



Universidad de Valladolid
Escuela de Ingeniería Informática
Trabajo Fin de Grado

Grado en Ingeniería Informática
Mención Tecnologías de la información

Electronic Money Token
Gestor de saldos de eventos

Autor:
D. Borja Rabadán Martín



Universidad de Valladolid

Escuela de Ingeniería Informática

Trabajo Fin de Grado

Grado en Ingeniería Informática
Mención Tecnologías de la información

Electronic Money Token

Gestor de saldos de eventos

Autor:

D. Borja Rabadán Martín

Tutor:

Dr. Joaquín Adiego Rodríguez

Agradecimientos

*A los profesores de la facultad de informática.
A mis padres por su esfuerzo y pagarme los estudios.
A mi novia Carme por acompañarme mi último año.
A mi hermano y mi cuñada por soportarme cada día.
A los multiversos de Telegram por conocer gente maravillosa.
Y a mis queridos compañeros y amigos de universidad.*

Resumen

El objetivo principal de este proyecto es el desarrollo de una aplicación móvil en el sistema operativo Android para la gestión de saldos de eventos, con el fin de facilitar a los usuarios la realización de sus transacciones en diferentes eventos de manera sencilla, rápida y eliminando los actuales inconvenientes.

Además de una aplicación web de administración de eventos, donde los creadores de éstos puedan gestionarlos de una manera sencilla, de tal modo que se puedan visualizar en cualquier PC y dispositivo inteligente.

Abstract

The main objective of this project is the development of a mobile application in the Android operating system for the management of event balances, in order to facilitate users to management their transactions in different events in a simple, fast and eliminating the current disadvantage.

In addition to a web application of event management, where the creators of these can manage them in a simple way, so that they can be viewed on any PC and smart device.

Índice general

Capítulo I	10
Introducción y contexto	10
1.1 Introducción	10
1.2 Objetivos	11
1.3 Motivación	11
1.4 Alternativas	12
1.4.1 Dinero	12
1.4.2 Tokens	13
1.4.3 Pulseras Cashless	14
1.4.4 Monederos electrónicos	15
1.4.5 Apps similares	16
1.5 Estructura de la memoria	17
1.6 Propuesta	17
Capítulo II	18
Herramientas utilizadas	18
2.1 Herramientas utilizadas	18
2.2 Tecnologías	18
2.2.1 Firebase	19
2.2.2 Java	20
2.2.3 Angular	21
2.2.4 Android	21
2.2.5 AES	23
Capítulo III	24
Electronic Money Token APP	24
3.1 Aplicación móvil	24
3.2 Plan de Desarrollo del Software	24
3.2.1 Alcance del proyecto	24
3.2.2 Metodología aplicada	25
3.2.3 Equipo	27
3.2.4 Plan de Riesgos	28
3.2.5 Gestión del proyecto	31
3.2.5.1 Estimación	31
3.2.5.2 Formación	31
3.2.5.3 Calendario de Trabajo	32
3.2.6 Gestión de configuraciones	33
3.2.7 Presupuesto del proyecto	34

3.2.7.1 Recursos Humanos	34
3.2.7.2 Recursos Hardware y Software	35
3.2.7.3 Costes Indirectos	35
3.3 Análisis	36
3.3.1 Requisitos	36
3.3.1.1 Requisitos funcionales	36
3.3.1.2 Requisitos no funcionales	37
3.3.2 Casos de uso	38
3.3.3 Modelo de dominio	46
3.3.4 Diagramas de secuencia	47
3.4 Diseño	54
3.4.1 Arquitectura	54
3.4.2 Patrones	57
3.4.3 Modelo no relacional	58
3.4.4 Diagrama de despliegue	59
3.5 Implementación	60
3.6 Pruebas	62
3.7 Manual de instalación	63
3.8 Guía de usuario	64
Capítulo IV	75
Electronic Money Token WEB	75
4.1 Aplicación Web	75
4.2 Plan de Desarrollo del Software	75
4.2.1 Alcance del proyecto	75
4.2.2 Metodología aplicada	75
4.2.3 Equipo	77
4.2.4 Plan de Riesgos	78
4.2.5 Gestión del proyecto	82
4.2.5.1 Estimación	82
4.2.5.2 Formación	82
4.2.5.3 Calendario de Trabajo	83
4.2.6 Gestión de configuraciones	84
4.2.7 Presupuesto del proyecto	84
4.2.7.1 Recursos Humanos	85
4.2.7.2 Recursos Hardware y Software	85
4.2.7.3 Costes Indirectos	86
4.2.7.4 Costes totales	86
4.3 Análisis	87
4.3.1 Requisitos	87
4.3.1.1 Requisitos Funcionales	87
4.3.1.2 Requisitos Funcionales	87

4.3.2 Casos de uso	88
4.3.3 Modelo de dominio	92
4.3.4 Diagramas de secuencia	93
4.4 Diseño	96
4.4.1 Arquitectura	96
4.4.2 Modelo no relacional	98
4.4.3 Diagrama de despliegue	99
4.5 Implementación	99
4.6 Pruebas	101
4.7 Manual de instalación	102
4.8 Guía de usuario	103
Capítulo V	108
Conclusiones y trabajo futuro	108
5.1 Conclusiones	108
5.2 Trabajo futuro	109
Capítulo VI	110
Webgrafía y Bibliografía	110
Parte I	113
Anexos	113
Anexo I: Estudio estadístico	114
Estudio	114
Conclusión	114
Anexo II: Selenium	117

Capítulo I

Introducción y contexto

1.1 Introducción

Actualmente se congregan decenas de miles de personas en diferentes eventos por todo el mundo, realizan diferentes actividades que van desde los congresos de las tecnologías del mañana, como puede ser el CES (Consumer Electronics Show) de Las Vegas con más 182.000 asistentes, o el mismísimo Tomorrowland donde se reúnen la mayoría de los artistas musicales del mundo de la música electrónica con más de 450.000 asistentes disfrutando de los espectáculos, hasta eventos pequeños como el Zlive en la localidad zamorana con 4000 asistentes.

Todos estos eventos generan una gran cantidad de beneficios, y no solo por el coste de la entrada al recinto, sino de los productos o servicios que se pueden adquirir dentro de ellos, éstos pueden ir desde la comida y bebida hasta el pago del parking en dichos espacios. Para ello los organizadores evitan el uso de dinero efectivo para aumentar la comodidad de los asistentes mediante el uso de los "tokens" (piezas de plástico o hechas a partir de papel) , como si se tratara de una divisa nueva en otro país, lo que permite a los asistentes mayor seguridad al no tener que usar dinero real, ya que solo lo pueden usar dentro del evento y no tendría validez en el exterior, así como la facilidad de dar el cambio a los usuarios al adquirir un producto.

Sin embargo el uso de los tokens da lugar a una gran cantidad de inconvenientes, empezando con el propio hecho de que para adquirirlos tienes que acudir a puntos de intercambio dentro de los propios recintos, por lo que nos lleva a que el propósito de la seguridad se ve anulado, además el hecho de hacer cola en estos puntos nos roba parte de nuestro tiempo. La cantidad de los tokens fabricados deben ser estudiados con gran determinación para no crear un exceso o una escasez de los mismos, lo que provocaría pérdidas, y suelen estar hechos de materiales no biodegradables por lo que no tiene una visión ecológica.

En la búsqueda de alternativas del token, se han llegado a otras vías de pago que aún pecan de los mismos errores llevando a un sobrecoste y a una mala gestión.

1.2 Objetivos

La propuesta del Trabajo Fin de Grado es realizar una aplicación en Android que puedas pagar y cobrar en los distintos eventos, mediante las tecnologías más actuales para su desarrollo. Además de una plataforma web donde se podrá gestionar los eventos creados y mostrar información acerca de ellos.

Este Trabajo Fin de Grado mostrará el análisis, desarrollo e implementación de ambos sistemas, y un estudio estadístico / reportaje de la utilidad que tendría la idea cara al público.

1.3 Motivación

La principal motivación es analizar e implementar un aplicación móvil, siendo ésta un elemento cotidiano en el usuario actual y utilizar tecnologías que ayuden a su desarrollo en el Trabajo Fin de Grado.

Desarrollar una aplicación web con el framework Angular, está siendo actualmente demanda por las empresas para el desarrollo de webs que se realizan en una sola página y además son más dinámicas.

Otra de las motivaciones es en base a la detección de un problema actual basado en la propia experiencia a la hora de asistir eventos, y presentar una solución mediante la informática.

1.4 Alternativas

1.4.1 Dinero



Figura 1: Euros

- **Elemento analizado:** Euros.

- **Análisis:**

Es el activo empleado por la Unión Europea como medio de pago, éste es el medio más cómodo para el asistente del evento, ya que no tiene que realizar ningún cambio de "divisa", te da mayor tranquilidad y tienes un control de lo que gastas. Sin embargo, uno de los motivos por el cual el organizador quiere suprimir el uso del dinero en efectivo es ayudar ofrecer a los usuarios alternativas más seguras y que resuelvan los tiempo de espera a la hora de dar el cambio como veremos más adelante.

- **Ventajas:**

- No tienes que cambiar tu dinero por tokens.
- Más confianza.
- Controlas lo que gastas.

- **Desventajas:**

- Menos seguridad.
- El cambio.
- Sin patrocinadores

1.4.2 Tokens

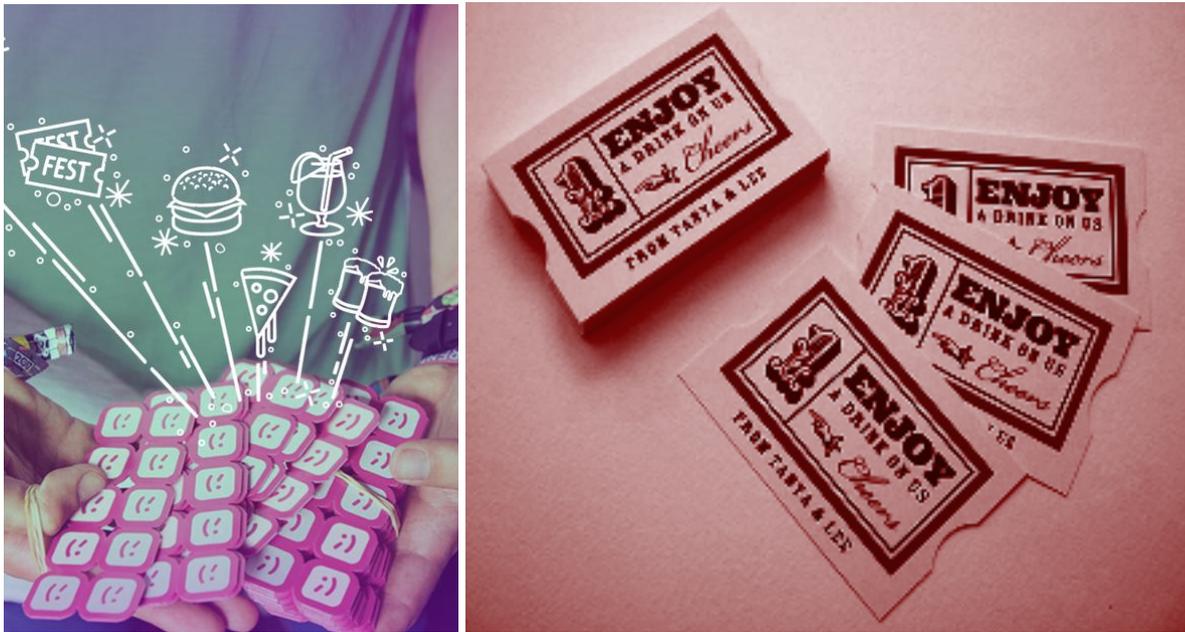


Figura 2: Tokens

- **Elemento analizado:** Tuent.

- **Análisis:**

Los organizadores de los eventos, implantaron los tokens con el fin de buscar alternativas al dinero y agilizar la venta de productos y garantizar su seguridad. Pero el propósito que más favorece al organizador es poder ofrecer “publicidad” en sus tokens, y conseguir un beneficio extra y/o que la empresa corra con los gastos de éstos. Por otro lado también provoca un mayor gasto al usuario, ya que puede llevar a una mala interpretación del valor verdadero de los productos transformándose en marketing engañoso.

El material de estos tokens no son biodegradables por lo que no se reutilizan una vez usados, con lo cual lleva un sobrecoste y perjudica el medio ambiente.

- **Ventajas:**

- Patrocinadores.
- El cambio.
- Merchandising.

- **Desventajas:**

- Acudir a un punto de cambio.

- No suelen ser biodegradables.
- No mejora la seguridad.
- Incómodo de llevar en grandes cantidades.

1.4.3 Pulseras Cashless



Figura 3: Cashless

- **Elemento analizado:** Cashless del Zlive.

- **Análisis:**

Los organizadores para mejorar el sistema anterior de Tokens, facilitar el transporte de saldo y aumentar la comodidad del usuario crearon unas pulseras con un chip NFC el cual registra el saldo a un usuario. Pero ésto no soluciona los problemas principales de los tokens y además están obligados a depender de máquinas de terceros para la lectura y cobro del saldo.

- **Ventajas:**

- El cambio.
- Merchandising.
- Más fácil de transportar el saldo.

- Desventajas:

- Acudir a un punto de cambio.
- No suelen ser biodegradables.
- No mejora la seguridad.
- Necesita dispositivos ajenos para la gestión del saldo.

1.4.4 Monederos electrónicos



Figura 4: Apps de pago

- **Elementos analizados:** PayPal, PayTM, Passbook.

Análisis:

Es un medio muy usado hoy en día por su comodidad, rapidez y pagas solo lo justo. Cada app tiene su medio de pago, mediante email, alias, QR o contactless, cada uno con sus pros y contras, pero sin embargo no se agrupan todos estos medios en una sola app. Tampoco está centrada para eventos y no tiene la posibilidad de patrocinadores.

Ventajas:

- Más seguro.
- El cambio.
- Más fácil de transportar el saldo.
- Más rápido de tener adquirir saldo.

Desventajas:

- Sin patrocinadores.
- No enfocadas a eventos.
- Más lento los pagos.

- No engloba varios modos de pago

1.4.5 Apps similares

No he encontrado aplicaciones del mismo estilo o que se usan hoy en dia en eventos.

1.5 Estructura de la memoria

Este trabajo está dividido en seis partes y, a su vez, en apartados y subapartados, abarcando así los distintos documentos recogidos en esta memoria:

1. Capítulo I: Introducción del cometido de la app y en el contexto por el cual se desarrolla.
2. Capítulo II: Tecnologías y dispositivos usados para desarrollar estas aplicaciones y los motivos de optar por ellas.
3. Capítulo III: Estará compuesto por varios procesos: plan de desarrollo, análisis, diseño, implementación y testeado acerca de la app móvil.
4. Capítulo IV: Estará compuesto por varios procesos: plan de desarrollo, análisis, diseño, implementación y testeado acerca de la aplicación web.
5. Capítulo V: Resultados obtenidos en el proceso del Trabajo Fin de Grado y mejoras futuras a implementar en el trabajo.
6. Capítulo VI: Enlaces de las páginas webs y libros de los recursos utilizados.
7. Anexo: Estudio realizado mediante Google Forms para obtener la opinión de la viabilidad de la app.

1.6 Propuesta

El proyecto se realizará en base a la propuesta de D. Borja Rabadán Martín bajo las indicaciones y recomendaciones del tutor del Trabajo Fin de Grado, el Dr. Joaquín Adiego Rodríguez, y como motivo de seguimiento del proyecto iniciado en la asignatura "Sistemas móviles" de 4º de carrera.

Capítulo II

Herramientas utilizadas

2.1 Herramientas utilizadas

Herramientas utilizadas en el desarrollo de las aplicaciones.

- Para la elaboración de la documentación, Microsoft Word 2016.
- Para copias de seguridad de documentos y ficheros, Google Drive.
- Para la historias de usuario, Restyaboard .
- Para el control de versiones, GitHub.
- Para la elaboración de diagramas de Gantt, Microsoft Project.
- Como IDE para el desarrollo, Android Studio 3.2.1.
- Como editor , Visual Studio Code 1.3.
- Navegador Web , Firefox.
- Editor de imagenes, PhotoShop CC 2015.
- Para realizar bocetos de la app y web, Mockup.
- Para la relación de diagramas de UML, Visual Paradigm Online y Astah.
- Para los test de pruebas, Sonarqube.

2.2 Tecnologías

En esta sección se indican las diferentes tecnologías usadas en este trabajo: Firebase, Android, Java , Angular y AES.

2.2.1 Firebase



Figura 5: Logo Firebase

Firebase es una plataforma de Google, para el desarrollo de aplicaciones web y aplicaciones móviles desarrollada en 2012 y adquirida por Google en 2014. Donde nos ofrece una plataforma para poder desarrollar nuestras aplicaciones con facilidad y ofrecer múltiples herramientas.

Firestore Analytics

Firestore Analytics es un servicio que nos ofrece datos estadísticos del uso de nuestras aplicaciones.

Firestore Cloud Messaging

Firestore Cloud Messaging es una plataforma para gestionar mensajes y notificaciones en los dispositivos.

Firestore Auth

Firestore Auth es un servicio que puede identificar a los usuarios mediante Facebook, GitHub, Twitter, Google y email. Además, nos ofrece mayor seguridad y restricciones mediante las reglas que se pueden establecer en firebase.

Firestore Hosting

Nos ofrece un alojamiento web para almacenar páginas webs, en este caso Angular, ofreciendo total seguridad con SSL y poca latencia sin importar la ubicación para una mayor fluidez de la página web.

Firestore Firestore

Firestore es un servicio que nos ofrece una base de datos NoSQL de tipo documento-colección. Se organiza en forma de documentos agrupados en colecciones, y en ellos se pueden incluir tanto campos de diversos tipos (cadenas de texto, números, puntos geográficos, referencias a la propia base de datos, arrays, booleanos, marcas de tiempo, e incluso objetos propios) como otras subcolecciones.

Almacena y sincroniza datos entre usuarios y dispositivos a escala global, además te ofrece sincronización en vivo, soporte sin conexión y consultas de datos eficaces.

Todas estas herramientas son gratuitas, pero limitadas. A cierto nivel de uso y mayor número de servicios, se puede seleccionar planes adaptados a tus necesidades.

2.2.2 Java

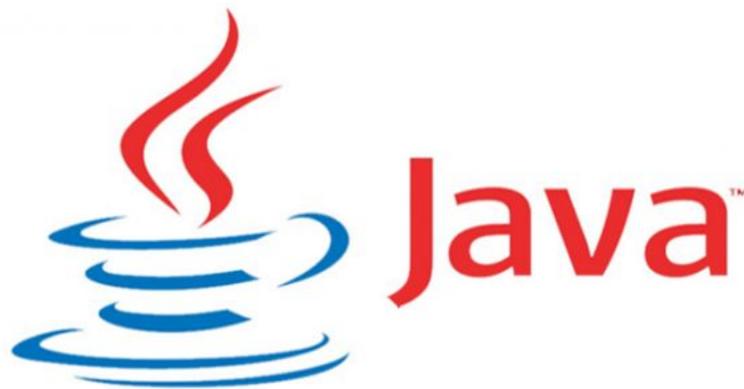


Figura 6: Logo Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo . Actualmente, Java tiene una de las comunidades más importantes y grandes del mundo.

Como alternativas para desarrollar el proyecto en Android, está el conocido Kotlin, desarrollado por JetBrains que destaca por su programación dedicada a procedimientos y es interoperable con código Java.

Pero debido a la curva de aprendizaje de Java y disponer de más información acerca de Java en Android que en Kotlin, opte por Java.

2.2.3 Angular



Figura 7: Logo Angular

Angular es un framework para aplicaciones web desarrollado en TypeScript, soportado por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Es predecesora de AngularJS pero sin compatibilidad con ella.

Al ser un framework, nos da multitud de ventajas a la hora de desarrollar y sin preocuparnos sobre otros aspectos, hoy en día es muy cotizada en las empresas, debido el uso de componentes web que nos ayuda a reutilizar partes en otros proyectos y el gran soporte de herramientas compatible con este framework.

2.2.4 Android



Figura 8: Logo Android

Android es un sistema operativo que fue creado especialmente para teléfonos con pantalla táctil, smartphones, tablets, smartwatch, televisores y sistemas de navegación de los coches. Adquirida en 2005 por Google y que continuó con su desarrollo a través de 10 versiones a lo largo de 11 años, dejando Android Q como la última versión del mercado.

Introducir una app en Android tiene más ventajas que en otros S.O, como iOS o Windows Phone, ocupan el 80% del mercado mundial de la telefonía. Nos permite más facilidad de distribuirla ya que solo se tiene que pagar una cuota de por vida de 25\$ y recibe multitud de opiniones gracias a Google Play.

Existen mayor número de desarrolladores y trabajar sobre un sistema de código abierto nos permite multitud de opciones y mejoras en nuestra aplicaciones.

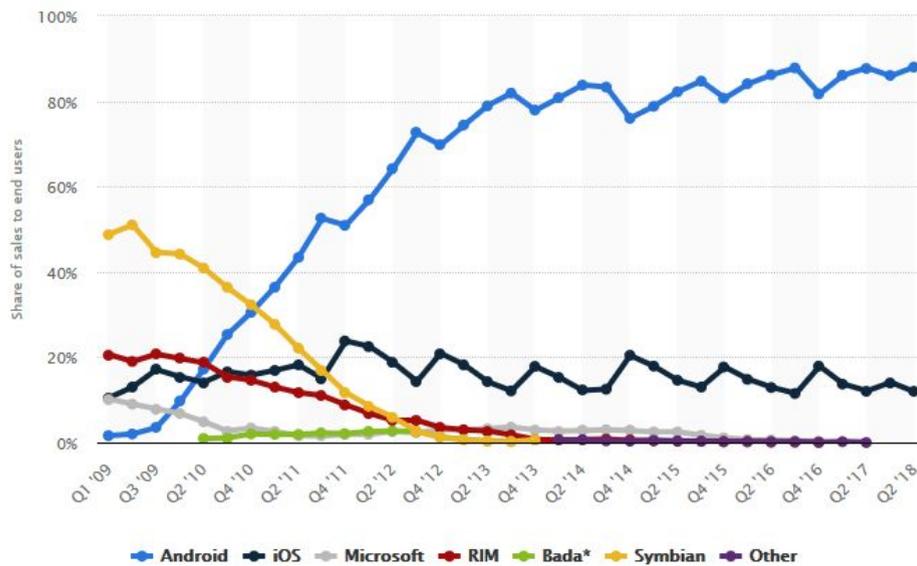


Figura 9: Evolución del número de dispositivos en el mundo

2.2.5 AES

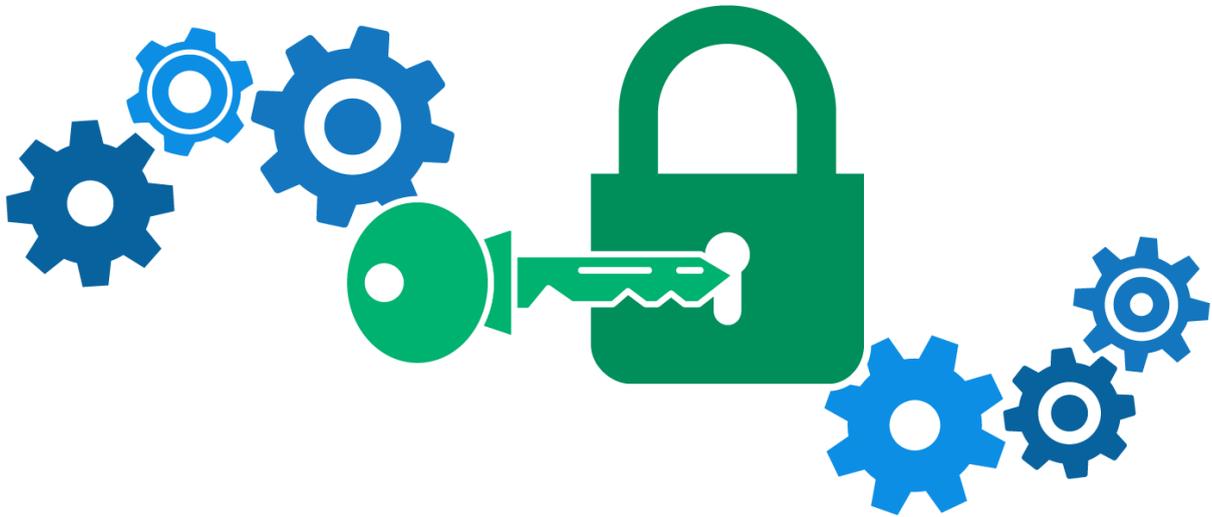


Figura 10: Encriptación

Advanced Encryption Standard es uno de los algoritmos de cifrado más utilizados. El algoritmo se basa en varias sustituciones, permutaciones y transformaciones lineales, cada una ejecutada en bloques de datos de 16 bytes, estas son ejecutadas en "Rounds". Durante cada "Round", una clave circular única se calcula a partir de la clave de cifrado y se incorpora en los cálculos. Basado en la estructura de bloques de AES, el cambio de un solo bit, ya sea en la clave, o en el bloque de texto sin cifrado, da como resultado un bloque de texto cifrado completamente diferente - una ventaja clara sobre los cifrados de flujo tradicionales. Existen tres variedades que son AES-128, AES-192 y AES-256, cuya diferencia es el tamaño de bits que está formado.

Capítulo III



Figura 11: Logo de EMT

Electronic Money Token APP

3.1 Aplicación móvil

3.2 Plan de Desarrollo del Software

Elaborar una aplicación móvil, mediante las tecnologías y herramientas comentadas en el capítulo anterior, llamada Electronic Money Token o EMT que gestione saldos con total funcionalidad y uso práctico.

3.2.1 Alcance del proyecto

El objetivo principal de este trabajo es realizar una aplicación con la que se puede transferir tokens entre usuarios mediante diferentes medios y mayor facilidad al usuario de gestionarlos mediante el móvil.

La aplicación podrá:

- Enviar tokens del mismo evento entre usuarios.
- Usar alternativas de transmisión.
- Recibir tokens del mismo evento entre usuarios.
- Poder añadir saldos de distintos eventos.
- Visualizar un historial de transacciones.
- Registrarse en el sistema.
- Identificarse en el sistema.
- Añadir fondos a los saldos del sistema.

3.2.2 Metodología aplicada

El proyecto se aplicará el Proceso Unificado, impartida en la Universidad de Valladolid, marca unas pautas a seguir para desarrollar un software muy planificado. Como metodología no ágil presenta estas características:

- Centrado en la arquitectura.
- Dirigido por los casos de uso.
- Iterativo e Incremental.
- Enfocado en los riesgos.

Decidí llevar a cabo esta metodología porque es la que más hemos usado durante la carrera y no está tan enfocada al trabajo en equipo como pueden ser otras metodologías, como Scrum, tampoco estamos en constante relación con un cliente y ante posibles evolutivos que van cambiando según las especificaciones del cliente.

El proceso unificado se divide en cuatro etapas distintas:

Inicio:

- Establecer los objetivos y los límites del proyecto.
- Identificar los casos de uso más importantes del sistema.
- Estimar el costo global y la planificación de todo el proyecto.
- Especificar los riesgos del proyecto.

Elaboración:

- Garantizar que la arquitectura, los requisitos y los planes son lo bastante estables.
- Obtener una visión refinada del proyecto a realizar.
- Producir un prototipo del proyecto.

Construcción:

- Minimizar los costos de desarrollo optimizando los recursos e impidiendo las

reconstrucciones y los fragmentos innecesarios.

- Conseguir una calidad adecuada de forma rápida y práctica.
- Conseguir versiones útiles.
- Completar el análisis, diseño, desarrollo y prueba de toda la funcionalidad necesaria.

Transición:

- Ejecutar el despliegue.
- Conseguir que el usuario sea capaz de mantener el producto.
- Conseguir la aceptación por el usuario (stakeholder) que lo entregado es completo y consistente con el criterio de evaluación fijado en la visión inicial del proyecto.
- Obtener un producto final tan rápido y eficiente respecto al coste como práctico.

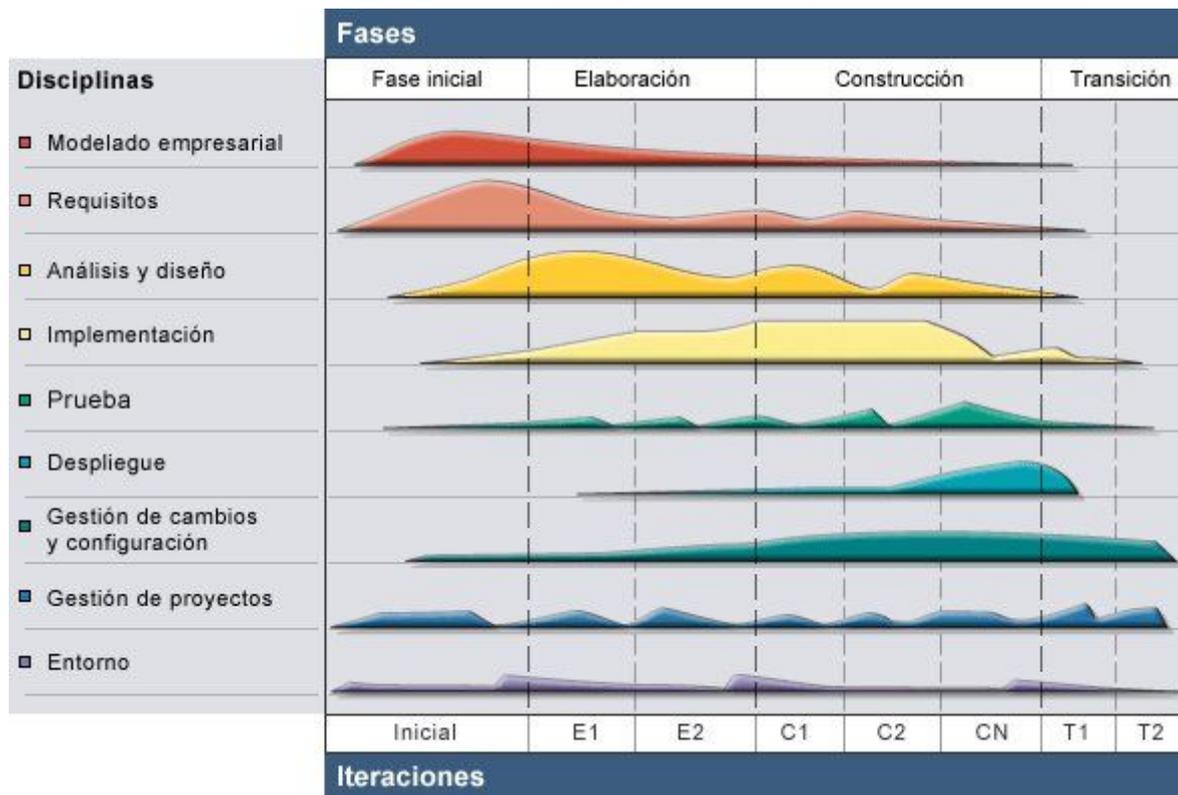


Figura 12: Fases de PU

Dentro de alguna de las etapa de desarrolla el método Waterfall/Cascada, donde se lleva el método más clásico del software.

