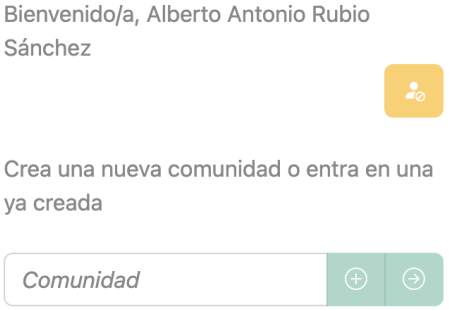
Darse de alta en la aplicación

Como primer paso tendremos una vista de login que lleva a la selección de cuenta de Google. El alta consiste en pulsar el botón y seleccionar una cuenta de Google del dispositivo.



Ingresar en una comunidad o darla de alta

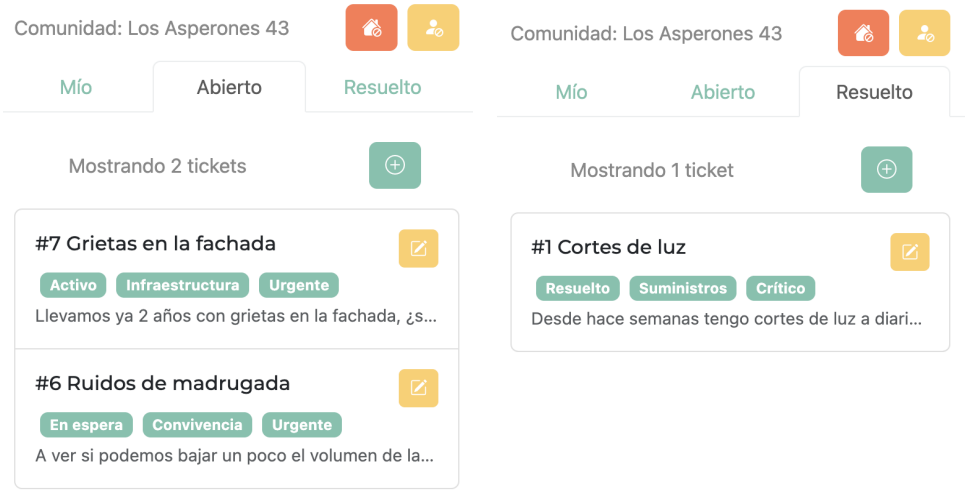
Una vez realizado el login es necesario asociar al usuario con una comunidad, ya sea creando una nueva o añadiéndose como miembro de una ya existente.



En esta vista, además de los botones para crear una comunidad o ingresar en una existente, contamos con un botón de sign-out, que permite desvincular el dispositivo utilizado del usuario, e ingresar con uno diferente si se desea.

Visualizar listado de tickets de la comunidad

Una vez dentro de una comunidad podremos ver la vista de listado, donde pueden verse todos los tickets de la comunidad con distintos filtros predefinidos: los que haya creado el usuario activo, los que se encuentren activos y los que se hayan resuelto ya.



Las vistas de listado cuentan con botones para salir de la comunidad, hacer sign-out o crear nuevos tickets. Además, en el caso de ser el administrador de la comunidad, se mostrará también un botón para editar y rectificar todos los valores del ticket.

Crear un nuevo ticket

Desde la vista de listado, pulsando el botón de añadir ticket accedemos a la vista de creación de ticket. En ella será necesario introducir un título y una descripción, y seleccionar una categoría y una prioridad.

Una vez se pulse el botón para guardar, el ticket quedará registrado con estado “enviado”.

Visualizar un ticket existente, añadir comentarios, suscribirse a un ticket

Pulsando sobre el marco de uno de los tickets de la vista de listado accederemos a la vista de detalle y podremos ver tanto la descripción completa y detalles del ticket como todos los comentarios que se hayan ido añadiendo.

Para añadir un comentario bastará con escribir en el cuadro de texto y pulsar el botón de guardar.

Marcar el ticket como favorito, pulsando el botón de la estrella, permitirá que recibamos notificaciones push cuando cambie de estado o se añada un nuevo comentario.

Si el usuario está identificado como administrador, este podrá, además, modificar el estado del ticket desde el primer desplegable, no apareciendo activo para otros usuarios.

