



**Universidad de Valladolid**

**ESCUELA DE INGENIERÍA INFORMÁTICA  
DE SEGOVIA**

**Grado en Ingeniería Informática  
de Servicios y Aplicaciones**

---

**Proa, Aplicación Distribuida para mejorar el mundo  
empresarial**

---

**Alumno: Ignacio de Frutos Rosa**

**Tutor: Francisco José Gonzalez Cabrera**



# Proa, Aplicación Distribuida para mejorar el mundo empresarial

Ignacio de Frutos Rosa



# Contents

<b>Índice general</b>	<b>i</b>
<b>Lista de figuras</b>	<b>v</b>
<b>Lista de tablas</b>	<b>vii</b>
<b>Resumen</b>	<b>xi</b>
<b>I Memoria del Proyecto</b>	<b>1</b>
<b>1 Descripción del proyecto</b>	<b>3</b>
1.1 Introducción . . . . .	3
1.2 Motivación . . . . .	5
1.3 Objetivos del trabajo . . . . .	5
1.4 Estado del arte . . . . .	6
1.4.1 BBVA Campus Wallet [2] . . . . .	6
1.4.2 Karma [7] . . . . .	6
1.4.3 Meeds token [10] . . . . .	7
<b>2 Metodología</b>	<b>9</b>
2.1 Proceso de desarrollo . . . . .	9
2.2 Herramientas utilizadas . . . . .	10
<b>II Memoria técnica del proyecto</b>	<b>13</b>
<b>3 Análisis</b>	<b>15</b>
3.1 Descripción de los Actores . . . . .	15
3.2 Requisitos . . . . .	15
3.2.1 Requisitos de negocio . . . . .	15
3.2.2 Requisitos de usuario . . . . .	16
3.2.3 Requisitos de información . . . . .	17
3.2.4 Requisitos Funcionales . . . . .	17

3.2.5	Requisitos de interfaz externa . . . . .	18
3.2.6	Requisitos no funcionales . . . . .	19
3.3	Casos de Uso . . . . .	21
3.3.1	Especificación de casos de Uso . . . . .	22
3.4	Estimación y presupuestos . . . . .	23
<b>4</b>	<b>Diseño e Implementación</b>	<b>27</b>
4.1	Arquitectura . . . . .	27
4.1.1	Arquitectura lógica . . . . .	27
4.1.2	Arquitectura física . . . . .	29
4.2	Diagrama de clases y Árbol de características . . . . .	29
4.3	Diseño de datos . . . . .	31
4.4	Diseño de la interfaz . . . . .	33
4.4.1	La barra de navegación . . . . .	35
4.4.2	Menú lateral . . . . .	36
4.4.3	Página de iniciativas . . . . .	36
4.4.4	Tienda . . . . .	37
4.4.5	Página mis compras . . . . .	37
4.4.6	Página Tramitar Compras . . . . .	40
4.4.7	Transacciones y Transacciones Empresa . . . . .	42
4.4.8	Administrar Empleados . . . . .	43
4.5	Pruebas . . . . .	45
4.5.1	Pruebas de los smart contract . . . . .	45
4.5.2	Pruebas de los componentes de React . . . . .	47
<b>5</b>	<b>Despliegue</b>	<b>49</b>
5.1	Despliegue real de la aplicación . . . . .	49
5.2	Coste de despliegue y operación . . . . .	49
<b>6</b>	<b>Ampliaciones y Conclusión</b>	<b>51</b>
6.1	Posibles ampliaciones y trabajo futuro . . . . .	51
6.2	Propuestas de mejora . . . . .	51
6.3	Conclusión . . . . .	52
<b>III</b>	<b>Manuales de la Aplicación</b>	<b>53</b>
<b>7</b>	<b>Manual de Instalación</b>	<b>55</b>
<b>8</b>	<b>Manual de Usuario</b>	<b>59</b>
8.1	Manual de Usuario . . . . .	59
8.1.1	Página iniciativas . . . . .	60
8.1.2	Tienda . . . . .	60
8.1.3	Mis Compras . . . . .	60

8.1.4	Transacciones . . . . .	60
8.2	Manual de Administración . . . . .	60
8.2.1	Tienda . . . . .	61
8.2.2	Transacciones Empresa . . . . .	61
8.2.3	Administrar Empleados . . . . .	61
8.2.4	Tramitar Compras . . . . .	61
<b>Bibliography</b>		<b>63</b>





# List of Figures

2.1	Modelo de desarrollo iterativo . . . . .	9
3.1	Diagrama de casos de uso . . . . .	21
4.1	Arquitectura lógica . . . . .	28
4.2	Arquitectura física . . . . .	29
4.3	Diagrama de clases . . . . .	30
4.4	Árbol de Características . . . . .	31
4.5	Diagrama Entidad-Relación . . . . .	32
4.6	Modelo de datos . . . . .	33
4.7	Vista general escritorio . . . . .	34
4.8	Visión general móvil . . . . .	34
4.9	Barra de navegación . . . . .	35
4.10	notificaciones . . . . .	35
4.11	Añadir Iniciativa . . . . .	37
4.12	Detalles Iniciativa . . . . .	38
4.18	Página Mis Compras . . . . .	38
4.13	Detalles Iniciativa Móvil . . . . .	39
4.14	Tienda . . . . .	39
4.15	Tienda versión móvil . . . . .	40
4.16	Añadir producto a la tienda . . . . .	41
4.19	Tramitar compras . . . . .	41
4.17	Editar producto de la tienda . . . . .	42
4.20	Transacciones Empresa . . . . .	43
4.21	Nueva Transferencia . . . . .	43
4.22	Administrar Empleados . . . . .	44
4.23	Datos Empleado . . . . .	44
4.24	Editar Empleado . . . . .	45
4.25	Interfaz de Remix para probar contratos . . . . .	46
7.1	Despliegue Contratos . . . . .	57
7.2	/tfginterfaz/src/App.jsx:59 . . . . .	58
8.1	Página usuario desconocido . . . . .	59



# List of Tables

3.1	Requisitos de negocio . . . . .	15
3.2	Requisitos de Empleado . . . . .	16
3.3	Requisitos de Administrador . . . . .	16
3.4	Requisitos de información . . . . .	17
3.5	Requisitos Funcionales . . . . .	18
3.6	Requisitos de interfaz Externa . . . . .	18
3.7	Requisitos No Funcionales de usabilidad . . . . .	19
3.8	Requisitos No Funcionales de Seguridad . . . . .	19
3.9	CU-01. Proponer Iniciativa . . . . .	22
3.10	CU-02. Tramitar Iniciativa . . . . .	23
3.11	CU-03. Login . . . . .	23
3.12	Presupuesto . . . . .	24
3.13	Coste . . . . .	24
4.1	PU-01. User . . . . .	46
4.2	PU-01. User . . . . .	47



# Agradecimientos

Gracias a todas aquellas personas que invierten su tiempo en publicar y resolver dudas en foros como stackoverflow y github. Sin ellos no solucionaría ni la mitad de los problemas que tengo en todos los proyectos.



# Resumen

Proa es una aplicación distribuida sobre la red de Ethereum que busca fomentar la proactividad entre los empleados de una empresa. Esto se consigue a través de la participación y propuesta de iniciativas grupales y la obtención de recompensas proporcionadas por la empresa a través de una tienda. El motor de estas iniciativas y tienda son tokens, que se utilizarán tanto para conseguir las recompensas como para apuntarse a ciertas iniciativas. Los tokens se consiguen en un tipo particular de iniciativa o comerciando con los compañeros.

**Palabras claves:** Blockchain, Ethereum, Dapp, smart contract, React, IPFS.





# Part I

## Memoria del Proyecto



# Chapter 1

## Descripción del proyecto

### 1.1 Introducción

Blockchain es una tecnología que, desde que se introdujo en 2008, ha ido ganando mucha popularidad, sobre todo debido a Bitcoin [1]. Sin embargo, aunque mucha gente lo desconoce, las criptomonedas no son la única aplicación de esta tecnología, Aquí es donde entra Ethereum, una plataforma lanzada en 2015, que popularizó los llamados smart contracts [4].

Un smart contract es un programa que se ejecuta sobre la red de blockchain de Ethereum. Es una colección de código y datos que reside en una dirección específica dentro de la blockchain. Estos contratos son un tipo de cuenta de Ethereum, por lo que tienen balance y pueden mandar transacciones a la red, sin embargo no son controlados por ningún usuario, son subidos a la red y actúan según se hayan programado. Esto permite a los usuarios interactuar con el contrato y ejecutar una función definida en el mismo [6].

