



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

Aplicación web para una comunidad
de ocio en Canarias (Canary Go)

*Web application for a leisure community in the Canary Islands
(Canary Go)*

Iván Josué Borges González

La Laguna, 6 de julio de 2020

D. **Vicente José Blanco Pérez**, con N.I.F. 42.171.808-C profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

CERTIFICA

Que la presente memoria titulada:

"Aplicación web para una comunidad de ocio en Canarias (Canary Go)"

ha sido realizada bajo su dirección por D. **Iván Josué Borges González**, con N.I.F. 43.835.785-Q.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 6 de julio de 2020

Agradecimientos

En primer lugar, quiero agradecer a mi tutor Vicente José Blanco Pérez por la continua orientación y preocupación de que el desarrollo del proyecto fuera el mejor posible y de los posibles problemas que podían surgir.

En segundo lugar, quiero agradecer a mi familia y amigos por el constante apoyo durante todos estos años.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.

Resumen

El ocio es una actividad que ha sido necesaria para el ser humano desde sus inicios hasta la actualidad. Aporta grandes beneficios tales como eliminar o disminuir el estrés, aumentar la energía positiva, mantener la salud a largo plazo y reducir el riesgo de tener enfermedades como diabetes e hipertensión.

Sin embargo, en los últimos años ha descendido la cantidad de tiempo empleado en el buen ocio considerablemente. Por ello, la idea de este proyecto es crear una plataforma web novedosa al alcance de todo el mundo (permitiendo su uso en cualquier dispositivo de escritorio o móvil), que tenga una modalidad de uso gratuito y que contenga una gran cantidad de eventos o servicios de ocio en un solo lugar, equilibrando la balanza.

Para el desarrollo de la aplicación web se han utilizado diferentes tecnologías. Para la parte de "front-end", se ha utilizado el framework Quasar CLI que se basa en Vue JS y para la parte de "back-end" se ha utilizado Node JS para desarrollar el servidor. Para garantizar la seguridad del proyecto en las conexiones entre servidor y cliente se ha utilizado el sistema JWT (JSON Web Token). El proyecto hace uso de Travis CI para la integración continua y prueba del código. La plataforma de despliegue utilizada es Heroku y se configuró el proyecto para que se realizaran los despliegues de forma automatizada.

La plataforma utilizada para el almacenamiento y gestión de datos es Firebase ya que permite tanto el almacenamiento de datos (mediante Firestore) como el almacenamiento de archivos (mediante Storage) y la gestión de usuarios (mediante Authentication) consiguiendo un sistema unificado sin depender de diferentes herramientas.

La plataforma utilizada para el testeo de la aplicación web es Cypress, un "framework" "todo en uno" para el testeo de aserciones, Mockups, pruebas y pruebas e2e automáticas.

Palabras clave: Ocio, Eventos, Canarias, Quasar, Vue JS, Vuex, Javascript, JWT, Leaflet, Google, Firebase, Firestore, Storage, Cypress, Heroku, SPA

Abstract

Leisure is an activity that has been necessary for the human being from its beginnings to the present day. It provides great benefits such as eliminating or reducing stress, increasing positive energy, maintaining long-term health and reducing the risk of having diseases such as diabetes and hypertension.

However, in recent years the amount of time spent on good leisure has decreased considerably. For this reason, the idea of this project is to create a web platform available to everyone (allowing its use on any desktop or mobile device), which has a free use modality and contains a large number of events or services leisure in one place. In this way we can "balance the balance".

Different technologies have been used to develop the web application. For the front-end part, the Quasar CLI framework based on Vue JS has been used, and for the back-end part, Node JS has been used to develop the server. To guarantee the security of the project in the connections between server and client, the JWT system (JSON Web Token) has been used. The project makes use of Travis CI for continuous integration and testing of the code. The deployment platform used is Heroku and the project was configured so that the deployments were carried out automatically.

The platform used for data storage and management is Firebase since it allows both data storage (through "Firestore") and file storage (through "Storage") and user management (through ".Authentication") achieving a unified system without relying on different tools. The platform used to test the web application is Cypress, an all-in-one framework for testing assertions, Mockups, tests and automatic e2e tests.

Keywords: Leisure, Events, Canary Islands, Quasar, Vue JS, Vuex, Javascript, JWT, Leaflet, Google, Firebase, Firestore, Storage, Cypress, Heroku, SPA

Índice general

1. Introducción	1
1.1. Antecedentes y estado actual del ocio	1
1.2. Idea principal y problemas que resuelve la aplicación web	2
1.3. Tareas realizadas durante el desarrollo	3
2. Entorno de desarrollo	6
2.1. IDE	6
2.1.1. Visual Studio Code	6
2.2. Front-end	6
2.2.1. Quasar	6
2.2.2. Leaflet	8
2.3. Back-end	8
2.3.1. Node JS	8
2.4. Almacenamiento y gestión de datos	8
2.4.1. Firebase Authentication	9
2.4.2. Firebase Firestore	9
2.4.3. Firebase Storage	9
2.5. Despliegue	10
2.5.1. Heroku	10
2.6. Control y pruebas	10
2.6.1. ESLint	10
2.6.2. Pivotal Tracker	10
2.6.3. GitHub	11
2.6.4. Travis CI	11
2.6.5. Cypress	11
3. Descripción de la aplicación (Canary Go)	12
3.1. Disponibilidad	12
3.2. Internacionalización	12
3.3. Modo oscuro	12
3.4. Sección de información de la web	13
3.5. Sección de registro	13
3.6. Sección de inicio de sesión	14
3.7. Sección de eventos	14
3.8. Sección de descripción del evento	14
3.9. Sección de añadir evento	15
3.10 Sección de actualizar el perfil de usuario	15

4. Desarrollo	23
4.1. Instalación y configuración general	23
4.2. Visual Studio Code	25
4.3. Travis CI	26
4.4. Heroku	26
4.5. Vue-Router	27
4.6. Sección de Información	27
4.7. Login y Registro	29
4.8. Editor WYSIWYG y selector de emojis	30
4.9. Internacionalización y selector de idioma	33
4.10Firestore y Storage	34
4.11Mapa	34
4.12Vuex (Store)	38
4.13Back-end	39
5. Conclusiones y líneas futuras	41
6. Summary and Conclusions	43
7. Presupuesto	45

Índice de Figuras

1.1. Evolución del ocio en Canarias [21]	2
2.1. Patrón MVVM[23]	7
2.2. Desarrollo en Quasar	8
3.1. Internacionalización	12
3.2. Modo oscuro versión móvil	13
3.3. Sección de información en escritorio y móvil	16
3.4. Sección de registro en escritorio y móvil	17
3.5. Sección de login escritorio y en móvil	18
3.6. Sección de eventos en escritorio y móvil	19
3.7. Sección descripción de evento en escritorio y móvil	20
3.8. Sección de añadir evento en escritorio y móvil	21
3.9. Sección de actualizar usuario en escritorio y móvil	22
4.1. Instalación de Linux en Windows	23
4.2. Configuración de Quasar para el proyecto	24
4.3. Ejecución de Quasar en modo desarrollador	25
4.4. Configuración de VS Code	25
4.5. Configuración y ejecución correcta de Travis CI para Node 10, 12 y 14	26
4.6. Configuración de Heroku	26
4.7. Uso de palabras clave	34
4.8. Modelo de datos de Firestore (eventos)	34
4.9. Modelo de datos de Firestore (usuarios)	35
4.10 Modelo de datos de Storage	35
4.11 Seguridad de JWT utilizada en el proyecto	40
7.1. Población de Canarias en 2020 por isla [22]	45
7.2. Presupuesto de Google para la v1 [7]	46
7.3. Presupuesto de Google para la v2 [7]	47

Índice de Tablas

7.1. Presupuesto actual	46
7.2. Presupuesto lanzamiento v1	46
7.3. Presupuesto lanzamiento v2	47

Capítulo 1

Introducción

1.1. Antecedentes y estado actual del ocio

El ocio es una actividad que ha sido necesaria para el ser humano desde sus inicios hasta la actualidad. Es un ejercicio de gran importancia ya que aporta grandes beneficios tales como eliminar o disminuir el estrés, aumentar la energía positiva o mantener la salud a largo plazo. Muchas investigaciones han demostrado que un poco de ocio diario reduce el riesgo de tener enfermedades como diabetes e hipertensión, mejora la salud física, mental y la calidad de vida.

Sin embargo, en los últimos años la falta de dedicar tiempo suficiente al ocio está siendo la causa más importante de tantos casos de ansiedad, teniendo en estos años el mayor crecimiento que haya registrado. Algunos ejemplos de ello serían: depresión, ataques de pánico, agorafobia, déficit de atención, hiperactividad, etc.

El mayor impedimento en nuestra época para “el ejercicio” del ocio saludable es la “escasez de tiempo”, ya que el tiempo disponible es usado (en mayor parte) en el entretenimiento que proponen los medios de comunicación. Debido a esto, el público se vuelve cada vez más pasivo, menos participativo, impidiendo que se concentren en las auténticas necesidades.

En estos últimos años la cantidad de tiempo y dinero invertido ha tenido una tendencia negativa (exceptuando el año 2018) tal como se muestra en la figura 1.1.

En cuanto a la situación actual del sector podemos decir que:

A nivel de las TIC e innovación el sector se encuentra favorecido y cuenta con factores tecnológicos en constante desarrollo que lo benefician tales como:

- La gran cantidad de portales que disponen para reservas.
- La gran conectividad que se dispone: WIFI, fibra, antenas 4G y 5G, etc.
- La aparición e inclusión de las valoraciones online en las webs.
- El uso de mapas electrónicos tales como Google Maps para planificar viajes y eventos.
- La gran cantidad de métodos de pago online, no solo el pago en efectivo.

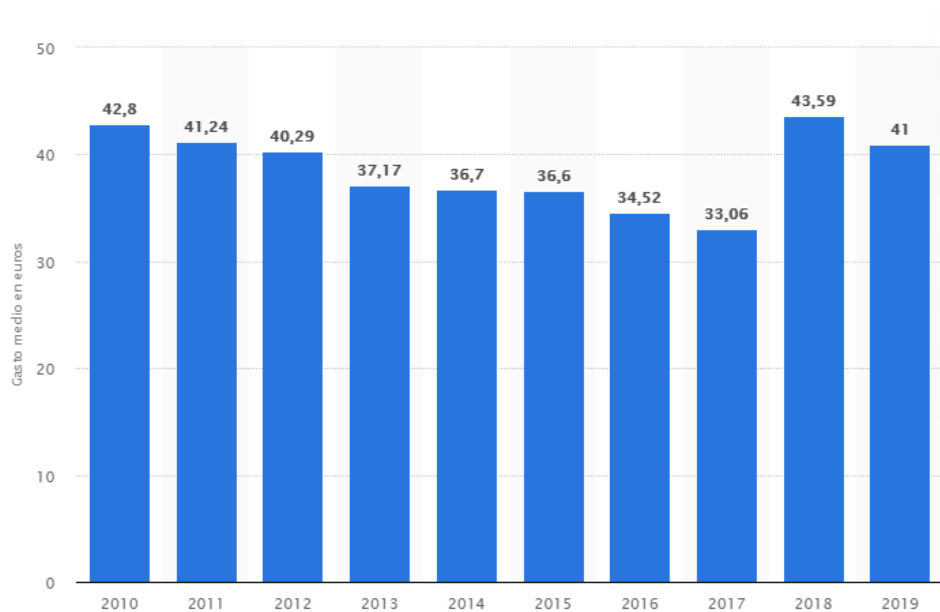


Figura 1.1: Evolución del ocio en Canarias [21]

- Un aumento dentro del ámbito de los servicios de transporte enfocados hacia lo “eco-friendly”, tales como: coches eléctricos, bicicletas y patinetes eléctricos.
- La extensión del móvil y la facilidad que supone tener todo al alcance de la mano.

A nivel de emprendimiento el sector al estar ampliamente relacionado con el turismo es uno de los sectores principales de Canarias, por lo que es uno de los que más inversión tiene y más proyectos se lleva a cabo. Por lo tanto, tenemos cierto grado de seguridad en nuestro proyecto, siempre y cuando se haga un proyecto de calidad y útil para el usuario.

1.2. Idea principal y problemas que resuelve la aplicación web

La aplicación web pretende dar soporte al sector del ocio en la comunidad de Canarias. La idea principal es crear una comunidad al alcance de todo el mundo (permitiendo su uso en cualquier dispositivo de escritorio o móvil).

Una comunidad en la que los usuarios se comparten eventos (localizados) de ocio con sus recomendaciones, consejos, opiniones o experiencias. Se podrán encontrar eventos de ocio gratuitos o de pago con sus respectivos descuentos. Los usuarios deciden cuáles son las ofertas que merecen la pena, votando en positivo o negativo en forma de referencia a la comunidad. También los comentarios ayudan a conocer más en profundidad sobre la calidad del evento o servicio.

La idea de la plataforma ha sido inspirada en varias aplicaciones web en las que el usuario genera contenido de interés para otros usuarios (creando una comunidad) tales con Airbnb[1], HouseTrip[12], Chollometro[2] o Vrbo[25].