Editar un ticket como administrador

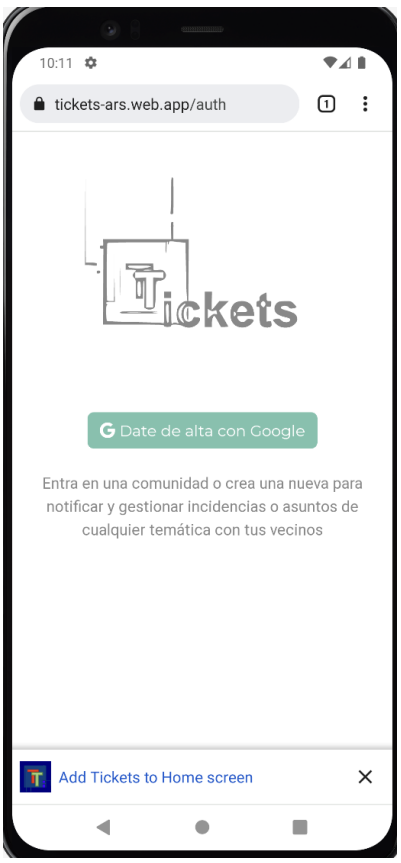
En la vista de listado, en el caso de que el usuario esté identificado como administrador, aparecerá un botón de edición en cada ticket. Pulsando en este botón accederemos a la vista de edición de administrador.

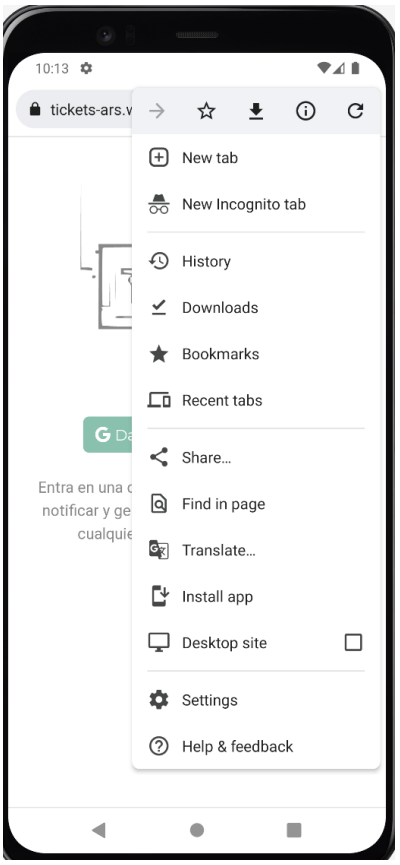
Esta vista permite modificar todos los valores del ticket (pulsando, modificando el valor y pulsando el botón de guardar) o eliminarlo completamente.

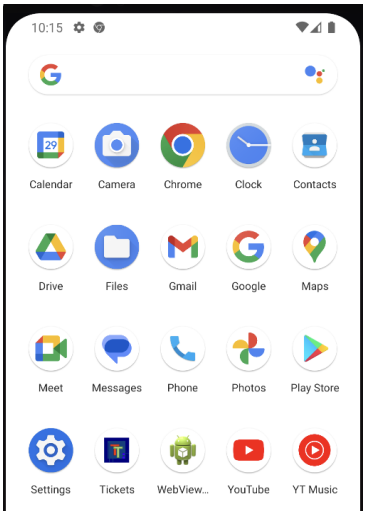
También permite modificar cualquier comentario o eliminarlo.

7.3 Instrucciones compilación y ejecución

La aplicación se encuentra desplegada y accesible en la dirección http://tickets-ars.web.app. Puede utilizarse directamente desde la web en cualquier dispositivo móvil, aunque está pensada para instalarse como PWA. Para instalarla como PWA podemos pulsar sobre la sugerencia de añadirla a la pantalla de inicio que aparece la primera vez que se accede a la página, o bien desde el menú de Google Chrome pulsando en “instalar aplicación”.







La aplicación aparecerá como una más en el listado de aplicaciones del dispositivo.

8. Presentación

https://drive.google.com/file/d/1pJxxJHs_yVpwhJweY1ndXpsyyQmTqk9T/view?usp=sharing

9. Conclusiones

9.1 Descripción de las conclusiones

Haber escogido una variedad de tecnologías sobre las que no tenía experiencia previa ha resultado una experiencia enriquecedora a la par de frustrante e irritante. Por un lado me ha permitido empezar a tocar áreas que tenía mucho interés por descubrir, pero por otro, ha sido una tarea llena de tropiezos y arreglos de última hora donde tampoco he podido profundizar demasiado en las distintas herramientas u obtener el producto óptimo que me hubiese gustado.

De manera general estoy muy satisfecho con el resultado, que pienso que puede considerarse una prueba de concepto de una idea con cierto interés social llevada a cabo con tecnologías de interés actual.