Estos smart contracts, al poder definir reglas y una lógica, se pueden utilizar para programar ciertas aplicaciones. Estas son las llamadas aplicaciones distribuidas (dapps a partir de ahora), que básicamente, son aplicaciones cuyo backend se ejecuta en una red peer-to-peer descentralizada, en este caso, la red de blockchain de Ethereum. Esta arquitectura, tiene muchas ventajas, siendo las principales[5]:

- **Zero downtime:** al ser una red distribuida, una vez el contrato se ha desplegado en la blockchain, la red siempre estará disponible para aquellos usuarios que quieran interactuar con el contrato.
- **Privacidad:** no es necesario dar datos privados para desplegar o interactuar con una dapp.
- **Resistencia a la censura:** ningún usuario de la red puede bloquear a otros de realizar transacciones, desplegar contratos o leer datos de la blockchain.

- **Integridad de datos completa:** los datos almacenados en la blockchain son inmutables e indisputables, gracias a la criptografía con la que funciona. Los datos publicados no pueden ser modificados.

Sin embargo, estas aplicaciones también tienen ciertas consideraciones a tener en cuenta[5]:

- **Mantenimiento:** las dapps pueden ser difíciles de mantener, ya que es complicado modificar su código
- **Escalado difícil:** todos los nodos de la red ejecutan y guardan todas las transacciones, además, la prueba de trabajo (este es el método de validación de transacciones que utiliza Ethereum, aunque se está trabajando en soluciones como la *proof of stake*) lleva su tiempo.
- **Congestión de la red:** si una sola dapp usa muchos recursos, la red entera se congestiona. Actualmente la red es capaz de procesar entre 10 y 15 transacciones por segundo.
- **Experiencia de usuario:** es complicado implementar experiencias user-friendly debido a la dificultad de preparar todas las herramientas necesarias para interactuar con la blockchain de forma segura. Además, muchas soluciones que sí que consiguen ser user-friendly, acaban pareciendo sistemas centralizados, en los que se guardan datos sensibles en el lado del servidor.

Esta tecnología es bastante interesante y tiene mucho potencial, ya que al final lleva solo 6 años desarrollándose. Por tanto, todavía hay mucho campo para la innovación y el desarrollo. Hasta ahora, las dapps más populares tienen que ver con las criptomonedas y tokens, estas son exchanges, aplicaciones para ver el valor de estos elementos, para la compra de NFTs o aplicaciones de juegos y apuestas para conseguir criptomonedas y tokens. Sin embargo creo que hay otras aplicaciones que pueden ser interesantes, como es el campo de los recursos humanos.

En este ámbito, las empresas siempre están buscando formas de motivar a los empleados para que estos estén más contentos trabajando y aumente su productividad. Una de las formas sociales que las empresas utilizan para aumentar la productividad, son las dinámicas de grupo, que además mejoran las relaciones en la empresa y favorecen el aprendizaje [3]. Otra de las formas más comunes de aumentar la felicidad entre los empleados es el otorgar premios a los mejores empleados, sin embargo, muchas veces estos premios se conceden basándose únicamente en baremos competitivos (por ejemplo, el número de ventas), lo que puede fomentar una competitividad excesiva antes que la colaboración, que es lo que de verdad están buscando[13].

¿Y si existiese una solución a este problema de las empresas que además funcionase con todo el potencial de las dapps? Para esto es este proyecto.

## 1.2 Motivación

Personalmente, la tecnología blockchain me parece muy interesante y creo que tiene un potencial muy alto y que todavía está por explotar. Por ello me surgió interés en formarme en los smart contracts y blockchain en general. Además, aunque en la carrera estudié javascript, css, html y php, creo que actualmente el mercado laboral demanda más los frameworks como pueden ser React o Angular, por lo que decidí hacer todo el desarrollo frontend de mi proyecto en uno de estos frameworks, React.

Este proyecto ha sido una forma estupenda de descubrir cuánto puedo aprender por mi cuenta sobre cualquier tema, lo empecé sin saber absolutamente nada ni de React ni de Solidity y, formándome de forma autodidacta, he acabado con un conocimiento que considero sólido de ambas tecnologías.

Concretamente, al inicio del proyecto me marqué tres objetivos personales para cumplir durante el proyecto:

- Aprender sobre la tecnología blockchain y, más concretamente, Ethereum como plataforma de desarrollo.
- Aprender a diseñar una aplicación web con una interfaz agradable para el usuario utilizando el framework React.
- Hacerme cargo de todo el peso de un proyecto totalmente nuevo y conseguir desarrollar mis habilidades en campos desconocidos y novedosos.

Creo que he cumplido todos estos objetivos y el resultado de todo ello es Proa, una aplicación de la que estoy orgulloso.

## 1.3 Objetivos del trabajo

El objetivo de este proyecto es crear Proa, una aplicación empresarial distribuida que permitirá a los empleados de una empresa tener un portal donde organizar iniciativas grupales y acceder a recompensas exclusivas proporcionadas por la empresa, así como reconocer el trabajo o el aprecio entre unos y otros a través del intercambio de tokens. Esto tendría como objetivo principal el mejorar la colaboración entre empleados, así como fomentar su participación en actividades que la empresa considere positivas para ellos y premiar a aquellos que demuestren mayor proactividad (que serían aquellos que conseguirían más tokens).

Este proyecto tiene varios objetivos y de distintos tipos, por lo que los voy a dividir entre objetivos personales y prácticos.

### **Objetivos Prácticos de la aplicación:**

- Dar a las empresas una plataforma con la que sus empleados puedan trabajar diariamente y así mejorar el ambiente de trabajo, cohesión del grupo e incentivar actividades beneficiosas para la empresa.
- Dar un producto funcional que se pueda aplicar a un entorno real

## 1.4 Estado del arte

Para ser tecnologías relativamente nuevas, hay ciertas empresas, que ya han adoptado una solución similar a lo planteado en este proyecto.

### 1.4.1 BBVA Campus Wallet [2]

El banco bilbaíno ha desarrollado un token con el objetivo de impulsar la formación continua entre sus empleados. Este token se utiliza a través de la plataforma de cursos y talleres online y presenciales llamada BBVA Campus Wallet.

Cada curso dentro de esta plataforma tiene un valor en tokens y se le han dado a cada empleado X tokens para que puedan comenzar a apuntarse a ellos. Para conseguir más tokens, un empleado puede realizar cursos gratuitos dentro de la misma plataforma, de esta forma recibe tokens por cada hora de formación invertida. Otra vía para conseguir tokens, es compartir conocimiento, es decir, que él mismo imparta cursos a sus compañeros, recibiendo así tokens por cada hora impartida y por cada alumno que tenga.

De esta forma se consigue un círculo en el que los mismos empleados forman a sus compañeros para conseguir tokens y esos tokens que consiguen los utilizan para seguir formándose ellos.

Los resultados de este proyecto han sido bastante positivos, el 88% de los empleados ha entrado a BBVA Campus Wallet y el 60% ya ha realizado cursos.

De momento éste es un proyecto piloto y está implementado en una red privada, sin embargo, están trabajando para desplegarlo sobre la red de ethereum y hacerlo un token público.

Esta aplicación es bastante similar a Proa, ya que los cursos podrían ser un caso particular de iniciativas dentro de la aplicación.

### 1.4.2 Karma [7]

Karma es un proyecto de Telefónica que identifica y formenta los comportamientos de un empleado que, sin estar recogidos en las funciones de su puesto, aportan valor a la empresa. Por ello, el empleado recibe una recompensa de tokens Karma que luego puede intercambiar por realizar ciertas actividades exclusivas.

Karma permite a las empresas incentivar comportamientos de valor en los empleados y aprovecha aspectos como la visibilidad, la colaboración o la productividad, claves en cualquier estrategia de RRHH centrada en las personas y difícilmente reconocibles o remunerables de manera continua en el día a día de las empresas. Por su lado, el empleado recibe incentivos no retributivos (y por tanto con menor coste para la empresa) en forma de actividades y experiencias exclusivas que recompensan sus acciones y redundan de manera beneficiosa en el clima laboral y su percepción de la empresa.

Este proyecto, está basado en la tecnología de TrustOS, un api para tokens.

Karma se parece a Proa en que tiene objetivos similares, aunque carece de las iniciativas para centrarse más en los intercambios.

### 1.4.3 Meeds token [10]

Meeds es un software que incluye un portal de actividad, un motor de gamificación y un motor de recompensa de tokens. Puede ser usado como base para la implementación de nuevo software o ser incluido en un software existente a través de APIs. El software es libre bajo licencia LGPL y distribuido gratuitamente.

El potencial de este sistema es el motor de gamificación, que captura la actividad del usuario en el sistema y le atribuye puntos en función de sus acciones. Esto permite recompensar a los empleados y motivarlos a través de trofeos y clasificaciones. Además, los empleados se pueden reconocer entre ellos como contribuidores, lo que permite una mayor visibilidad a través de grupos de trabajo relevantes.

La otra característica fuerte de este software es la recompensa a través de tokens, que se consiguen a través de los puntos mencionados en el anterior párrafo y que pueden ser intercambiados por dinero real

A diferencia de Proa, cuyo núcleo son las iniciativas, Meeds se enfoca más en la gamificación automática.