¿Por qué es interesante este aplicativo dentro del sector? Los principales problemas que resuelve son:

1. **Existencia de eventos:** Muchas veces debido a que el evento se hace de forma local o en otra isla, se desconoce de la existencia del mismo (por muy bueno que fuese).
2. **Calidad de los eventos:** Gracias a las aportaciones de la gran comunidad que hay detrás se podrá saber si el evento se adecúa a lo que se está buscando, lo bueno y lo malo. Todo esto gracias al sistema de consejos, opiniones o experiencias de la comunidad.
3. **Calendario de eventos:** Los eventos se ordenarán por ubicación (isla) y preferencias de forma que se podrá estar informado de los eventos actuales y todos los próximos por llegar.
4. **“Meetings“:** Conoce gente nueva dentro de la comunidad.
5. **Sedentarismo y beneficios en la salud:** Con toda esta cantidad de estos eventos también se está contribuyendo a que el usuario sea un usuario activo y se mueva de un lado a otro, mejorando así su salud. Además, todos los beneficios del ocio mencionados anteriormente.
6. **Mejora en la gestión de viajes:** Muchas veces debido al desconocimiento de los eventos o actividades que existen e incluso la poca información que existe, es difícil para planificar. De esta forma, se está promoviendo que sea mucho más sencillo este proceso.
7. **Mayor competencia en el transporte:** A largo plazo, el conocimiento aportado por los usuarios y la gran cantidad de eventos hará que las movilizaciones sean mayores. De esta forma a más compañías de transporte (aéreo, terrestre o marítimo) les interesaría competir en el sector dentro de Canarias, por lo que estaríamos hablando de mejores precios en el transporte, tanto de forma externa como de forma interna (entre islas).

1.3. Tareas realizadas durante el desarrollo

Este proyecto consiste en la creación de una aplicación web mediante las tecnologías **MEVN** (MongoDB, Express, Vue.js, Node JS [13]), con un acercamiento a la creación de Single Page Application (SPA) y con el uso del framework **Quasar CLI**[20] el cual está basado en **Vue JS** [26].

Para la conclusión satisfactoria del proyecto se realizaron diferentes actividades, descritas a continuación:

- Se realizó un estudio en profundidad del “framework” Quasar CLI[20] y de Vue.
- Se hizo el **“Setup”** de la infraestructura mediante diferentes herramientas y técnicas de desarrollo:
 - Se utilizó **Pivotal Tracker** [24] para definir correctamente las tareas a realizar y llevar una buena planificación y seguimiento del proyecto.
 - Se creó un repositorio en la herramienta **GitHub** [6] para llevar el proyecto bajo un control de versiones y poder gestionar mejor las diferentes funcionalidades mediante ramas.

- Se incluyó el uso **Travis CI [3]** y **Cypress [4]** para realizar el “testing”. Cypress [4] se utilizó para el testeo e2e de la aplicación web (“framework” “todo en uno” para el testeo de aserciones, Mockups, pruebas y pruebas e2e automáticas).
 - Se configuró el despliegue y una integración continua de la aplicación para funcionar de una forma automatizada.
- Se desarrolló el **Front-end** con diferentes funcionalidades:
 - Se realizó la configuración del router (navegación entre las páginas disponibles).
 - Se realizó el diseño “front-end”.
 - Se incluyó una sección de información para mostrar el funcionamiento de la página y los aspectos que cubre.
 - Se realizó el login y registro de usuarios (incluyendo el extra de la integración con el “login” de Google).
 - Se incluyó una sección para la actualización y borrado de datos del usuario registrado.
 - Se implementó un selector de archivos para móvil y escritorio.
 - Se realizó el diseño y desarrollo de una sección que incluye una lista de eventos dinámica.
 - Se implementó un sistema de votos y ranking para los eventos.
 - Se implementó una barra de búsqueda de eventos.
 - Se añadió e implementó una barra para filtrar los eventos por isla, destacados o novedades.
 - Se incluyó la sección de añadir eventos por parte de los usuarios registrados.
 - Se realizó la integración de la localización de los eventos con la API de Leaflet [17] para Vue (actualmente Maps es de pago). Se implementaron funcionalidades extras que no existían en Leaflet [17] para Vue tales como: un buscador de lugares, mostrar información de la coordenada seleccionada y localizar la isla seleccionada.
 - Se añadió una sección para ver la descripción y comentarios de cada evento.
 - Se añadió un editor para el texto de los comentarios.
 - Se implementó un selector de emojis personalizado.
 - Se implementó un modo oscuro para la web.
 - Se incluyó internacionalización con soporte al inglés a todo.
 - Se realizó la integración con el “back-end”.
 - La aplicación web fue pensada para para un uso repartido entre dispositivos móviles y versión web.
 - Se desarrolló el **Backend**, con las siguientes funcionalidades:
 - Se configuró el proyecto en Firebase.
 - Se configuró Firebase Authentication [10].

- Se creó un modelo de datos en Firestore [8] y Storage [9].
 - Se creó un servidor basado en Node JS [13] para la gestión de las consultas del cliente.
- Se configuró **Heroku [11]** para funcionar con Node JS [13] y como plataforma de **despliegue**.

Capítulo 2

Entorno de desarrollo

2.1. IDE

2.1.1. Visual Studio Code

Es el editor de código fuente (IDE) elegido para el desarrollo del proyecto. Fue desarrollado por Microsoft y tiene grandes ventajas como:

- Es multiplataforma y gratuito.
- Tiene integrado GitHub [6].
- Tiene una gran cantidad de extensiones que facilitan la vida al programador (por ejemplo: ESLint [5] para la detección de errores, Live Share para trabajo en equipo, etc.).
- Tiene la terminal integrada, incluso dando la opción de abrir diferentes terminales en el mismo espacio de trabajo, dando muchísima flexibilidad.
- Incluye la funcionalidad IntelliSense, la cual ofrece el autocompletado de código en muchos lenguajes.

2.2. Front-end

2.2.1. Quasar

Quasar [20] es el framework de desarrollo utilizado para el desarrollo del proyecto. Se trata un “framework” de código abierto basado en Vue JS [26]. Al estar basado en Vue JS [26], hereda todas sus ventajas:

- **Uso del patrón MVVM:** Este patrón de diseño permite pasar datos desde el código JavaScript (modelo) al código HTML (vista) de una manera sencilla, sin utilizar etiquetas HTML desde JavaScript. Esto permite que el desarrollador no tenga que preocuparse por cómo o cuando renderiza un modelo en pantalla (ya que ese trabajo lo realiza la librería) sino en la lógica que gestiona ese modelo. Tenemos 2 tipos de relaciones en este patrón:

- **Relación Modelo-Vista:** Introduce una funcionalidad muy interesante, llamada “data-binding”. Esta funcionalidad permite que la vista se actualice automáticamente, cada vez que se realice algún cambio. Cada vez que una de las variables modifica su valor, Vue ordena que se genere nuevamente la vista con los nuevos valores.
- **Relación Vista-Modelo:** En este caso ocurre lo contrario, el modelo recoge información de la plantilla como datos de formularios o eventos de JavaScript.

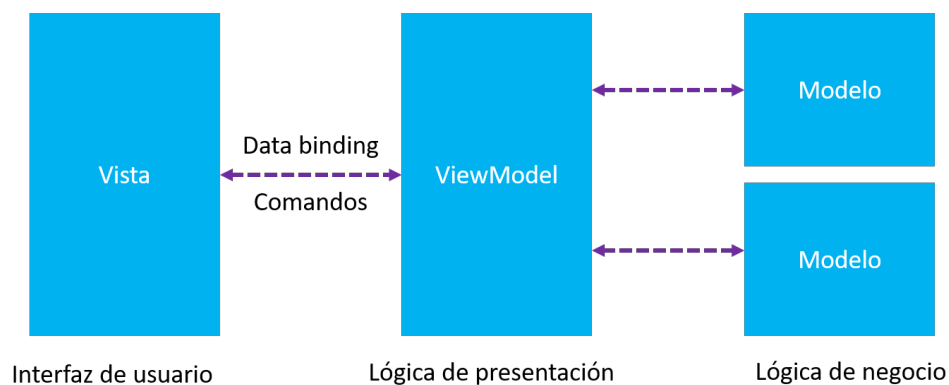


Figura 2.1: Patrón MVVM[23]

- **Uso basado en componentes:** pudiendo emplear las implementaciones por separado y de manera independiente, según la necesidad.
- **Curva de aprendizaje no elevada y progresiva.**
- **Comunidad activa.**
- **Extensión para la detección de errores.**

Entonces, ¿por qué utilizar un “framework” de código abierto adicional y nuevo? ¿Por qué no utilizar Vue JS [26] directamente?

Quasar nos ofrece una gran ventaja adicional. Con la implementación de un solo código podemos desarrollar para diferentes plataformas:

- Aplicaciones web (SPA, PWA, SSR).
- Aplicaciones móviles IOS (se requiere un sistema Mac OS) y Android.
- Aplicaciones de escritorio para Windows, Linux Y Mac OS (se requiere un sistema Mac OS).

Aunque el proyecto va enfocado a una aplicación web de tipo SPA, me parece una funcionalidad muy interesante, que tener en cuenta y en la que invertir tiempo para futuros proyectos. El ahorro de tiempo de desarrollos extras puede ser muy importante.

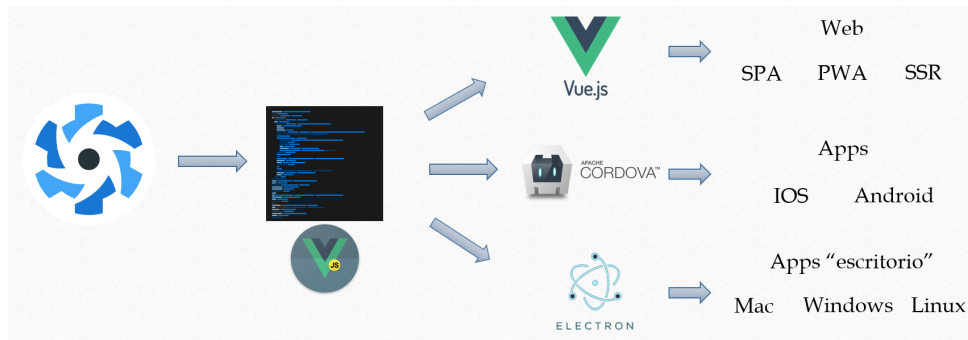


Figura 2.2: Desarrollo en Quasar

2.2.2. Leaflet

Leaflet [16] es la biblioteca para mapeo web utilizada en el proyecto. Leaflet [16] es de código abierto por lo que junto a Google Maps y OpenLayers es una de las bibliotecas más utilizadas teniendo buen rendimiento y simplicidad. De esta forma, también nos encontramos con soporte para Vue como Vue Leaflet [17], para el proyecto. Sin embargo, aunque las características básicas de la versión original están presentes, escasea de funcionalidades extras como un buscador, localización de marcadores y geolocalización (funcionalidades que se han implementado adicionalmente en este proyecto).

2.3. Back-end

2.3.1. Node JS

Node JS [13] es el entorno en tiempo de ejecución utilizado para el back-end y la implementación de un servidor para la aplicación. La versión utilizada fue la 14.2.0 siguiendo con la rama que utiliza las últimas características. Las ventajas de utilizar Node JS [13] son:

- Incluye una buena gestión de paquetes a través de NPM [19].
- Está basado en eventos asíncronos.
- Tiene una gran comunidad de soporte.
- Permite utilizar JavaScript como lenguaje.
- Tiene buen rendimiento y gran escalabilidad.

2.4. Almacenamiento y gestión de datos

Tal como se mencionó al inicio, la plataforma utilizada para el almacenamiento y gestión de datos fue Firebase ya que permite tanto el almacenamiento de datos (mediante Firestore [8]) como el almacenamiento de archivos (mediante Storage [9]) y la gestión de usuarios (mediante Authentication [10]) consiguiendo un sistema unificado sin depender de diferentes herramientas. Hablemos un poco de cada una.

2.4.1. Firebase Authentication

Authentication [10] es la plataforma utilizada para la creación y gestión de usuarios de la aplicación web. Authentication [10] tiene servicios para “back-end” fáciles de usar y bibliotecas de IU ya elaboradas para autenticación. Los métodos de autenticación que proporciona son los siguientes:

- Autenticación basada en correo y contraseña.
- Autenticación basada en otras plataformas como Google, Facebook, GitHub [6], Twitter, etc.
- Autenticación con número de teléfono.
- Autenticación anónima.
- Autenticación personalizada (JWT).

2.4.2. Firebase Firestore

Firestore [8] es la plataforma utilizada para la creación, gestión y el almacenamiento de datos de usuarios en la aplicación web. Firestore [8] es una base de datos NoSQL flexible, escalable y en la nube que permite almacenar y sincronizar datos para el cliente y el servidor. Además, proporciona una serie de características extras como:

- Mantiene los datos sincronizados a través de agentes de escucha en tiempo real.
- Ofrece modo sin conexión para dispositivos móviles y Web.
- Tiene un escalado automático que mantiene el rendimiento.
- Tiene bastante flexibilidad.

2.4.3. Firebase Storage

Storage [9] es la plataforma utilizada para el almacenamiento de archivos. Storage [9] es un servicio de almacenamiento de objetos potente y a la vez simple. Sus principales características son:

- Se mantiene la seguridad de las operaciones de carga y descarga de archivos mediante el uso de tokens.
- Se puede almacenar todo tipo de imágenes, audio, vídeo y otros tipos de contenido generado por el usuario.
- Ofrece gran escalabilidad.