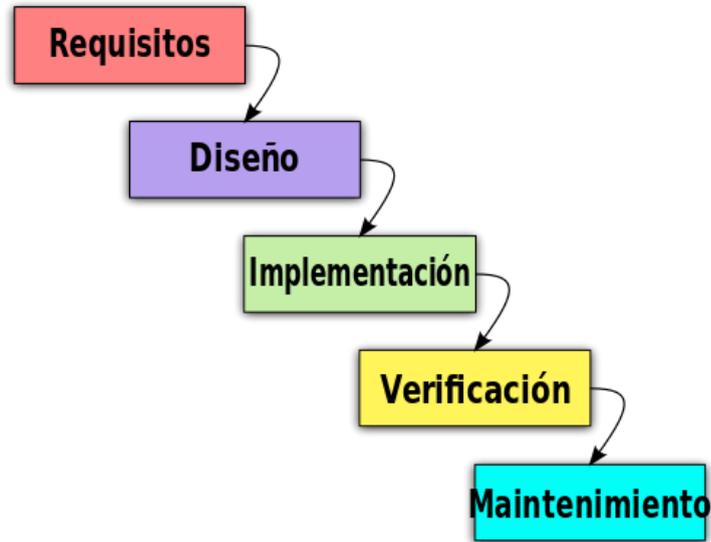


Figura 13: Fases del método en cascada

3.2.3 Equipo

Los distintos roles que se pueden encontrar en el proceso unificado, y que desempeñan en el proyecto son:

Decidí llevar a cabo esta metodología porque es el método más usado durante la carrera.

Rol	Desempeña
Analista	Borja Rabadán Martín
Desarrollador	Borja Rabadán Martín
Gestores	Borja Rabadán Martín
Pruebas	Borja Rabadán Martín

Tabla 1: Roles

- **Analista:** Encargado de analizar los requisitos, el sistema y el proyecto, para plantear un modelo de negocio acorde al alcance del proyecto.
- **Desarrollador:** Se dedica a desarrollar el software del sistema, escribiendo el código en función de los requisitos planteados por el analista.
- **Gestores:** Llevan la gestión del proyecto, el reparto de recursos y seguimiento del mismo.
- **Pruebas:** Realizan test en los códigos escritos, tratan errores y optimizan el código.

3.2.4 Plan de Riesgos

En este apartado se plantean posibles riesgos que pueden surgir a lo largo del desarrollo del proyecto, así como los planes de contingencia para cada uno de ellos.

Se realiza una matriz de análisis de riesgo para representar la relación entre la probabilidad que ocurra, y decidir una actuación en función de la tabla creada.

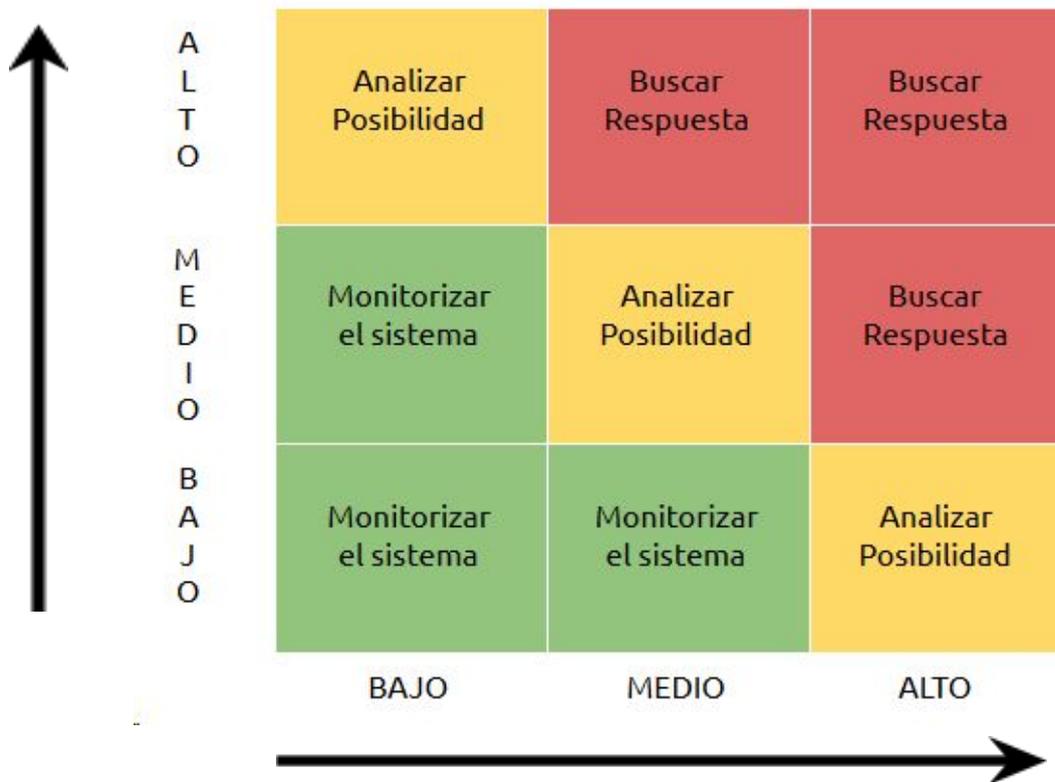


Figura 14: Tabla de riesgos

Lista de Riesgos

RA-01	Falta de experiencia del equipo de desarrollo
Descripción	La falta de experiencia por parte del equipo de desarrollo
Probabilidad	Media
Consecuencia	Mayor costes y retraso en la entrega
Impacto	Alto
Gestión del riesgo	

Estrategia	Reservar el riesgo
Plan de mitigación	Dedicar tiempo y dinero en formación
Plan de contingencia	Aumentar recursos

Tabla 2: Riesgo App 1

RA-02	Pérdida de datos
Descripción	Pueden perderse datos debido al fallo del hardware
Probabilidad	Bajo
Consecuencia	Repetir el trabajo perdido
Impacto	Alto
Gestión del riesgo	
Estrategia	Evitación del Riesgo
Plan de mitigación	No se aplica
Plan de contingencia	Subir copias de seguridad a Github

Tabla 3: Riesgo App 2

RA-03	Falta de disponibilidad del hardware
Descripción	No disponer de dispositivos para hacer pruebas
Probabilidad	Alto
Consecuencia	No poder probar algunas funcionalidades
Impacto	Medio
Gestión del riesgo	
Estrategia	Evitación del Riesgo
Plan de mitigación	No se aplica
Plan de contingencia	Solicitar dispositivos para la realización de pruebas

Tabla 4: Riesgo App 3

RA-04	Caída de Servidor
Descripción	Caída del servidor donde está la base de datos de la app
Probabilidad	Bajo
Consecuencia	No poder realizar la demostración

Impacto	Alto
Gestión del riesgo	
Estrategia	Reservar Riesgo
Plan de mitigación	Ejecutarlo en local
Plan de contingencia	Preparar una versión en local

Tabla 5: Riesgo App 4

RA-05	Diseño incorrecto
Descripción	No realizar un diseño que ayude a la implementación
Probabilidad	Medio
Consecuencia	Retraso en el proyecto
Impacto	Medio
Gestión del riesgo	
Estrategia	Protegerse del Riesgo
Plan de mitigación	Dedicar más tiempo en el diseño
Plan de contingencia	Rehacer el diseño

Tabla 7: Riesgo App 5

RA-06	Falta de personal
Descripción	Debido a la enfermedad o motivos personales
Probabilidad	Medio
Consecuencia	Aumento del coste y de retraso de la planificación
Impacto	Medio
Gestión del riesgo	
Estrategia	Asumir el Riesgo
Plan de mitigación	Contratar personal
Plan de contingencia	Aceptar el riesgo

Tabla 8: Riesgo App 6

3.2.5 Gestión del proyecto

3.2.5.1 Estimación

Teniendo en cuenta que este proyecto es la continuación de la práctica final de la asignatura de Sistemas Móviles, el tiempo que se ha tardado en realizarlo será mucho menor, ya que parto de una base pero no bien definidos.

Etapa	Fecha	Duración
Inicio	21 de enero	2 Semanas
Elaboración	4 de febrero	2 Semanas
Construcción	18 de febrero	7 Semanas
Transición	8 de abril	1 Semana

Tabla 9: Tiempo estimado del TFG

Dedicación diaria como máximo 2 horas, puede que le dedique menos tiempo algún día o incluso nulo, para ello usaba cronómetro para contabilizar el tiempo y que avisara del tiempo límite establecido.

3.2.5.2 Formación

La mayor parte de los conocimientos han sido adquiridos durante la carrera, sin embargo para el desarrollo de ciertas partes de android y el uso de firebase ha requerido cursos de formación.

Formación	Medio	Duración
Curso Avanzado Android	Udemy	5 horas
NFC y QR	Web Oficial	2 horas
Implementación Firebase	Web oficial	10 horas

Tabla 10: Tiempo dedicado a la formación

3.2.5.3 Calendario de Trabajo

La tabla de actividades realizadas en el plan de trabajo con un numero recursos igual a 1.

Número de Actividad	Actividad	Coste
1	Reunión inicial	1 hora
2	Esquema de la documentación del TFG	3 horas
3	Investigar acerca de Android, Java, AES y Firebase	6 horas
4	Investigar acerca de métodos de pago	4 horas
5	Investigar Herramientas de trabajo	3 horas
6	Realizar encuesta	2 horas
Entrega 1º		
7	Estructura del proyecto	2 hora
8	Alcance del proyecto	2 hora
9	Metodología aplicada	3 horas
10	Equipo de realización	2 horas
11	Plan de riesgos	3 horas
12	Presupuesto del proyecto	4 horas
Entrega 2º		
13	Requisitos Funcionales	5 horas
14	Requisitos No Funcionales	4 horas
15	Casos de uso	6 horas
16	Diagramas de secuencia	8 horas
17	Arquitectura y patrones	4 horas
18	BBDD	3 horas
19	Dominio	4 horas

20	Despliegue	3 hora
Entrega 3°		
21	Implementación-Login	3 horas
22	Implementación-Registro	3 horas
23	Implementación-Home	7 horas
24	Implementación-Recibir	5 horas
25	Implementación-Saldo	11 horas
26	Implementación-Enviar	9 horas
27	Implementación-Transferencia	5 horas
28	Implementación-Amigos	6 horas
29	Implementación-Configuración	4 horas
Entrega 4°		
30	Pruebas	6 horas
31	Depuración	3 horas
32	Conclusiones y posibles mejoras	1 hora

Tabla 11: Tiempo dedicado en el TFG

3.2.6 Gestión de configuraciones

Se realizan periódicamente copias de seguridad del software construido, para evitar la pérdida de información. Del mismo modo, también se sube configuraciones, documentación y archivos de instalación en caso de extraviar su contenido.

Usando una cuenta de GitHub privada proporcionada a los estudiantes, se sube cada día productivo una copia del trabajo actual, con commit que indique la fecha y el trabajo realizado ese día.

3.2.7 Presupuesto del proyecto

Se puede estimar dos visiones del presupuesto, una llevada al mundo laboral que se acerca más a la realidad y otra en la situación del estudiante, donde los costes son en mayor parte nulos o pequeños costes. Nos centraremos en los costes reales, con estimaciones obtenidas mediante estudios proporcionados por *Indeed* sobre salario medio de los distintos cargos en España.

Como en este proyecto se realizan dos aplicaciones, separo el presupuesto de ambos elementos, sin embargo algunos de ellos serán comunes para su planificación, como puede ser el mismo Hardware.

3.2.7.1 Recursos Humanos

Sólo contamos las horas necesarias para el proyecto que son 170 horas, por lo que calculamos según los sueldos anuales, el coste por hora de cada participante en el proyecto.

Rol	Salario Anual	€/Hora	Horas estimadas	Costes Total
Desarrollador Android	30.692 €	15,98 €/H	170 Horas	3675€*
Analista	27.560 €	14,35 €/H		
Diseñador	18.230 €	9,49 €/H		
Jefe de proyecto	34.834 €	18,14 €/H		

Tabla 12: Presupuesto para los roles

*Costes totales redondeados según el porcentaje de participación empleado por cada uno.

3.2.7.2 Recursos Hardware y Software

Costes Hardware

Partiendo de la amortización del equipo usado en el proyecto según la Agencia tributaria, tendremos que aplicar un coeficiente lineal del 25%, y como sólo se usará para 3 meses del proyecto, se reducen los costes.

Dispositivo	Precio	Coste al año	Coste al mes	3 Meses de Proyecto
Xiaomi MI6 6GB	240€	60 €/Año	5 €/Mes	15€
Lenovo G550	400€	100 €/Año	8,33 €/Mes	25€

Tabla 13: Presupuesto del hardware

Costes Software

La mayor parte del software usado es de código abierto, open source y gratuito, otros se han usado versiones de prueba.

Software	Coste al año	Coste al mes	3 Meses de Proyecto
Office 2013	105,6 €/Año	8,8 €/Mes	26,4€
GitHub	108 €/Año	9 €/Mes	27€
Google Play	22,50€ (De por vida)		
Firebase	268,68 €/Año	22.39€/Mes	67,17€

Tabla 14: Presupuesto del software

3.2.7.3 Costes Indirectos

Costes que están presentes en el proyecto de manera indirecta, como el espacio de trabajo.

Software	Detalles	Coste al mes	3 Meses de Proyecto
Oficina	Centro de ciudad(27m2)	180€/Mes	540€
Internet+llamadas	Fibra simétrica 300mb y llamadas ilimitadas	37 €/Mes	111€
Luz+Calefacción	uso normal	13,82€/Mes	41,46€

Tabla 15: Presupuesto de los costes indirectos

3.3 Análisis

3.3.1 Requisitos

3.3.1.1 Requisitos funcionales

Requisito	Descripción
RF-01	El sistema permitirá iniciar sesión.
RF-02	El sistema permitirá cerrar sesión.
RF-03	El sistema permitirá registrar a un usuario.
RF-04	El sistema notificará el acceso incorrecto.
RF-05	El sistema notificará el registro incorrecto.
RF-06	El sistema mostrará el evento más próximo.
RF-07	El sistema mostrará la última transacción.
RF-08	El sistema notificará el estado de una transacción.
RF-09	El sistema permitirá añadir saldos de eventos.
RF-10	El sistema permitirá mostrar todos los saldos del usuario.
RF-11	El sistema permitirá activar el lector QR.
RF-12	El sistema permitirá activar el lector NFC.
RF-13	El sistema permitirá agregar saldo al evento.
RF-14	El sistema permitirá eliminar saldos de evento.
RF-15	El sistema notificará al eliminar el saldo de un evento.
RF-16	El sistema mostrará una lista de todas las transacciones.
RF-17	El sistema permitirá generar un código QR.
RF-18	El sistema permitirá enviar tokens.

RF-19	El sistema permitirá recibir tokens.
RF-20	El sistema permitirá buscar por alias.

Tabla 16: Lista de Requisitos Funcionales

3.3.1.2 Requisitos no funcionales

Requisito	Descripción
RNF-01	No se podrá enviar tokens sin saldo.
RNF-02	No se podrá recibir/enviar tokens de diferente eventos.
RNF-03	Un identificador único para cada usuario.
RNF-04	No se podrá añadir el mismo saldo al sistema.
RNF-05	Se notificara del pago.
RNF-06	No se podrá enviar tokens, si no existe el usuario.
RNF-07	El sistema deberá estar operativo al 99,99%.
RNF-08	El sistema pedirá permisos de la cámara.
RNF-09	El sistema pedirá permisos de la tecnología NFC.
RNF-10	El sistema usará bibliotecas externas para generar el código QR.
RNF-11	El sistema requiere la API 23 como mínimo (Android 6.0)
RNF-12	El sistema usará el sistema de autenticación de firebase.
RNF-13	El sistema usará el sistema de notificación de firebase.
RNF-14	El sistema está realizado de acuerdo a la ley de protección de datos.

Tabla 17: Lista de Requisitos No Funcionales

3.3.2 Casos de uso

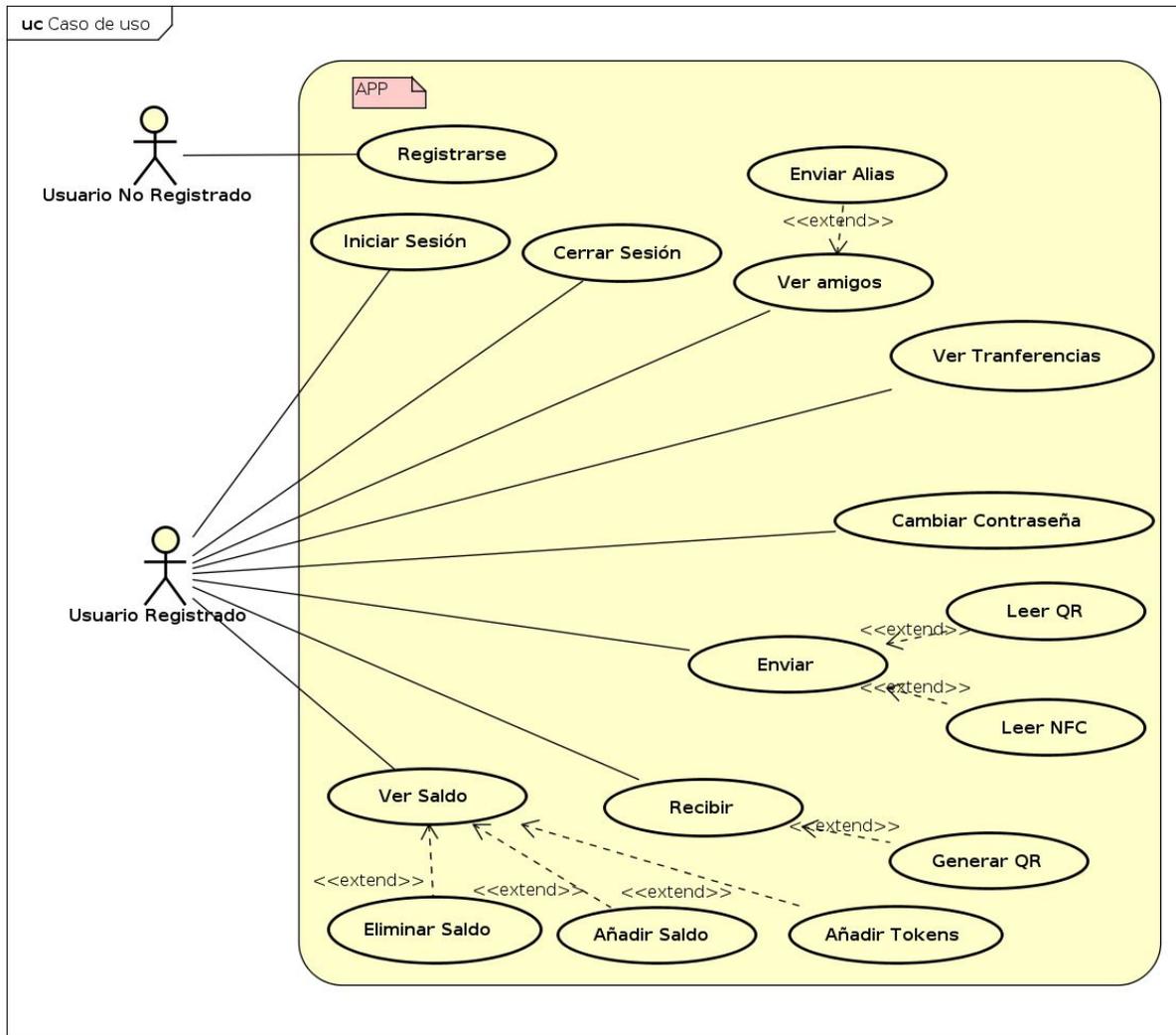


Figura 15: Casos de uso

CU-1	Iniciar Sesión
Actor	Usuario Registrado
Precondición	Estar registrado en el sistema
Postcondición	Entrar en el sistema
Flujo principal	<ol style="list-style-type: none"> 1. El usuario entra en la app 2. El sistema pide correo y usuario 3. Introduce los datos 4. Selecciona el botón "login" 5. El sistema le identifica y le envía al menú principal. 6. El sistema muestra la información correspondiente del usuario
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) 2) Cancela la operación, y no se realiza ninguna acción • 3a) No existe el usuario en el sistema, y vuelve al paso 1

	<ul style="list-style-type: none"> • 3b) La contraseña es correcta , y vuelve al paso 1 • 3c) No cumple con el formato del correo , y vuelve al paso 1
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 18: Caso de uso 1

CU-2	Cerrar Sesión
Actor	Usuario Registrado
Precondición	Estar dentro del sistema
Postcondición	Salir del sistema
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Configuración" 2. El usuario selecciona "Cerrar sesión" 3. El sistema elimina la sesión actual del usuario 4. El sistema le lleva al login
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) Cancela la operación, y no se realiza ninguna acción

Tabla 19: Caso de uso 2

CU-3	Recibir
Actor	Usuario Registrado
Precondición	Estar dentro del sistema
Postcondición	Mostrar los eventos que puede recibir tokens.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Recibir" 2. El sistema mostrará una lista de saldos del usuario
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) Cancela la operación, y no se realiza ninguna acción

Tabla 20: Caso de uso 3

CU-4	Ver Amigos
Actor	Usuario Registrado
Precondición	Estar dentro del sistema
Postcondición	Mostrar los eventos que puede recibir tokens.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Amigos" 2. El sistema mostrará una lista de saldos del usuario
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) Cancela la operación, y no se realiza ninguna acción

Tabla 21: Caso de uso 4

CU-5	Ver saldos
Actor	Usuario Registrado
Precondición	Estar dentro del sistema
Postcondición	Mostrar una lista de saldos con su información.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Saldo" 2. El sistema muestra una lista de saldos con su información correspondiente
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) Cancela la operación, y no se realiza ninguna acción

Tabla 22: Caso de uso 5

CU-6	Añadir saldos
Actor	Usuario Registrado
Precondición	Estar dentro del sistema
Postcondición	Añadir un nuevo saldo de un evento.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Saldo" 2. El sistema muestra una lista de saldos con su información correspondiente 3. El usuario selecciona "Añadir" 4. El sistema abre la cámara 5. El sistema lee el código QR con la cámara 6. El sistema comprueba que no tenga ese evento 7. El sistema agrega ese saldo
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) 3a) Cancela la operación, y no se realiza ninguna acción • 4a) La cámara no está operativa • 5a) Ya disponga de ese saldo

Tabla 23: Caso de uso 6

CU-7	Ver transferencias
Actor	Usuario Registrado
Precondición	Estar dentro del sistema
Postcondición	Mostrar una lista de transacciones con su información.
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona "Transferencias"

principal	2. El sistema muestra una lista de transferencias con su información correspondiente
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) Cancela la operación, y no se realiza ninguna acción

Tabla 24: Caso de uso 7

CU-8	Generar QR
Actor	Usuario Registrado
Precondición	Estar dentro del sistema
Postcondición	Visualizar un código QR
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Recibir" 2. El sistema muestra una lista de eventos 3. El sistema pide una cantidad 4. El usuario introduce los datos 5. El usuario selecciona "QR" 6. El sistema genera un código QR
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) 5a) Cancela la operación, y no se realiza ninguna acción • 4a) Es negativo la cantidad, y el sistema lo notifica

Tabla 25: Caso de uso 8

CU-9	Enviar
Actor	Usuario Registrado
Precondición	Estar dentro del sistema
Postcondición	Menú de enviar
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Enviar" 2. El sistema muestra una lista de eventos y posibles metodos de envio
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) Cancela la operación, y no se realiza ninguna acción

Tabla 26: Caso de uso 9

CU-10	Leer QR
Actor	Usuario Registrado
Precondición	Estar dentro del sistema
Postcondición	Enviar Tokens
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Enviar" 2. El sistema muestra una lista de eventos y posibles metodos de envio 3. El usuario selecciona "QR" 4. El sistema abre la camara 5. El sistema lee el código QR con la cámara 6. El sistema comprueba que tengan el mismo eventos 7. El sistema comprueba que disponga de saldo 8. El sistema envía los tokens 9. El sistema crea sus correspondiente transacciones
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) 2a) Cancela la operación, y no se realiza ninguna acción • 4a) La camara no esta disponible, no abre el lector • 5a) El código no corresponde a nada • 7a) 6a) Se notifica de que no tienen el saldo

Tabla 27: Caso de uso 10

CU-11	Leer NFC
Actor	Usuario Registrado
Precondición	Estar dentro del sistema
Postcondición	Enviar Tokens
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Enviar" 2. El sistema muestra una lista de eventos y posibles metodos de envio 3. El usuario selecciona un evento 4. El usuario selecciona "NFC" 5. El sistema comprueba que tengan el mismo eventos 6. El sistema comprueba que disponga de saldo 7. El sistema envía los tokens 8. El sistema crea sus correspondiente transacciones
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) 2a) Cancela la operación, y no se realiza ninguna acción • 4a) NFC no está disponible, no realiza nada • 5a) El NFC no corresponde a nada • 7a) 6a) Se notifica de que no tienen el saldo

Tabla 28: Caso de uso 11

CU-12	Añadir Tokens
Actor	Usuario Registrado
Precondición	Estar dentro del sistema
Postcondición	Sumar una cantidad de tokens al saldo
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Saldo" 2. El sistema muestra una lista de saldos con su información correspondiente 3. El usuario selecciona un saldo de la lista 4. El sistema pedirá una cantidad 5. El usuario introduce la cantidad 6. El sistema comprueba la cantidad 7. El sistema añade la cantidad
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) 3a) Cancela la operación, y no se realiza ninguna acción • 5 a) La cantidad no corresponde y lo notifica

Tabla 29: Caso de uso 12

CU-13	Eliminar Saldo
Actor	Usuario Registrado
Precondición	Estar dentro del sistema
Postcondición	Eliminar un saldo
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Saldo" 2. El sistema muestra una lista de saldos con su información correspondiente 3. El usuario desliza un saldo de la lista 4. El sistema notificará que se quiere borrar el evento 5. El usuario selecciona una opción 6. El sistema comprueba y realiza los cambios
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) 3a) Cancela la operación, y no se realiza ninguna acción • 5 a) Rechaza y vuelve al estado original

Tabla 30: Caso de uso 13

CU-14	Registrar Usuario
Actor	Usuario sin registrar
Precondición	-
Postcondición	Registrar el usuario en el sistema
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Sign up" 2. El sistema pide un usuario y contraseña 3. El usuario introduce los datos 4. El sistema comprueba los datos 5. El sistema notificará de se ha realizado y le devolverá al menú de inicio
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) 3a) Cancela la operación, y no se realiza ninguna acción • 4 a) Ya existe el usuario y lo notifica • 4 b) la contraseña no es válida y lo notifica

Tabla 31: Caso de uso 14

CU-15	Enviar Alias
Actor	Usuario Registrado
Precondición	Estar dentro del sistema
Postcondición	Enviar saldo a otro usuario
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Amigos" 2. El sistema muestra una lista de saldos 3. El usuario selecciona un saldo de la lista 4. El sistema pedirá una cantidad y un alias 5. El usuario introduce los datos 6. El sistema comprueba que exista el usuario 7. El sistema comprueba que tengan el mismo eventos 8. El sistema comprueba que disponga de saldo 9. El sistema envía los tokens 10. El sistema crea sus correspondiente transacciones
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) 3a) Cancela la operación, y no se realiza ninguna acción • 5 a) La cantidad no corresponde y lo notifica • 9a) Se notifica de que no tienen el saldo • 6a) Se notifica de que no existe el usuario • 7a) Se notifica de que no tiene el mismo evento en común

Tabla 32: Caso de uso 15

CU-16	Cambiar contraseña
Actor	Usuario Registrado
Precondición	Estar dentro del sistema
Postcondición	Modificar la contraseña actual
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Configuración" 2. El sistema le pide los datos 3. El usuario introduce los datos 4. El sistema comprueba los datos 5. El sistema realiza los cambios
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) 3a) Cancela la operación, y no se realiza ninguna acción • 4 a) La contraseña no coincide, vuelve a empezar • 4 b) No son iguales las contraseña nuevas, vuelve a empezar