# Chapter 2

## Metodología

### 2.1 Proceso de desarrollo

El proceso de desarrollo, al ser un proyecto de una sola persona, ha sido bastante informal y simplificado, aunque basado en un desarrollo iterativo como el que se muestra en la figura 2.1. Primero se hizo una planificación inicial, a partir de la cual se obtuvieron unos requisitos iniciales. Estos requisitos se ordenaron por prioridad y orden de implementación y, muchas veces, durante la fase de implementación o de diseño se podían llegar a modificar para ajustarlos a la especificación cambiante del proyecto. La especificación ha cambiado mucho durante el desarrollo debido al desconocimiento inicial de las tecnologías con las que se trabajaba y por tanto también a la falta de experiencia a la hora de estimar tiempos trabajando con dichas tecnologías.

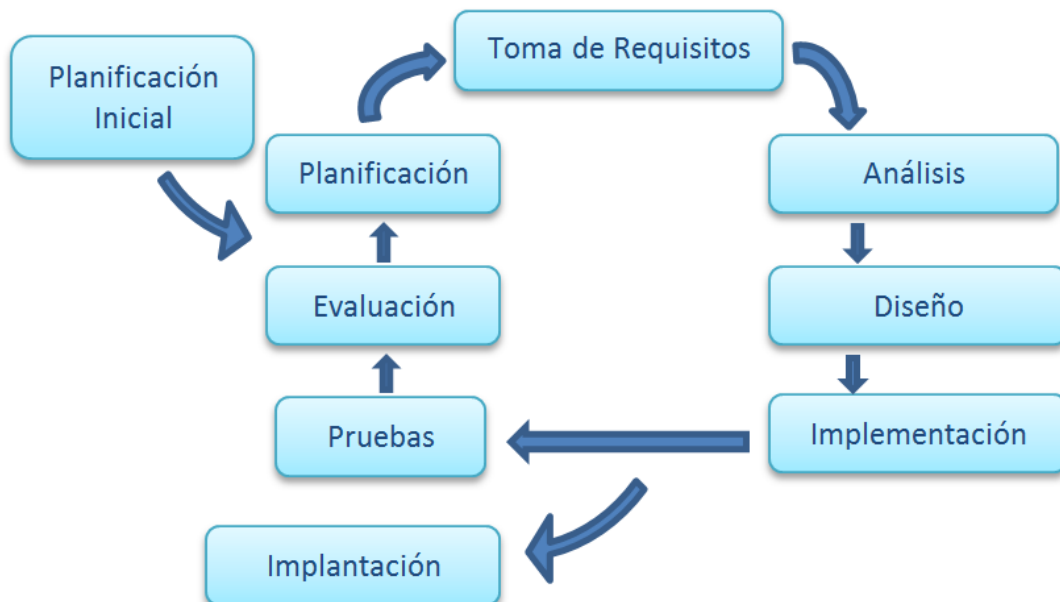


Figure 2.1: Modelo de desarrollo iterativo

## 2.2 Herramientas utilizadas

- **Solidity:** Lenguaje de programación utilizado para el despliegue de smart contracts en la red de ethereum.
- **React:** Framework de JavaScript utilizado para la creación de la interfaz de usuario y la parte fron-end de la aplicación. Se ha utilizado este framework por su alta compatibilidad con web3JS e IPFS.
- **Microsoft To-Do:** aplicación de tareas que se ha utilizado para marcar las tareas pendientes del proyecto así como las ya completadas. Muy útil para la planificación del proyecto
- **Overleaf:** editor de LaTeX utilizado para la documentación y elaboración de la memoria
- **Procreate:** aplicación de IOS utilizada para el diseño primitivo de la interfaz de usuario a través de bocetos.
- **Visual Studio Code:** IDE de programación utilizado tanto para Solidity como para React
- **Truffle** servidor de desarrollo para blockchain, en el que se despliegan los contratos de la aplicación y proveedor de Ethereum para la aplicación.
- **Ganache** provee de una interfaz gráfica para interactuar con Truffle.
- **Github:** para el control de versiones y backup del desarrollo.
- **Nodo Infura de IPFS:** para utilizar IPFS (Concepto explicado en el punto 4.3) es necesario establecer un nodo desde el que acceder a la red. Si bien es posible desplegarlo en un servidor propio, se ha creído conveniente utilizar un nodo de Infura.io, que ofrece un API para interactuar con él cómodamente y se ocupan del mantenimiento.
- **Material UI:** librería de componentes de React que permite seguir la filosofía de diseño de interfaz Material Design
- **Web3JS:** librería de javascript utilizada para interactuar con los contratos desplegados en la red de Ethereum.
- **Remix IDE:** IDE online de solidity utilizado para desplegar los contratos al principio y hacer pruebas con ellos, ya que permite hacer pruebas fácilmente sin necesidad de interactuar con los contratos a través de web3js.
- **Draw.io:** aplicación para hacer diagramas de todo tipo.
- **StarUML:** aplicación utilizada para el diagrama de casos de uso.

- **npm:** gestor de paquetes de NodeJS.



## Part II

# Memoria técnica del proyecto



# Chapter 3

## Análisis

### 3.1 Descripción de los Actores

En Proa hay 2 actores completamente diferenciados, los empleados y los administradores. Los empleados son la base de los usuarios, que tienen acceso a un saldo de tokens privado para comercializar e intercambiar. Además pueden proponer iniciativas y beneficiarse de la tienda incorporada en la aplicación.

Los administradores son un tipo especial de empleado que se encarga de manejar los fondos de la empresa, aprobar las iniciativas que proponen el resto de empleados, así como de proponer las iniciativas propias de la empresa y de administrar a los propios empleados de la empresa.

### 3.2 Requisitos

#### 3.2.1 Requisitos de negocio

RN-01	Incentivar la participación de los trabajadores en actividades formativas y de interés para la empresa
RN-02	Motivar a los empleados a través de premios
RN-03	Incentivar la ayuda mutua entre empleados

Table 3.1: Requisitos de negocio

### 3.2.2 Requisitos de usuario

Empleado	
RU-01	Acceder al sistema a través de su cuenta de Metamask
RU-02	Usar la tienda del sistema
RU-03	Podrá ver su historial de transacciones así como su saldo
RU-04	Podrá intercambiar tokens con el resto de empleados
RU-05	Podrá apuntarse a iniciativas creadas
RU-06	Podrá proponer una iniciativa
RU-07	Podrá editar o cancelar una iniciativa propuesta por él
RU-08	Podrá eliminar las notificaciones que haya visto
RU-09	Podrá cerrar una iniciativa de coste que haya sido propuesta por él, confirmando los empleados apuntados que hayan asistido

Table 3.2: Requisitos de Empleado

Administrador	
RU-10	Podrá acceder a todas las funciones de empleado
RU-11	Añadir, editar o eliminar un empleado al sistema
RU-12	Enviar tokens de la empresa a un empleado
RU-13	Podrá añadir, editar o eliminar recompensas de la tienda
RU-14	Podrá aprobar o rechazar las iniciativas propuestas por los empleados
RU-15	Podrá aprobar o rechazar la compra realizada por un empleado
RU-16	Podrá marcar como acabada una iniciativa recompensada, confirmando los empleados apuntados que recibirán tokens
RU-17	Podrá cancelar o editar cualquier iniciativa

Table 3.3: Requisitos de Administrador



### 3.2.3 Requisitos de información

RI-01	Todos los datos de las transacciones quedan registrados en la blockchain, ligados a cada contrato distinto (token, iniciativa, usuario, etc.)
RI-02	Los datos pesados de la aplicación así como los que no son estrictamente necesarios para manejar los contratos se guardarán en archivos JSON dentro de un nodo IPFS

Table 3.4: Requisitos de información

### 3.2.4 Requisitos Funcionales

RF-01	El sistema reflejará las transacciones en la blockchain
RF-02	El sistema se encargará de mandar los tokens establecidos en una iniciativa con recompensa a los participantes cuando esta se cierre
RF-03	Una iniciativa podrá tener los siguientes estados: En trámite, Aprobada, Rechazada, Cancelada, Cerrada
RF-04	Una iniciativa en trámite solo puede ser vista por su creador y por los administradores
RF-05	Una iniciativa en Aprobada, podrá ser vista por todos los usuarios si no está cerrada
RF-06	Si una iniciativa está cerrada solo la podrán ver aquellos que hayan participado en ella y el creador
RF-07	Una iniciativa en Rechazada solo podrá ser vista por su creador
RF-08	Una iniciativa en Cancelada solo podrá ser vista por su creador y los usuarios que estaban apuntados
RF-09	Una iniciativa en trámite solo puede ser vista por su creador y por los administradores
RF-10	Una iniciativa podrá tener un coste en tokens o entregar tokens como recompensa por participar
RF-11	La recompensa en tokens de una iniciativa solo podrá ser modificada por un admin
RF-12	Una iniciativa que otorga tokens como recompensa, solo puede ser cerrada por un admin y otorgará siempre como recompensa tokens de la empresa
RF-13	Una iniciativa que cuesta tokens pagará estos al usuario creador si este no es admin, si lo es, se los pagará a la empresa