9.2 Reflexión crítica sobre el logro de los objetivos

Considerando mi inexperiencia en general con este tipo de desarrollos, y con las herramientas utilizadas en particular, me siento especialmente orgulloso de haber podido cubrir todos los objetivos propuestos en la planificación y el diseño con cierta solvencia.

No obstante, y debido precisamente a esa falta de experiencia, la calidad de la solución no es la mejor posible y cuenta con algunas carencias en ámbitos no previstos durante las etapas previas a la implementación. Lo más relevante son las limitaciones en el ámbito de seguridad que tiene la aplicación, la cual puede tener algunas vulnerabilidades accesibles para un usuario con conocimientos técnicos avanzados.

9.3 Análisis crítico del seguimiento de la planificación y la metodología

Pienso que, en general, la planificación ha podido seguirse con ajustes mínimos, y esto se ha ido trasladando en todo momento a los tutores del proyecto. Por otro lado, la sensación de haber subestimado cada una de las tareas planteadas ha sido constante. Han sido muchas herramientas diferentes sin tener más que una idea previa de para qué servían, y ha resultado difícil ponerlas juntas a funcionar sin haber trabajado con ellas antes.

Ha sido de ayuda, aunque pueda no ser lo ideal, el plantear una metodología flexible desde el primer momento. Sin esa libertad, los tiempos para ir sacando adelante cada elemento del proyecto podría haber sido insostenible. A pesar de dicha flexibilidad, se ha mantenido un control escrupuloso del desarrollo modularizado y copias de seguridad mediante Git, que considero que se ha llevado a cabo de manera solvente.

9.4 Líneas de trabajo futuro

Seguridad en la aplicación

La prioridad en el hipotético trabajo futuro a partir de la aplicación sería mejorar el aspecto de su seguridad. Como se ha indicado antes, este apartado tiene algunas carencias y puede mejorarse dedicando algo de tiempo.

Por cómo se ha ideado la arquitectura de la aplicación y tratando de sacarle el máximo partido a los frameworks y herramientas utilizadas, prácticamente todo el contenido de la aplicación se encuentra en el cliente. Esto, que facilita muchas partes del desarrollo, también puede suponer un arma de doble filo, permitiendo a usuarios malintencionados alterar y ejecutar ciertas llamadas a base de datos.

La aplicación en su estado actual impide que un usuario no autenticado actúe sobre la base de datos, y también valida en cliente que el usuario tenga el rol de administrador para ejecutar las acciones previstas para este tipo de usuario, pero esto no es suficiente para asegurar que no se abuse del sistema. La solución pasa por implementar reglas en Cloud Firestore Security Rules que validen el token de autenticación del usuario contra sus permisos en base de datos según la acción que vaya a realizar, de manera que no sea posible que cualquier otro usuario pueda alterar el código de la aplicación para tocar la base de datos.

Por otro lado, y de manera menos crítica, debería mejorarse la manera en que se permite o impide el acceso a ciertas acciones de administración, mediante código que también podría alterarse en ejecución.

Acceso a comunidades por invitación, restringir el acceso

Una vez se crea una comunidad, esta es accesible por cualquier usuario registrado en el momento en que este conozca el nombre.

Merecería la pena añadir dos funcionalidades a este respecto: las solicitudes de acceso en lugar de la entrada directa a la comunidad, permitiendo al administrador de la comunidad validar si permite o no entrar al usuario; y las invitaciones, donde puedas mandar un enlace para acceder directamente a alguien mediante cualquier manera de compartir que permita el dispositivo (email, Whatsapp, Telegram…).

Gestión de la administración de una comunidad

Tal como funciona en este momento la aplicación, el usuario que crea una comunidad pasa a ser su administrador, lo cual es inmutable para el usuario.

La aplicación podría contar con una vista de gestión de administración, que permitiera traspasar o duplicar los permisos de administración a otros usuarios pertenecientes a la comunidad.

Por otro lado, sería muy interesante explorar la posibilidad de varios grados de administración, pudiendo contar con roles diferenciados dentro de una comunidad. Por ejemplo, un superadministrador que pueda modificar todos los detalles de la comunidad, y varios usuarios administradores con permiso sólo para cambiar de estado y cerrar tickets.

Herramientas de moderación

Aunque la aplicación permite editar y eliminar tickets y comentarios, no permite bloquear o echar a usuarios de una comunidad. Este tipo de cosas puede ser indispensable según el tipo de usuario que encontremos.