Tabla 33: Caso de uso 16

3.3.3 Modelo de dominio

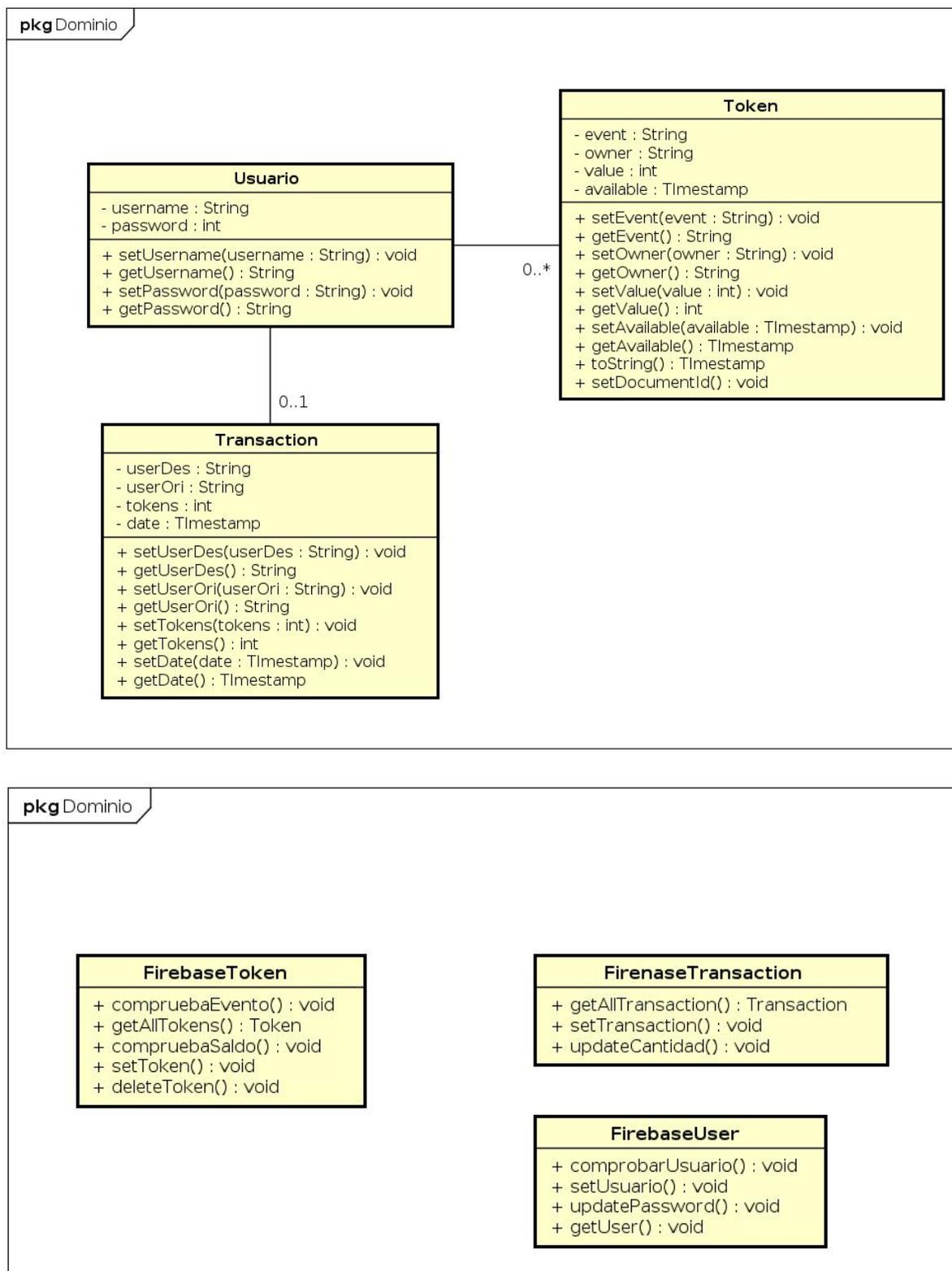


Figura 16: Modelo de dominio

3.3.4 Diagramas de secuencia

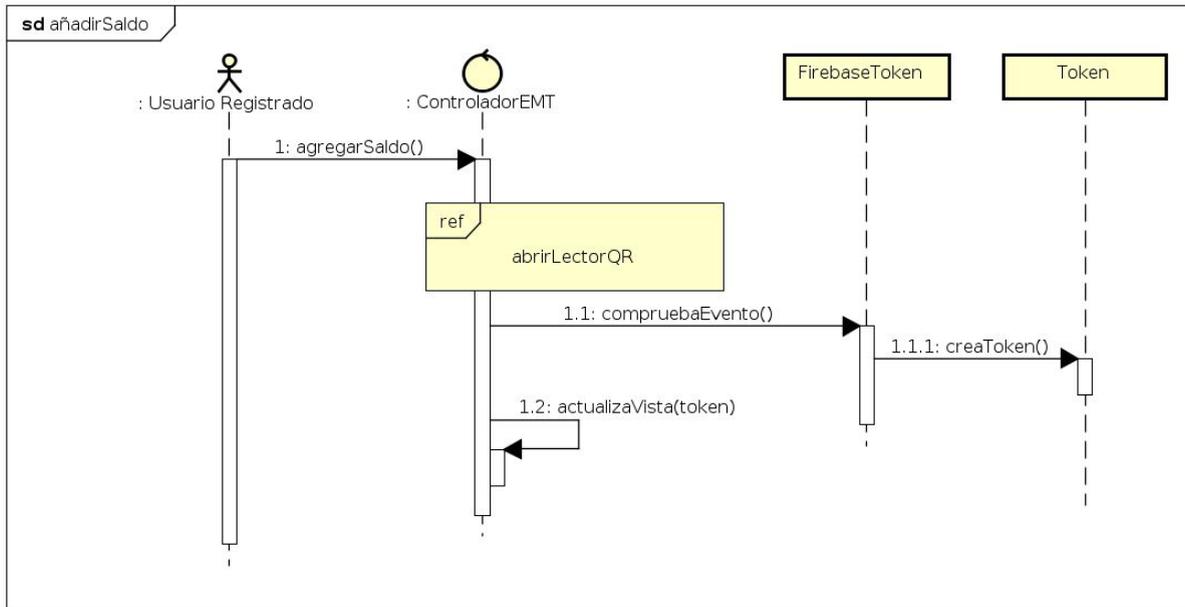


Figura 17: DS de añadir Saldo

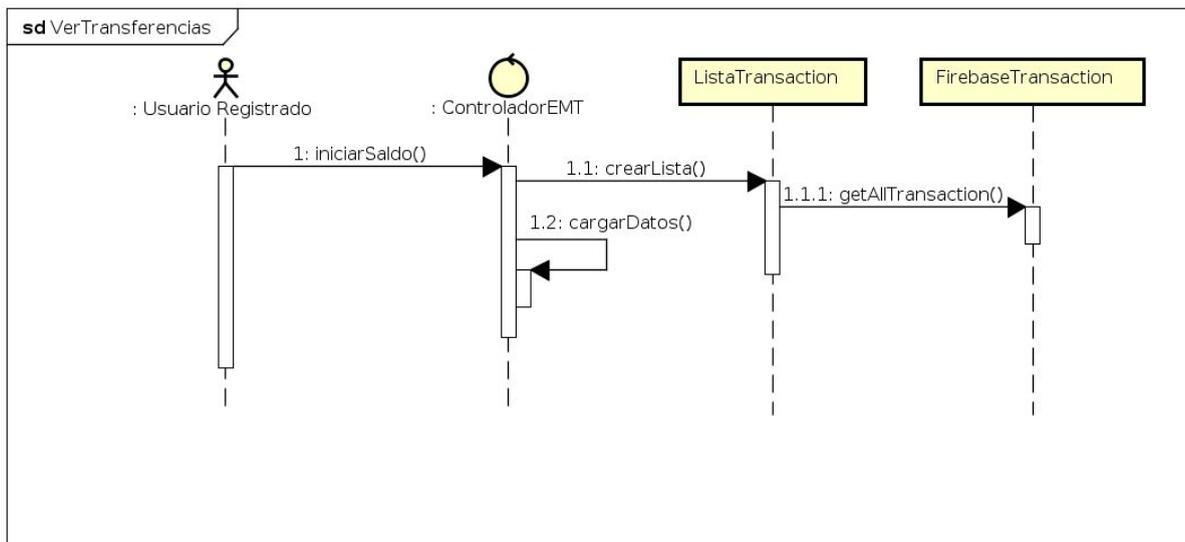


Figura 18: DS de ver transferencias

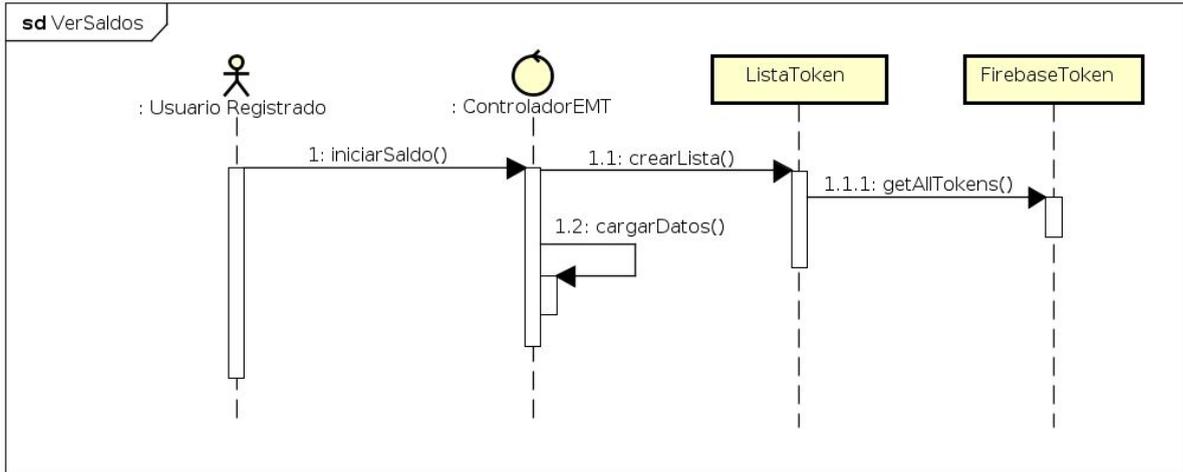


Figura 19: DS de ver Saldo

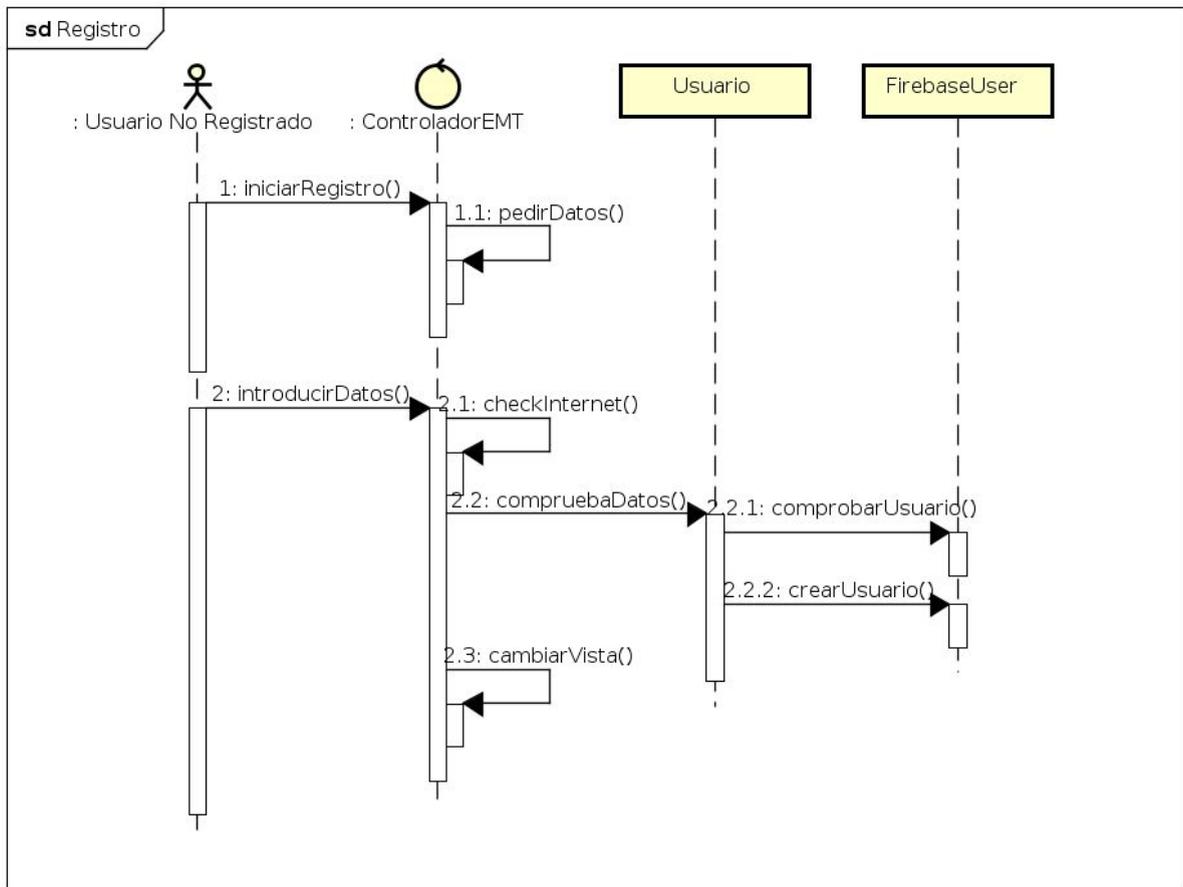


Figura 20: DS de Registro

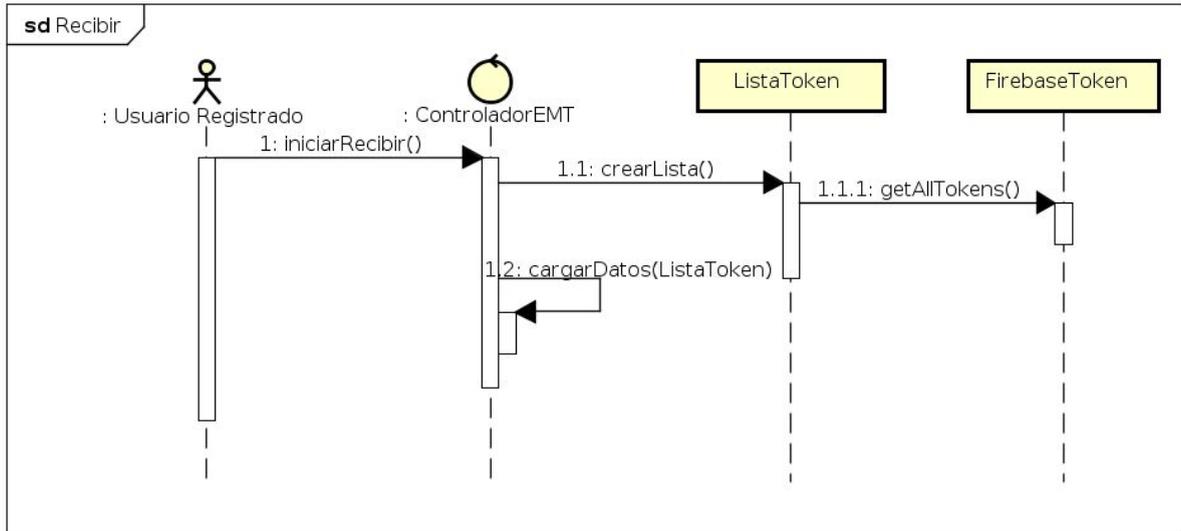


Figura 21: DS de Recibir

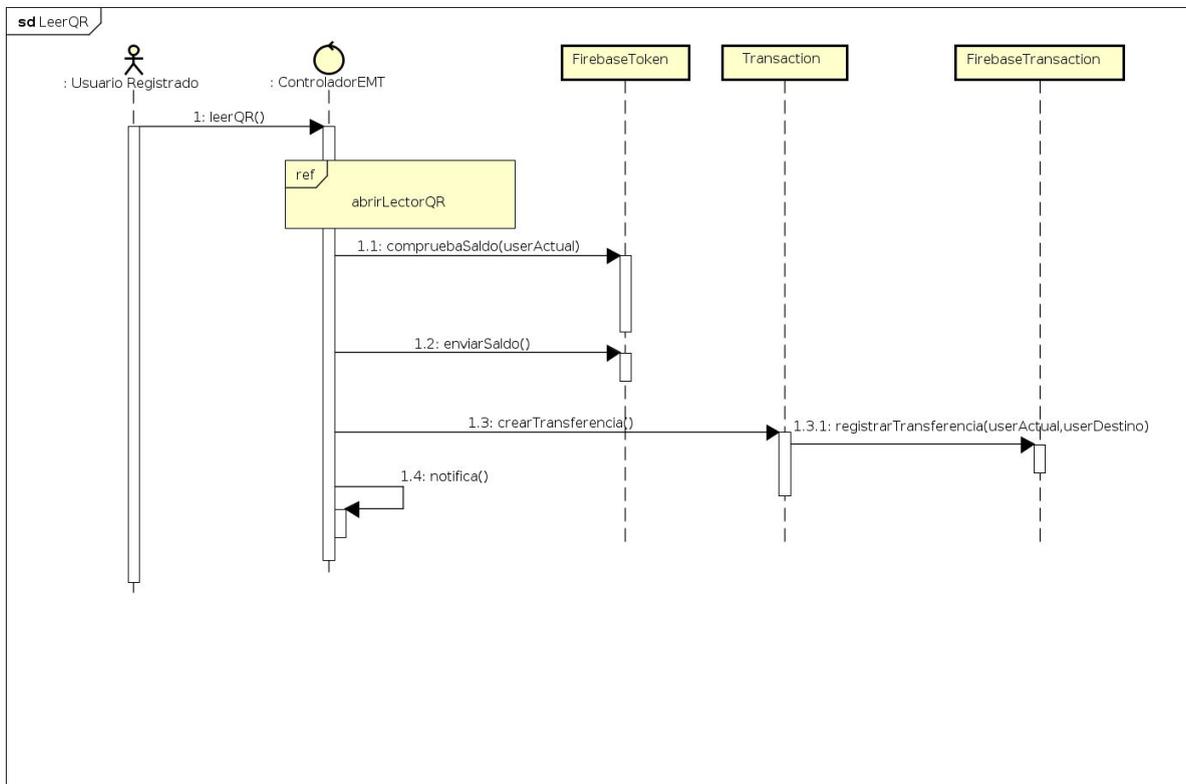


Figura 22: DS de LeerQR

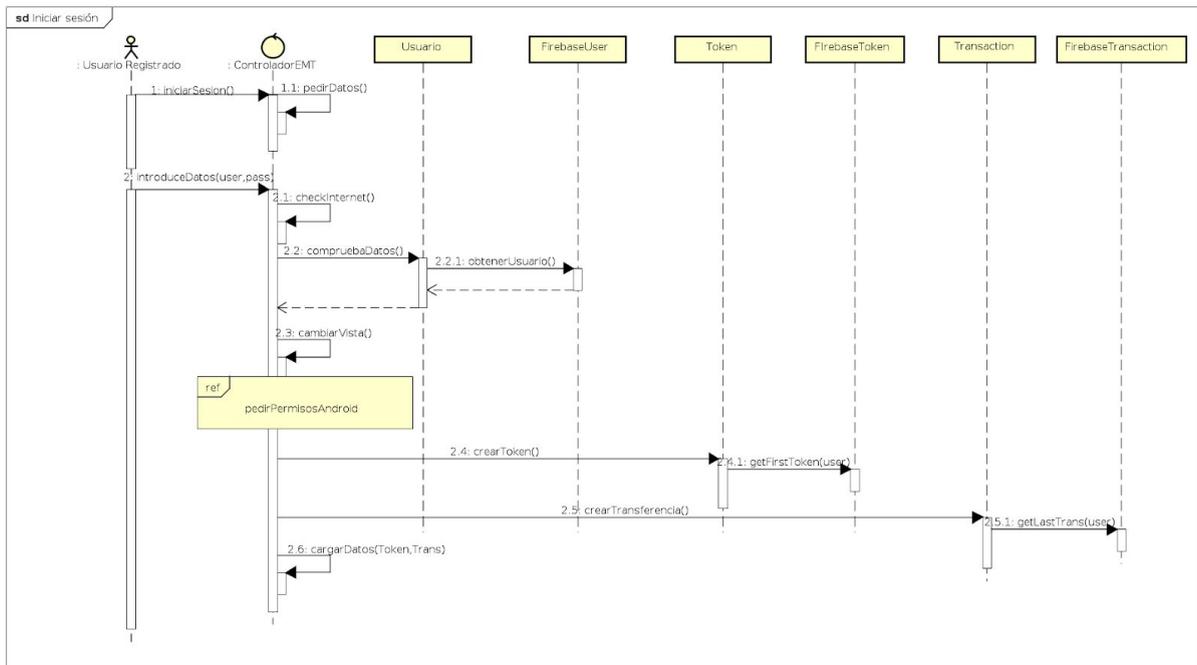


Figura 23: DS de Iniciar sesión

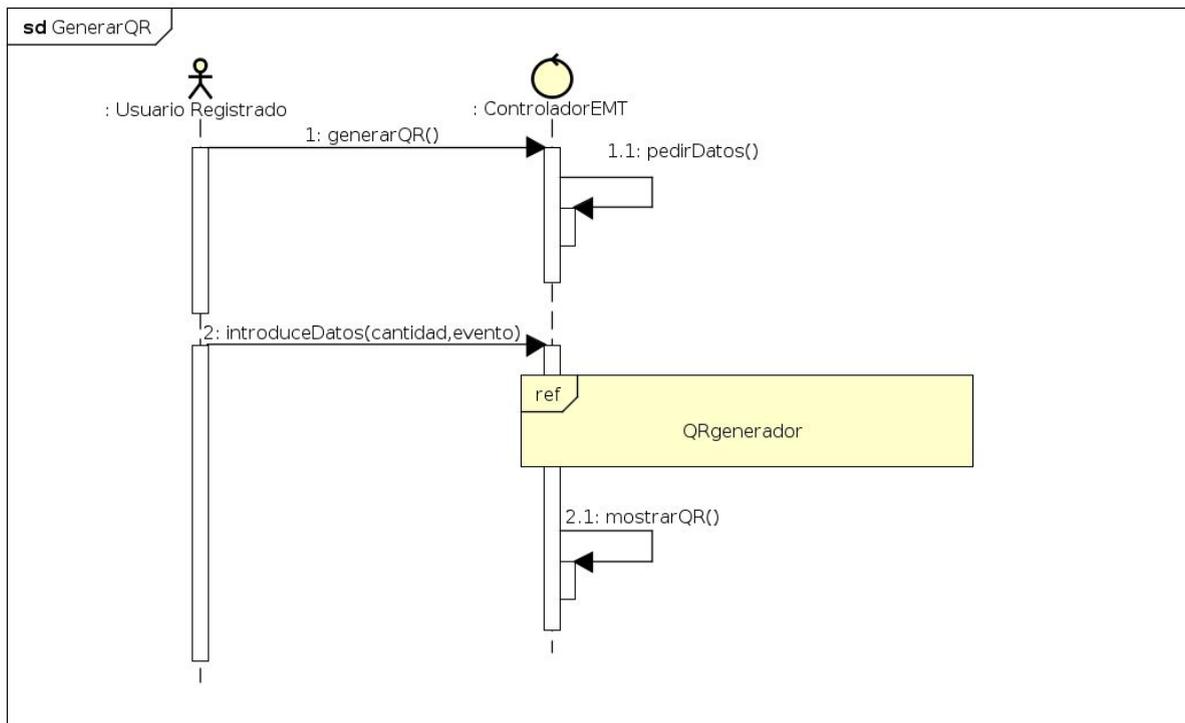


Figura 24: DS de GenerarQR

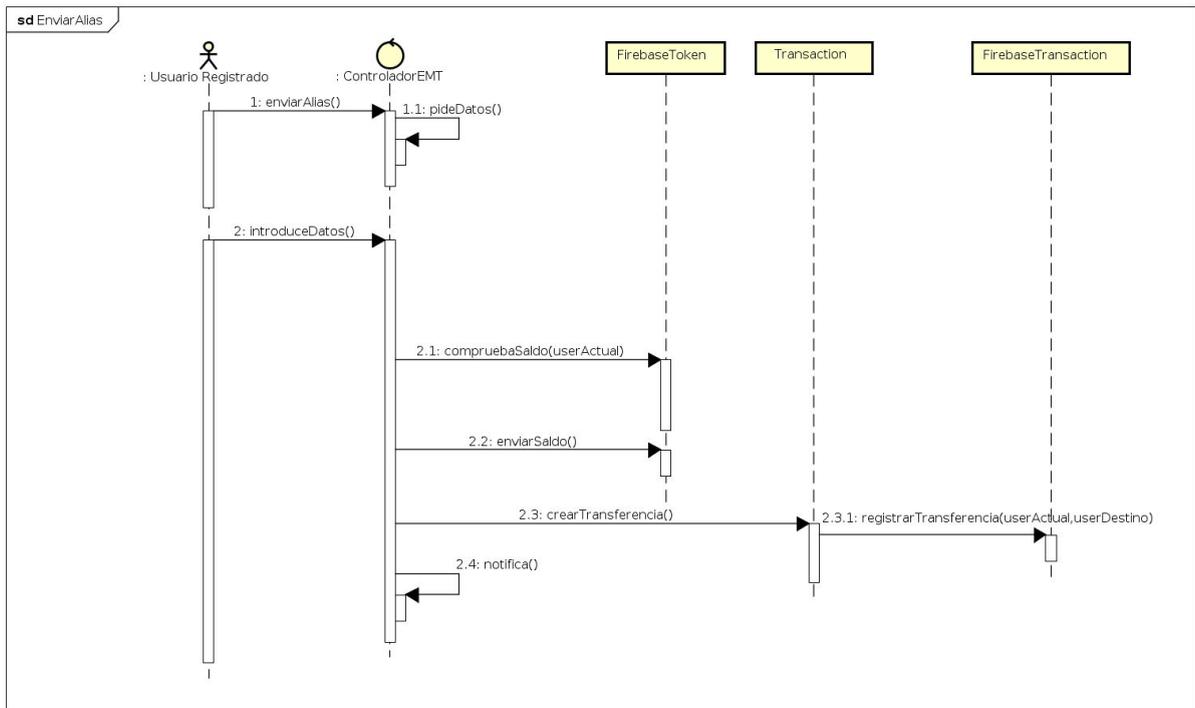


Figura 25: DS de EnviarAlias

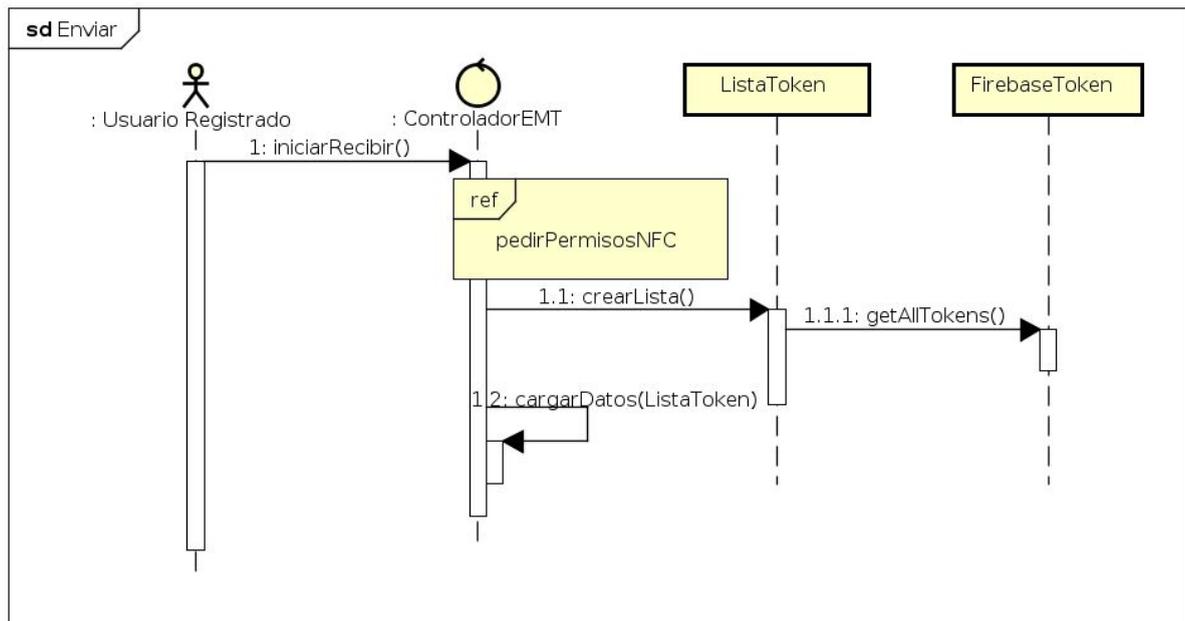


Figura 26: DS de Enviar

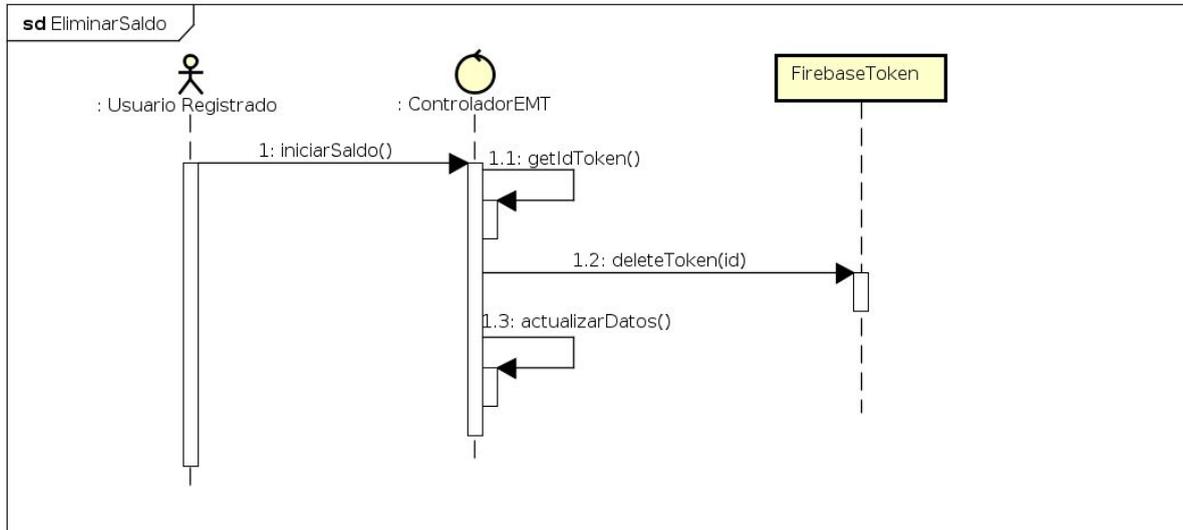


Figura 27: DS de EliminarSaldo

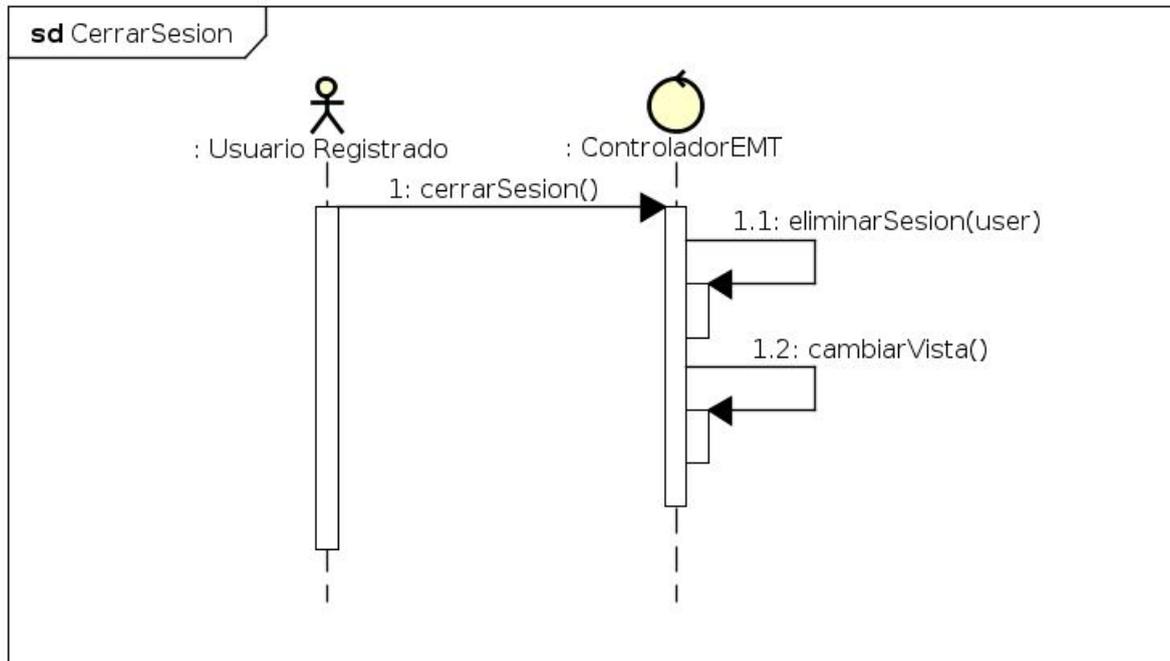


Figura 28: DS de CerrarSesion

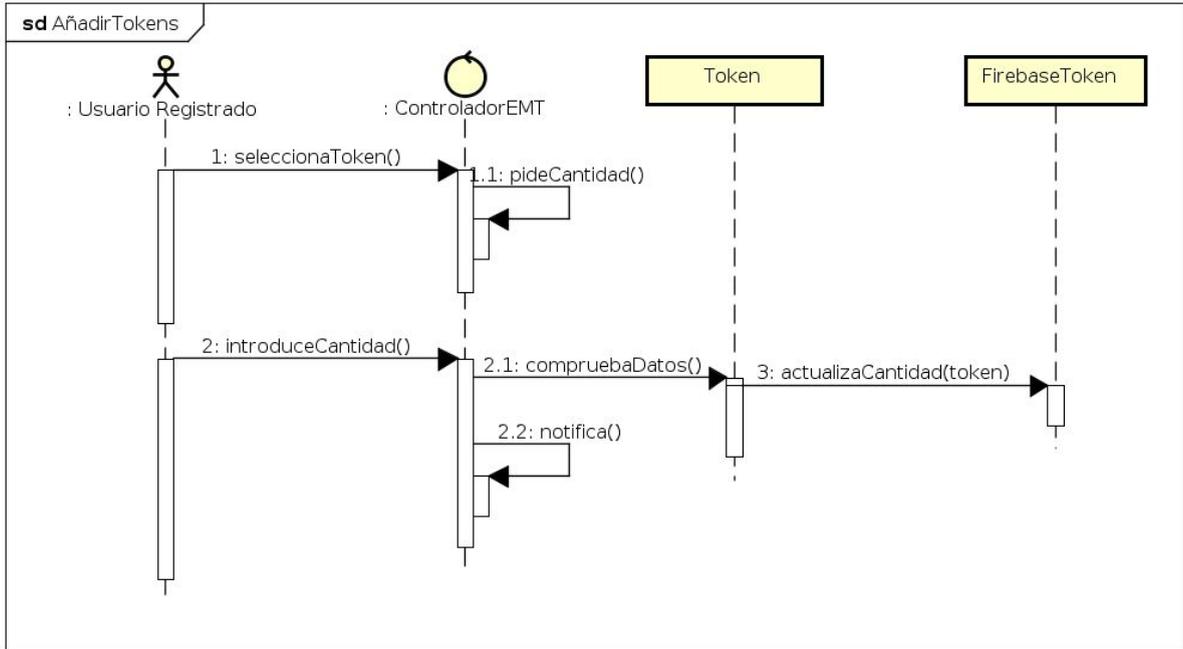


Figura 29: DS de AñadirTokens

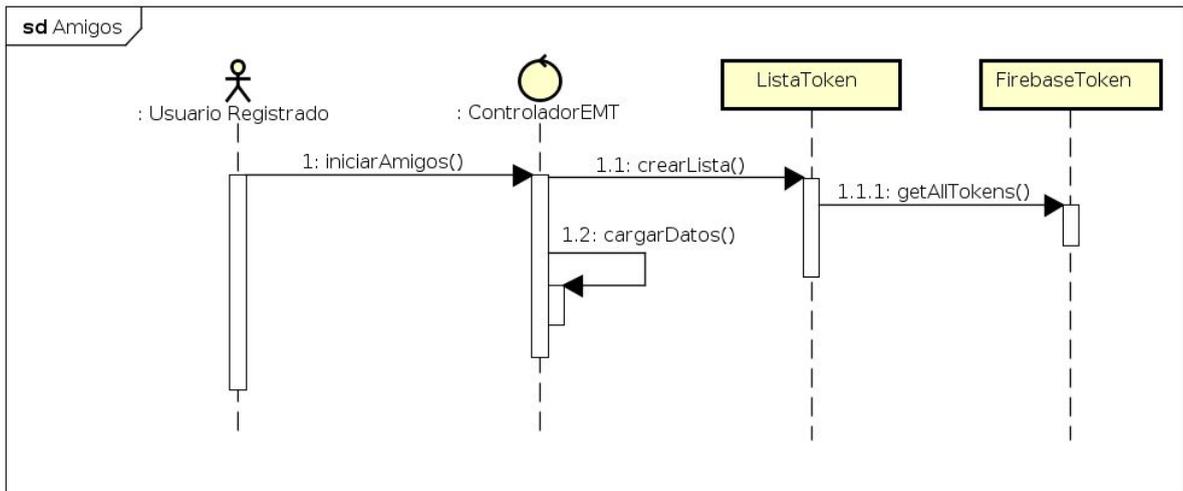


Figura 30: DS de Amigos

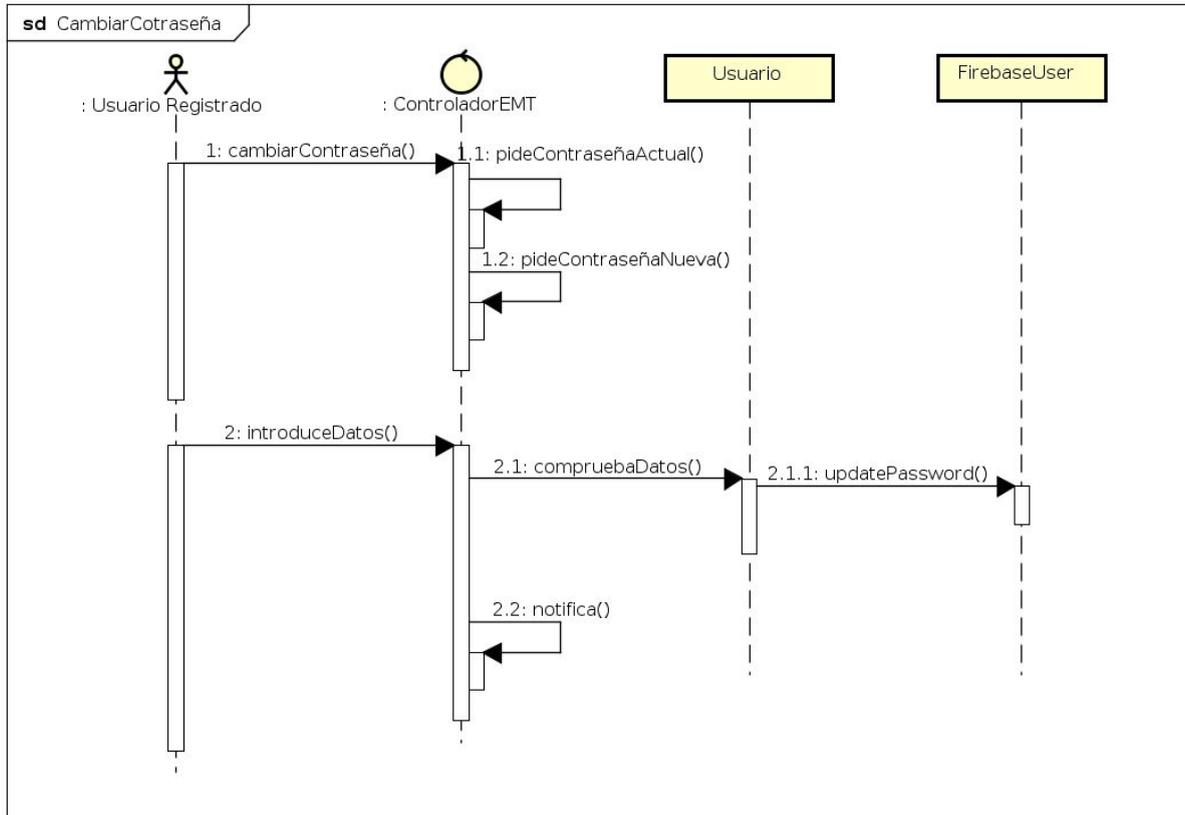


Figura 31: DS de CambiarContraseña

3.4 Diseño

3.4.1 Arquitectura

Model View Controller (MVC): Permite acometer cambios en la presentación de los datos con mayor facilidad, dando lugar a un software más mantenible. Mientras que la Vista y el Controlador se encontrarán en la arquitectura cliente (capa de presentación), el modelo permanecerá en los servidores o en el servidor central (capa de dominio).