RF-14	Una iniciativa que cuesta tokens y no ha sido creada por un admin solo puede ser cerrada por el creador
RF-15	El sistema permitirá a los usuarios filtrar los diferentes tipos de iniciativa, así como por fecha
RF-16	El sistema notificará al creador cuando se apruebe o rechace una iniciativa propuesta por él
RF-17	El sistema notificará al comprador cuando se apruebe o rechace la compra en la tienda a través de un administrador
RF-18	El sistema notificará a los usuarios cuando reciban tokens
RF-19	El sistema notificará a los usuarios apuntados a una iniciativa si esta cambia sus condiciones
RF-20	Cuando un usuario se apunta a una actividad propuesta por un usuario y que cuesta tokens, el sistema le concede allowance en la cuenta del usuario por valor del precio de la actividad al creador
RF-21	Si una actividad que requería allowance a un usuario se cancela, el allowance también se cancelará
RF-22	Si una actividad que tenía un coste se intenta cerrar y algún usuario no tiene saldo suficiente, se le notificará al usuario que está cerrando la actividad para que no deje que los usuarios sin saldo para pagarla participen

Table 3.5: Requisitos Funcionales

### 3.2.5 Requisitos de interfaz externa

RIE-01	El sistema establecerá una conexión con el wallet meta-mask para autenticar usuarios, gestionar los tokens y las operaciones con los contratos
RIE-02	El nodo de IPFS será proporcionado por Infura.io

Table 3.6: Requisitos de interfaz Externa

### 3.2.6 Requisitos no funcionales

Requisitos de usabilidad	
RNF-01	El sistema será accesible desde cualquier dispositivo con un navegador
RNF-02	La interfaz del sistema se ajustará automáticamente en función del rol del usuario
RNF-03	La interfaz del sistema se ajustará automáticamente en función del dispositivo desde el que se acceda al mismo
RNF-04	El sistema mostrará en todo momento un panel informativo al usuario con su nombre, apellidos, avatar y saldo disponible

Table 3.7: Requisitos No Funcionales de usabilidad

Requisitos de seguridad	
RNF-05	El sistema validará todas las transacciones antes de llevarlas a cabo
RNF-06	El sistema solo permitirá acceso a los usuarios autenticados con metamask y cuya dirección esté registrada en el contrato User
RNF-07	Los contratos solo permitirán ejecutar sus funciones a direcciones registradas en el contrato User
RNF-08	Los contratos solo permitirán ejecutar funciones sensibles a los administradores
RNF-09	Solo la cuenta de la empresa puede iniciar por primera vez la aplicación, hasta que crea más administradores.
RNF-10	El número total de tokens es fijo y solo lo puede modificar la empresa

Table 3.8: Requisitos No Funcionales de Seguridad



### 3.3 Casos de Uso

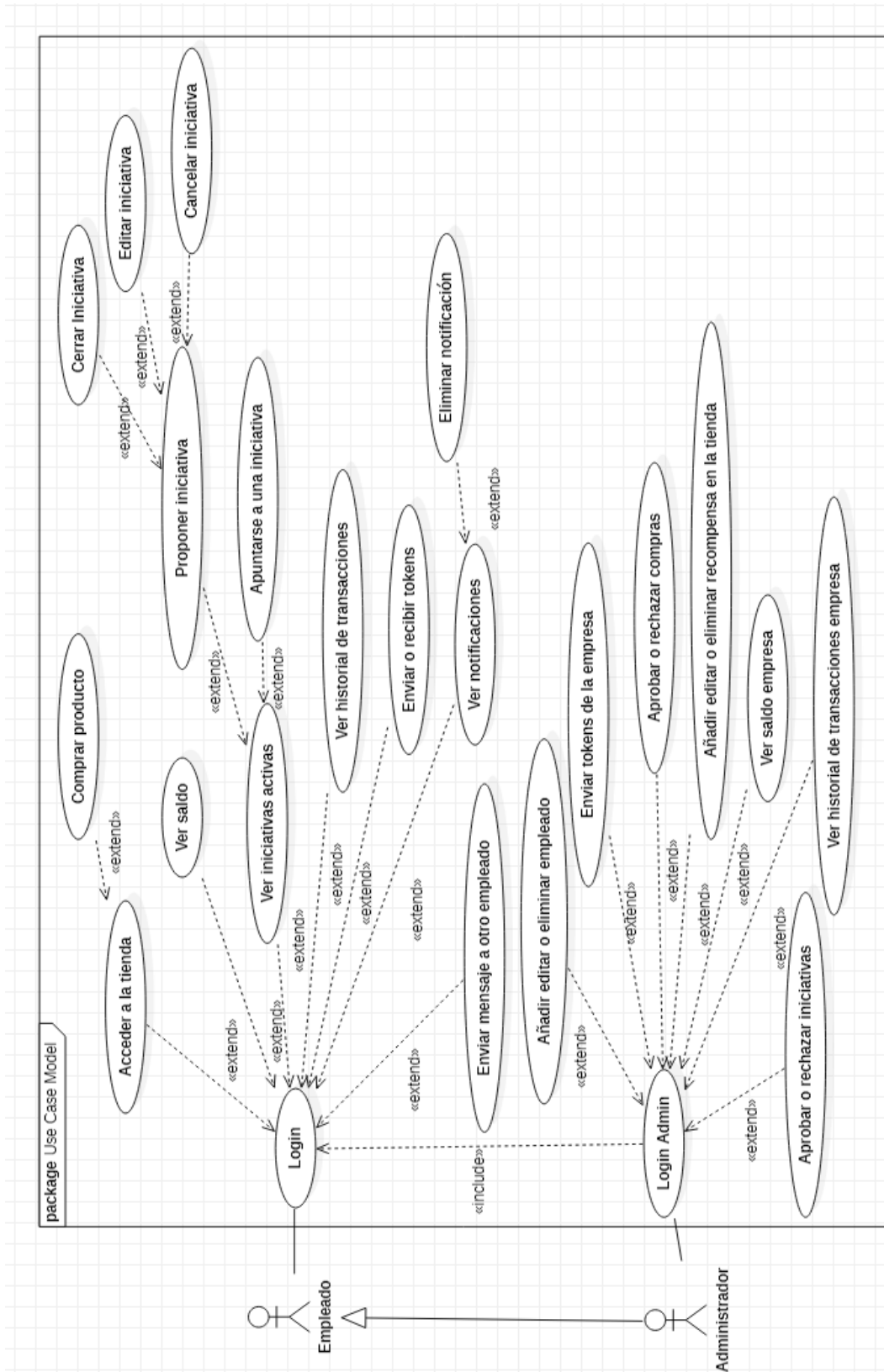


Figure 3.1: Diagrama de casos de uso

## 3.3.1 Especificación de casos de Uso

<b>Nombre e ID del CU</b>	CU-01. Proponer Iniciativa
<b>Actor</b>	Usuario
<b>Descripción</b>	El usuario rellena un formulario con los datos de la iniciativa a proponer y, si todo está correcto, pasará al estado tramitando
<b>Precondiciones</b>	PRE-1. El usuario está identificado en el sistema.
<b>Postcondiciones</b>	POST-1. La iniciativa queda almacenada en el sistema pendiente de ser tramitada.
<b>Flujo normal</b>	<p>FN1 El actor introduce los datos en el formulario proporcionado por el sistema.</p> <p>FN2 El sistema comprueba los datos introducidos.</p> <p>FN3 Si los datos son correctos, el sistema almacena la iniciativa y la muestra con el resto con la etiqueta "tramitando".</p>
<b>Flujo alternativo 1</b>	<p>FA3 Si los datos son incorrectos se informa al usuario del error y no se procede a almacenar la iniciativa.</p>
<b>Prioridad</b>	Alta

Table 3.9: CU-01. Proponer Iniciativa

<b>Nombre e ID del CU</b>	CU-02. Tramitar Iniciativa
<b>Actor</b>	Administrador
<b>Descripción</b>	El actor aprueba o rechaza una iniciativa con el estado "tramitando"
<b>Precondiciones</b>	<p>PRE-1 El usuario está identificado en el sistema como administrador.</p> <p>PRE-2 El usuario está en la página de iniciativas.</p>
<b>Postcondiciones</b>	POST-1. La iniciativa queda almacenada como aprobada o rechazada.
<b>Flujo normal</b>	<p>FN1 El actor selecciona la iniciativa con la etiqueta "tramitando".</p> <p>FN2 El actor selecciona la opción "Aprobar".</p> <p>FN3 El sistema actualiza la iniciativa y pasa a mostrársela a todos los usuarios para que se apunten.</p>