2.5. Despliegue

2.5.1. Heroku

Heroku [11] es la herramienta que se ha utilizado para el despliegue de la aplicación web. Se trata de un servicio de computación en la nube que soporta varios lenguajes de programación y está orientado para desarrolladores, startups y empresas pequeñas. Algunas ventajas que proporciona son:

- Ofrece un plan gratuito lo suficientemente bueno.
- La infraestructura utiliza AWS (Amazon Web Services), lo que proporciona una gran estabilidad y rendimiento.
- Está basado en dynos y es bastante escalable.
- Se puede utilizar con GitHub [6] o con un Git privado propio de Heroku [11], los que permite los despliegues automatizados.
- Es multiplataforma y está orientado a desarrolladores, por lo que ofrece buena retroalimentación al momento del despliegue.

2.6. Control y pruebas

2.6.1. ESLint

ESLint [5] es una herramienta de análisis de código que permite identificar problemas que existan en código JavaScript. Esta herramienta permite ahorrar mucho tiempo ya que detecta automáticamente los errores que podrían provocar problemas de compilación o futuros bugs en nuestro desarrollo y permite solucionar automáticamente los errores de sintaxis.

En este caso fue configurado para garantizar la calidad del código (como la detección de variables o funciones que no se estén utilizando) y junto a Vetur [28] (extensión de VS Code [18]) para garantizar el formato correcto del código (espaciado, saltos de línea, etc.). De esta forma, cuando encuentra algún error nos advierte de ello a través de distintos mensajes.

2.6.2. Pivotal Tracker

Pivotal Tracker [24] es una herramienta que permite la gestión y colaboración ágil de proyectos. Pivotal Tracker [24] permite desglosar los requisitos del proyecto en pequeñas tareas concretas y organizarlas en sección compartida en orden de prioridad, utilizando una interfaz de arrastrar y soltar. Esta característica es bastante útil ya que te permite organizarte y dividir el proyecto en “mini proyectos”. Sin embargo, al tratarse de un proyecto de trabajo individual resultó poco útil para algunos aspectos concretos.

2.6.3. GitHub

GitHub [6] es la plataforma utilizada para el almacenamiento del código del proyecto. GitHub [6] es una plataforma gratuita para el desarrollo colaborativo que permite alojar proyectos y hacer un seguimiento de los mismo gracias al control de versiones Git.

Actualmente es una de las comunidades mas grande en la que los usuarios aprenden, comparten y trabajan juntos para crear software, con más de 100 millones de repositorios.

El proyecto utilizó un total de 8 ramas: master, index, login, signup, eventos, internacionalización, test y back-end. El repositorio utilizado es el siguiente: <https://github.com/ULL-TFGyMs-vblanco/TFG-1920-IvanBorges-CanaryGo>.

2.6.4. Travis CI

Travis CI [3] es el servicio utilizado para la integración continua del proyecto. Travis CI [3] permite probar proyectos de software que estén alojados en GitHub [6]. Además, tiene una serie de características que hace que destaque:

- Integración sencilla con los repositorios públicos y privados de GitHub [6].
- Se puede configurar para diferentes ejecuciones (por ejemplo, para probar diferentes versiones de Node JS [13]). Esto es una gran ventaja porque permite probar diferentes configuraciones sin deber tenerlas instaladas localmente.
- Soporta bastantes lenguajes y su número aumenta constantemente (30 actualmente).
- Se puede configurar de forma que si la prueba falla no se despliegue en Heroku [11] (como es el caso de este proyecto).

2.6.5. Cypress

Cypress [4] es otra herramienta utilizada para las pruebas e2e del proyecto. En este caso, Cypress [4] es un framework “todo en uno” para el testeo de aserciones, Mockups, pruebas (en general) y pruebas e2e automáticas. Al permitir todo tipo de pruebas se puede utilizar tanto para pruebas e2e o para alguna prueba adicional de tipo unitaria, por ejemplo, sin necesidad de tener distintas herramientas. Ofrece muchas ventajas adicionales como:

- Es fácil de configurar, sin dependencias externas.
- Las pruebas son fáciles de escribir y se ejecutan en tiempo real en un navegador directamente.
- Hace capturas de pantalla de cada prueba por lo que se puede ver la información de qué es lo que se esperaba y qué fue lo que sucedió.

Capítulo 3

Descripción de la aplicación (Canary Go)

A continuación, se procederá a explicar las diferentes secciones y funcionalidades que tiene la aplicación web Canary Go.

3.1. Disponibilidad

Aunque la aplicación está orientada a ser gratuita de forma mundial, estableció la necesidad de que los usuarios estuvieran registrados y con sesión iniciada para su utilización. De esta forma, se puede controlar mejor que la información aportada por la comunidad sea siempre verídica y castigar a los usuarios que no lo cumplen, ya que cada publicación queda registrada.

3.2. Internacionalización

Aunque el proyecto va dirigido inicialmente a la comunidad de Canarias, se le ha dado soporte al inglés a toda la web. Cambiar el idioma es muy sencillo ya que está integrado en un selector en la parte superior que siempre está visible en la web, por lo que se puede cambiar en cualquier momento y en cualquier dispositivo.



Figura 3.1: Internacionalización

3.3. Modo oscuro

Se ha implementado un modo oscuro en toda la web, para entornos con poca luz en los que se desea una menor cantidad de blancos o para reducir la fatiga visual en casos de un uso elevado de la aplicación web. Está implementado de tal modo que detecta la preferencia del usuario y si en el dispositivo (ya sea de escritorio o móvil) tiene el sistema establecido en modo oscuro, la web cargará automáticamente en modo oscuro. Sin embargo, siempre estará la opción de cambiarlo manualmente.

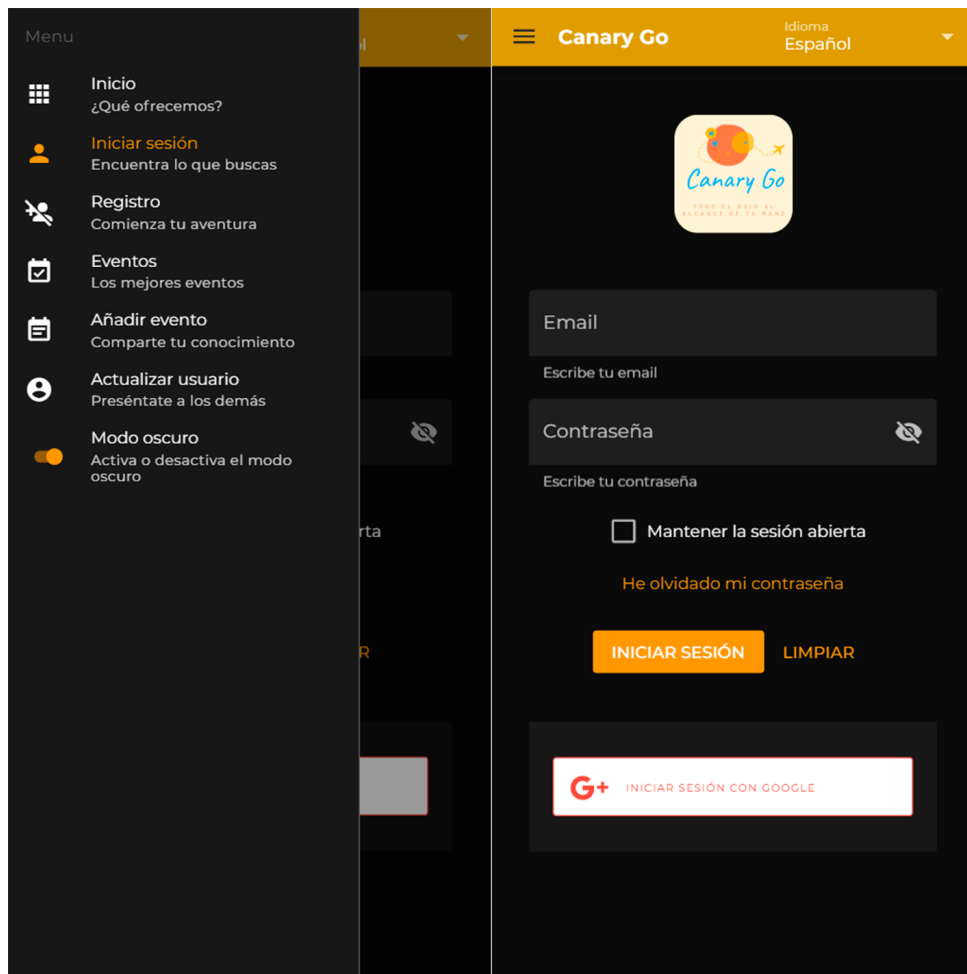


Figura 3.2: Modo oscuro versión móvil

3.4. Sección de información de la web

En esta sección el usuario podrá informarse de cuales son los objetivos principales que la aplicación web cumple. Esta sección está dividida en 4 apartados: eventos, calendario, comunidad y gestión de viajes. Se recomienda visitar la web para ver las diferentes secciones.

Enlace a la aplicación web: <https://canarygo.herokuapp.com/home>

3.5. Sección de registro

Esta sección está dedicada para el registro de nuevos usuarios que no quieran utilizar la plataforma de Google y prefieran un método más tradicional. Lo datos requeridos son los básicos: nombre, usuario público, género, fecha de nacimiento, email y contraseña. Una vez registrado el usuario deberá confirmar la activación de la cuenta en el mensaje que le llegará al correo electrónico. Al registrarse se asigna un avatar predeterminado de perfil en función del género.

3.6. Sección de inicio de sesión

Esta sección está dedicada para el inicio de sesión de los usuarios que se hayan registrado anteriormente. Además, permite la opción de iniciar sesión directamente a través de Google. En caso de usar credenciales erróneas, el sistema es capaz de reconocer si el fallo está en el correo o en la contraseña.

3.7. Sección de eventos

Una vez que el usuario haya iniciado sesión podrá acceder a la lista de eventos que están por llegar. Además de los eventos disponibles, en esta sección se muestra una bienvenida junto al nombre del usuario “logueado”, la foto de perfil del usuario (al pulsar encima se muestra la posibilidad de cambiar el perfil o cerrar sesión) y un filtro para los eventos. Se permite filtrar los eventos por ubicación (isla) y cambiar el “ranking” según lo que se busque: los más votados o los más nuevos. Por último, también se puede filtrar por texto mediante la barra de búsqueda situada en la parte superior de la sección.

Además, cada evento dispone de un sistema de “ranking” mediante votos. Una vez que el usuario haya votado ya sea en positivo o en negativo, el sistema registra el voto y se bloquea la opción de votar otra vez o anular el voto. De esta forma, se busca aumentar más aún la veracidad de las votaciones. El usuario podrá ver siempre cuál fue la opción que votó y la cantidad de votos positivos que tiene cada evento.

Para cada evento el usuario podrá ver rápidamente quién lo publicó, los votos que tiene (teniendo la opción de votar), la fecha en la que empieza, el nombre del evento, la isla en la que se realiza, el precio que tiene, la cantidad de comentarios que tiene, un enlace a Google Maps de la ubicación indicada y un enlace a la descripción del evento.

3.8. Sección de descripción del evento

En esta sección, el usuario podrá ver una descripción más detallada de cada evento. Los datos que se muestran son: el nombre del evento, la isla en la que se realiza, el precio que tiene, la cantidad de votos que tiene y el voto realizado por el usuario, una pequeña descripción del evento añadida por el usuario que lo ha publicado, el código de descuento (en caso de ser aplicable), las fechas de inicio y fin del evento, un mapa básico para ver rápidamente la ubicación del evento (junto con las coordenadas), el avatar del usuario que publicó el evento, un enlace a Google Maps de la ubicación indicada, un enlace para el registro en la web del evento (en caso que sea necesario), la cantidad de comentarios que tiene y una sección de comentarios.

Esta sección consta de un editor WYSIWYG (What You See Is What You Get) y un selector de emojis que se han implementado. Este editor permite seleccionar los diferentes tipos de alineado y el uso de negrita, subrayado, tachado y cursiva. Este editor introdujo la dificultad de detectar la posición del cursor ya que no funciona como un “textarea” normal y no es una tarea sencilla. Por otro lado, el selector de emojis permite filtrar por categorías, buscar un emoji en concreto y cambiar el color. Ambos con su versión en inglés.

3.9. Sección de añadir evento

En esta sección el usuario que haya iniciado sesión podrá añadir eventos a la comunidad. Los datos que se requieren son: foto de presentación (mediante un selector de archivos compatible con dispositivos de escritorio y móviles), nombre del evento, localización del evento (para ello se dispone del mapa implementado con Leaflet [17] y una vez seleccionada una ubicación se añadirán las coordenadas automáticamente), fecha de inicio del evento, fecha de finalización del evento, precio, código de descuento (opcionalmente si lo hubiera), el enlace a la web para el registro (en caso de ser necesario) y una pequeña descripción de este. El evento será publicado con el "avatar" y nombre de usuario del redactor.

3.10. Sección de actualizar el perfil de usuario

En esta sección se mostrarán los datos actuales del usuario incluyendo la foto de perfil. Los datos que tendrá disponible para actualizar cada usuario variarán en función del método de registro que utilizara el usuario. Si fue utilizado el registro tradicional, se podrán cambiar el nombre, el usuario, el género, la fecha de nacimiento, el email y la contraseña. Si por el contrario se utilizó Google como método de registro, no estarán disponibles las opciones de cambiar contraseña y cambiar email, ya que dependen de la cuenta de Google. Además, también se incluye la opción de borrar por completo la cuenta para ambos tipos de usuarios.