Otra medida más sofisticada también interesante pueden ser las de establecer tiempos entre un comentario y otro, o a la hora de realizar accesos a la base de datos, para evitar spam.

Existen también herramientas de auto-moderación que podrían evaluar si el contenido de un mensaje es legítimo o no, por ejemplo, según contenga cierto tipo de enlaces o estructuras de texto. Añadir este tipo de funcionalidades a la aplicación podría ser muy positivo para su uso.

Detalle del perfil de usuario y comunidad

Actualmente no se guarda apenas detalle del usuario o la comunidad, básicamente sus identificadores y nombres. Añadiendo algunos detalles estéticos y unas pocas vistas se podría personalizar la experiencia de usuario haciendo la aplicación mucho más amigable y atractiva.

Respecto al usuario, es bastante sencillo recuperar el enlace a su foto del perfil de Google y mostrarla en los comentarios a modo de avatar. Se podría establecer también una vista de perfil de usuario al pulsar en su foto, o en su nombre, que mostrase datos biográficos introducidos por el usuario. Esto, en el contexto de esta aplicación puede ser especialmente útil para entender el rol del usuario concreto dentro de la aplicación cuando este ha indicado que ha realizado alguna acción concreta sobre una incidencia.

Por la parte de la comunidad, podría desligarse el nombre de la identificación, permitiendo cambiar el nombre por parte del administrador. Más allá de esto también puede ser interesante contar con más detalles, como una lista de usuarios, teléfonos de emergencia u otros datos que podrían estar en un tablón de anuncios físicamente, pero accesible desde esta aplicación

Añadir más formas de autenticación

La autenticación mediante Google puede ser la más estandarizada ahora mismo, pero prácticamente todas las aplicaciones permiten varias maneras de identificarse además de una básica mediante usuario y contraseña.

Firebase Authentication permite añadir proveedores a su sistema de autenticación, aunque cada uno se controla de manera ligeramente distinta.

Añadir la aplicación a tiendas de aplicaciones

En el ecosistema actual de aplicaciones, limitarse al acceso mediante una web reduce considerablemente el impacto que pueda tener una aplicación nueva. Si se pretende su uso en gran escala es necesario tener presencia en las tiendas de aplicaciones y realizar alguna inversión económica en promoción.

10. Glosario

PWA (Progressive Web App): aplicación web que utiliza tecnologías modernas para ofrecer una experiencia de usuario similar a la de una aplicación móvil nativa.

Ticketing: En el contexto de gestión de incidencias, el ticketing se refiere al proceso de registro, seguimiento y resolución de incidencias o problemas técnicos que pueden surgir en una organización. El ticketing permite a los usuarios y clientes reportar problemas, solicitar soporte o hacer consultas a través de un sistema de gestión de incidencias, donde se les asigna un número de ticket único para su seguimiento. Los tickets pueden ser asignados a un equipo o técnico específico para su resolución, y se puede hacer un seguimiento de su estado y progreso a lo largo del tiempo. El objetivo del ticketing en la gestión de incidencias es garantizar que los problemas sean resueltos de manera oportuna y eficiente, y que los usuarios sean informados sobre el progreso y la resolución de sus problemas.

SFC (Single File Component): Tipo de componente que se utiliza comúnmente en el desarrollo de aplicaciones web utilizando el framework Vue.js. En un SFC, todo el código necesario para definir el componente, incluyendo su plantilla HTML, su estilo CSS y su lógica de JavaScript, se encuentra en un mismo archivo. Estos tres componentes se identifican, respectivamente, con las etiquetas *template*, *style* y *script*.

11. Bibliografía

The Modern JavaScript Tutorial. (s.f.). Recuperado el 29 de mayo de 2023, de https://javascript.info/

Firebase. (2023). Documentation. Recuperado el 29 de mayo de 2023, de https://firebase.google.com/docs

Vite. (s.f.). Guide. Recuperado el 29 de mayo de 2023, de https://vitejs.dev/guide/

Vue.js. (2023). Documentation. Recuperado el 29 de mayo de 2023, de https://vuejs.org/v2/guide/

Pinia. (2023). Documentation. Recuperado el 29 de mayo de 2023, de https://pinia.esm.dev/

Bootstrap. (2023). Documentation. Recuperado el 29 de mayo de 2023, de https://getbootstrap.com/docs/5.0/

12. Anexos

Documento de presentación para la presentación en vídeo del proyecto:

https://docs.google.com/presentation/d/1AO3qi7sEvxFXPlpHQIoiRcvNFcpGmhNFagRCYi8gN2I/edit?usp=sharing