<b>Flujo alternativo 1</b>	<p>FA2 Si el actor selecciona la opción "Rechazar".</p> <p>FA3 El sistema actualiza la iniciativa y la marca como rechazada.</p> <p>FA4 El sistema actualiza la lista de las iniciativas, eliminando la que ha sido rechazada.</p>
<b>Prioridad</b>	Alta

Table 3.10: CU-02. Tramitar Iniciativa

<b>Nombre e ID del CU</b>	CU-03. Login
<b>Actor</b>	Empleado o Administrador
<b>Descripción</b>	El actor se loguea en el sistema a través de la extensión de Metamask
<b>Precondiciones</b>	PRE-1. El usuario no está identificado en el sistema.
<b>Postcondiciones</b>	POST-1. El usuario accede a la página de iniciativas si es identificado o a la de desconocido si no lo es.
<b>Flujo normal</b>	<p>FN1 La extensión de Metamask detecta automáticamente que ha entrado en la aplicación y pide al usuario que seleccione una cuenta.</p> <p>FN2 El usuario introduce su cuenta.</p> <p>FN3 El sistema reconoce el address proporcionado por metamask y carga la página de iniciativas.</p>
<b>Flujo alternativo 1</b>	<p>FA3 Si el sistema no reconoce el address proporcionado por metamask, se carga la página de desconocido.</p>
<b>Prioridad</b>	Alta

Table 3.11: CU-03. Login

### 3.4 Estimación y presupuestos

Dado que gran parte de la dificultad del proyecto ha sido el aprender a utilizar nuevas tecnologías, la estimación del esfuerzo necesario para sacar el proyecto adelante ha sido muy complicada. Aún así, cuando comencé el proyecto estimé unas 306h de trabajo entre formación y desarrollo durante 4 meses (mayo-agosto). Este número de horas es el resultado de dedicar 18h semanales durante las 17 semanas que calculé inicialmente para el proyecto. Además, el coste de cada hora de trabajo ha sido establecido en 10,63€, que

es el salario medio de un desarrollador software junior en España [12]. A esto se le añade un 50% sobre el valor del trabajo como beneficio, este porcentaje viene dado porque son tecnologías muy nuevas en las que cuesta encontrar gente experta. Por tanto la estimación de presupuesto del proyecto es el descrito en la tabla 3.12.

Presupuesto			
Gasto	Coste (Mes)	Meses	Total
Estación de trabajo	coste total: 2.000€ vida útil: 5 años	4	133,4€
Internet	70€	4	280€
Luz	45€	4	180€
Oficina	250€	4	1.000€
RRHH	10,63€ (hora)	306 horas	3.252,78€
Beneficio	-	-	1.626,39€
<b>Total</b>	-	-	<b>6.472,50€</b>

Table 3.12: Presupuesto

Sin embargo, la realidad ha sido muy distinta, debido a complicaciones durante el desarrollo del proyecto, este se ha extendido 3 semanas más de lo previsto y ha ocasionado casi 100h más de trabajo, haciendo un total de 400h. Debido a esto, el coste real del proyecto ha sido el de la tabla 3.13.

Coste			
Gasto	Coste (Mes)	Meses	Total
Estación de trabajo	coste total: 2.000€ vida útil: 5 años	5	133,4€
Internet	70€	5	350€
Luz	45€	5	225€
Oficina	250€	5	1.250€
RRHH	10,63€ (hora)	400 horas	4.252€
Beneficio	-	-	2.126€
<b>Total</b>	-	-	<b>8.369,67€</b>

Table 3.13: Coste

Como podemos observar, la falta de experiencia en la tecnología ha ocasionado un fallo en la planificación que nos ha supuesto 3 semanas de retraso y 100 horas más de



trabajo, lo que supone un sobre coste de 1.897,67€



# Chapter 4

## Diseño e Implementación

### 4.1 Arquitectura

#### 4.1.1 Arquitectura lógica

La arquitectura lógica se explica en el diagrama 4.1, que muestra cómo React es capaz de interactuar con Metamask y la blockchain de Ethereum a través de la librería Web3JS. Por otra parte, el nodo de IPFS se comunica directamente con la aplicación.

Dentro de la aplicación React, se ha incluido un resumen de los componentes principales. El componente más grande de la aplicación es el context provider, que guardará los datos de los contratos, address e IPFS para que el resto de los componentes puedan llamar a los servicios externos.

Dentro del context provider tenemos dos grandes componentes, PaginaDesconocido es simplemente una página estática que se muestra a los usuarios que no se haya podido identificar a través de Metamask. PaginaUsuario es para aquellos usuarios que sí han sido autenticados por el sistema. Se compone de 4 componentes principales: NavBar y Drawer son los menús que rodean la aplicación y sirven para navegar por las diferentes páginas, mostrar las notificaciones y mostrar los datos del usuario logeado.

Los otros dos componentes son los dedicados al Empleado y al Admin, que tienen dentro las páginas necesarias para que ambos puedan realizar sus funciones. Los administradores tienen acceso a las páginas de Empleado y además verán en ellas funciones únicas para ellos.

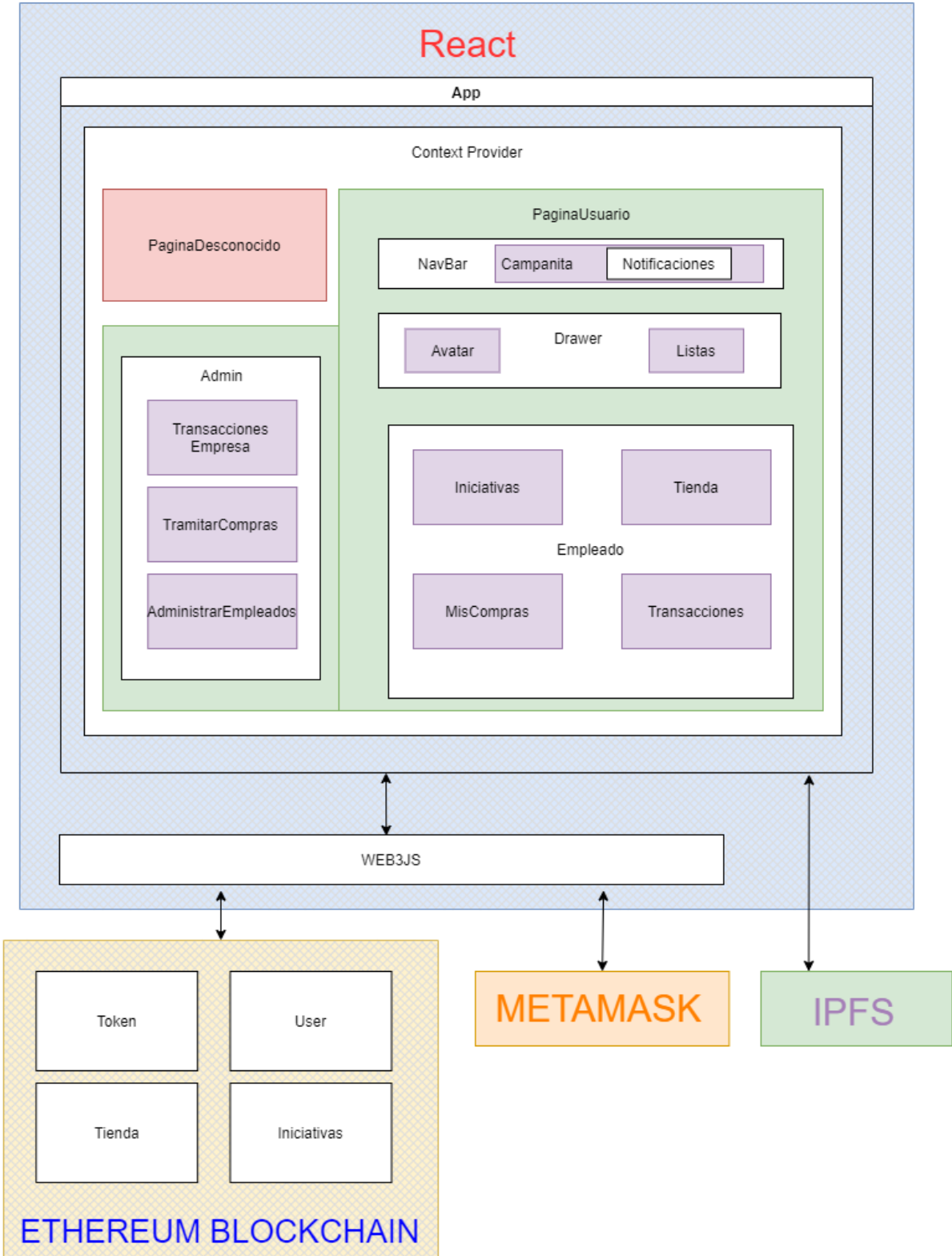


Figure 4.1: Arquitectura lógica

### 4.1.2 Arquitectura física

La arquitectura física de la aplicación que se describe en la figura 4.2 no es muy compleja ya que la mayor parte de ella no depende de nosotros. Lo único que necesitamos proveer es de un servidor web sencillo que permita al usuario acceder a la aplicación React. Una vez el usuario ha obtenido la aplicación de ese servidor, el cliente se comunica directamente con la red de Ethereum y con el nodo infura de IPFS.