- Menu
- Inicio
¿Qué ofrecemos?
- Iniciar sesión
Encuentra lo que buscas
- Registro
Comienza tu aventura
- Eventos
Los mejores eventos
- Añadir evento
Comparte tu conocimiento
- Actualizar usuario
Presentate a los demás
- Modo oscuro
Activa o desactiva el modo oscuro

Canary Go Idioma Español

Encuentra los eventos que tanto buscas

MÁS INFORMACIÓN

NO MÁS EVENTOS OLVIDADOS

Muchas veces debido a que el evento se hace de forma local o en otra isla, se desconoce de la existencia del mismo (por muy bueno que sea). Olvídate de ese momento con CanaryGo.

OPINIONES DE GENTE COMO TÚ

Gracias a las aportaciones de la gran comunidad podrás saber si el evento se adecúa a lo que se está buscando, lo bueno y lo malo.

Canary Go Idioma Español

Encuentra los eventos que tanto buscas

MÁS INFORMACIÓN

NO MAS EVENTOS OLVIDADOS

Muchas veces debido a que el evento se hace de forma local o en otra isla, se desconoce de la existencia del mismo (por muy bueno que sea). Olvídate de ese momento con CanaryGo.

OPINIONES DE GENTE COMO TÚ

Gracias a las aportaciones de la gran comunidad podrás saber si el evento se adecua a lo que se está buscando, lo bueno y lo malo.

Figura 3.3: Sección de información en escritorio y móvil

- Menu
- Inicio
¿Qué ofrecemos?
 - Iniciar sesión
Encuentra lo que buscas
 - Registro
Comienza tu aventura
 - Eventos
Los mejores eventos
 - Añadir evento
Comparte tu conocimiento
 - Actualizar usuario
Presentate a los demás
 - Modo oscuro
Activa o desactiva el modo oscuro


The desktop registration form for Canary Go features a yellow header with the logo and language selection. The form includes fields for Name, Username, Gender, Birth Date, Email, Password, and Password Repeat. A checkbox for terms and conditions is located below the password fields, with a 'REGISTRO' button and a 'LIMPIAR' button at the bottom.

The mobile registration form for Canary Go is a vertical version of the desktop form, featuring the same fields and layout adapted for a smaller screen. It includes a yellow header with the logo and language selection, and a 'REGISTRO' button at the bottom.

Figura 3.4: Sección de registro en escritorio y móvil

- Menu
- Inicio**
¿Qué ofrecemos?
 - Iniciar sesión**
Encuentra lo que buscas
 - Registro**
Comienza tu aventura
 - Eventos**
Los mejores eventos
 - Añadir evento**
Comparte tu conocimiento
 - Actualizar usuario**
Preséntate a los demás
 - Modo oscuro**
Activa o desactiva el modo oscuro

Canary Go Idioma Español




Email
Escribe tu email


Contraseña
Escribe tu contraseña

Mantener la sesión abierta
He olvidado mi contraseña

INICIAR SESIÓN LIMPIAR

 INICIAR SESIÓN CON GOOGLE

Canary Go Idioma Español



Email
Escribe tu email

Contraseña
Escribe tu contraseña

Mantener la sesión abierta
He olvidado mi contraseña

INICIAR SESIÓN LIMPIAR


 INICIAR SESIÓN CON GOOGLE

Figura 3.5: Sección de login escritorio y en móvil

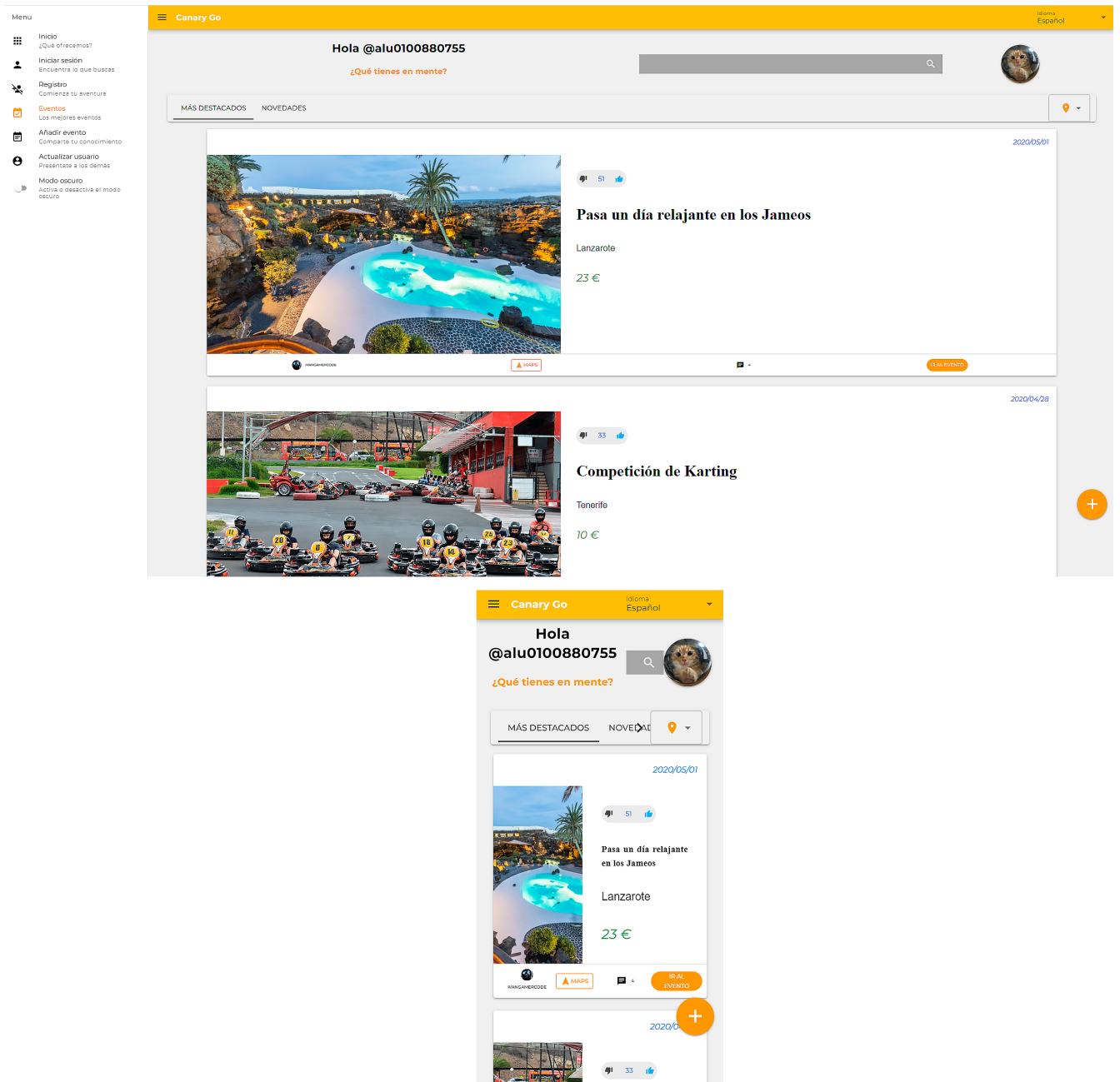


Figura 3.6: Sección de eventos en escritorio y móvil

- Menu
- Inicio
¿Qué ofrecemos?
- Iniciar sesión
Encuentra lo que buscas
- Registro
Comienza tu aventura
- Eventos
Los mejores eventos
- Añadir evento
Comparte tu conocimiento
- Actualizar usuario
Presentate a los demás
- Modo oscuro
Activa o desactiva el modo oscuro

The desktop interface features a yellow header with the 'Canary Go' logo and a language dropdown set to 'Español'. Below the header is a map showing a coastal area with a blue location pin. A navigation bar contains the user profile 'IVANGAMERCODE', a 'MAPS' button, a notification icon with '4', and an 'IR AL REGISTRO' button. The main content area is titled 'Comentarios' and includes a 'Publica un comentario' section with a rich text editor. Below this, several user comments are displayed, such as 'A pasar un buen día se ha dicho' by IvanGamerCode and 'Ya tengo excusa para ir a visitar la isla' by @alu0100880755.

The mobile interface shows an event listing for 'Pasa un día relajante en los Jameos' in Lanzarote, priced at 23 €. The event details include a description: 'Disfruta de espacios y rincones impactantes como un lago natural con aguas transparentes y una bóveda de cañón, que recuerda a la nave de una catedral, con un agujero en la parte superior por donde entra un impresionante haz de luz de forma central.' It also lists the discount code 'LNZ2020' and the event duration from 2020/05/01 to 2020/05/01. The interface includes a 'gu' icon, a '51' count, and a thumbs-up icon.

Figura 3.7: Sección descripción de evento en escritorio y móvil

- Menu
- Inicio
¿Que ofrecemos?
 - Iniciar sesión
Encuentra lo que buscas
 - Registro
Comienza tu aventura
 - Eventos
Los mejores eventos
 - Añadir evento
Comparte tu conocimiento
 - Actualizar usuario
Presentate a los demás
 - Modo oscuro
Activa o desactiva el modo oscuro

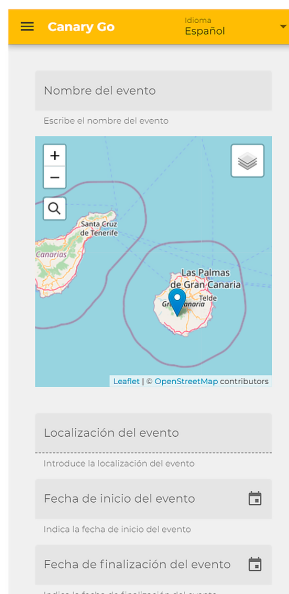
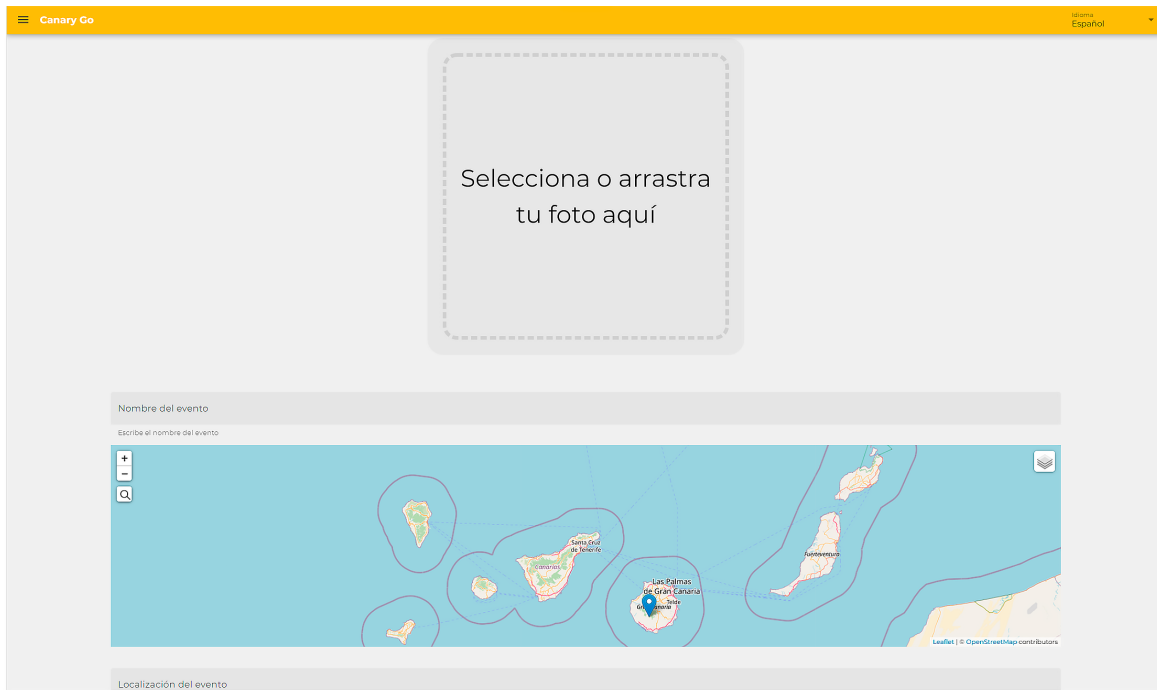


Figura 3.8: Sección de añadir evento en escritorio y móvil

- Inicio
¿Que ofrecemos?
- Iniciar sesión
Encuentra lo que buscas
- Registro
Comienza tu aventura
- Eventos
Los mejores eventos
- Añadir evento
Comparte tu conocimiento
- Actualizar usuario
Presentate a los demás
- Modo oscuro
Activa o desactiva el modo oscuro

Canary Go Idioma Español

Selecciona o arrastra tu foto aquí

Nombre
Iván Clez
Escribe tu nombre

Usuario
@alu0100880755
Elige tu usuario

Género
Masculino
Selecciona tu género

Fecha de nacimiento
1996/07/01
Indica tu fecha de nacimiento

Email
alu0100880755@ull.edu.es
Escribe tu email

Contraseña
Escribe tu contraseña

Canary Go Idioma Español

Nombre
Iván Clez
Escribe tu nombre

Usuario
@alu0100880755
Elige tu usuario

Género
Masculino
Selecciona tu género

Fecha de nacimiento
1996/07/01
Indica tu fecha de nacimiento

Email
alu0100880755@ull.edu.es
Escribe tu email

Contraseña
Escribe tu contraseña

[Cambiar la contraseña](#)

[ACTUALIZAR](#) [LIMPIAR](#)

[Borrar usuario](#)

Figura 3.9: Sección de actualizar usuario en escritorio y móvil

Capítulo 4

Desarrollo

A continuación, se mostrarán brevemente algunos de los pasos seguidos más destacados para el desarrollo del proyecto. Se recomienda visualizar todo el código en el repositorio:

<https://github.com/ULL-TFGyMs-vblanco/TFG-1920-IvanBorges-CanaryGo>

4.1. Instalación y configuración general

Como cualquier proyecto, al inicio se tuvieron que instalar todas las herramientas necesarias para el desarrollo de este. El desarrollo del proyecto se llevó a cabo en un equipo con el sistema operativo Windows. Sin embargo, recientemente Windows incluyó la característica de utilizar Linux como subsistema en las opciones de desarrollador, por lo que se obtienen los beneficios de ambos. De esta forma algunas instalaciones se realizaron bajo Windows y otras para Linux. Veámoslo.