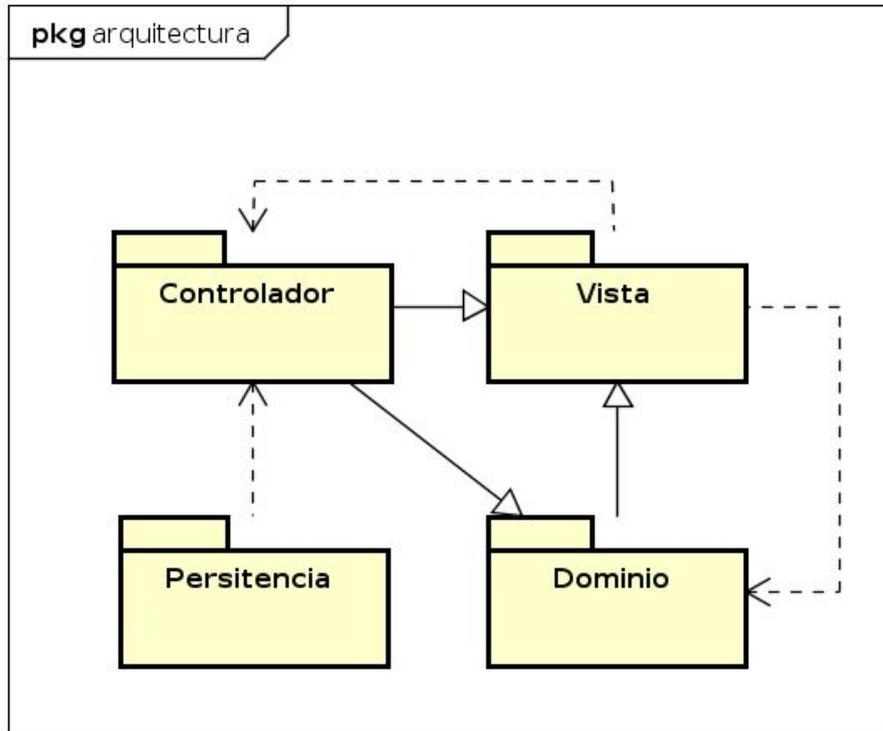


Figura 32: DP de la arquitectura

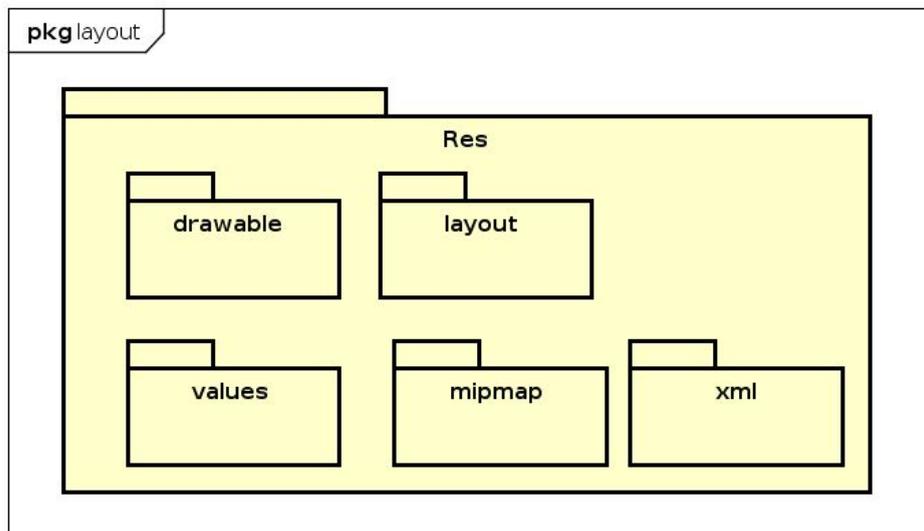


Figura 33: DP de los Recursos

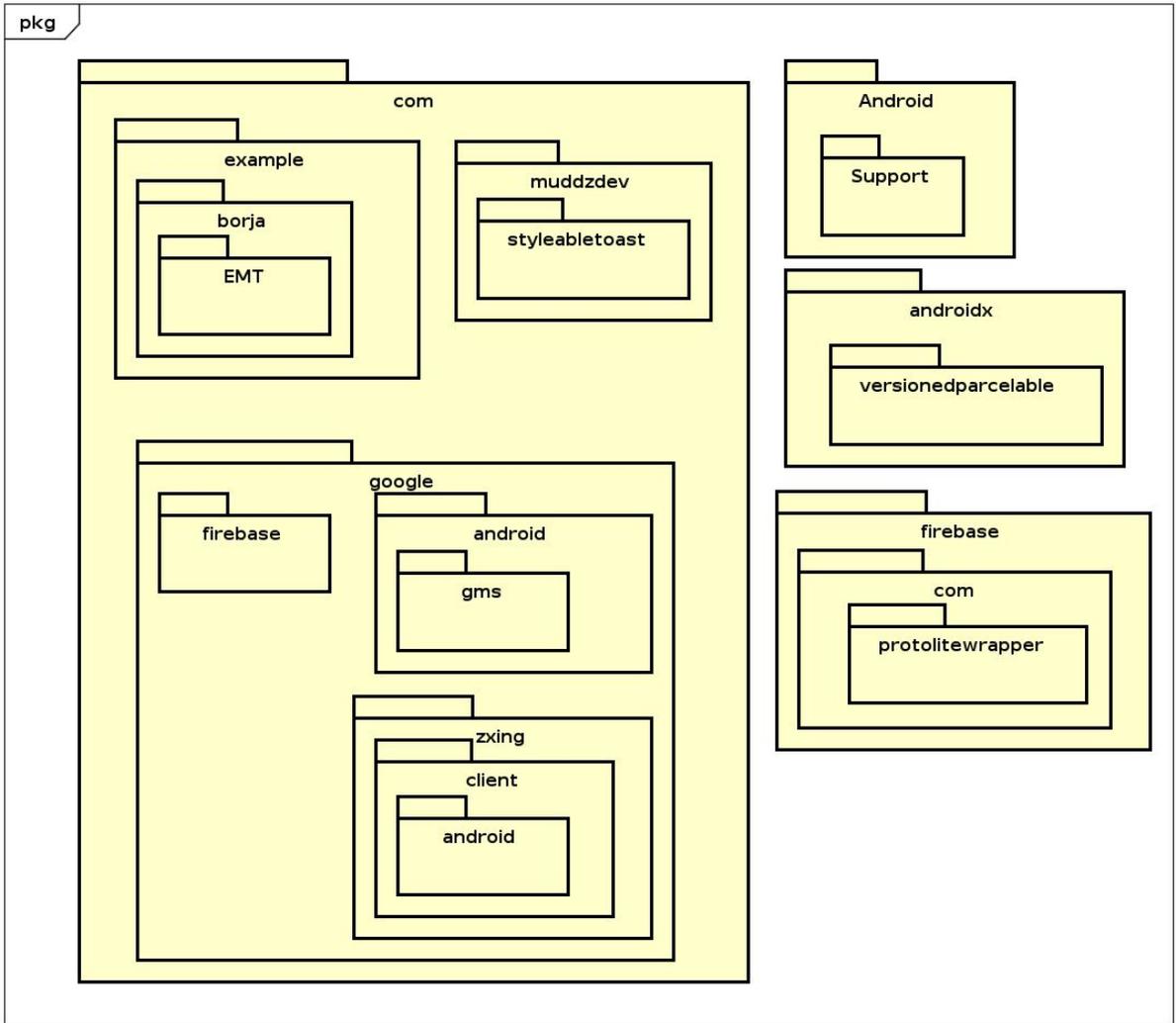


Figura 34: DP de la app

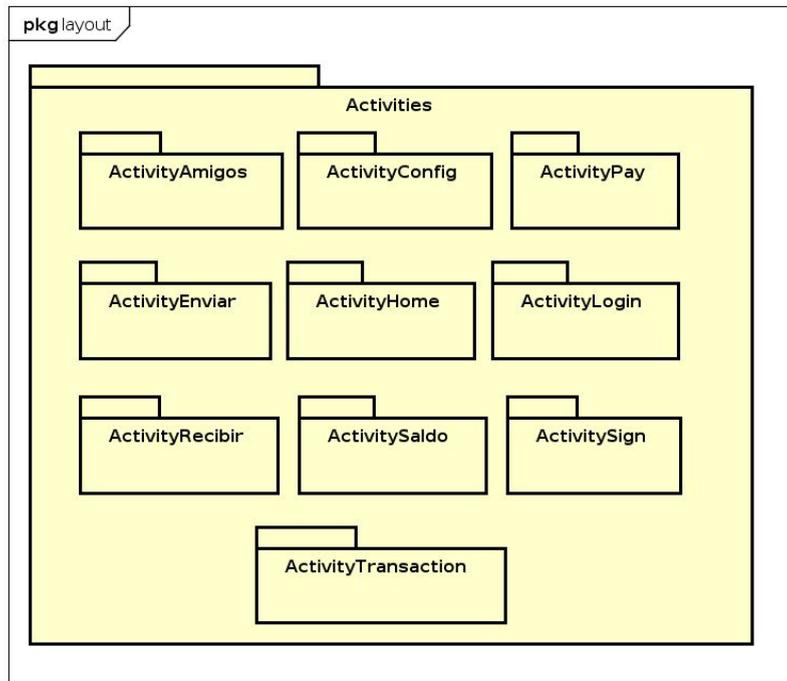


Figura 35: DP de la Activities

3.4.2 Patrones

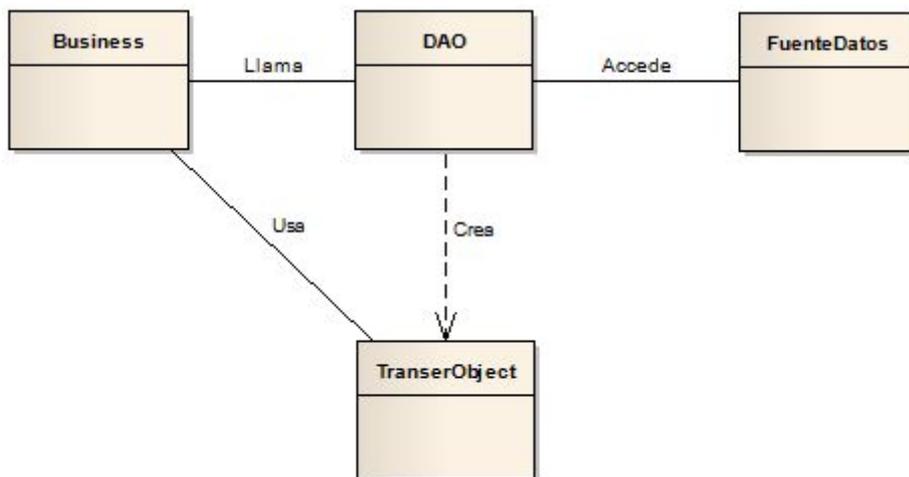


Figura 35: Diagrama del patrón DAO

Patrón DAO : Abstrae y encapsula el acceso a las fuentes de datos en una capa separada, proporciona una interfaz abstracta a bases de datos y otros mecanismos de persistencia, por lo que se gestiona la conexión con la fuente de datos para obtener y almacenar los datos. Se divide en GET, UPDATE, CREATE Y DELETE.

3.4.3 Modelo no relacional

Firestore usa actualmente una base de datos no relacional (NoSQL) de tipo Colección-Documento, mediante la documentación que proporciona Firebase acerca de Firebase Firestore y los conceptos básicos dados en Administración de base de datos de 4º de Ing Informática, he realizado este esquema.

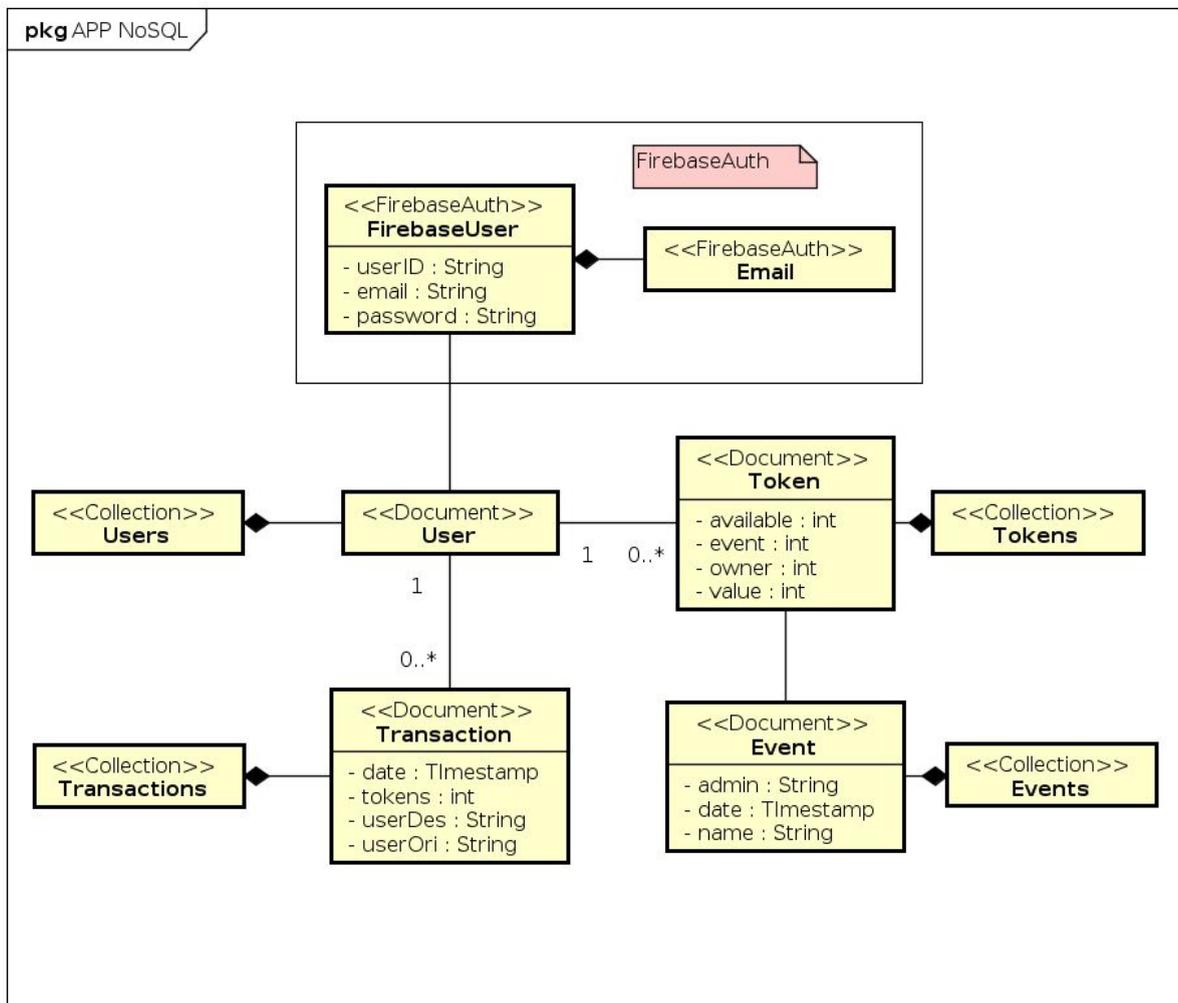


Figura 36: Modelo no relacional en firebase

3.4.4 Diagrama de despliegue

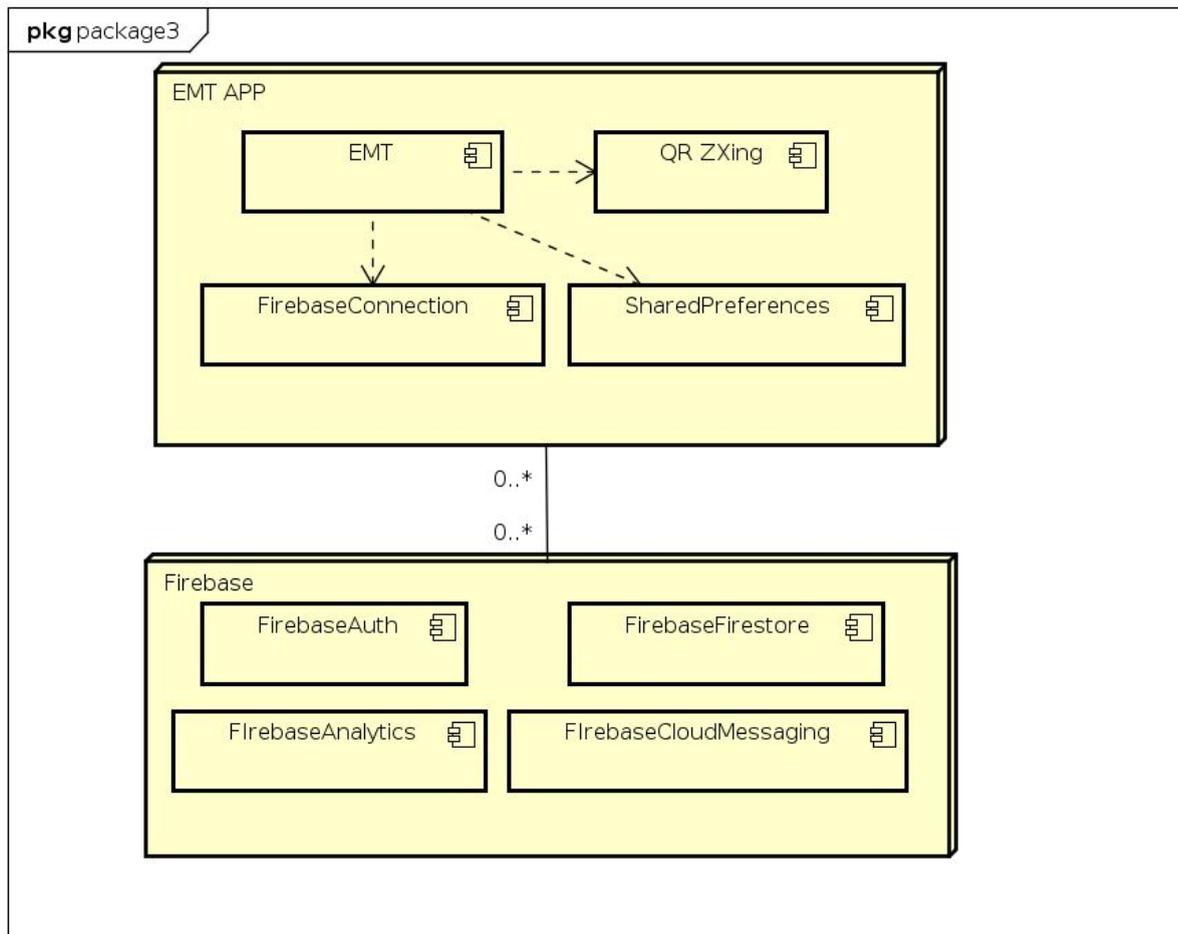


Figura 37: Diagrama del despliegue

3.5 Implementación

La implementación se ha estructurado en diferentes actividades, y dentro de ellas sus correspondientes funcionalidades. Siguiendo el MVC en Android, se muestra la vista implementada en XML, hasta el controlador gestionado por la actividad, e interactuar con el modelo de negocio y realizar peticiones en la base de datos.