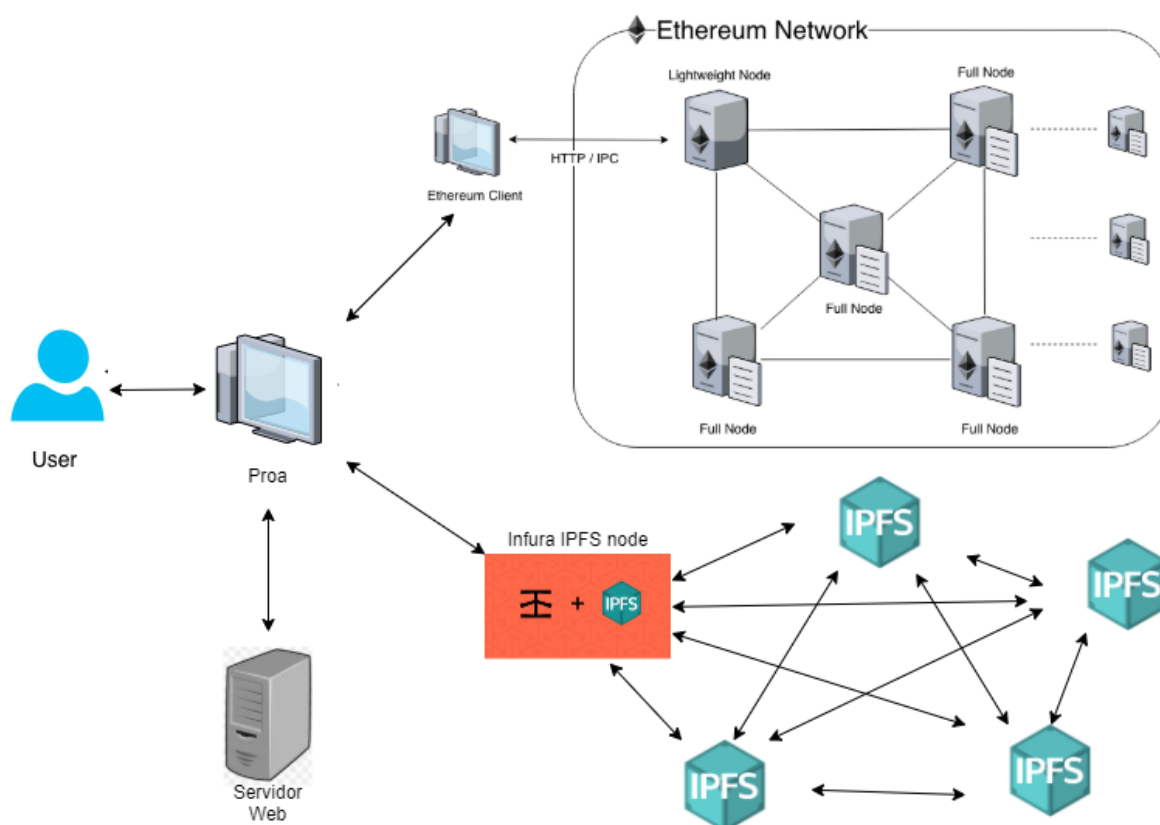


Figure 4.2: Arquitectura física

## 4.2 Diagrama de clases y Árbol de características

En el diagrama de clases de la figura 4.3 se pueden apreciar los 4 contratos que se han desplegado en la red de Ethereum: User, Iniciativas, Tienda y Token. En cada uno de estos contratos se han coloreado distintos campos en función de su tipo: Amarillo para los atributos, verde para los eventos y azul para los modifiers.

# Chapter 4. Diseño e Implementación

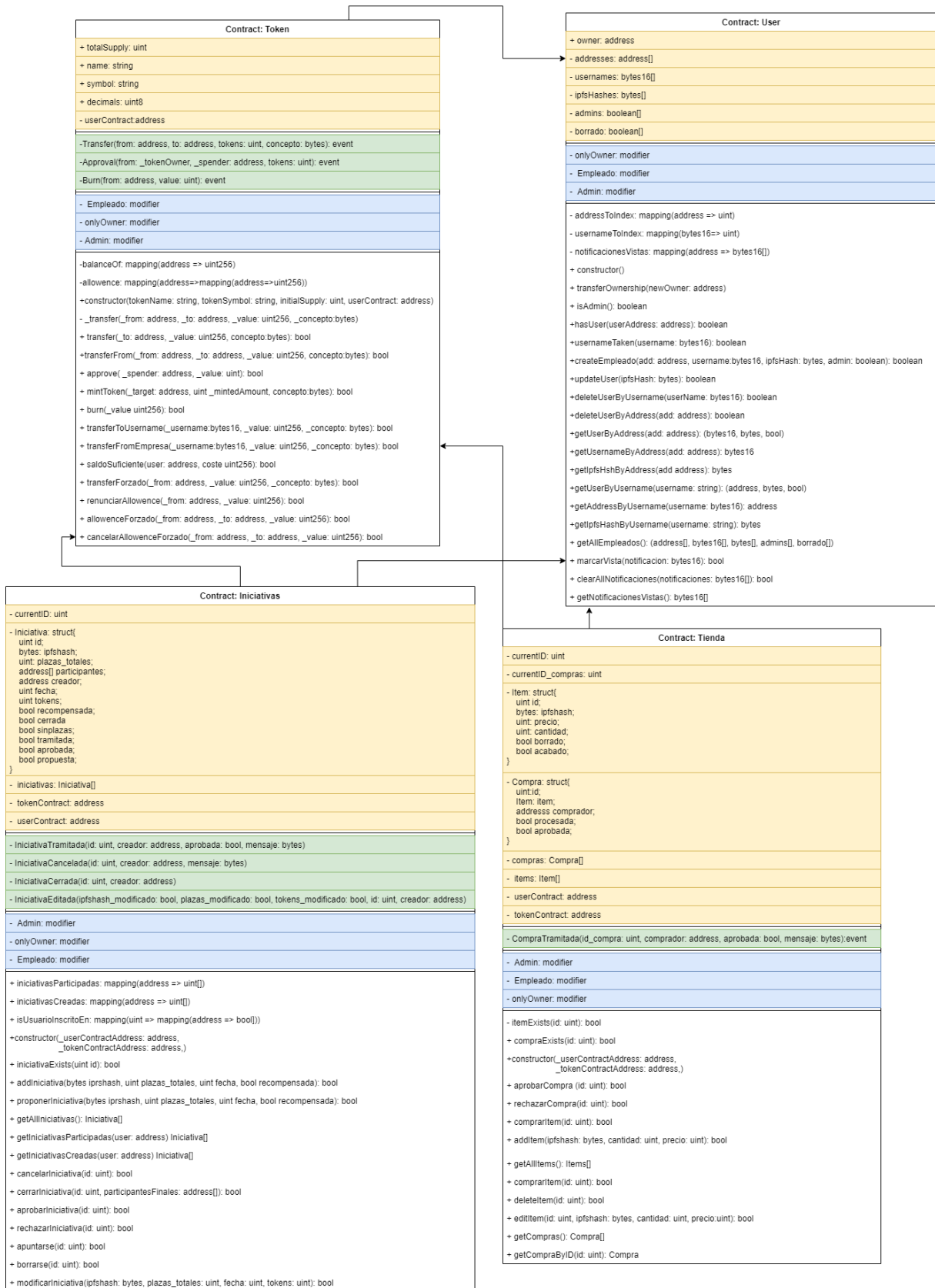


Figure 4.3: Diagrama de clases

El árbol de características de la figura 4.4 representa las características principales de Proa: las iniciativas, los tokens y las notificaciones.