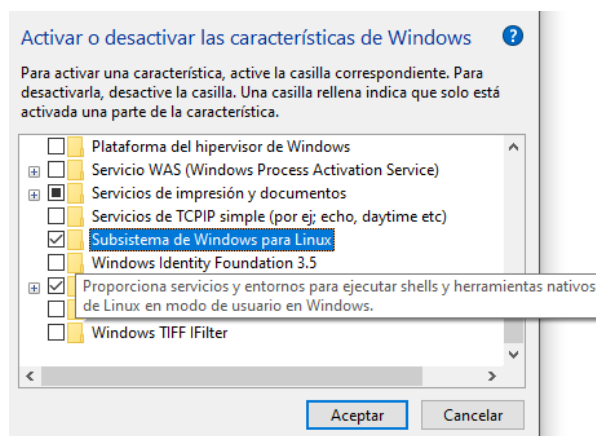


Figura 4.1: Instalación de Linux en Windows

El primer paso al comenzar fue la creación de las cuentas correspondientes a cada herramienta, en caso de no tenerla creada. A continuación, se procedió con la creación del repositorio en GitHub [6], en el que posteriormente se incluyeron las ramas y el código. Después, se comenzó a instalar NPM [19] y Node JS [13] (en este caso en la versión que incluye las últimas características) bajo Linux, por lo que se utilizó la terminal. Para la

utilización de Quasar [20] no se recomiendan las versiones impares de Node JS [13]. En este caso se usó la 14.

```
1 sudo apt install npm
2 sudo npm install -g n
3 sudo n latest
4 sudo apt-get update
5 sudo apt-get upgrade
```

Listing 4.1: Instalación de NPM y Node Js

Seguidamente se procedió a instalar Quasar [20]. Se utilizó NPM [19] para su instalación.

```
1 npm install -g @quasar/cli
```

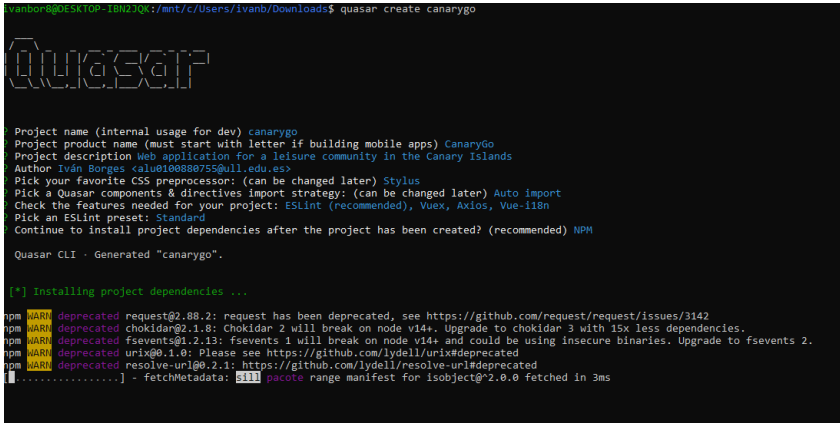
Listing 4.2: Instalación de Quasar

A continuación, se comenzó con la configuración del mismo.

```
1 quasar create canarygo
```

Listing 4.3: Configuración de Quasar

A continuación, el instalador pidió una serie de características para tener en cuenta para el proyecto. En este caso, en la configuración se incluyeron ESLint [5] (seleccionando el “preset” Standard), Vuex (para permitir la gestión del estado en Vue JS [26]), Axios (para permitir la comunicación con el servidor) y Vue-i18n (para la internacionalización del proyecto). Además, se configuró NPM [19] como el gestor de dependencias por defecto.



```
ivanb@DESKTOP-1BN23Q:~/mnt/C/Users/ivanb/Downloads$ quasar create canarygo
QUASAR

Project name (internal usage for dev) canarygo
Project product name (must start with letter if building mobile apps) CanaryGo
Project description web application for a leisure community in the Canary Islands
Author Iván Borges <alu0100880755@ull.edu.es>
Pick your favorite CSS preprocessor: (can be changed later) Stylus
Pick a Quasar components & directives import strategy: (can be changed later) Auto import
Check the features needed for your project: ESLint (recommended), Vuex, Axios, Vue-i18n
Pick an ESLint preset: Standard
Continue to install project dependencies after the project has been created? (recommended) NPM
Quasar CLI - Generated "canarygo".

[*] Installing project dependencies ...
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated chokidar@2.1.8: Chokidar 2 will break on node v14+. Upgrade to chokidar 3 with 15x less dependencies.
npm WARN deprecated fsevents@1.2.13: fsevents 1 will break on node v14+ and could be using insecure binaries. Upgrade to fsevents 2.
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
[.....] - fetchMetadata: 511 packets range manifest for isobject@2.0.0 fetched in 3ms
```

Figura 4.2: Configuración de Quasar para el proyecto

Una vez instalado, para poder ejecutarlo bastaba con el siguiente comando. Una vez terminaba de compilar, abría automáticamente el navegador por defecto en la dirección “localhost:8080”, con el código de ejemplo funcionando.

```
1 quasar dev
```

Listing 4.4: Ejecutar Quasar en modo desarrollador (por defecto SPA)

```
DONE Compiled successfully in 10041ms

N App dir..... /mnt/c/Users/ivanb/Downloads/canarygo
  App URL..... http://localhost:8080
  Dev mode..... spa
  Pkg quasar..... v1.12.8
  Pkg @quasar/app... v2.0.1
  Transpiled JS..... yes (Babel)

[wds] Project is running at http://0.0.0.0:8080/
[wds] webpack output is served from
App · Opening default browser at http://localhost:8080
```

Figura 4.3: Ejecución de Quasar en modo desarrollador

4.2. Visual Studio Code

Para instalar esta herramienta es muy sencillo ya que se utilizó el instalador disponible para Windows. Posteriormente se procedió a configurar las extensiones necesarias y configuración interna del formateador para que el código se mostrara correctamente y se mostraran los errores que ESLint [5] detectara. La línea de autoguardado fue activada a gusto del programador, pero muy recomendable activarla para evitar posibles pérdidas posteriormente.

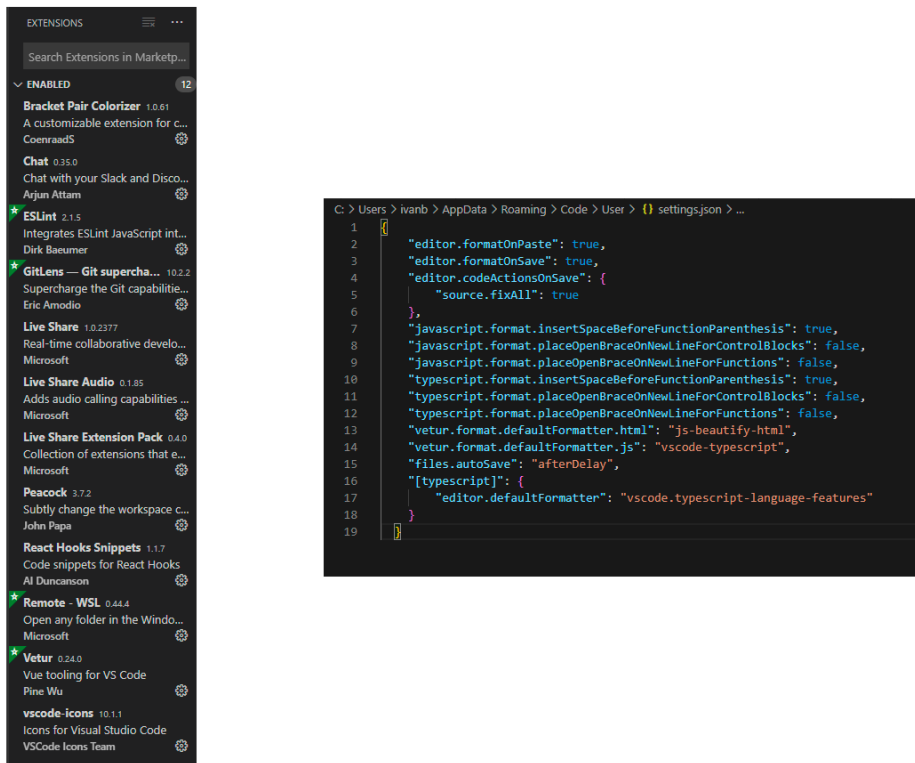


Figura 4.4: Configuración de VS Code

4.3. Travis CI

Para configurar la integración continua mediante Travis CI [3], esta plataforma requería de una pequeña configuración en función del lenguaje que se esté utilizando. En este caso, se probó que funcionara para las versiones pares a partir de las 10. Para ello, se creó el fichero “.travis.yml” con la siguiente configuración.

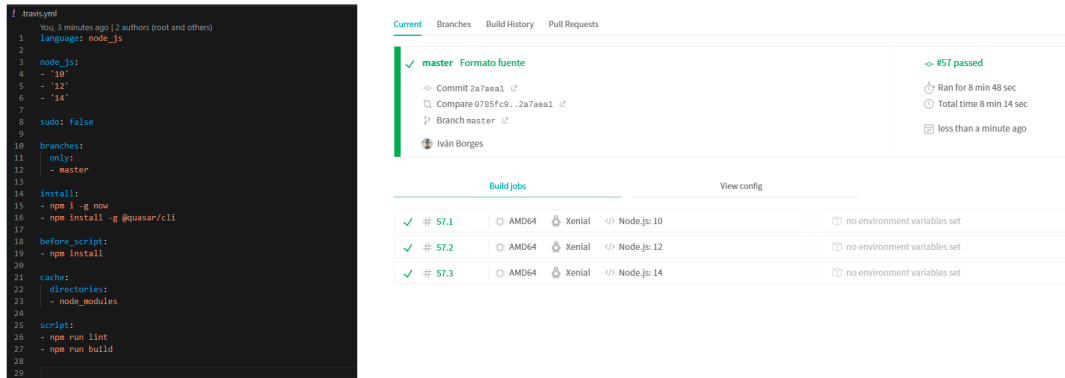


Figura 4.5: Configuración y ejecución correcta de Travis CI para Node 10, 12 y 14

4.4. Heroku

Para configurar Heroku [11] de forma que desplegara los cambios automáticamente, se vinculó Heroku [11] con la cuenta de GitHub [6], el repositorio utilizado y la rama deseada. Además, fue necesario añadir la carpeta “dist” al repositorio y definir un dyno para el servidor para que Heroku [11] funcionara correctamente. Para ello, había que eliminar la carpeta “dist” del archivo “.gitignore”.

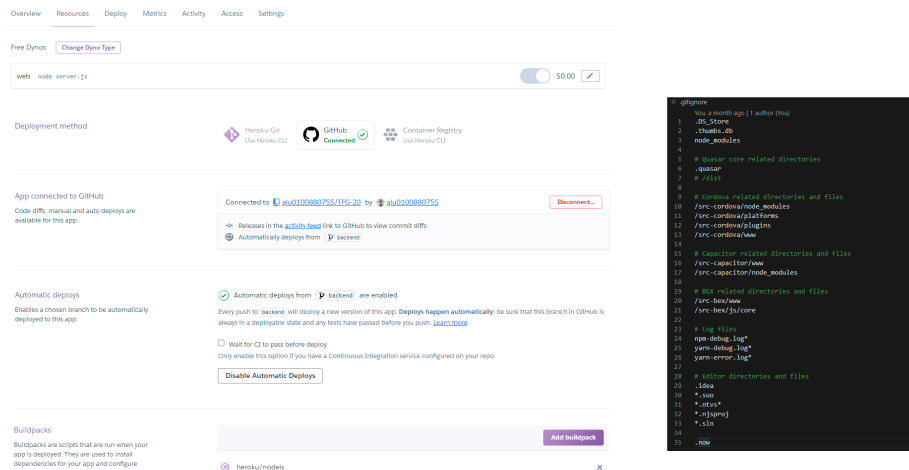


Figura 4.6: Configuración de Heroku

4.5. Vue-Router

Una vez configuradas las herramientas para que funcionaran de manera automatizada, se comenzó con la configuración de la navegación del “front-end”. Para ello, se utilizó Vue-Router[14]. Para que esto funcionara había que configurar el fichero routes.js (ubicado en **canarygo/src/router/routes.js**).

```
1 const routes = [  
2   {  
3     path: '/',  
4     exact: true,  
5     component: () => import('layouts/MainLayout.vue'),  
6     children: [  
7       { path: '/', redirect: '/home' },  
8       { path: '/home', component: () => import('pages/Index.vue') },  
9       { path: '/login', component: () => import('pages/Login.vue') },  
10      { path: '/signup', component: () => import('pages/SignUp.vue') },  
11      { path: '/new', component: () => import('pages/NuevoEvento.vue') },  
12      { path: '/update', component: () => import('pages/ActualizarUsuario.vue') },  
13      { path: '/reset', component: () => import('pages/Restablecer.vue') },  
14    ],  
15    {  
16      path: '/events',  
17      component: () => import('pages/Eventos.vue')  
18    },  
19    { path: '/error', component: () => import('pages/Error404.vue') },  
20    { path: ':id', name: 'evento_detallado', component: () => import('pages/  
21      Descripcionevento.vue'), prop: true }  
22  ]  
23 }  
24 ]  
25 if (process.env.MODE !== 'ssr') {  
26   routes.push({  
27     path: '*',  
28     component: () => import('pages/Error404.vue')  
29   })  
30 }  
31 export default routes
```