Se han desarrollado las vistas mediante XML usando los layouts, que nos dan una gran posibilidad de realizar interfaces de manera fácil y adaptable.

Principalmente se ha utilizado el *ConstraintLayout*, es el más recomendado y se adapta más a las vistas que deseo implementar, algunas de estas vistas se han tenido que realizar usando otros Layout para mostrar el contenido de una manera más específica, como ya veremos en los saldos y transferencias.

Los controladores se han construido mediante actividades que llevan el ciclo de vida de las vistas y procesos, donde éstas puedan terminar una vez ya hayan sido utilizadas y pasar el control a otra actividad o dejarlas en pausa para su posterior uso. Dichos controladores han sido desarrollados en java mediante inyección de vistas, y así recibir los datos de los parámetro de entradas de éstas, para su posterior tratamiento. Mediante funciones se han construido las acciones de los principales botones para ejecutar las funcionalidades.

Para el acceso a la base de datos de firebase, son llamadas asíncronas las cuales obtenemos los datos de manera inmediata y podremos obtener actualizaciones al instante sobre la información solicitada

Login: gracias a Firebase y a su método de identificación mediante *FirebaseAuth*, podemos garantizar la seguridad de la cuenta de usuario y que nos devuelva los datos correspondientes, se han comprobado si existe el usuario, y si la contraseña es correcta. También se comprueba si existe conexión a internet para poder acceder a la aplicación.

Registro: al igual que el login, *FirebaseAuth* permite registrarse con total seguridad y crear un usuario único y con un id asociado. Se registra mediante usuario/password, sin embargo permite más opciones (no implementadas), también se comprueba que la contraseña esté bien escrita y si ya existe ese usuario.

Home: nos devuelve mediante distintas consultas: la última transferencia y el evento más próximo del usuario. Estas opciones son actualizadas al instante sin

tener que cambiar de actividad para ver los cambios gracias al snapshot de firebase. Las demás acciones son *intents* para cambiar de actividad según el propósito de la funcionalidad.

Configuración: tenemos dos posibilidades, cambiar la contraseña del usuario mediante una comprobación de datos, para una posterior actualización de la contraseña nueva, y cerrar la sesión que sólo es eliminar el usuario actual y llevar la actividad mediante un intent al login y cerrar todas las actividades abiertas mediante un flag.

Transferencias: mediante *Adapters*, y usando *CardView* combinado con *RecyclerView* podemos generar objetos de la base datos de manera dinámica y siguiendo una estructura.

Saldo: al igual que las Transferencias, usamos los mismos métodos pero añadiendo la posibilidad de que se puedan añadir acciones, en este caso la de deslizar mediante *swipe* y *clickListener* para borrar y cambiar de actividad. Además se realiza un builder con la librería de *Zxing*, que abre la cámara y reconoce el código QR, el resultado será procesado en un *Result* donde se comprueba si dispone de ese saldo, y en caso satisfactorio se le añade.

Amigos: mediante *spinner* se muestran los saldos del usuario, se realizan comprobaciones de los datos introducidos, para la transmisión de los mismos y así comprobar la disponibilidad de ese evento y si dispone de saldo.

Enviar: se reutilizan elementos del activity de Amigos, añadiendo el lector QR de *Zxing* para procesar códigos QR pero creados por la misma app, añadiendo encriptación en el código, para aumentar la seguridad . Para NFC se tienen dos versiones implementadas: BEAM y HCE, he usado las dos versiones de la app en cada tecnología de NFC, mediante NDEF son transmitidos los mensajes.

Recibir: volviendo a implementar el uso de *spinner* y usando la librería de *Zxing* para generar un QR de un tamaño específico que combinado con la encriptación se asegura el saldo que se envía.

BBDD: Usando las transacciones de Firebase nos permite mayor seguridad a la hora de realizar cambios en la base de datos. Todas las modificaciones erróneas en la base de datos volverían al estado anterior("Rollback"), y en caso de que la app funcione correctamente o la base de datos esté operativa, se realizaron los cambios ("Commit").

3.6 Pruebas

Para las pruebas, use Junit4 para hacer test unitarios y test de integración en android studio, y para averiguar la cobertura obtenida use Sonarqube.



Figura 38: Logo de sonarqube

Sonarqube es un software de código libre que nos permite buscar fallos en nuestro código y también posibles mejoras, además de poder comprobar la cobertura del código a través de los test de Junit implementados en android studio.

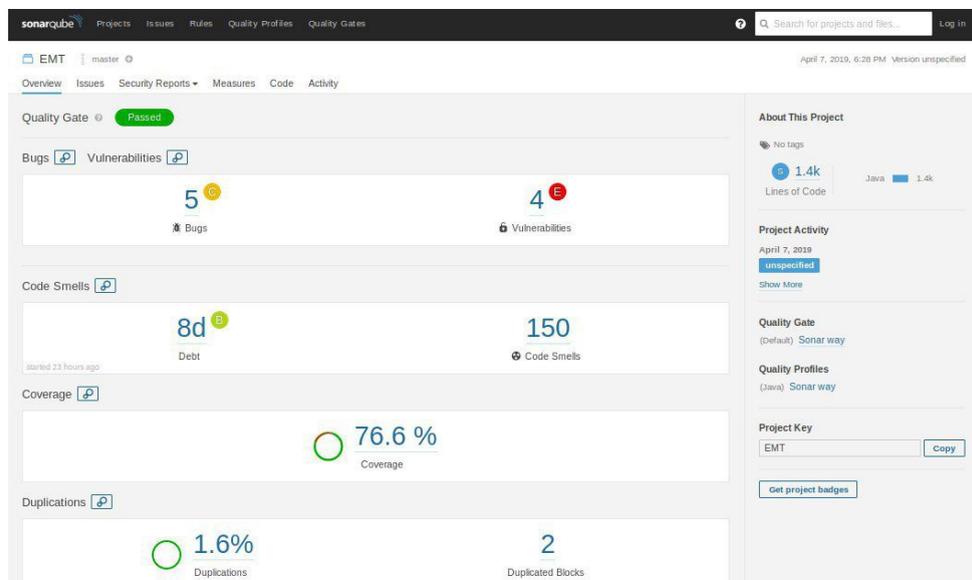


Figura 39: Resultados del sonarqube

Los resultados obtenidos son un porcentaje decente de cobertura(76,6%), lo recomendado por el programa es un 80% y a nivel laboral se suele pedir ese mínimo, también nos muestra bugs en el código y posibles vulnerabilidades, la mayoría son posibles nullpointer en la asignación de variables.

Los code smells, suelen ser más bien recomendaciones o buenas prácticas de desarrollo de código, donde te avisa de los posibles cambios ante malas prácticas de desarrollo de código y su motivo.

Por último, te indica si existe código repetido en el programa, para evitar líneas duplicadas que se podrían realizar en una misma implementación, o menos líneas.

Para la instalación tan sólo requiere iniciar el servidor de Sonarqube y añadir las dependencias en el gradle del proyecto de Android.

3.7 Manual de instalación

La instalación de la aplicación móvil tendrá que ser instalada mediante un APK fuera de Google Play. Una vez que se disponga el APK en el móvil, se tendrá que dar permisos de instalación de software desconocido:

1. Acceder a Ajustes.
2. Ir al apartado de Seguridad.
3. Seleccionar "Orígenes desconocidos".

Una vez realizado estos pasos, se podrá instalar la app.

Requisitos previos

- **Versión mínima:** Android 6.0 (Marshmallow).
- **Almacenamiento:** Se requiere un mínimo de 8 MBytes.
- **Conexión a internet:** Mediante wifi o Datos.

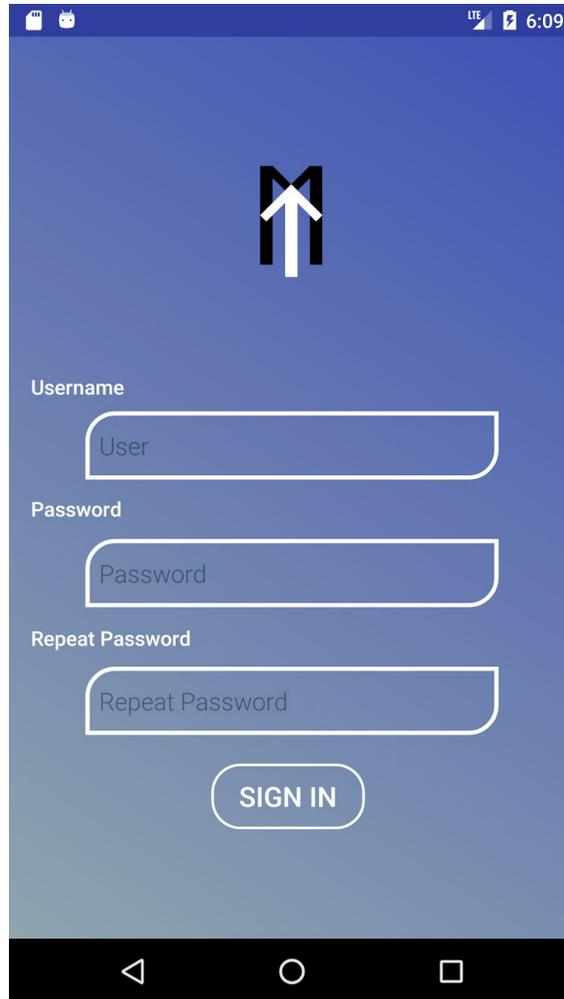
3.8 Guía de usuario

Inicio: al acceder a la aplicación nos aparecerá un método de identificación mediante usuario y contraseña, donde nos redireccionará a la cuenta. También podemos acceder a la pantalla de registro de EMT.



Figura 40: Captura del login

Registro: un registro estándar, con comprobaciones de contraseñas coincidan y el usuario ya exista en la BBDD .



The image shows a mobile application registration screen. At the top, there is a status bar with icons for signal strength, battery, and the time 6:09. Below the status bar is a dark blue header with a white logo consisting of a stylized 'M' with an upward-pointing arrow. The main content area has a light blue background and contains three input fields: 'Username' with the placeholder text 'User', 'Password' with the placeholder text 'Password', and 'Repeat Password' with the placeholder text 'Repeat Password'. Below these fields is a white button with rounded corners and the text 'SIGN IN' in blue. At the bottom of the screen is a black navigation bar with three white icons: a back arrow, a circle, and a square.

Figura 41: Captura del Registro

Home: EMT nos devolverá el saldo del evento más próximo y la última transacción realizada o recibida. Desde el home podremos acceder a las principales funciones de la aplicación:

- **Saldo**
- **Transferencia**
- **Configuración**
- **Enviar**
- **Amigos**
- **Recibir**

*Si es la primera vez que se instala en el dispositivo, se pedirá al usuario permisos para usar la cámara y la tecnología NFC para realizar algunas de las funciones de la aplicación, debido a que Android a partir de la versión 6.0 es obligatorio pedir permiso al usuario.



Figura 42: Captura del Home

Lista de transferencias: Nos devolverá el historial de operaciones realizadas entre los usuarios, fecha, participantes y cantidad.

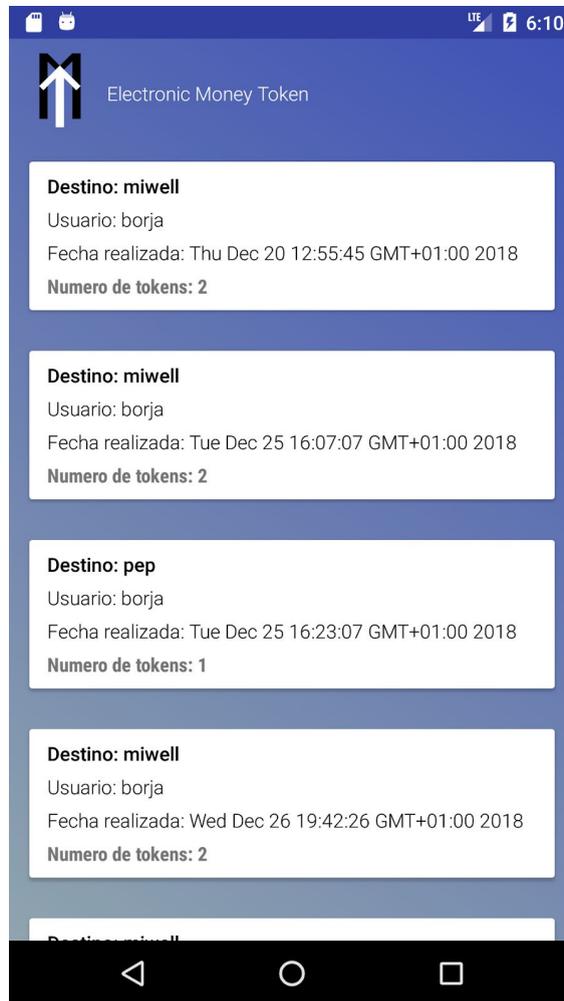


Figura 43: Captura de las Transferencias

Lista de saldos: Nos devolverá los saldos registrados por el usuario, con la cantidad disponible y su disponibilidad (a partir de la fecha fijada no se pueden usar). Deslizándolo el dedo (hacia la izquierda o derecha) sobre un saldo, podremos eliminarlo, nos avisará si queremos llevar a cabo la acción. Si seleccionamos un saldo, nos llevará a la pantalla de agregar tokens y si queremos añadir un saldo, con dar a AÑADIR abre el lector QR.

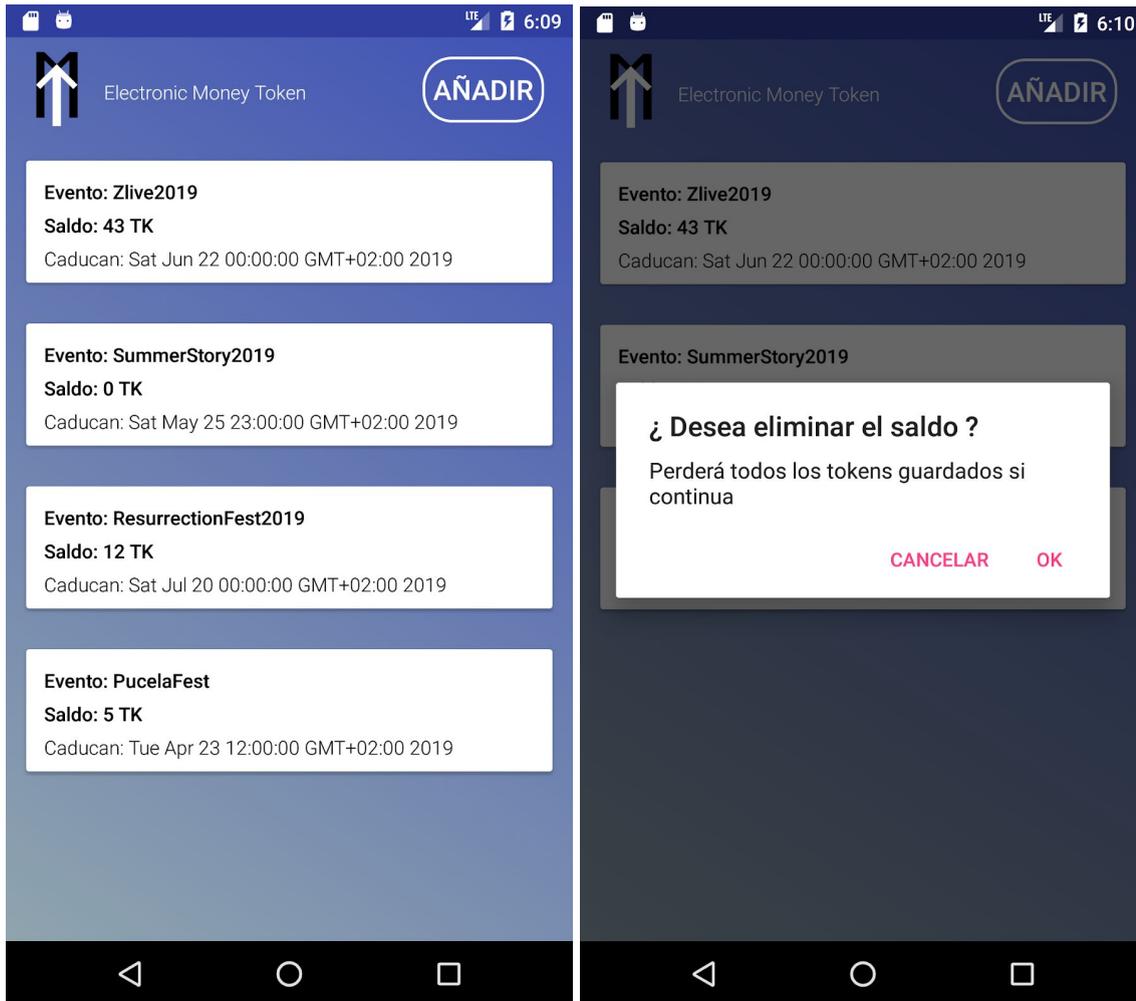


Figura 44: Captura del Saldo

Lector QR: un lector QR que nos permite escanear códigos QR en las funciones de añadir saldo y de enviar mediante QR.

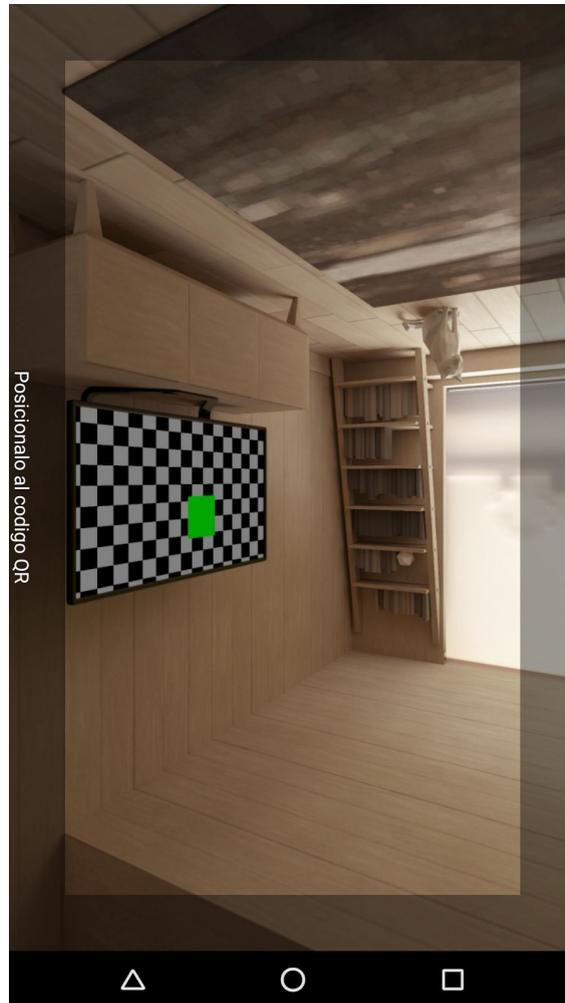


Figura 45: Captura del lector QR

Añadir tokens al saldo: tras pulsar sobre un saldo nos permite añadir una cantidad que sumará al saldo actual del evento.

*Simula una pasarela de pago, tanto Tarjeta de crédito y PayPal.



Figura 46: Captura de añadir de los tokens

Recibir: si queremos pedir a otro usuario una cantidad de tokens para un festival, podremos generar un código QR que será leído en el apartado de ENVIAR, mediante el lector QR. Tenemos que elegir entre los saldos que disponemos y una cantidad, tras seleccionar QR se nos genera el código QR.

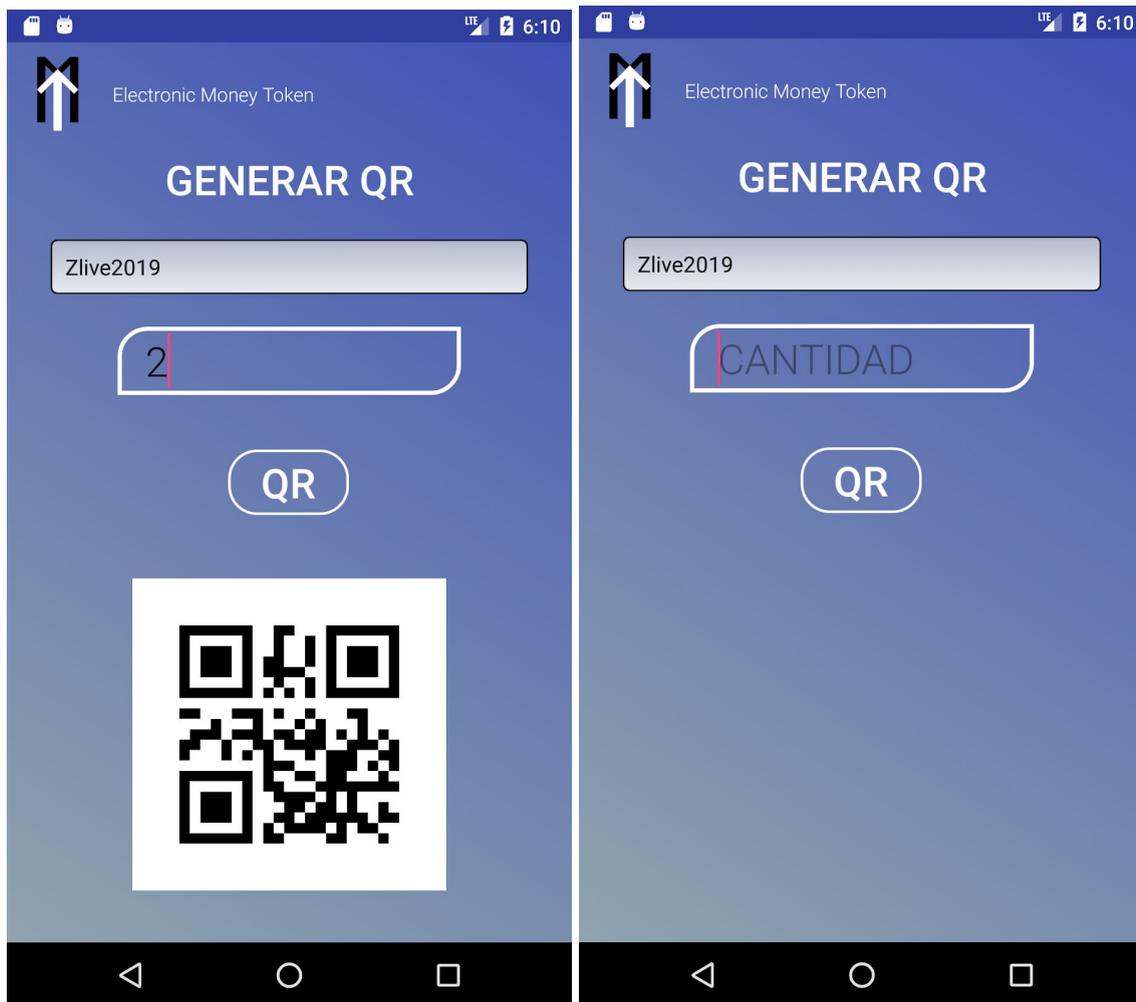


Figura 47: Captura del Generador QR

Enviar: se podrá elegir entre dos modalidades.

1. **QR** : usando este método, se abrirá un lector QR que sólo reconocerá dicho código creado por la aplicación. Una vez leído comprueba si disponemos de saldo y de ese evento, en caso contrario saltará un mensaje informando al usuario.
2. **NFC** : usando este método, se tendrá que seleccionar la cantidad y el evento que queremos enviar tokens. La aplicación comprueba si el móvil dispone de NFC y si está encendido. Ambos móviles tienen que estar en la misma pantalla para poder enviarlo, una vez leído comprueba si disponemos de saldo y de ese evento, en caso contrario saltará un mensaje informando al usuario.

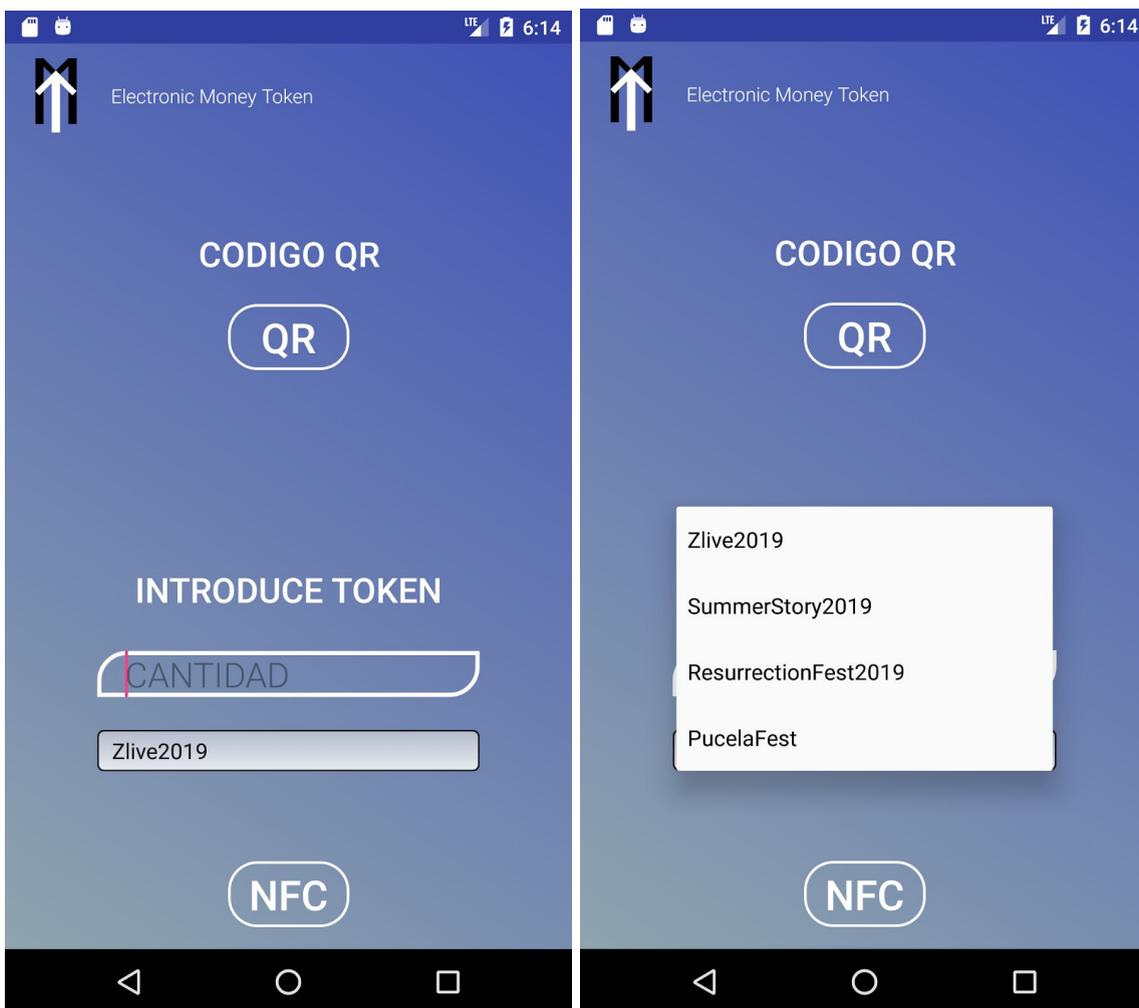


Figura 48: Captura de enviar

Enviar mediante alias: si conocemos el nombre del usuario, podremos enviarle tokens directamente, siguiendo el mismo procedimiento de añadir un nombre, cantidad y saldo de evento. En caso de que algún dato sea incorrecto se muestra un mensaje de error.



Figura 49: Captura de alias

Configuración: donde están los ajustes de la cuenta del usuario, como son el cambio de contraseña o el cierre de sesión .



Figura 50: Captura de Configuración

Capítulo IV

Electronic Money Token WEB

4.1 Aplicación Web

4.2 Plan de Desarrollo del Software

Con el objetivo de crear un proyecto completo, se ha incluido un apartado de gestión web para realizar un proyecto más realista, donde realizaremos gestiones complementarias a la app móvil descrita en el capítulo anterior.

4.2.1 Alcance del proyecto

El objetivo principal es crear una aplicación web que se pueda acceder desde un enlace público mediante cualquier navegador web, y que realice las principales funciones de administración de eventos para la app móvil.