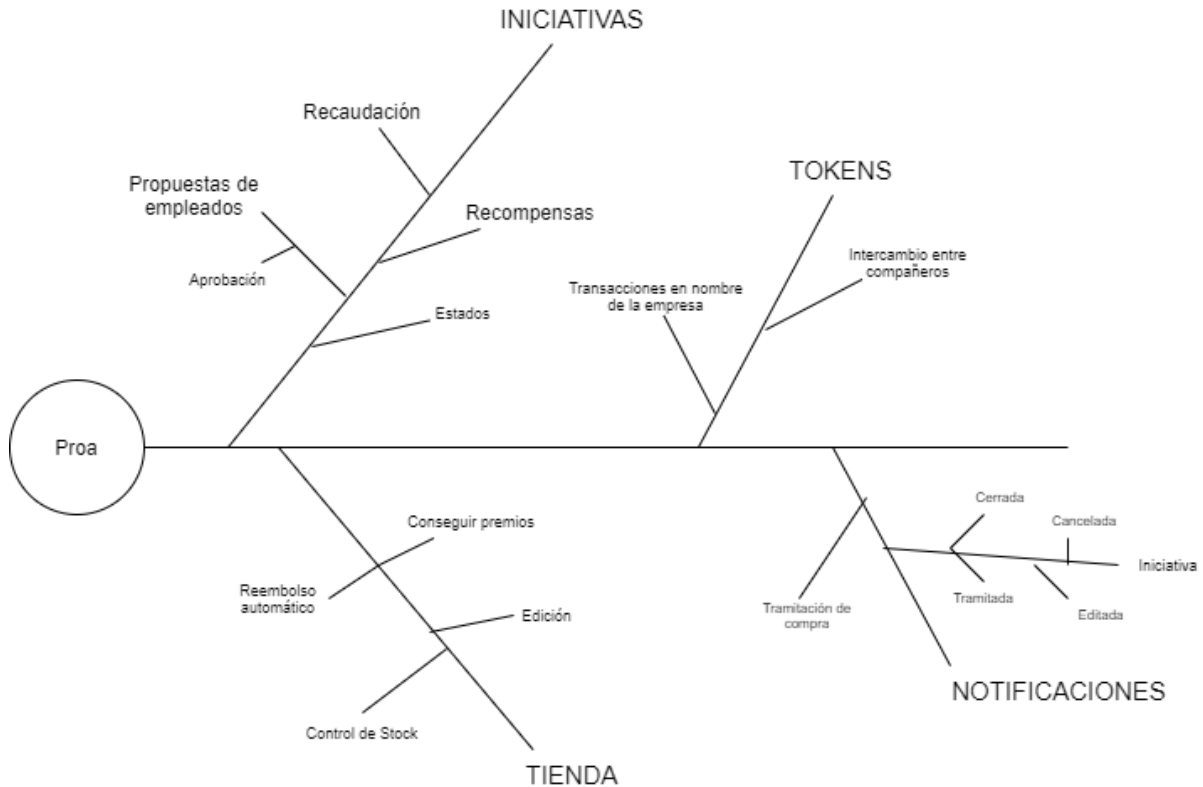


Figure 4.4: Árbol de Características

## 4.3 Diseño de datos

En este proyecto, los datos tienen un tratamiento especial dado que no se va a utilizar ninguna base de datos tradicional. Como la intención del proyecto es una aplicación totalmente distribuida, no tendría sentido centralizar nuestros datos en una base de datos, por ello vamos a aprovechar el almacenamiento que nos proporciona la tecnología blockchain.

Los principales problemas de este enfoque, es que los datos almacenados en la blockchain son difícilmente modificables y además, que es caro almacenar grandes cantidades de datos, por ello, se ha optado por utilizar un almacenamiento complementario a la blockchain para almacenar los datos no necesarios para la realización de operaciones en los contratos y para almacenar los datos pesados (imágenes). Este sistema complementario es IPFS, un sistema distribuido de archivos que nos permite subir cualquier archivos a un nodo de la red peer-to-peer y recuperarlo más adelante introduciendo una clave hash (que es la que sí almacenaremos en la blockchain).

En el Diagrama Entidad-Relación de la figura 4.5 podemos ver que los atributos tienen

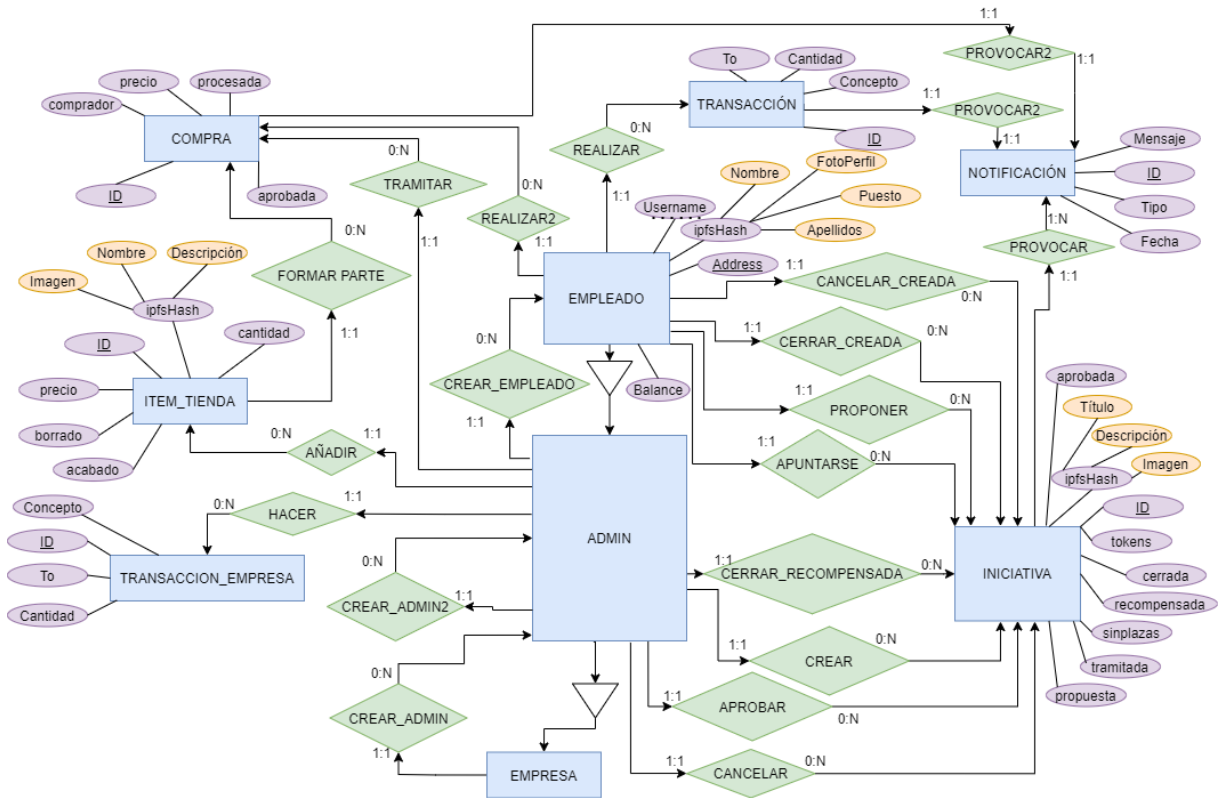


Figure 4.5: Diagrama Entidad-Relación

dos posibles colores, los atributos morados son aquellos que se utilizan dentro de los contratos, por lo que necesariamente tienen que estar en la blockchain. Los atributos naranjas son los que almacenaremos en IPFS.

El modelo conceptual de los datos se puede establecer sin mucha dificultad, sin embargo, cuando pasamos al modelo lógico de los datos no es tan sencillo plasmar el almacenamiento de la aplicación. Para explicar el modelo lógico he creado el diagrama mostrado en la figura 4.6. Aquí podemos ver los 4 contratos que se han implementado: Tienda, Tokens, Iniciativas y Tienda. Para cada contrato, se pueden ver los datos que se guardan en forma de variables, habiendo una variable algo especial: ipfsHash (marcada en azul). Esta variable es el hash que se le pasará a IPFS para recuperar los datos indicados dentro de esa sección en forma de archivo JSON, las imágenes serán almacenadas como una url originada tras subirlas independientemente a IPFS.

Los Mappings marcados con el color verde son objetos que relacionan dos datos, por ejemplo en el mapping addressToIndex del contrato User, nos relaciona un address con el índice en el que está guardado el usuario.

Por último, los eventos (marcados en rojo) son producidos por el contrato cuando se ejecutan ciertas funciones y guardan en la blockchain información sobre una transacción. Es lo que la aplicación utilizará para mostrar las notificaciones a los usuarios.