Listing 4.5: Canary Go routes

4.6. Sección de Información

Una vez implementados todos los elementos básicos, se comenzó con la parte informativa de la web. Muchos de los componentes predeterminados de Quasar[20] facilitaron el trabajo para crear el efecto Parallax.

```
1 .....  
2 <div class="total1">  
3   <!-- SECCION1 -->  
4   <q-parallax  
5     id="seccionhtml"
```

```

6      :height="1050"
7      :speed="1"
8      src="../../../assets/images/index/beach1853939.jpg"
9      >
10     
14     <br />
15     <div id="seccionh3-index" class="text-bold text-center" >{{$(
introduccion1)}}</div>
16     <br />
17     <!-- ACCIONES -->
18     <q-card-actions>
19       <q-btn
20         color="black"
21         :label="$t('more_info')"
22         round
23         flat
24         dense
25         id="expanded1"
26         :icon="expanded ? 'keyboard_arrow_up' : 'keyboard_arrow_down'"
27         @click="onClick(1)"
28       />
29     </q-card-actions>
30     <!-- /ACCIONES -->
31     <q-slide-transition :duration="2000">
32       <div v-show="expanded">
33         <Seccioninfo
34           v-for="seccion in seccion1"
35           :key="seccion.titulo"
36           v-bind="seccion"
37           v-model="$i18n.locale"
38         />
39       </div>
40     </q-slide-transition>
41   </q-parallax>
42 </div>
43 .....
44 < script>
45 import Seccioninfo from '../components/Seccioninfo.vue'
46
47 export default {
48   name: 'Index',
49   components: {
50     Seccioninfo
51   },
52   data: function () {
53     return {
54       introduccion1: 'introduction1',
55       seccion1: [
56         {
57           icono: 'event',
58           titulo: 'title1',
59           descripcion: 'description1'
60         },
61         {
62           icono: 'people',

```

```

63     titulo: 'title1_1',
64     descripcion: 'description1_1'
65   }
66 ],
67 .....

```

Listing 4.6: Porciones de código de la página Index

4.7. Login y Registro

A continuación, se continuó trabajando en los formularios de registro e inicio de sesión. Siempre incluyendo la internacionalización y las comprobaciones de reglas en los campos requeridos.

```

1   .....
2   <form
3     id="Form"
4     @submit.prevent.stop="onSubmit"
5     @reset.prevent.stop="onReset"
6     class="q-gutter-md"
7   >
8     <q-input
9       ref="email"
10      class="Email"
11      filled
12      v-model="email"
13      :label="$t('email') "
14      :hint="$t('email_hint')"
15      type="email"
16      lazy-rules
17      :rules="[ val => val && val.length > 0 && val.indexOf('@') >= 0 || $t('
email_fail')]"
18    />
19    <q-input
20      ref="contrasena"
21      class="Contrasena"
22      filled
23      v-model="contrasena"
24      :type="isPwd ? 'password' : 'text'"
25      :label="$t('password')"
26      :hint="$t('password_hint')"
27      lazy-rules
28      :rules="[
29        val => val !== null && val !== '' || $t('password_fail'),
30      ]"
31    >
32      <template v-slot:append>
33        <q-icon
34          :name="isPwd ? 'visibility_off' : 'visibility'"
35          class="cursor-pointer"
36          @click="isPwd = !isPwd"
37        />
38      </template>
39    </q-input>
40    <div class="text-center">

```



```

11     preCaretRange.selectNodeContents(element)
12     preCaretRange.setEnd(range.endContainer, range.endOffset)
13     caretOffset = preCaretRange.toString().length
14 }
15 } else if ((sel = doc.selection) && sel.type !== 'Control') {
16     var textRange = sel.createRange()
17     var preCaretTextRange = doc.body.createTextRange()
18     preCaretTextRange.moveToElementText(element)
19     preCaretTextRange.setEndPoint('EndToEnd', textRange)
20     caretOffset = preCaretTextRange.text.length
21 }
22 return caretOffset - this.diferencia_cursor
23 }
24 .....
25     html2text (html) {
26         // Transformamos a texto simple
27         html = html.replace(new RegExp('&nbsp;', 'g'), ' ')
28         html = html.replace(new RegExp('<div>', 'g'), '\n')
29         html = html.replace(new RegExp('</div>', 'g'), '')
30         html = html.replace(new RegExp('<div/>', 'g'), '\n')
31         html = html.replace(new RegExp('<br>', 'g'), '\n')
32
33         // Contamos los espacios de lineas que afectan al cursor y los añadimos (1
espacio equivale a 6 en HTML text)
34         let contador = 0
35         let htmlextra = false
36         let i = 0
37         let final = this.cursor
38
39         while (i < final) {
40             if (html[i] === ' ') {
41                 contador = contador + 1
42             } else if ((html[i] === '<') || (html[i] === '>')) {
43                 final++
44                 htmlextra = !htmlextra
45             } else if (htmlextra) {
46                 final++
47             }
48             i += 1
49         }
50
51         // console.log('espacios detectados -> ', contador)
52         this.cursor += contador * 6
53
54         // Calculamos longitud de solo texto
55         this.cantidad_letras = html.length
56
57         return html
58     }

```

Listing 4.8: Porciones de código del editor WYSIWYG

```

1 <template>
2 .....
3   <!-- Emojis -->
4     <div id="caretPos"></div>
5     <picker
6       class="selectoremojis"
7       set='messenger'
8       v-show="mostrar_emoji"
9       v-model="comentario"
10      :defaultSkin=3
11      :title=this.title
12      emoji="point_up"
13      @select="SeleccionarEmoji"
14      :i18n="{
15        search: this.search_results,
16        notfound: this.notfound,
17        categories: {
18          search: this.search,
19          recent: this.recent,
20          people: this.people,
21          nature: this.nature,
22          foods: this.foods,
23          activity: this.activity,
24          places: this.places,
25          objects: this.objects,
26          symbols: this.symbols,
27          flags: this.flags
28        } }"
29      />
30   <!-- -->
31 .....
32 </template>
33 .....
34 SeleccionarEmoji (emoji) {
35   // Transformar a texto plano para insertar
36   this.comentario = this.html2text(this.comentario)
37
38   // Devolver a rich html
39   this.comentario = this.text2HTML(this.comentario)
40   const cursorPosition = this.cursor
41
42   const inicio = this.comentario.substr(0, cursorPosition)
43   const fin = this.comentario.substring(cursorPosition, this.comentario.length)
44
45   const text = inicio + emoji.native + fin
46   this.comentario = text
47 }
48 .....
49   PublicarComentario () {
50     // console.log('DATOS', this.id)
51     if (this.comentario !== '') {
52       axios({
53         method: 'put',
54         url: 'https://canarygo.herokuapp.com/eventos',
55         data: {
56           tipo: 'Comentario',
57           nombre: this.$store.state.store.datosUsuario.displayName,

```

```

58     avatar: this.$store.state.store.datosUsuario.photoURL,
59     hora: new Date().getHours() + ':' + String(new Date().getMinutes()).
padStart(2, '0'),
60     dia: new Date().getDate() + '/' + (new Date().getMonth() + 1) + '/' +
new Date().getFullYear(),
61     texto: this.comentario,
62     id: this.id,
63     token: this.$store.state.store.token
64   }
65 })
66 .....

```

Listing 4.9: Porciones de código del selector de emojis

4.9. Internacionalización y selector de idioma

Otra parte muy importante fue la correcta configuración de la internacionalización. Para ello, se requirió de la implementación de un selector de idioma y la utilización de palabras claves para cualquier texto utilizado. De esta forma, cada palabra clave tenía su correspondiente traducción en cada idioma.

```

1 <template>
2   <div class="language">
3     <q-select
4       v-model="lang"
5       :options="langOptions"
6       :label="$t('language')"
7       dense
8       borderless
9       emit-value
10      map-options
11      options-dense
12      style="min-width: 150px"
13    />
14  </div>
15 </template>
16
17 <script>
18 export default {
19   data () {
20     return {
21       lang: this.$i18n.locale,
22       langOptions: [
23         { value: 'es', label: 'Español' },
24         { value: 'en', label: 'English' }
25       ]
26     }
27   },
28   watch: {
29     lang (lang) {
30       this.$i18n.locale = lang
31     }
32     .....

```

Listing 4.10: Selector de idioma

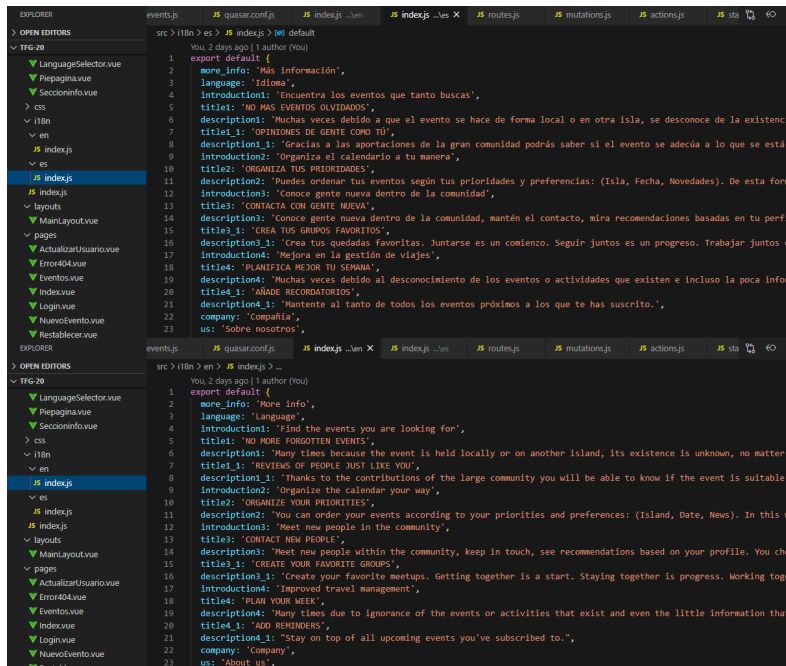


Figura 4.7: Uso de palabras clave

4.10. Firestore y Storage

Para el almacenamiento de datos y archivos multimedia se dividió en 2 modelos de datos: un modelo para los datos de los eventos (datos y foto de cada evento, usuarios que han votado y comentarios) y otro modelo para los datos de los usuarios (datos y foto de perfil).

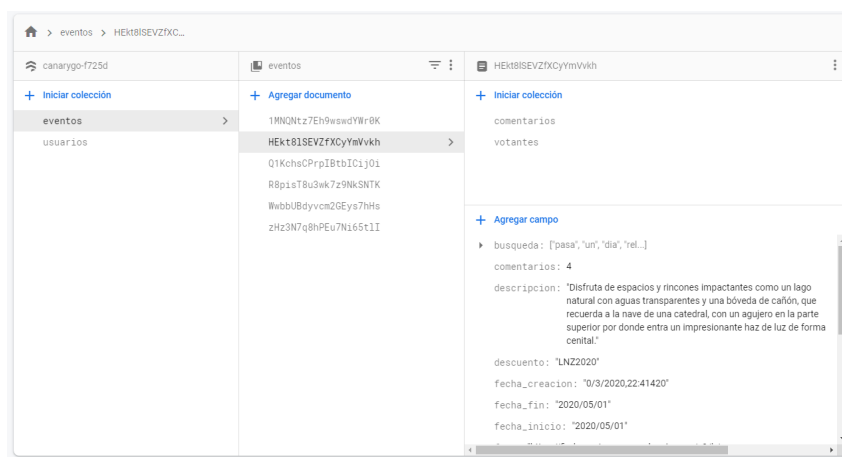


Figura 4.8: Modelo de datos de Firestore (eventos)

4.11. Mapa

El mapa para la elección y localización de los eventos fue otro desafío, ya que aunque Leaflet[17] para Vue[26] traía las funcionalidades básicas. Algunas como la localización de la isla seleccionada, la geocodificación inversa o la creación de marcadores en el mapa

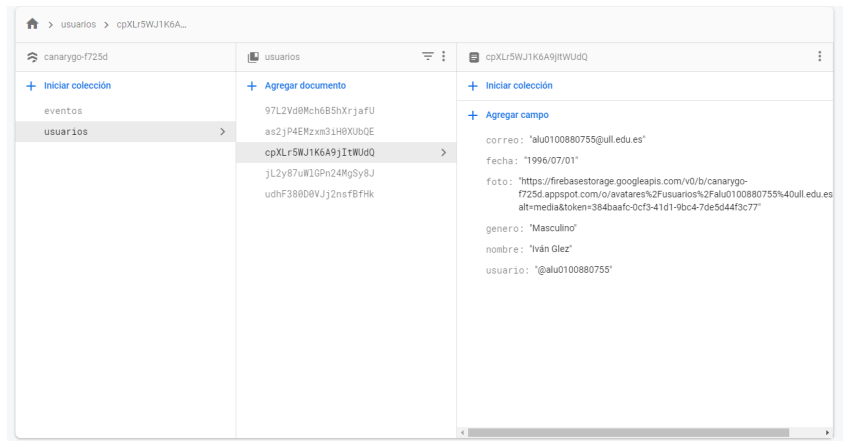


Figura 4.9: Modelo de datos de Firestore (usuarios)