La web debe permitir unas funciones básicas :

- Añadir eventos.
- Eliminar eventos.
- Consultar eventos.

4.2.2 Metodología aplicada

El proyecto se aplicará el Proceso Unificado, impartida en la Universidad de Valladolid, marca unas pautas a seguir para desarrollar un software muy planificado. Como metodología no ágil presenta estas características:

- Centrado en la arquitectura.

- Dirigido por los casos de uso.
- Iterativo e Incremental.
- Enfocado en los riesgos.

Decidí llevar a cabo esta metodología porque es la que más hemos usado durante la carrera y no está tan enfocada al trabajo en equipo como pueden ser otras metodologías, como Scrum, tampoco estamos en constante relación con un cliente y ante posibles evolutivos que van cambiando según las especificaciones del cliente.

El proceso unificado se divide en cuatro etapas distintas:

Inicio/Concepción:

- Establecer los objetivos y los límites del proyecto.
- Identificar los casos de uso más importantes del sistema.
- Estimar el costo global y la planificación de todo el proyecto.
- Especificar los riesgos del proyecto.

Elaboración:

- Garantizar que la arquitectura, los requisitos y los planes son lo bastante estables.
- Obtener una visión refinada del proyecto a realizar.
- Producir un prototipo del proyecto.

Construcción:

- Minimizar los costos de desarrollo optimizando los recursos e impidiendo las reconstrucciones y los fragmentos innecesarios.
- Conseguir una calidad adecuada de forma rápida y práctica.
- Conseguir versiones útiles
- Completar el análisis, diseño, desarrollo y prueba de toda la funcionalidad necesaria.

Transición:

- Ejecutar el despliegue.
- Conseguir que el usuario sea capaz de mantener el producto.
- Conseguir la aceptación por el usuario que lo entregado es completo y consistente con el criterio de evaluación fijado en la visión inicial del proyecto.
- Obtener un producto final tan rápido y eficiente respecto al coste como práctico.

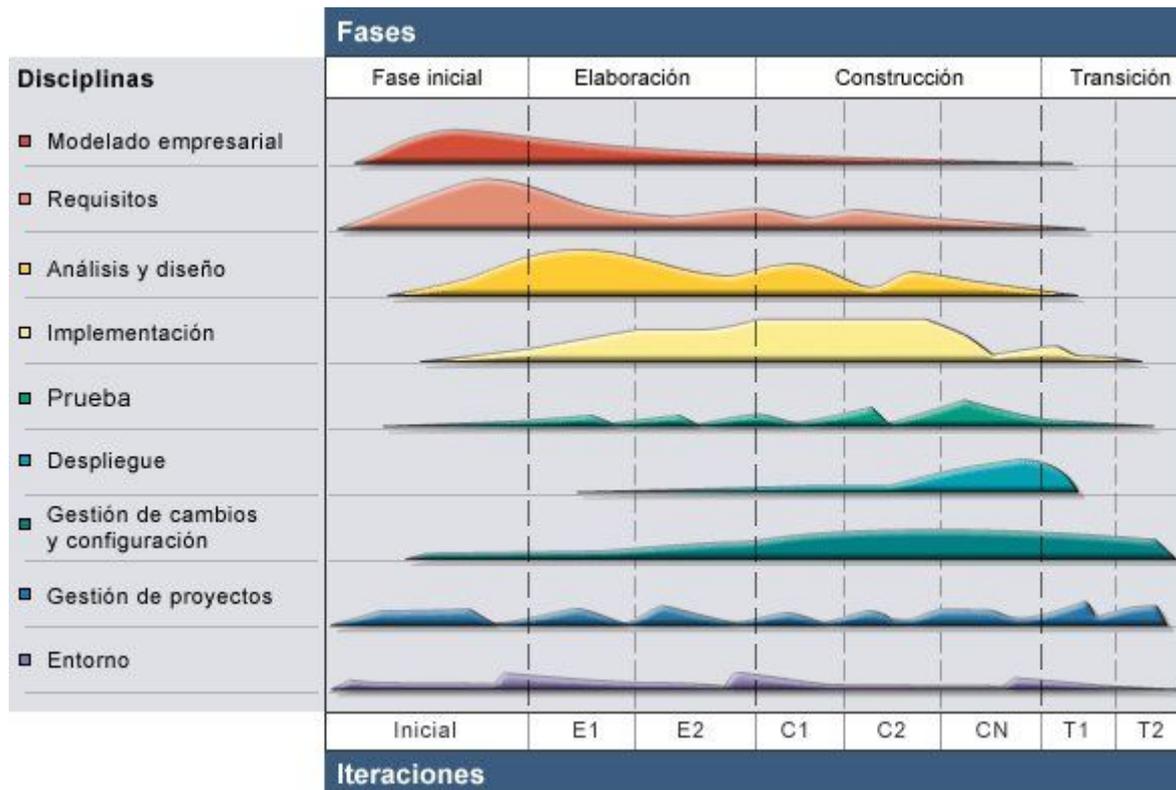


Figura 12: Fases PU

Dentro de alguna de las etapa de desarrolla el método Waterfall/Cascada, donde se lleva el método más clásico del software.

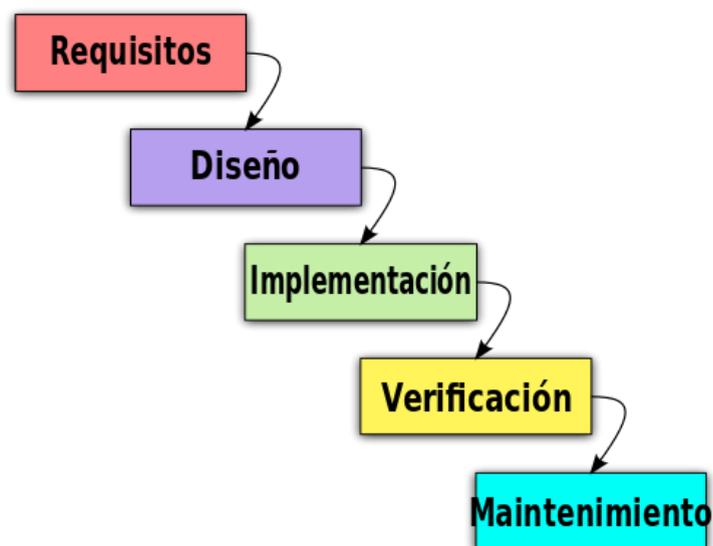


Figura 13: Fases del método en cascada

4.2.3 Equipo

Los distintos roles que se pueden encontrar en el proceso unificado, y que desempeño en el proyecto son:

Rol	Desempeña
Analista	Borja Rabadán Martín
Desarrollador	Borja Rabadán Martín
Diseñador	Borja Rabadán Martín
Gestores	Borja Rabadán Martín
Pruebas	Borja Rabadán Martín

Tabla 1: Roles

- **Analista:** Encargado de analizar los requisitos, el sistema y el proyecto, para plantear un modelo de negocio acorde al alcance del proyecto.
- **Desarrollador:** Se dedica a desarrollar el software del sistema, escribiendo el código en función de los requisitos planteados por el analista.
- **Gestores:** Llevan la gestión del proyecto, el reparto de recursos y seguimiento del mismo.
- **Pruebas:** Realizan test en los códigos escritos, tratan errores y optimizan el código.

4.2.4 Plan de Riesgos

En este apartado se plantean posibles riesgos que pueden surgir a lo largo del desarrollo del proyecto, así como los planes de contingencia para cada uno de ellos.

Se realiza una matriz de análisis de riesgo para representar la relación entre la probabilidad que ocurra, y decidir una actuación en función de la tabla creada.

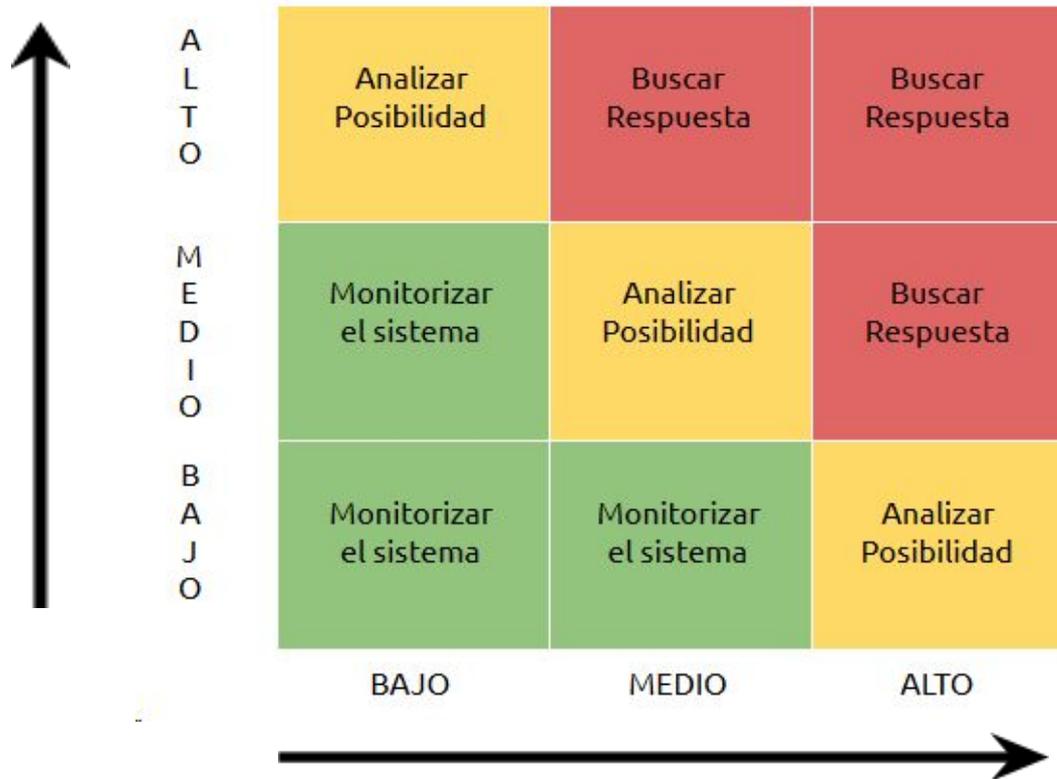


Figura 14: Tabla de riesgos

Lista de Riesgos

RW-01	Falta de experiencia del equipo de desarrollo
Descripción	La falta de experiencia por parte del equipo de desarrollo
Probabilidad	Media
Consecuencia	Mayor costes y retrasos en la entrega
Impacto	Alto
Gestión del riesgo	
Estrategia	Reservar el riesgo
Plan de mitigación	Dedicar tiempo y dinero en formación
Plan de contingencia	Aumentar recursos

Tabla 34: Riesgo web 1

RW-02	Pérdida de datos
Descripción	Pueden perderse datos debido al fallo del hardware
Probabilidad	Bajo
Consecuencia	Repetir el trabajo perdido
Impacto	Alto
Gestión del riesgo	
Estrategia	Evitar el Riesgo
Plan de mitigación	No se aplica
Plan de contingencia	Subir copias de seguridad a Github

Tabla 35: Riesgo web 2

RW-03	Falta de disponibilidad del hardware
Descripción	No disponer de dispositivos para hacer pruebas
Probabilidad	Alto
Consecuencia	No poder probar algunas funcionalidades
Impacto	Medio
Gestión del riesgo	
Estrategia	Evitar el Riesgo
Plan de mitigación	No se aplica
Plan de contingencia	Solicitar dispositivos para la realización de pruebas

Tabla 36: Riesgo web 3

RW-04	Caída de Servidor
Descripción	Caída del servidor donde está la base de datos de la app
Probabilidad	Bajo
Consecuencia	No poder realizar la demostración
Impacto	Alto
Gestión del riesgo	
Estrategia	Reservar Riesgo

Plan de mitigación	Ejecutarlo en local
Plan de contingencia	Preparar una versión en local

Tabla 37: Riesgo web 4

RW-05	Diseño incorrecto
Descripción	No realizar un diseño que ayude a la implementación
Probabilidad	Medio
Consecuencia	Retraso en el proyecto
Impacto	Medio
Gestión del riesgo	
Estrategia	Protegerse del Riesgo
Plan de mitigación	Dedicar más tiempo en el diseño
Plan de contingencia	Rehacer el diseño

Tabla 38: Riesgo web 5

RW-06	Falta de personal
Descripción	Debido a enfermedad o motivos personales
Probabilidad	Medio
Consecuencia	Aumento del coste y de retraso de la planificación
Impacto	Medio
Gestión del riesgo	
Estrategia	Asumir el Riesgo
Plan de mitigación	Contratar personal
Plan de contingencia	Aceptar el riesgo

Tabla 39: Riesgo web 6

RW-07	Documentación insuficiente de Angular
Descripción	Debido a que es un <i>framework</i> reciente, no dispone de tanta información
Probabilidad	Alta

Consecuencia	Retrasos en el proyecto
Impacto	Medio
Gestión del riesgo	
Estrategia	Asumir el Riesgo
Plan de mitigación	Obtener cursos de Angular2
Plan de contingencia	Solicitar ayuda acerca de la materia

Tabla 40: Riesgo web 7

4.2.5 Gestión del proyecto

4.2.5.1 Estimación

Para completar el proyecto, se ha incluido una estimación del tiempo dedicado para el desarrollo de la web administrativa.

Etapa	Fecha	Duración
Inicio	15 de abril	2 Semanas
Elaboración	29 de abril	2 Semanas
Construcción	20 de mayo	3 Semanas
Transición	3 de junio	2 Semanas

Tabla 41: Tiempo estimado para la web

Dedicación diaria como máximo de 2 horas, puede que le dedique menos tiempo algún día o incluso nulo, para ello usaba cronómetro para contabilizar el tiempo y que avisara del tiempo límite establecido.

4.2.5.2 Formación

El desarrollo de la web mediante HTML5, CSS3 y Bootstrap no ha requerido conocimientos, ya que se han impartido en Sistemas Web en la Universidad. Sin embargo, el desarrollo con el framework de Angular y el uso de TypeScript se ha tenido que aplicar cursos de formación.

Formación	Medio	Duración
Curso básico Angular2	Udemy	25 horas

Implementación Firebase Angular	Web oficial	3 horas
---------------------------------	-------------	---------

Tabla 42: Tiempo dedicado a la formación

4.2.5.3 Calendario de Trabajo

Número de Actividad	Actividad	Coste
1	Reunión acerca de la web Administración	1 hora
2	Esquema de la documentación del TFG	2 horas
3	Investigar acerca de Angular2	3 horas
4	Investigar Herramientas de trabajo	2 horas
Entrega 5º		
5	Estructura del proyecto	1 hora
6	Alcance del proyecto	1 hora
7	Plan de riesgos	3 horas
8	Presupuesto del proyecto	3 horas
Entrega 6º		
9	Requisitos Funcionales	3 horas
11	Requisitos No Funcionales	2 horas
12	Casos de uso	3 horas
13	Diagramas de secuencia	5 horas
14	Arquitectura y patrones	2 horas
15	BBDD	2 hora
16	Dominio	3 horas
17	Despliegue	2 horas
Entrega7º		

18	Implementación-Login	3 horas
19	Implementación-Navbar	4 horas
20	Implementación-Añadir evento	5 horas
21	Implementación-Eliminar evento	7 horas
22	Implementación-Generar QR	8 horas
23	Implementación-Cerrar Sesión	5 horas
Entrega 8º		
24	Pruebas	3 horas
25	Depuración	1 hora
26	Conclusiones y posibles mejoras	2 horas
Entrega Final		
27	Crear presentación	5 horas
28	Preparar presentación	2 horas

Tabla 43: Tiempo dedicado al TFG

4.2.6 Gestión de configuraciones

Se realizan periódicamente copias de seguridad del software construido, para evitar la pérdida de información. Del mismo modo, también se suben configuraciones, documentación y archivos de instalación en caso de extraviar su contenido.

Usando una cuenta de GitHub privada proporcionada a los estudiantes, se sube cada día productivo una copia del trabajo actual, con *commit* que indique la fecha y el trabajo realizado ese día.

4.2.7 Presupuesto del proyecto

Se pueden estimar dos visiones del presupuesto, una llevada al mundo laboral que se acerca más a la realidad y otra en la situación del estudiante, donde los costes son en mayor parte nulos o pequeños costes. Nos centraremos en los costes reales, con estimaciones obtenidas mediante estudios proporcionados por *Indeed* sobre salario medio de los distintos cargos en España.

4.2.7.1 Recursos Humanos

Sólo contamos las horas necesarias para el proyecto que son 130 horas, por lo que calculamos según los sueldos anuales, el coste por hora de cada participante en el proyecto.

Rol	Salario Anual	€/Hora	Horas estimadas	Costes Total
Desarrollador Web	22.541 €	11,71 €/H	130 Horas	1350€*
Analista	27.560 €	14,35 €/H		
Diseñador	18.230 €	9,49 €/H		
Jefe de proyecto	34.834 €	18,14 €/H		

Tabla 44: Presupuesto de los roles

*Costes totales redondeados según el porcentaje de participación empleados por cada uno.

4.2.7.2 Recursos Hardware y Software

Costes Hardware

Partiendo de la amortización del equipo usado en el proyecto según la Agencia tributaria, tendremos que aplicar un coeficiente lineal del 25%, y como sólo se usará para 3 meses del proyecto, se reducen los costes.

Dispositivo	Precio	Coste al año	Coste al mes	3 Meses de Proyecto
Xiaomi MI6 6GB	240€	60 €/Año	5 €/Mes	15€
Lenovo G550	400€	100 €/Año	8,33 €/Mes	25€

Tabla 45: Presupuesto del hardware

Costes Software

La mayor parte del software usado es de código abierto, open source y gratuito, otros han usado versiones de prueba.

Software	Coste al año	Coste al mes	3 Meses de Proyecto
Office 2013	105,6 €/Año	8,8 €/Mes	26,4€
GitHub	108 €/Año	9 €/Mes	27€
Dominio	12 €/Año	1€/Mes	3€
Firebase	268,68 €/Año	22.39€/Mes	67,17€

Tabla 46: Presupuesto del software

4.2.7.3 Costes Indirectos

Costes que están presentes en el proyecto de manera indirecta, como el espacio de trabajo.

Software	Detalles	Coste al mes	3 Meses de Proyecto
Oficina	Centro de ciudad(27m2)	180€/Mes	540€
Internet+llamadas	Fibra simétrica 300mb y llamadas ilimitadas	37 €/Mes	111€
Luz+Calefacción	uso normal	13,82€/Mes	41,46€

Tabla 47: Presupuesto de los costes indirectos

4.2.7.4 Costes totales

El presupuesto para el desarrollo de la app móvil y de la página web requiere un total de **7384,47 €**

4.3 Análisis

4.3.1 Requisitos

4.3.1.1 Requisitos Funcionales

Requisito	Descripción
RF-01	El sistema permitirá iniciar sesión.
RF-02	El sistema permitirá cerrar sesión.
RF-03	El sistema permitirá consultar los eventos de un usuarios.
RF-04	El sistema permitirá mostrar el código QR de los eventos de un usuario.
RF-05	El sistema permitirá eliminar los eventos de un usuario.
RF-06	El sistema notificará al usuario de la eliminación del evento.
RF-07	El sistema permitirá añadir eventos.
RF-08	El sistema notificará al usuario la agregación al evento.
RF-09	El sistema notificará el acceso incorrecto.

Tabla 48: Lista de Requisitos Funcionales

4.3.1.2 Requisitos Funcionales

Requisito	Descripción
RNF-01	No se podrán enviar tokens sin saldo.
RNF-02	No se podrán recibir/enviar tokens de diferente eventos.
RNF-03	Un identificador único para cada usuario.
RNF-04	No se podrá añadir el mismo saldo al sistema.
RNF-05	Se notificará el pago.
RNF-06	No se podrán enviar tokens, si no existe el usuario.
RNF-07	El sistema deberá estar operativo al 99,99%.

RNF-08	El sistema pedirá permisos de la cámara.
RNF-09	El sistema pedirá permisos de la tecnología NFC.
RNF-10	El sistema usará bibliotecas externas para generar el código QR.
RNF-11	El sistema requiere la API 23 como mínimo (Android 6.0)
RNF-12	El sistema usará los servicios de autenticación de firebase.
RNF-13	El sistema usará los servicios de notificación de firebase
RNF-14	El sistema está realizado de acuerdo a la ley de protección de datos.

Tabla 49: Lista de Requisitos No Funcionales

4.3.2 Casos de uso

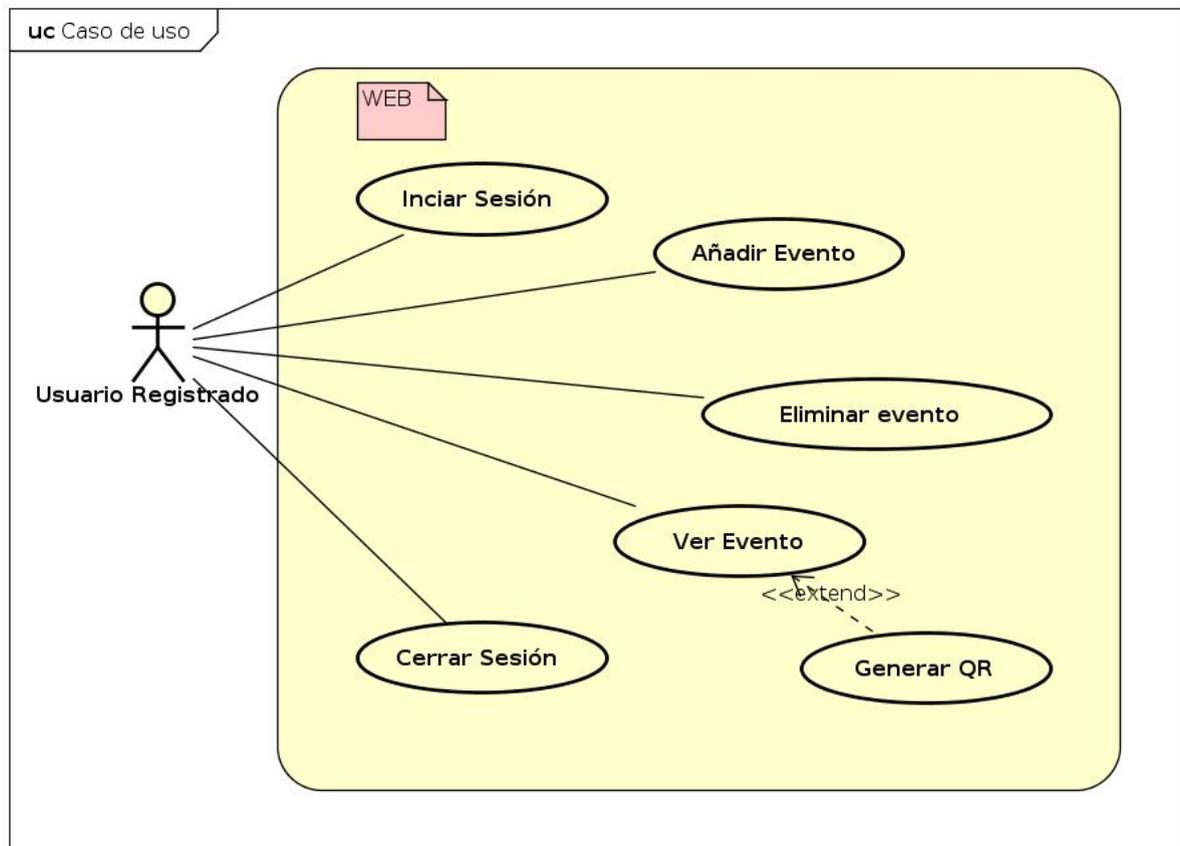


Figura 51: Casos de uso

CU-1	Iniciar Sesión
Actor	Administrador
Precondición	Estar registrado en el sistema

Postcondición	Entrar en el sistema
Flujo principal	<ol style="list-style-type: none"> 1. El usuario entra en la web 2. El sistema pide correo y usuario 3. Introduce los datos 4. Selecciona el botón "login" 5. Sistema le identifica y le envía al menú principal.
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) 2) Cancela la operación, y no se realiza ninguna acción • 3a) No existe el usuario en el sistema, y vuelve al paso 1 • 3b) La contraseña es correcta, y vuelve al paso 1 • 3c) No cumple con el formato del correo, y vuelve al paso 1

Tabla 50: Caso de uso web 1

CU-2	Cerrar Sesión
Actor	Administrador
Precondición	Estar dentro del sistema
Postcondición	Salir del sistema
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Logout" 2. El sistema elimina la sesión actual del usuario 3. El sistema le lleva al login
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) Cancela la operación, y no se realiza ninguna acción

Tabla 51: Caso de uso web 2

CU-3	Añadir Evento
Actor	Administrador
Precondición	Estar dentro del sistema
Postcondición	Crear un evento
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Añadir Evento" 2. El sistema pedirá al usuario un nombre y una fecha 3. El usuario introduce los datos. 4. El sistema comprueba los datos. 5. El sistema crea el evento y genera un mensaje de confirmación que desaparece a los 5 segundos
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) 3a) Cancela la operación, y no se realiza ninguna acción • 4a) Ya existe el evento en el sistema, y vuelve al paso 1 • 4b) La fecha es anterior a la actual, y vuelve al paso 1 • 5b) Elimina el mensaje de confirmación.

Tabla 52: Caso de uso web 3

CU-4	Ver evento
Actor	Administrador
Precondición	Estar dentro del sistema
Postcondición	Mostrar una lista de eventos
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Ver eventos" 2. El sistema muestra una lista de eventos creados por el usuario
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) Cancela la operación, y no se realiza ninguna acción

Tabla 53: Caso de uso web 4

CU-5	Generar QR
Actor	Administrador
Precondición	Estar dentro del sistema
Postcondición	Visualizar un código QR
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Ver eventos" 2. El sistema muestra una lista de eventos 3. El usuario selecciona "Generar QR" 4. El sistema genera un código QR
Flujo Alternativo	<ul style="list-style-type: none"> • 2a) Cancela la operación, y no se realiza ninguna acción

Tabla 54: Caso de uso web 5

CU-6	Eliminar evento
Actor	Administrador
Precondición	Estar dentro del sistema
Postcondición	Quitar un evento perteneciente al usuario
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Eliminar eventos" 2. El sistema muestra una lista de eventos 3. El usuario selecciona "eliminar" 4. El sistema borra del sistema ese evento
Flujo Alternativo	<ul style="list-style-type: none"> • 1a) 3a) Cancela la operación, y no se realiza ninguna acción