Figura 4.10: Modelo de datos de Storage

tuvieron que ser implementadas.

```

1  marker: true,
2      geosearchOptions: {
3          provider: new OpenStreetMapProvider()
4      },
5      url: 'https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',
6      zoom: 8,
7      center: [28.4682400, -16.2546200],
8      markerLatLng: L.latLng(27.8682400, -15.6546200),
9      markerLatLng_backup: '',
10     cambiar: true,
11     markerinfo: '',
12     tileProviders: [
13         {
14             name: this.$t('map0'),
15             visible: true,
16             attribution:
17                 '&copy; <a target="_blank" href="http://osm.org/copyright">OpenStreetMap
18                 </a> contributors',
19             url: 'https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png'
20         },
21         {
22             name: this.$t('map1'),
23             visible: false,
24             url: 'https://{s}.tile.opentopomap.org/{z}/{x}/{y}.png',
25             attribution:
26                 'Map data: &copy; <a href="http://www.openstreetmap.org/copyright">
27                 OpenStreetMap</a>, <a href="http://viewfinderpanoramas.org">SRTM</a> | Map style

```

```

: &copy; <a href="https://opentopomap.org">OpenTopoMap</a> (<a href="https://
creativecommons.org/licenses/by-sa/3.0/">CC-BY-SA</a>) '
26     }
27   ],
28   isla: '', // Lat y Lng islas
29   // Lanzarote
30   lnz_lat_min: 28.820834,
31   lnz_lat_max: 29.463718,
32   lnz_lng_min: -13.892249,
33   lnz_lng_max: -13.354280,
34
35   // Fuerteventura
36   ftv_lat_min: 27.994879,
37   ftv_lat_max: 28.780818,
38   ftv_lng_min: -14.563415,
39   ftv_lng_max: -13.761317,
40
41   // Gran Canaria
42   gcn_lat_min: 27.702862,
43   gcn_lat_max: 28.231178,
44   gcn_lng_min: -15.854814,
45   gcn_lng_max: -15.314903,
46
47   // Tenerife
48   tnf_lat_min: 27.935335,
49   tnf_lat_max: 28.646747,
50   tnf_lng_min: -16.944341,
51   tnf_lng_max: -16.089806,
52
53   // La Gomera
54   gmr_lat_min: 27.986798,
55   gmr_lat_max: 28.243741,
56   gmr_lng_min: -17.375494,
57   gmr_lng_max: -17.068637,
58
59   // La Palma
60   plm_lat_min: 28.433482,
61   plm_lat_max: 28.881690,
62   plm_lng_min: -18.031932,
63   plm_lng_max: -17.670697,
64
65   // El Hierro
66   hro_lat_min: 27.600233,
67   hro_lat_max: 27.890715,
68   hro_lng_min: -18.197013,
69   hro_lng_max: -17.835778
70 }
71 .....

```

Listing 4.11: Porciones de código de la implementación del mapa

```

1 .....
2   addMarker (event) {
3     // Datos
4     const busqueda = document.getElementsByClassName('leaflet-bar-part leaflet-bar
-part-single')[0]
5     const reset = document.getElementsByClassName('reset')[0]
6

```

```

7      // Establecer idioma
8      document.getElementsByClassName('leaflet-bar-part leaflet-bar-part-single')
9      [0].setAttribute('title', this.$t('direction'))
10     document.getElementsByClassName('glass')[0].setAttribute('placeholder', this.
11     $t('direction'))
12
13     if (document.getElementsByClassName('leaflet-marker-icon leaflet-zoom-animated
14     leaflet-interactive')[1]) {
15         document.getElementsByClassName('leaflet-marker-icon leaflet-zoom-animated
16         leaflet-interactive')[1].click()
17         const ubicacion = document.getElementsByClassName('leaflet-popup-content')
18         [1].outerText
19         this.Geocode(ubicacion)
20         reset.click()
21         document.getElementsByClassName('leaflet-marker-icon leaflet-zoom-animated
22         leaflet-interactive')[0].click()
23     }
24
25     // Marcador
26     this.markerLatLng = L.latLng(event.latlng.lat, event.latlng.lng)
27     document.getElementsByClassName('leaflet-marker-icon leaflet-zoom-animated
28     leaflet-interactive')[0].click()
29     this.reverseGeocode()
30
31     // Cerrar búsqueda
32     reset.onclick = function () {
33         busqueda.click()
34         if (document.getElementsByClassName('leaflet-marker-icon leaflet-zoom-
35         animated leaflet-interactive')[1]) {
36             document.getElementsByClassName('leaflet-marker-icon leaflet-zoom-animated
37             leaflet-interactive')[1].removeChild()
38         }
39     }
40     this.Localizar_isla()
41     this.$emit('clicked', this.markerLatLng, this.isla)
42 },
43 reverseGeocode () {
44     const Nominatim = require('nominatim-geocoder')
45     const geocoder = new Nominatim()
46     const ubicacion = this.markerLatLng.lat + ', ' + this.markerLatLng.lng
47
48     geocoder.search({ q: ubicacion })
49     .then((response) => {
50         if (response.length >= 1) {
51             this.markerinfo = response[0].display_name
52         } else {
53             this.markerinfo = this.$t('no_location')
54         }
55     })
56     .catch((error) => {
57         console.log(error)
58     })
59 },
60 Geocode (ubicacion) {
61     const Nominatim = require('nominatim-geocoder')
62     const geocoder = new Nominatim()
63
64     geocoder.search({ q: ubicacion })

```

```

56     .then((response) => {
57         if (response.length >= 1) {
58             this.markerLatLng = L.latLng(response[0].lat, response[0].lon)
59             this.markerinfo = ubicacion
60         } else {
61             // No se cambia
62         }
63     }).catch((error) => {
64         console.log(error)
65     })
66 }
67 .....

```

Listing 4.12: Porciones de código de la implementación del mapa

4.12. Vuex (Store)

Para centralizar la información del usuario para que fuera accesible a través de cualquier componente, se utilizó Vuex[27]. Se usaron dos modelos de datos, uno para el token y otro para los datos personales obtenidos a través del token. Desde que el token fuera inválido, se borraban todos los datos centralizados y se requería de un inicio de sesión.

```

1  const state = {
2    datosUsuario: [],
3    token: ''
4  }
5
6  export default state

```

Listing 4.13: State de Canary Go

```

1  const actions = {
2    anadirUsuario (context, payload) {
3      setTimeout(function () {
4        context.commit('anadirUsuario', payload)
5      }, 500)
6    },
7
8    anadirToken (context, payload) {
9      setTimeout(function () {
10       context.commit('anadirToken', payload)
11     }, 500)
12   },
13
14   borrarUsuario (context) {
15     setTimeout(function () {
16       context.commit('borrarUsuario')
17     }, 500)
18   }
19 }
20 }
21
22 export default actions

```

Listing 4.14: Actions de Canary Go

```

1 const mutations = {
2   anadirUsuario (state, payload) {
3     state.datosUsuario = payload
4   },
5
6   anadirToken (state, payload) {
7     state.token = payload
8   },
9
10  borrarUsuario (state) {
11    state.datosUsuario = []
12    state.token = []
13  }
14 }
15
16 export default mutations

```

Listing 4.15: Mutations de Canary Go

4.13. Back-end

Tal como se indicó previamente, el “back-end” se dividió en dos modelos: uno para la gestión de las peticiones relacionada con los usuarios (creación de usuario, actualización, login, etc.) y otro para la gestión de las peticiones de los eventos (creación de eventos, obtención de la información de los eventos registrados, etc.). Todas estas peticiones sólo eran posibles de realizar con un token JWT[15] válido generado por el back-end y otorgado al usuario durante el “login”. Para ello se utilizó un usuario administrador de Firebase.

```

1 module.exports = function (app) {
2   const { firebaseAuth, firebaseDb, firebaseStg } = require('../config/firebase')
3   const { admin } = require('../config/firebaseadmin')
4   const { OAuth2Client } = require('google-auth-library')
5
6   // ////////////////////////////////// IDENTIFICACIÓN //////////////////////////////////
7   .....
8   // Put (Login - Registro)
9   app.put('/autorizar', function (req, res) {
10
11     // ////////////////////////////////// LOGIN //////////////////////////////////
12     if (req.body.tipo === 'Login') {
13       if (req.body.tipo2 === 'Estandar') {
14         let errorcodes = ''
15
16         firebaseAuth.signInWithEmailAndPassword(req.body.correo, req.body.contrasena
17         )
18         .catch(function (error) {
19
20           // Errores
21           var errorCode = error.code
22           var errorMessage = error.message
23           errorcodes = errorCode
24         })
25         .then(() => {
26           const user = firebaseAuth.currentUser

```

```

26     firebaseAuth.signOut()
27
28     if (errorcodes === 'auth/user-not-found') {
29         res.send(errorcodes)
30     } else if (errorcodes === 'auth/wrong-password') {
31         res.send(errorcodes)
32     } else if (errorcodes === 'auth/too-many-requests') {
33         res.send(errorcodes)
34     } else {
35         // Creamos token de sesion
36         if (user.emailVerified === true) {
37             // Generamos el token del cliente
38             admin.auth().createCustomToken(user.uid)
39                 .then(function (customToken) {
40                     // Enviamos token al cliente
41                     res.send('Usuario correcto:' + customToken)
42                 })
43                 .catch(function (error) {
44                     console.log('Error al crear token:', error)
45                     res.send('auth/user-not-found')
46                 })
47         } else {
48             res.send('auth/must-verify')
49         }
50     }
51 })
52 } else if (req.body.tipo2 === 'OAuth') {
53     IniciarSesionOAuth(req, res)
54 }

```

Listing 4.16: Porción de código para la generación de JWT durante el login en el backend



Figura 4.11: Seguridad de JWT utilizada en el proyecto

Capítulo 5

Conclusiones y líneas futuras

Aunque el proyecto ha llevado un duro trabajo diario, ha sido bastante bueno para el aprendizaje y formación como desarrollador FullStack, lo cual es de gran utilidad para el futuro. Muchas de las herramientas y tecnologías mencionadas (Authentication, Storage, Firestore, Quasar, Cypress, no habían sido utilizadas anteriormente y fueron “un soplo de aire fresco” de nuevo conocimiento.

Por un lado, el uso de Quasar como “framework” de desarrollo del proyecto, me ha parecido bastante interesante y útil ya que ofrece una gran versatilidad en cuanto al despliegue de aplicaciones. Además, ofrece algunos componentes predefinidos muy bien logrados como, por ejemplo: Parallax en imágenes, alertas o notificaciones, avatares, etc.). Aunque Quasar es relativamente nuevo, continuamente están añadiendo componentes y mejoras, con lo que con el paso del tiempo dará más facilidades como “framework” y será cada vez más fácil desarrollar en él.

Por otro lado, como Quasar está basado en Vue JS, ha sido una buena experiencia de contacto. Anteriormente no había desarrollado nada en Vue y ahora se me abre un mundo de posibilidades mucho más amplio como desarrollador al tener un poco de experiencia en todo.

Con respecto a las líneas futuras, si el proyecto tuviese éxito y fuera apoyado como una de las webs oficiales para ocio y turismo en Canarias, se podrían añadir otras funcionalidades como:

- Posibilidad de ver en una sección llamada mis eventos, todos los eventos en los que el usuario se haya registrado.
- Posibilidad de descargar los eventos, junto con la información (como la descripción, hora, fechas, etc.) e importarlo al calendario del usuario.
- Añadir notificaciones vía email o “push”, para los eventos que contengan las palabras claves utilizadas por el usuario.

También se podría optar por desplegar la aplicación en dispositivos móviles IOS, Android e incluso para dispositivos de escritorio como Windows, Linux y Mac OS, ya que el “framework” lo permite (con la correcta configuración). Para desplegar en los sistemas

operativos de IOS y Mac OS, se requiere un equipo Mac de Apple para la instalación y utilización de Xcode.

Por otro lado, habría que terminar de dar soporte para el idioma en inglés para la zona de comentarios, cuando se añada un evento y cuando se muestre. De forma que no dependa del idioma en el que se escriba el mismo, sino que toda la información sea soportada en todos los idiomas. Incluso si el proyecto se expande lo suficiente añadir mucho más soporte en otros idiomas.

El proyecto siempre estará abierto a mejoras que se propusieran y que no fueran las mencionadas anteriormente.

Capítulo 6

Summary and Conclusions

Although the project has taken hard and daily work, it has been quite good for learning and training as a FullStack developer, which is very useful for the future. Many of the tools and technologies mentioned (Authentication, Storage, Firestore, Quasar, Cypress), had not been used before and were “a breath of fresh air” to acquire new knowledge.

On the one hand, the use of Quasar as a project development framework has seemed quite interesting and useful to me since it offers great versatility in terms of application deployment. In addition, it offers some very well-made predefined components such as components for Parallax in images, alerts or notifications, avatars, etc.). Although Quasar is relatively new, they are continually adding components and improvements, with which over time it will give more facilities as a framework and it will be easier to develop on it.