Tabla 55: Caso de uso web 6

4.3.3 Modelo de dominio

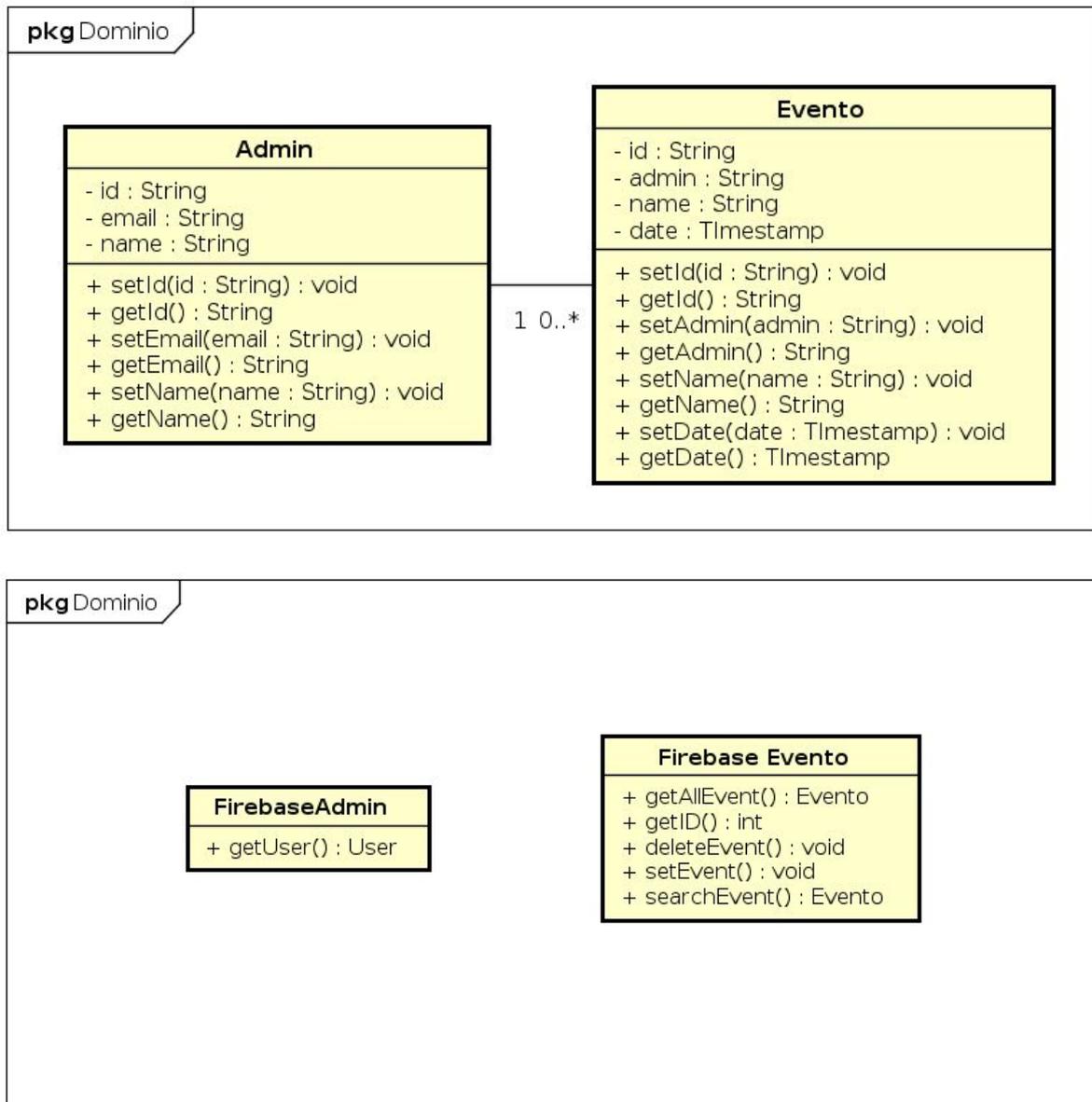


Figura 52: Modelo de dominio

4.3.4 Diagramas de secuencia

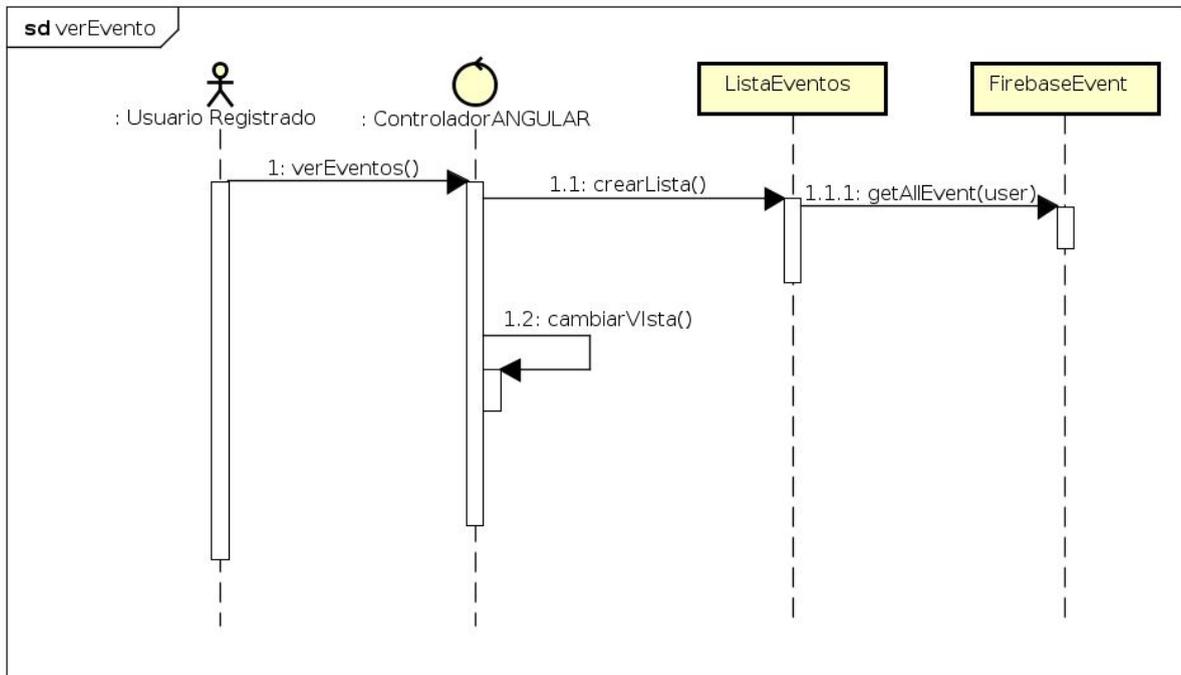


Figura 53: DS de ver evento

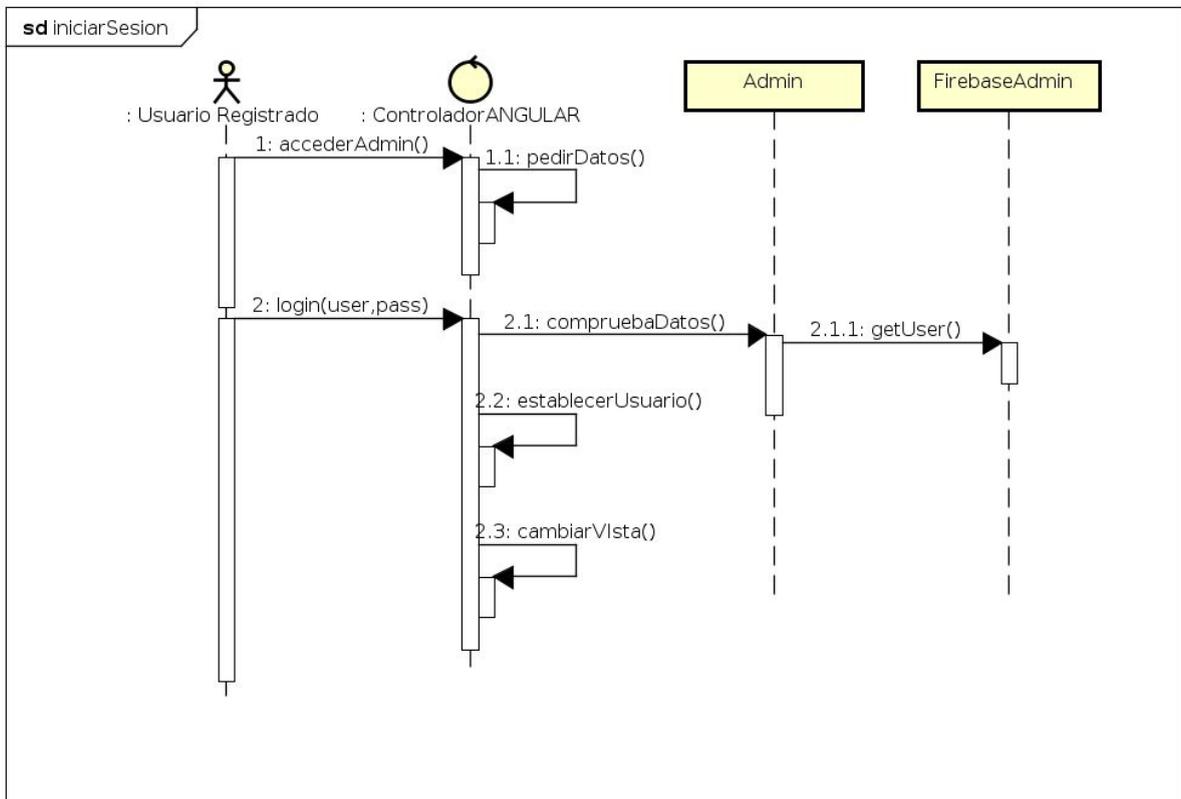


Figura 54: DS de iniciar sesión

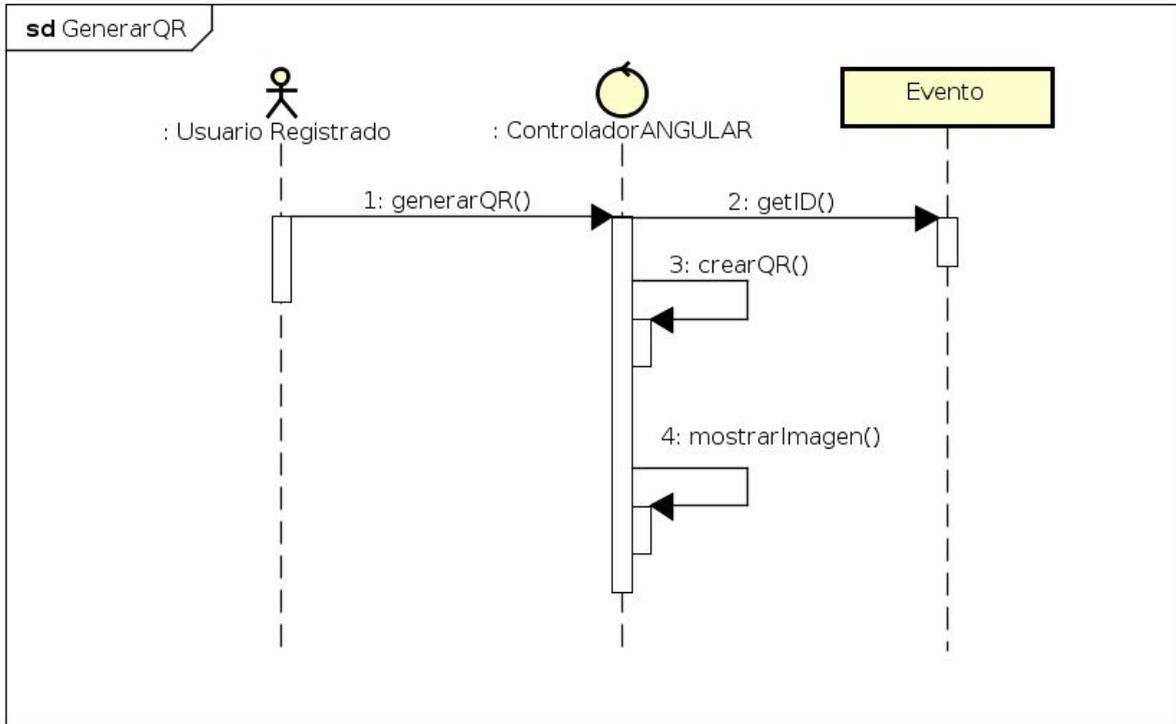


Figura 55: DS de generarQR

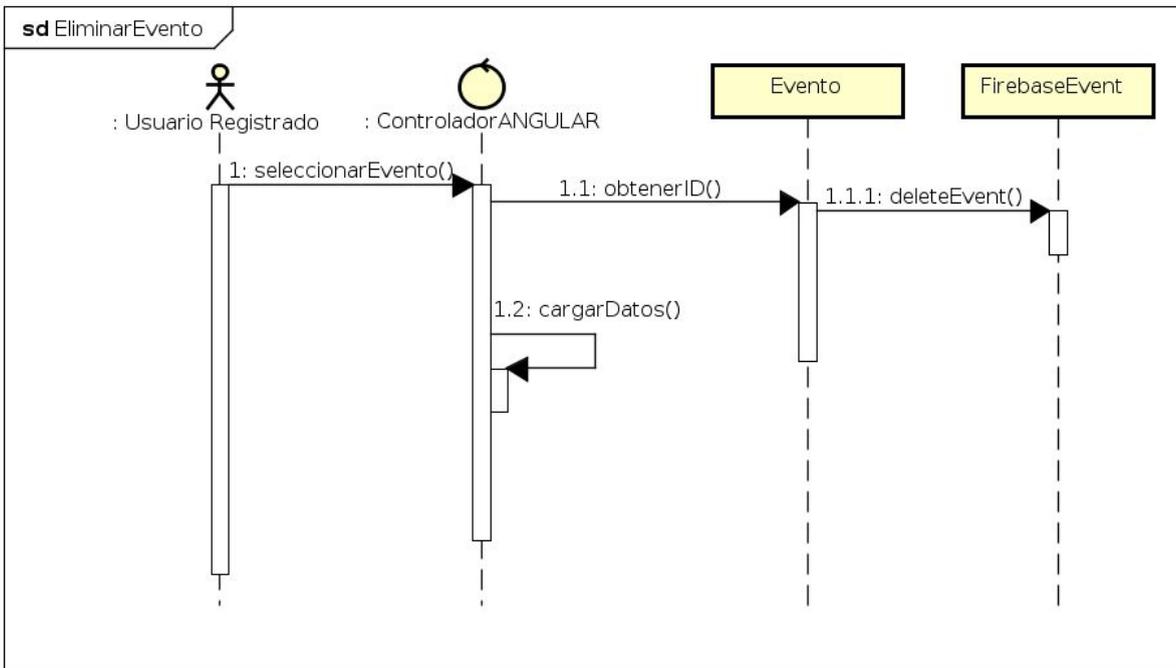


Figura 56: DS de eliminar evento

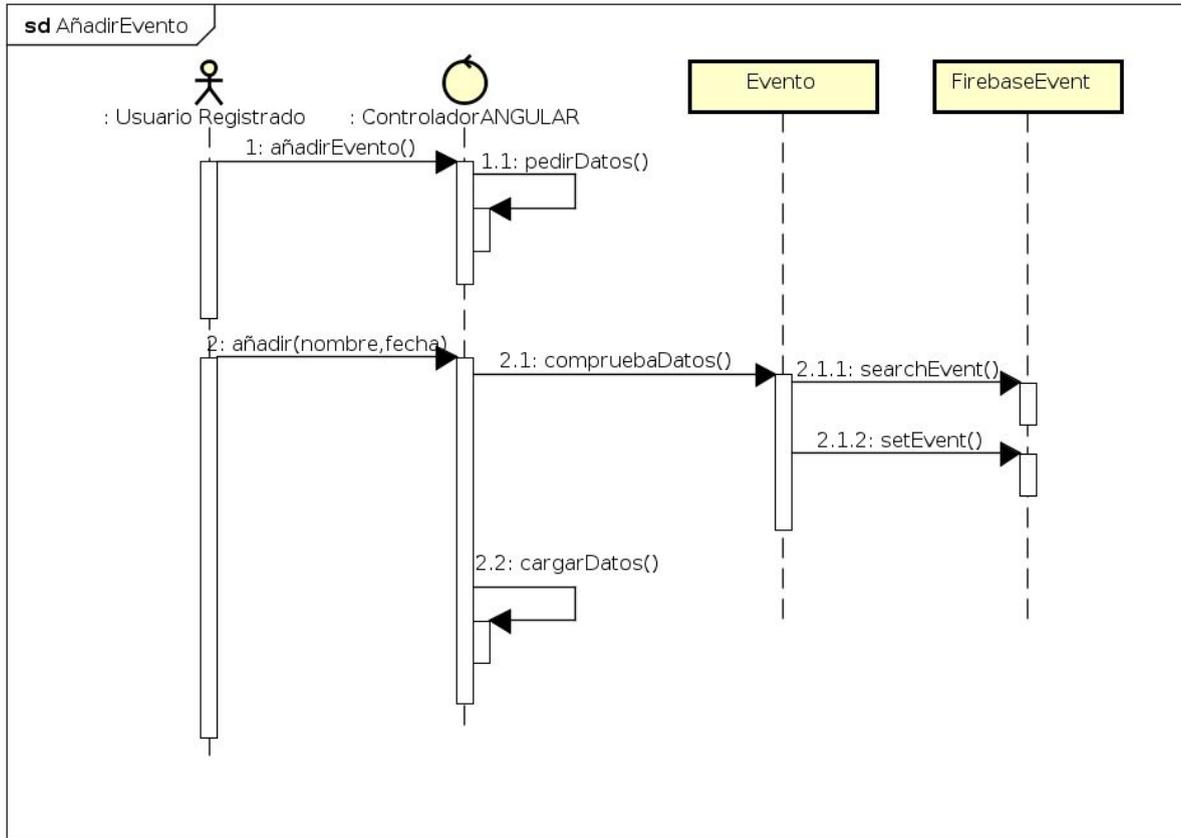


Figura 57: DS de añadir evento

4.4 Diseño

4.4.1 Arquitectura

Model View Controller (MVC): Permite acometer cambios en la presentación de los datos con mayor facilidad, dando lugar a software más mantenible. Mientras que la Vista y el Controlador se encontrarán en la arquitectura cliente (capa de presentación), el modelo permanecerá en los servidores o servidor central (capa de dominio).

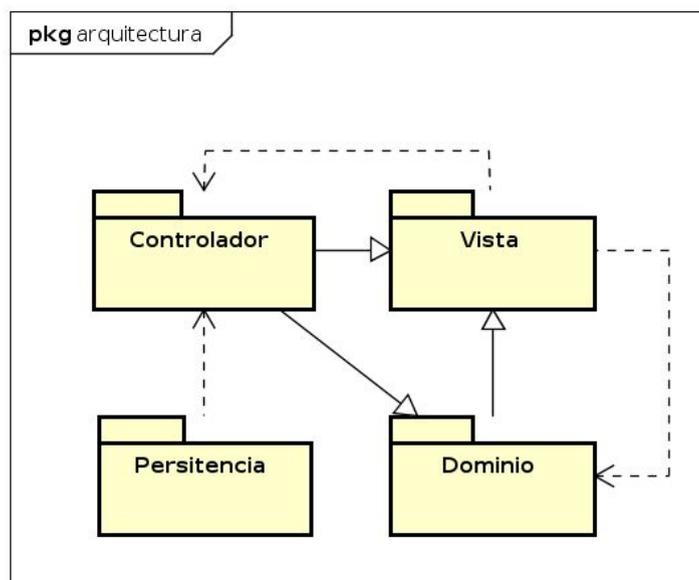


Figura 58: DP de arquitectura

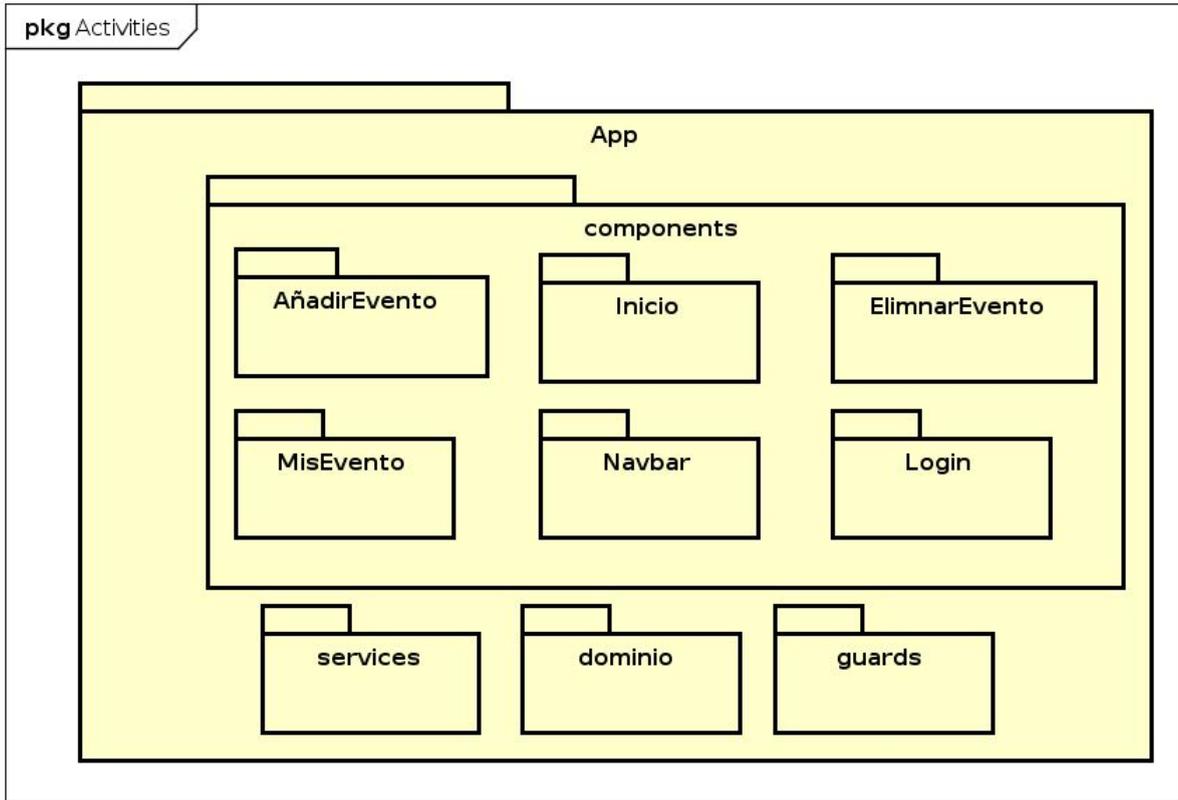


Figura 59: DP de Web Angular

4.4.2 Modelo no relacional

Firestore usa actualmente una base de datos no relacional (NoSQL) de tipo Colección-Documento, mediante la documentación que proporciona Firebase acerca de Firebase Firestore y los conceptos básicos dados en Administración de base de datos de 4º de Ing Informática, he realizado este esquema.

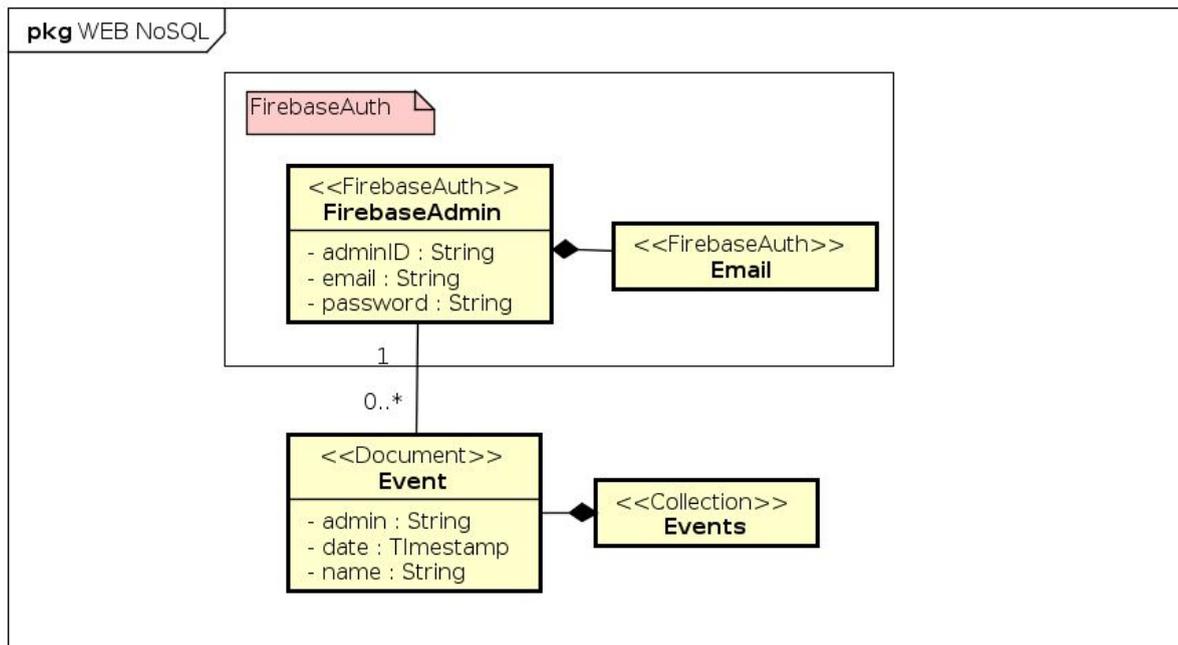


Figura 60: Modelo no relacional en firebase

4.4.3 Diagrama de despliegue

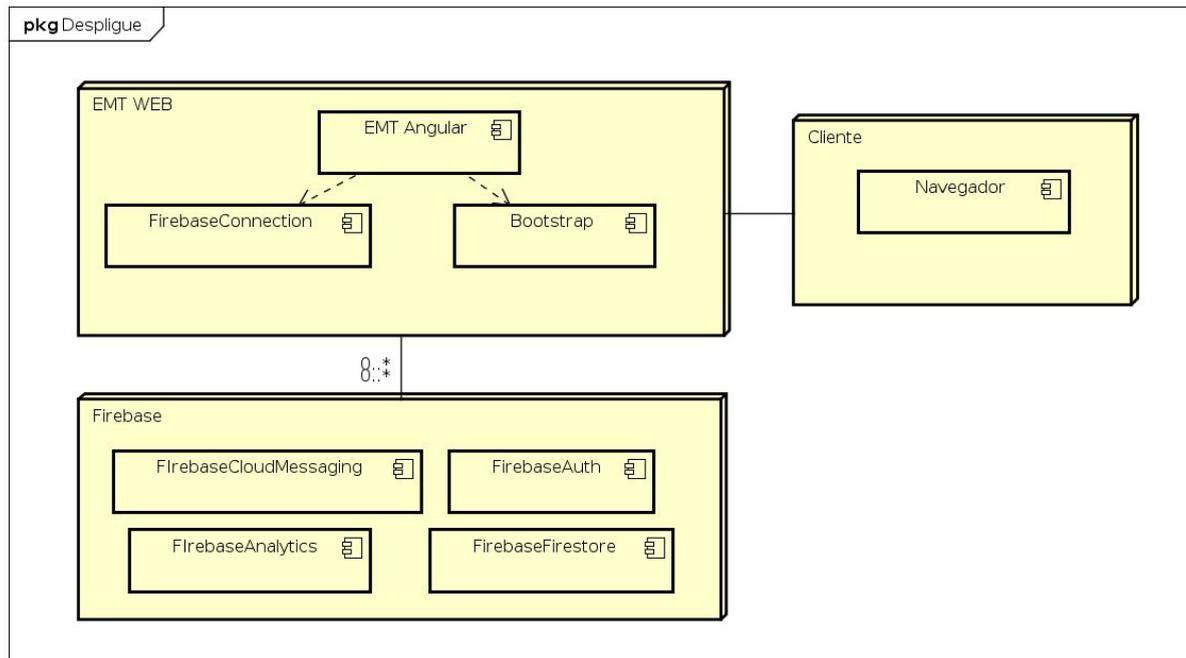


Figura 61: DP de Angular

4.5 Implementación

La página web se ha separado en la vista (CSS, HTML Y BOOTSTRAP), el controlador y el modelo de negocio con TypeScript, y la persistencia de Firebase.

Las vistas se han usado elementos básicos de formularios como el datepicker y el uso de elementos proporcionados por Bootstrap que son únicos para Angular que son los NG, que nos permite realizar funciones y condiciones específicas. Como el paso usado para mostrar los códigos QR con NG-SHOW, que nos permite interactuar directamente desde el controlador a la vista.

Angular usa tres elementos que son los servicios, components y guards, donde en cada uno se realiza las distintas tareas, como pueden ser los componentes que se encuentran la vista y el controlador de la vista.

Servicios realiza la función más genérica que controla los diferentes componentes como puede ser el acceso a la base de datos.

Y Guards guarda configuraciones como la de Firebase para darle los credenciales. Aparte se crea una carpeta que será el modelo de negocio de nuestro web, en este caso eventos y administradores.

Los componentes están formado por 6 elementos:

- Login/Inicio de sesión
- NavBar/Barra de navegación
- Home
- Añadir Evento
- Eliminar Evento
- Ver Evento

Mientras los servicios se han repartido en 2:

- Acceso de los usuarios
- Persistencia de Firebase

Por último el modelo de negocio comentado anteriormente.

4.6 Pruebas

Las pruebas de la web han sido testadas mediante un software específico para automatizar tareas, **Selenium**.



Figura 61: Logo de Selenium

Selenium nos permite desarrollar pruebas automatizadas mediante una IDE para grabar acciones y reproducirlas. También puede desarrollarse en varios lenguajes gracias a su API, en este caso he optado por hacerlo en Python.

Mediante Selenium WebDriver con el navegador de Google he realizado un test de los casos de uso en un solo script de manera secuencial. En el Anexo se encuentra el código del script y para su uso sólo requiere disponer de estos elementos:

- Python 3.X
- WebDriver de Chrome

Luego se tendría que ejecutar como un programa de python *py testEMT.py*.

4.7 Manual de instalación

La web actualmente se encuentra desplegada en Firebase gracias a su servicio de hosting, a la que podremos acceder mediante este [enlace].

Estos son los pasos que se han realizado:

1. `ng build --prod "EMT"` , para construir la web.
2. `firebase init` , para seleccionar la carpeta que se va a subir al servidor.
3. `firebase deploy` , para subir los archivos y desplegar la web.

Requisitos previos

- **Tener Firebase CLI instalado.**
- **Una cuenta en Firebase.**
- **Conexión a internet.**

En caso de querer desplegar la web en local, con el comando `ng serve -o`, realiza el despliegue de la web en el puerto 4200.

En otros servidores, se deberá construir la web con `ng build --prod` y mover los archivos a la carpeta correspondiente del servidor(Tomcat -- webapps).

4.8 Guía de usuario

Accediendo a la web mediante este [\[Enlace\]](#), nos encontramos con una página para identificarse (con usuarios creados previamente en Firebase).

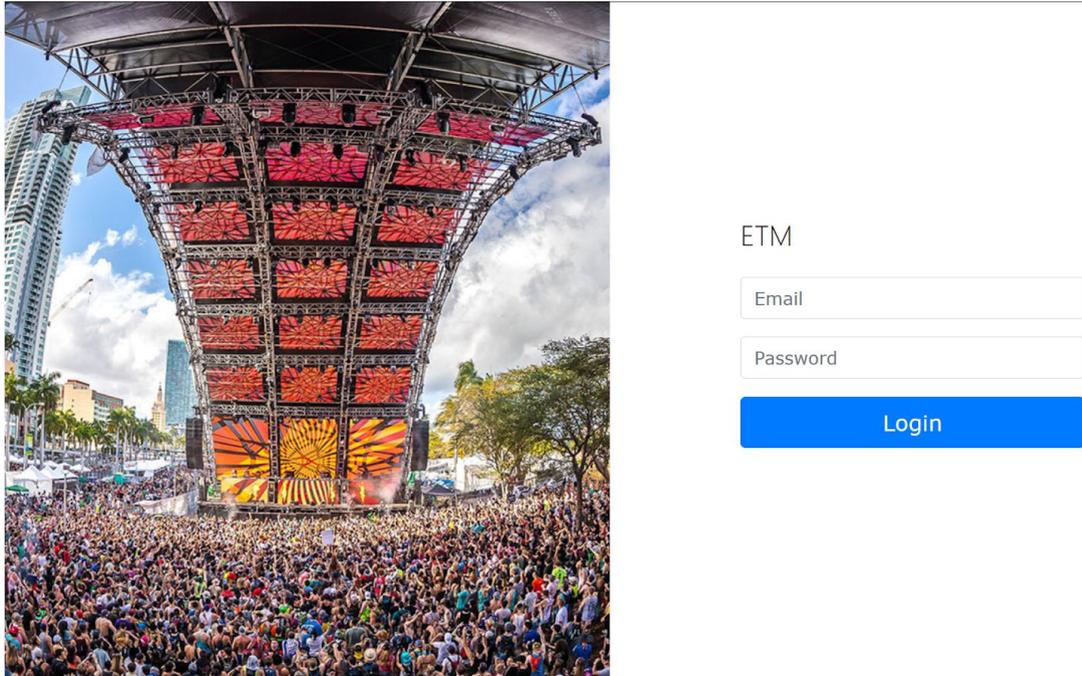


Figura 62: Captura del login de la web administrativa

Una vez accedido a la web de manera correcta, se accederá a la web de inicio del Administrador.