On the other hand, since Quasar is based on Vue JS, it has been a good experience of contact with Vue JS technology. Previously I had not developed anything in Vue. Now I have a much wider margin as a developer by having a little experience in everything.

Regarding future lines, if the project turned out to be a success and was supported as one of the official websites for leisure and tourism in the Canary Islands, other features could be added, such as:

- Possibility to see in a section called my events, all the events in which the user has registered.
- Ability to download events, along with information (such as description, time, dates, etc.) and import it into the user’s calendar.
- Add notifications via email or push, for events that contain the keywords used by the user.

You could also choose to deploy the application on IOS, Android mobile devices and even for desktop devices such as Windows, Linux and Mac OS, since the framework allows it (with the correct configuration). To deploy to IOS and Mac OS operating systems, an Apple Mac is required for the installation and use of Xcode.

On the other hand, it would be necessary to end the support for the English language for the comments area, when an event is added and when it is displayed. So that it does not

depend on the language in which it is written, but that all the information is supported in all languages. Even if the project expands enough add much more support in other languages.

The project will always be open to improvements that were proposed and were not those mentioned above.

Capítulo 7

Presupuesto

Como no se conoce la cantidad de usuarios inicial que tendría el proyecto en caso de ser lanzado al mercado se harán 2 tipo de estimaciones: estimación de usuarios máxima y estimación de usuario estándar basado en la población de Canarias actual (aproximadamente 2.24 millones de personas).

Población de la comunidad autónoma de Canarias en 2020

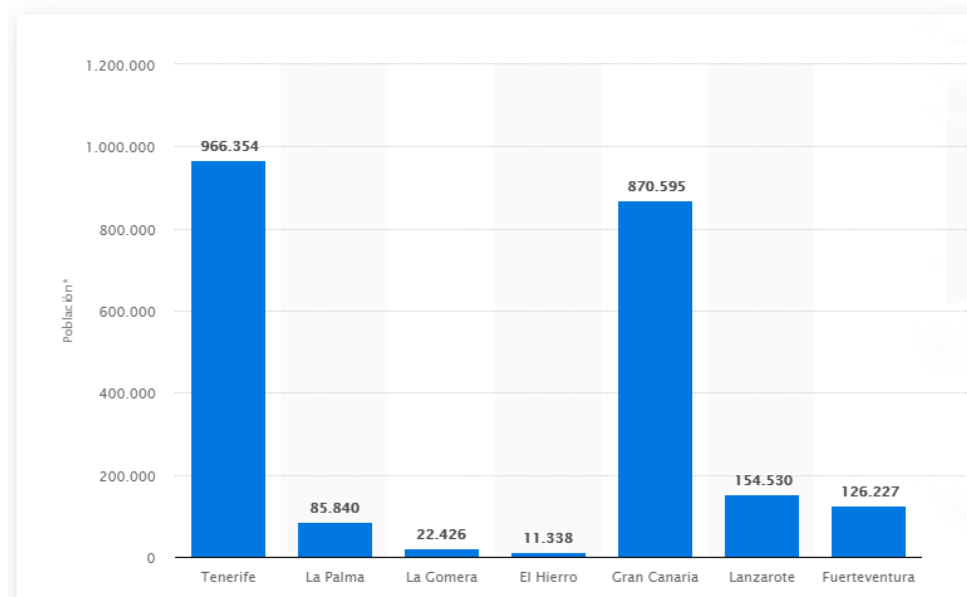


Figura 7.1: Población de Canarias en 2020 por isla [22]

Al estar basado en una gran comunidad (2.24 millones de usuarios) en la que se está continuamente compartiendo datos, se requiere de una gran cantidad de almacenamiento de datos, ya sean archivos multimedia o texto y operaciones de lectura y escritura. Al utilizarse Firebase para este proyecto, vamos a preparar el presupuesto basándonos en los costos de la plataforma de Google. Además, para el “hosting” se utilizará Heroku [11]. Se tiene en cuenta la implantación de las mejoras.

Presupuesto actual: desarrollo de la aplicación actual sin disposición al público.

Servicio	Tiempo	Precio	Precio Final
Programador	1056 horas	8,00 € / hora	8448 €
Firestore [8]	25550 horas (3.5 meses)	0 € / mes	0 €
Storage [9]	25550 horas (3.5 meses)	0 € / mes	0 €
Heroku [11]	25550 horas (3.5 meses)	0 € / mes	0 €
Precio final combinado			8448 €

Tabla 7.1: Presupuesto actual

Presupuesto lanzamiento v1: lanzamiento de la aplicación web con estimación de uso máximo del 50 % de la población, unos 1.12 millones de usuarios. Se incluye la implementación de las mejoras mencionadas anteriormente. La estimación se basa en que todos los usuarios añadan eventos con un máximo de 10 eventos (10 operaciones de escritura por usuario).

Servicio	Tiempo	Precio	Precio Final
Programador 1	150 horas	8,00 € / hora	1200 €
Programador 2	150 horas	8,00 € / hora	1200 €
Firestore [8]	1 año	297,44 € / mes	3569.28€
Storage [9]	1 año	17,52 € / mes	210.24€
Heroku [11]	1 año	222,71 € / mes	2672.52€
Precio final combinado			8852.04€

Tabla 7.2: Presupuesto lanzamiento v1

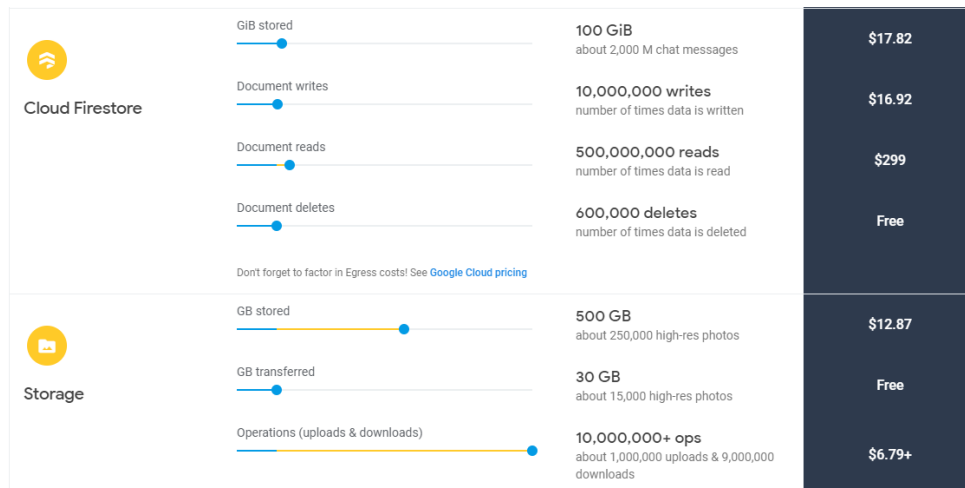


Figura 7.2: Presupuesto de Google para la v1 [7]

Presupuesto lanzamiento v2: lanzamiento de la aplicación web con estimación de uso máximo del 100% de la población, unos 2.24 millones de usuarios. Se incluye la implementación de las mejoras mencionadas anteriormente. La estimación se basa en que todos los usuarios añadan eventos con un máximo de 10 eventos (10 operaciones de escritura por usuario).

Servicio	Tiempo	Precio	Precio Final
Programador 1	150 horas	8,00 € / hora	1200 €
Programador 2	150 horas	8,00 € / hora	1200 €
Firestore [8]	1 año	639,66 € / mes	7675.92€
Storage [9]	1 año	35,53 € / mes	426 €
Heroku [11]	1 año	445,42 € / mes	5345.04€
Precio final combinado			15 846.96€

Tabla 7.3: Presupuesto lanzamiento v2

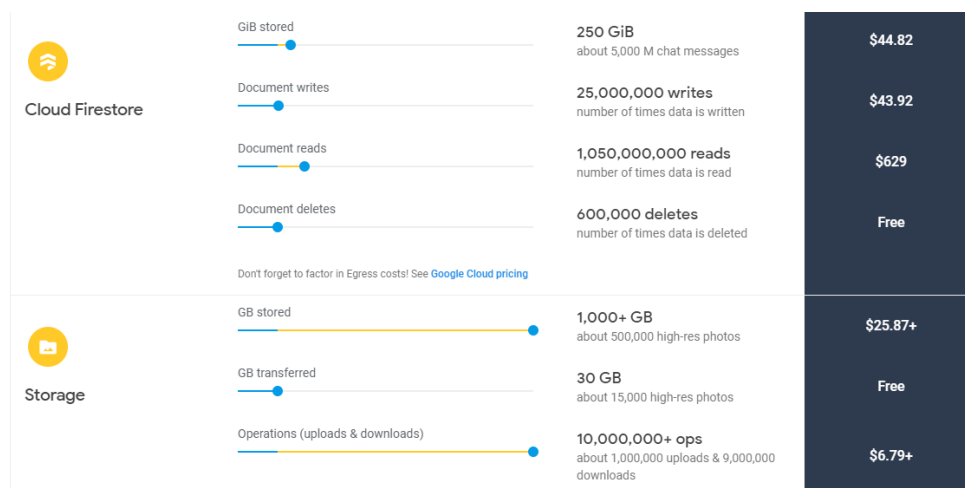


Figura 7.3: Presupuesto de Google para la v2 [7]

Bibliografía

- [1] Airbnb. Lo más importante está más cerca de lo que crees. <https://es-l.airbnb.com/>. Accessed: 2020-07-05.
- [2] Chollometro. Chollometro. <https://www.chollometro.com/>. Accessed: 2020-07-05.
- [3] Travis CI. The simplest way to test and deploy your projects. <https://travis-ci.com/>. Accessed: 2020-07-05.
- [4] Cypress. The web has evolved. finally, testing has too. <https://www.cypress.io/>. Accessed: 2020-07-05.
- [5] ESLint. Find and fix problems in your javascript code. <https://eslint.org/>. Accessed: 2020-07-05.
- [6] GitHub. Github is how people build software. <https://github.com/about>. Accessed: 2020-07-05.
- [7] Google. Calculadora del plan blaze. <https://firebase.google.com/pricing?authuser=0#blaze-calculator>. Accessed: 2020-07-05.
- [8] Google. Cloud firestore. 7. <https://firebase.google.com/docs/firestore?hl=es>. Accessed: 2020-07-05.
- [9] Google. Cloud storage. <https://firebase.google.com/docs/storage?hl=es>. Accessed: 2020-07-05.
- [10] Google. Firebase authentication. <https://firebase.google.com/docs/auth?hl=es>. Accessed: 2020-07-05.
- [11] Heroku. Heroku is a cloud platform that lets companies build, deliver, monitor and scale apps – we’re the fastest way to go from idea to url, bypassing all those infrastructure headaches. <https://www.heroku.com/what>. Accessed: 2020-07-05.
- [12] Housetrip. Encuentra un sitio donde quedarte. <https://www.housetrip.es/>. Accessed: 2020-07-05.
- [13] Node JS. Acerca de node.js®. <https://nodejs.org/es/about/>. Accessed: 2020-07-05.
- [14] Vue JS. Introduction. <https://router.vuejs.org/>. Accessed: 2020-07-05.
- [15] JWT. Introduction to json web tokens. <https://jwt.io/introduction/>. Accessed: 2020-07-05.

- [16] Leaflet. an open-source javascript library for mobile-friendly interactive maps. <https://leafletjs.com/>. Accessed: 2020-07-05.
- [17] Vue Leaflet. Harness the power of leaflet in vuejs. <https://vue2-leaflet.netlify.app/>. Accessed: 2020-07-05.
- [18] Microsoft. Code editing. redefined. https://code.visualstudio.com/?wt.mc_id=DX_841432. Accessed: 2020-07-05.
- [19] NPM. Build amazing things. <https://www.npmjs.com/>. Accessed: 2020-07-05.
- [20] Quasar. Introduction to quasar. <https://quasar.dev/introduction-to-quasar>. Accessed: 2020-07-05.
- [21] Statista. Evolución anual del gasto medio por viaje en actividades de ocio de los turistas que visitaron canarias de 2010 a 2019r. <https://es.statista.com/estadisticas/524090/gasto-medio-por-viaje-en-actividades-de-ocio-de-los-turistas-que-visitaron-canarias/>. Accessed: 2020-07-05.
- [22] Statista. Población de la comunidad autónoma de canarias en 2020, por isla. <https://es.statista.com/estadisticas/474029/poblacion-de-canarias-por-isla/#:~:text=Esta%20estad%C3%ADstica%20muestra%20la%20poblaci%C3%B3n,poblaci%C3%B3n%20ascend%C3%ADa%20unas%20870.000%20personas>. Accessed: 2020-07-05.
- [23] Cast the code. El patrón mvvm - introducción. <https://theshallowbay.github.io/tutoriales/2017/07/02/patron-mvvm-introduccion/>. Accessed: 2020-07-05.
- [24] Pivotal Tracker. Pivotal tracker is changing how teams build software—one story at a time. <https://www.pivotaltracker.com/>. Accessed: 2020-07-05.
- [25] Vrbo. Casas rurales, en la playa y apartamentos en todo el mundo. <https://www.vrbo.com/es-es>. Accessed: 2020-07-05.
- [26] Vue. Qué es vue.js? <https://es.vuejs.org/v2/guide/>. Accessed: 2020-07-05.
- [27] Vuex. What is vuex? <https://vuex.vuejs.org/>. Accessed: 2020-07-05.
- [28] Pine Wu. Vetur. <https://marketplace.visualstudio.com/items?itemName=octref.vetur>. Accessed: 2020-07-05.