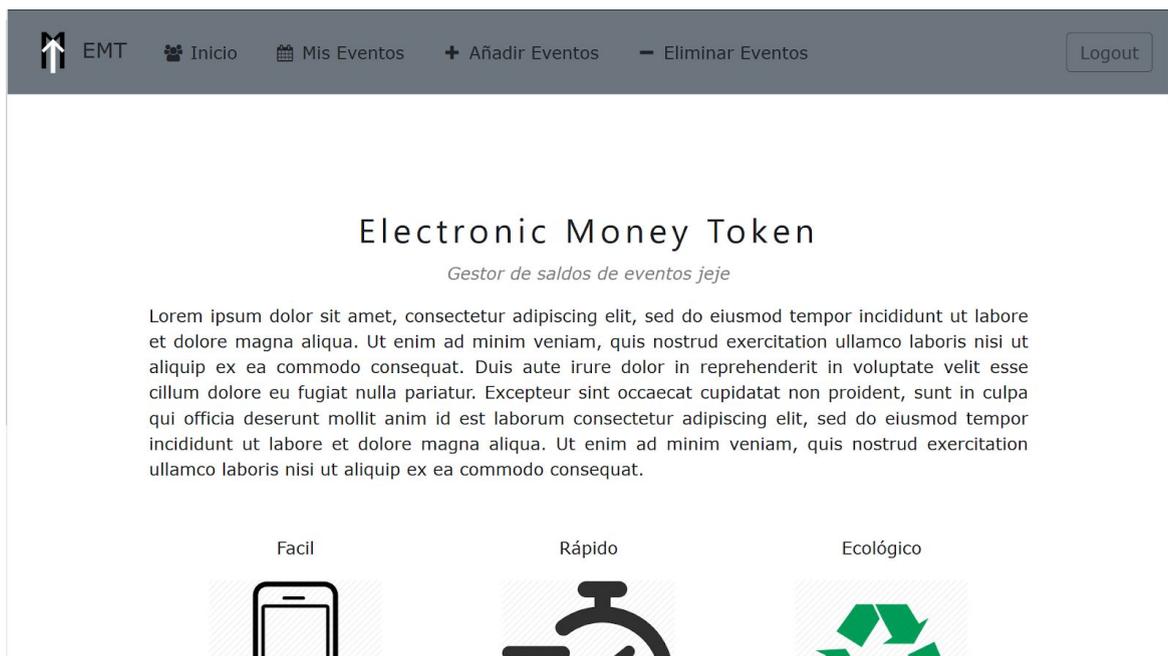


Figura 63: Captura de la página de inicio

Donde nos encontramos con una introducción de la app y un menú de navegación con 5 elementos:

- Inicio
- Añadir evento
- Eliminar evento
- Mis eventos
- Logout(salir)

Inicio: nos lleva a la página principal mostrada anteriormente.

Mis eventos: muestra la lista de eventos que hemos creado, con un botón de “Mostrar QR” que nos permite ver el QR perteneciente a ese evento, para su posible distribución en diferentes medios para que los usuarios puedan añadir sus correspondientes saldos.

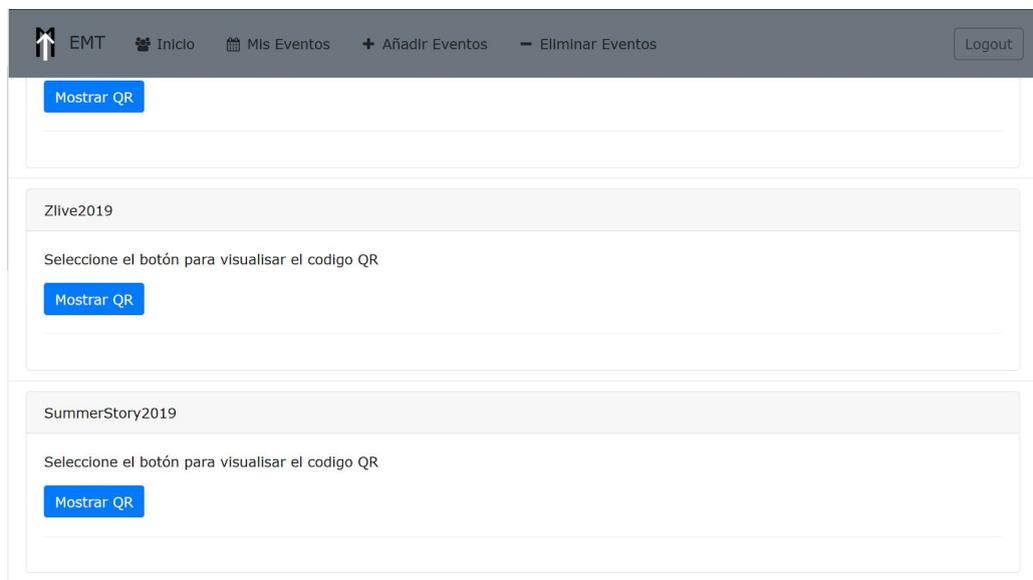


Figura 64: Captura de la página de los eventos

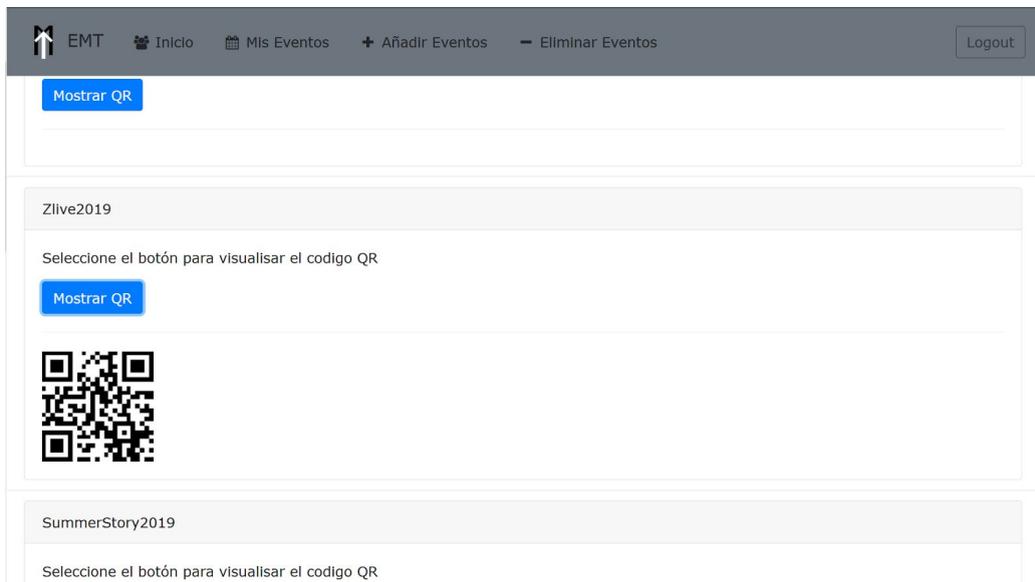


Figura 65: Captura de los eventos con su correspondiente QR

Eliminar evento: muestra la lista de eventos que disponemos, con un botón de “Eliminar” que nos permite quitar evento, para mayor seguridad, mostramos un mensaje de advertencia para asegurar que quiere eliminar el evento.

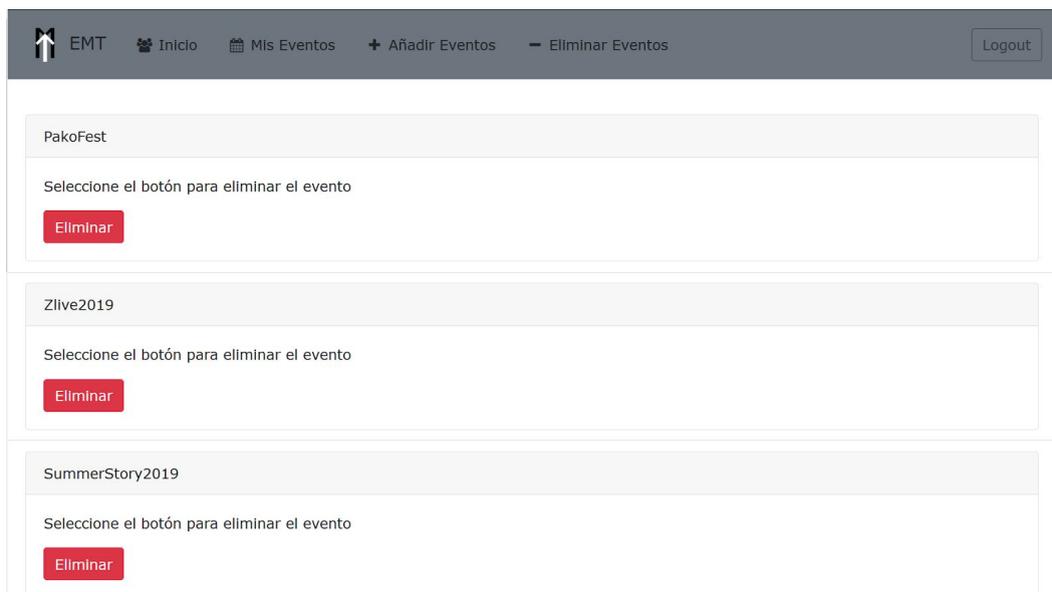


Figura 66: Captura de los eventos a eliminar

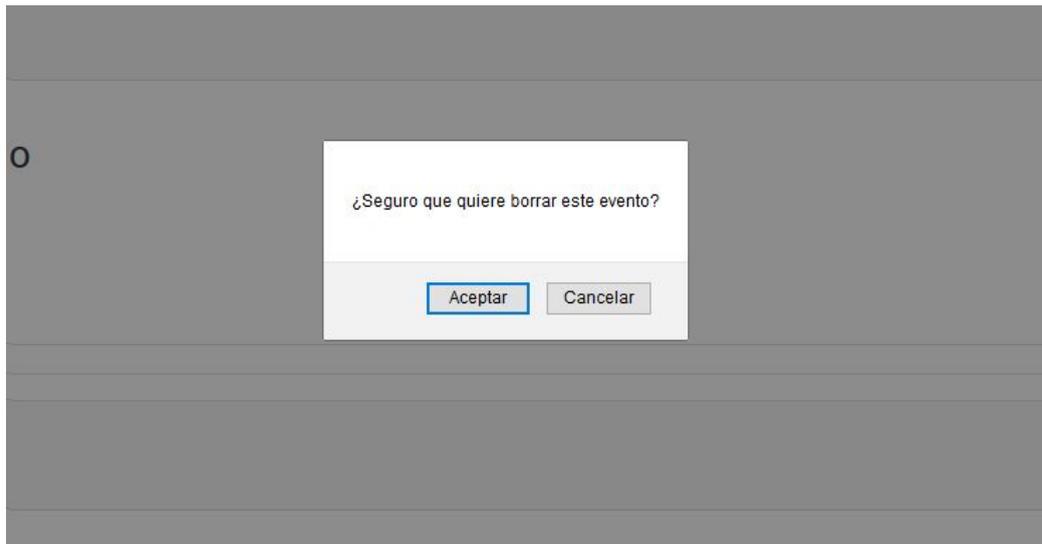


Figura 67: Captura del aviso para eliminar el evento

Añadir evento: mostramos un formulario simple, el nombre del evento y la fecha fin del evento. Tras seleccionar "Añadir", nos muestra que se ha añadido correctamente el evento.

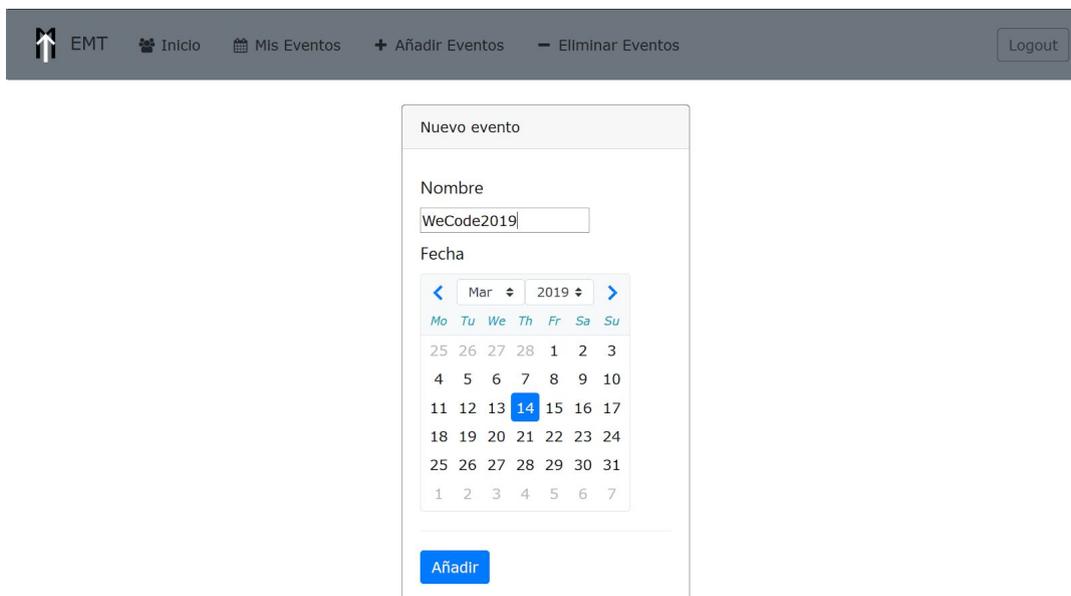


Figura 68: Captura del formulario para añadir eventos

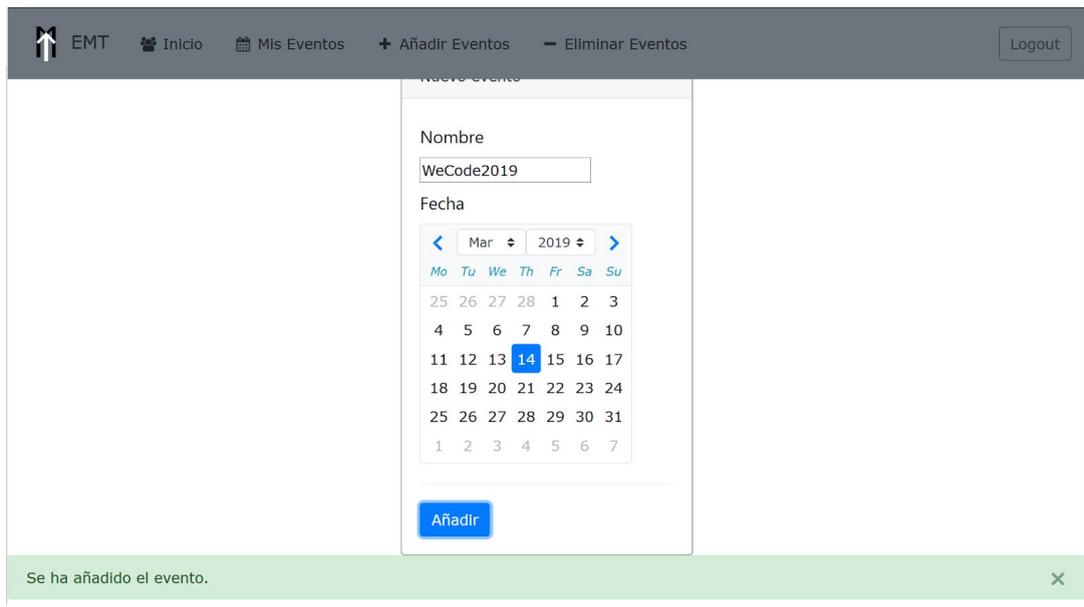


Figura 69: Captura de la confirmación del evento

Logout: Nos devolverá a la página de identificación, saliendo de manera exitosa de la sesión del usuario.

Capítulo V

Conclusiones y trabajo futuro

5.1 Conclusiones

Conclusiones de este Trabajo Fin de Grado:

1. El empleo de varias tecnologías para la implementación del TFG marcadas por java para el desarrollo de la app, de NoSql por parte del servidor de Firebase y Angular para crear webs dinámicas en una sólo página.
2. Dar una solución a un problema actual con las tecnologías que disponemos hoy en día y facilitar el acceso desde cualquier punto con el móvil, desde la visión de un ingeniero .
3. Usar diferentes métodos para transmitir el saldo y aplicar protocolos de seguridad para asegurar el saldo enviado, para asegurar el correcto funcionamiento de la aplicación, la seguridad de los datos y facilitar su comunicación.
4. Aplicar los conocimientos obtenidos a lo largo de la carrera, durante el desarrollo de proyecto y en la vida laboral.
5. Analizar el mercado y la opinión pública para ir mejorando nuestra aplicación mediante encuestas.

5.2 Trabajo futuro

Se proponen las siguientes ideas para extender las funcionalidades de la aplicación:

1. Añadir empleados al evento, donde se podrán monitorizar sus ventas en los eventos.
2. Añadir la opción de productos/servicios predefinidos en la app, para poder facilitar la venta por parte de los empleados del evento, es decir, que sea más personalizable por cada evento. En función del evento, se podría añadir (una vez implementado el punto anterior respecto a los empleados) un menú con el precio de cada producto, con sólo pulsarlo se obtendrá el precio y no haría falta meter la cantidad de manera manual.
3. Implementar la app en dispositivos IOS y Windows Phone, para cubrir el mayor número de usuarios.
4. Traducir la app a varios idiomas.
5. Poder identificarse mediante facebook, google, etcétera.
6. Añadir patrocinadores en los saldos, para que los organizadores obtengan un beneficio extra.
7. Implementar la opción de reembolso, en caso de que el evento lo permita, se podrá reembolsar los tokens no usados .
8. Introducir distintos medios de pago para añadir el saldo(Paypal, tarjetas de crédito, bitcoins...).
9. Lanzar la app al mercado.
10. Sistema de notificaciones.

Capítulo VI

Webgrafía y Bibliografía

596941857074829. (2017, April 03). Architecture patterns in Android. Retrieved from

<https://android.jlelse.eu/architecture-patterns-in-android-abf99f2b6f70>

Angular 6 tutorial - create your own Angular Bootstrap app, step by step. (n.d.). Retrieved from

<https://mdbootstrap.com/education/angular/>

Anonymous. (2018, November 28). El euro. Retrieved from

https://europa.eu/european-union/about-eu/euro_es

Arlow, J., & Neustadt, I. (2006). *UML 2*. Anaya Multimedia.

Bandi, A., & Bandi, A. (2018, October 03). Web Scraping Using Selenium - Python. Retrieved

<https://towardsdatascience.com/web-scraping-using-selenium-python-8a60f4cf40ab>

Businessinsider. (2019, April 18). How to create and customize a Google Form for polls and questionnaires.

<https://www.businessinsider.es/how-to-create-google-form?r=US&IR=T>

CES by the Numbers. (n.d.). Retrieved from

<https://www.ces.tech/About-CES/CES-by-the-Numbers.aspx>

Crónica del Z! Live Rock Fest 2018. (2018, June 19). Retrieved from

<https://mariskalrock.com/cronicas/cronica-del-z-live-rock-fest-2018/>

Documentación | Firebase. (n.d.). Retrieved from *<https://firebase.google.com/docs/?hl=es-419>*

Favre-Bulle, P., & Favre-Bulle, P. (2018, January 06). Security Best Practices: Symmetric Encryption with AES in Java and Android.

<https://proandroiddev.com/security-best-practices-symmetric-encryption-with-aes-in-java-7616b6aaade9>

Fichas para festivales. (n.d.). Retrieved from

<https://www.b-token.es/products/tokens/fichas-para-festivales>

Forde, E. (2017, January 30). Where concert ticket money goes: Who's getting rich off live music's golden age? Retrieved from

<https://www.theguardian.com/music/2017/jan/30/where-does-concert-ticket-money-go>

Fowler, M. (2015). *Patterns of enterprise application architecture*. Addison-Wesley.

How to Start a Profitable Music Festival - Eventbrite. (2018, November 20). Retrieved from

<https://www.eventbrite.com/blog/music-festival-profit-ds00/>

Jacobson, I., Booch, G., & Rumbaugh, J. (2006). *El proceso unificado de desarrollo de software*. Addison Wesley.

Larrouy, D., & Larrouy, D. (2018, August 25). 'Tokens', la moneda de plástico de los festivales que se cuenta al peso. Retrieved from

https://www.eldiario.es/economia/Tokens-moneda-curso-festivales-cuenta_0_807019491.html

Nieto, J. G., & Nieto, J. G. (2019, January 07). Así es como Android se ha comido el mercado en diez años. Retrieved from

<https://www.xatakamovil.com/sistemas-operativos/asi-como-android-se-ha-comido-mercado-diez-anos>

Ortiz, S. J., Ortiz, S. J., Ortiz, S. J., & Ortiz, S. J. (2015, May 20). Cómo Usar Passbook y Añadir

Tarjetas Paso a Paso. Retrieved from <https://www.ipadizate.es/2015/05/24/tutorial-passbook/>

Programador/a web. (n.d.). Retrieved from

<https://www.indeed.es/salaries/Programador/a-web-Salaries>

Sistema de pago cashless con NFC para eventos. (2019, February 20). Retrieved from

<https://intelify.net/sistema-de-pago-cashless-con-nfc-para-eventos/>

Sonar, la herramienta para escanear fallos de seguridad en páginas Web. (2017, October 30).

<https://www.redeszone.net/2017/11/01/sonar-la-herramienta-escanear-fallos-seguridad-paginas-web/>

These Tomorrowland 2018 Numbers Will Blow You Away. (2018, July 16). Retrieved from

<https://edmmaniac.com/tomorrowland-2018-by-the-numbers/>

Vardhan, J., Singh, J., Sharma, Y., Prasad, S., Singh, J., Upadhyay, H., . . . Vardhan, J. (2019,

January 18). Paytm is testing interoperable UPI QR Code for merchant payment: Why white-label route? Retrieved from

<https://entrackr.com/2019/01/paytm-nteroperable-upi-qr-code/>

Zxing. (2019, April 18). Zxing/zxing. Retrieved from *<https://github.com/zxing/zxing>*

Parte I

Anexos

Anexo I: Estudio estadístico

I. Estudio

Se realizó una encuesta acerca del proyecto, con una serie de preguntas básicas, descripción de la app y su viabilidad, y el motivo por el cual aceptan o rechazan su uso. Distribuida mediante RRSS, grupos de mensajería y foros destinados a eventos musicales para obtener la opinión del usuario sobre la app mediante Google Forms.

No se ha podido recoger la opinión de organizadores de eventos, la cual es una opinión fundamental para implementación de este sistema en sus eventos.

II. Conclusión

Con los datos obtenidos, se puede ver que el público más joven usaría más la app en distintos eventos donde destacan los conciertos/festivales, los jóvenes están más acostumbrados a las nuevas tecnologías y a las ventajas que aportan.

El gasto medio no es elevado, por lo que podemos ver que la gente se gasta una cantidad inferior a 50 €, y a través del número de personas que han participado en la encuesta se puede estimar el número de transacciones que tiene que soportar la app.

Y por último, el uso de esta app en los eventos, por una parte se puede ver como una buena idea (40 %), pero otros la rechazan por ser una app, o incluso preferirían pagar en efectivo, y casi el 50% opta por la indecisión, debido a los factores de seguridad, que tendrían que probarla, si los eventos la aceptarían, y más cuestiones que se han recogido sobre la app.

¿Usarías la app?

123 respuestas

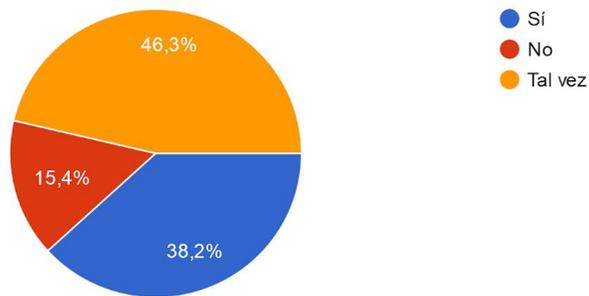


Figura 70: Resultados de la usabilidad de la app

Edad

122 respuestas

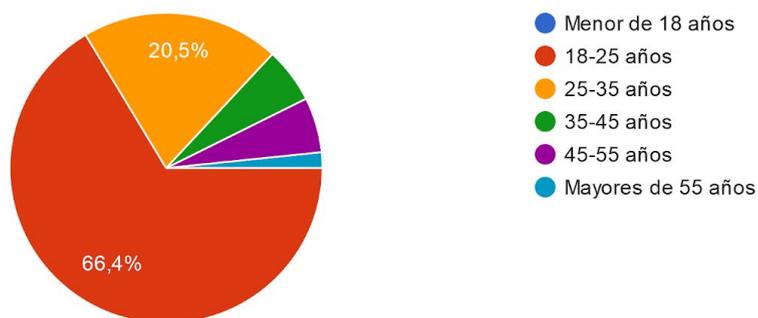


Figura 71: Resultados de la edad de los encuestados

¿Cuánto dinero sueles gastar dentro de estos eventos?

123 respuestas

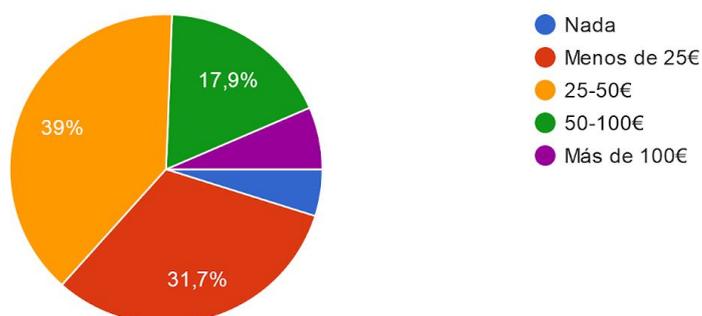


Figura 72: Resultados del dinero que se gasta en los eventos

¿A qué tipo de eventos acudes?

123 respuestas

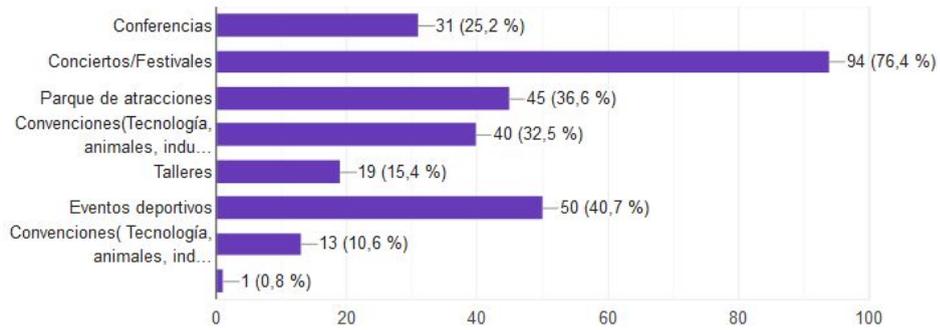


Figura 73: Resultados de los diversos eventos

Enlace a la encuesta:

<https://goo.gl/forms/Q1aLa9B5lrKlxFg63>

Enlace a los resultados:

https://docs.google.com/spreadsheets/d/1TQHWlTErLHWlN3gDB7yTBUVq6meTysF3kpPSu_iuq48/edit?usp=sharing

Anexo II: Selenium

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from time import gmtime, strftime
import sys
import os.path
import pickle
import time
import random
import re
import json
from bs4 import BeautifulSoup

def menu(opcion):
    switcher = {
1:verEvento
2:eliminarElemento
3:añadirElemento
    }

def verEvento():
driver.find_element_by_xpath(
    "//a[contains(text(),'Mis Eventos')]").click()
driver.find_element_by_xpath(
    "//ul[1]//li[1]//div[1]//div[2]//button[1]").click()
return "Proceso terminado"

def eliminarElemento():
driver.find_element_by_xpath(
    "//a[contains(text(),'Eliminar Eventos')]").click()
```

```

driver.find_element_by_xpath(
    "//ul[1]//li[1]//div[1]//div[2]//button[1]").click()
    return "Proceso terminado"

def añadirElemento():
driver.find_element_by_xpath(
    "//a[contains(text(),'Añadir Eventos')]").click()
driver.find_element_by_xpath(
    "//input[@id='nombre']").sendKeys("Festival_2019")
driver.find_element_by_xpath(
    "//button[@class='btn btn-primary']").click()
    return "Proceso terminado"

if len (sys.argv) != 2 :
    print ("Falta argumentos [servicio]")
    sys.exit (1)
driver = webdriver.Chrome()

driver.get('https://empapp-a871f.firebaseio.com/sign-in')
time.sleep(aleatorio())
user = driver.find_element_by_xpath("//input[@id='email']")
time.sleep(aleatorio())
user.send_keys('admin@zlive.es')
key = driver.find_element_by_xpath("//input[@id='password']")
time.sleep(aleatorio())
key.send_keys('totibero')

log_in_button = driver
    .find_element_by_xpath("//button[@class='btn btn-lg btn-primary btn-block']")
time.sleep(aleatorio())
log_in_button.click()

func = switcher.get(sys.argv[1], lambda: "Opcion invalida)

print func();

```