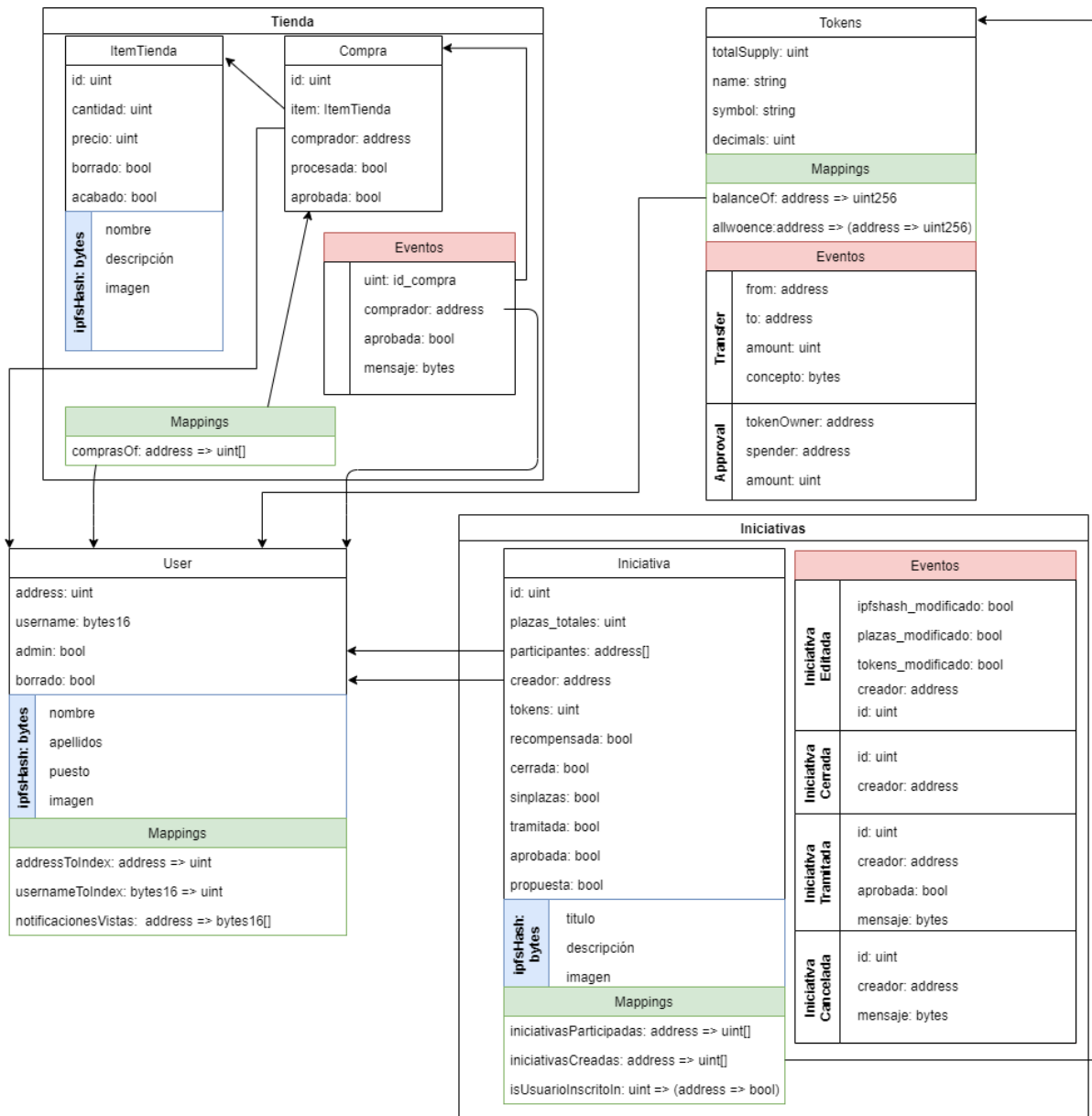


Figure 4.6: Modelo de datos

## 4.4 Diseño de la interfaz

En el diseño de la interfaz se ha seguido la filosofía de diseño de Material Design establecida por Google en 2014 [9]. Para ello, se ha utilizado la librería de componentes de React Material-UI, que proporciona algunos componentes diseñados con esta filosofía. Además, la interfaz se adapta automáticamente en función del dispositivo que usemos. En la figura 4.7 podemos ver la aplicación en su versión de escritorio y en la figura 4.8 podemos ver la versión móvil de la aplicación

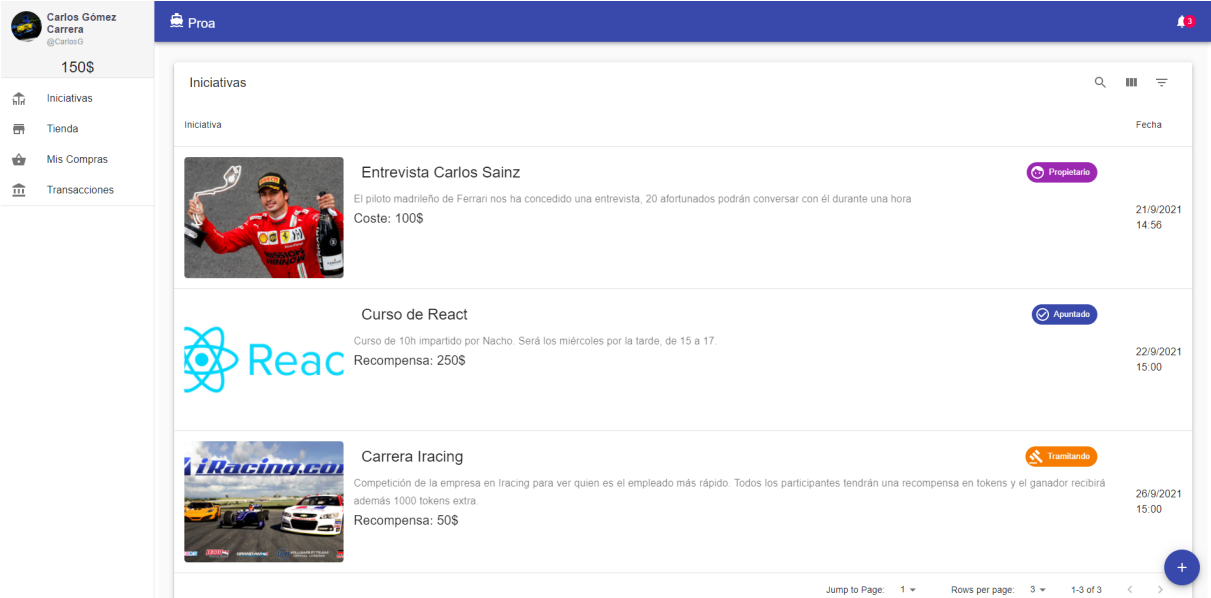
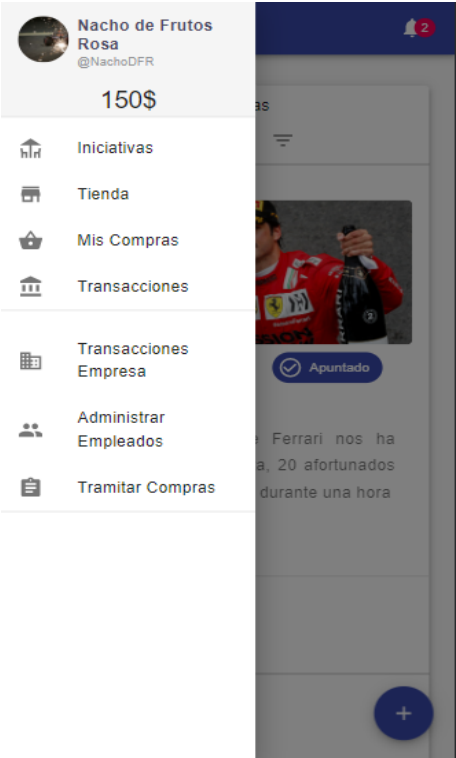


Figure 4.7: Vista general escritorio



(a) Vista general móvil (menú desplegado)



(b) Vista general móvil (menú oculto)

Figure 4.8: Visión general móvil

La Interfaz se puede dividir en tres partes: la barra de navegación, el menú lateral y el espacio para el contenido. Este último, es simplemente el espacio en el que se renderizarán las distintas páginas de la aplicación

#### 4.4.1 La barra de navegación

La barra de navegación, figura 4.9, se compone de tres elementos:

- Botón de despliegue del menú lateral: Este botón solo aparecerá para dispositivos pequeños, ya que es cuando el menú lateral no aparece visible por defecto.
- Icono y nombre de la aplicación
- Notificaciones: A la derecha del todo, se muestra un icono de notificaciones, que indicará el número de notificaciones que el usuario tiene pendientes. Si se hace click, se mostrarán todas estas notificaciones (como se muestra en la figura 4.10, que el usuario podrá desestimar. Además, si el usuario pasa el cursor por encima de una notificación con mensaje, este saldrá en un pequeño popup.



Figure 4.9: Barra de navegación



Figure 4.10: notificaciones

### 4.4.2 Menú lateral

El menú lateral, como ya hemos visto en las figuras 4.7 y 4.8, puede ser permanente (versión de escritorio) u ocultarse en el lateral izquierdo (versión móvil). Este componente esta compuesto por 3 partes:

- Información personal: Este es el apartado de la parte superior del menú, muestra el avatar del usuario, su nombre y apellidos, su nombre de usuario y el saldo disponible.
- Lista de acciones para empleados: Este componente está formado de 4 botones que permiten al usuario navegar por distintas páginas de la aplicación. Se muestra a todos los usuarios
- Lista de acciones para administradores: Este componente es muy similar al anterior, solo que contiene únicamente las acciones propias de los administradores. Para apreciar la diferencia, podemos ver como en la figura 4.7 solo se muestra la lista de empleados, mientras que en la figura 4.8 se muestran todas las opciones

### 4.4.3 Página de iniciativas

Esta es la base de la aplicación y la que aparece de forma predeterminada. Se puede apreciar en las figuras 4.7 y 4.8b. Consta de una tabla, que contiene todas las iniciativas y su fecha y de un botón, a través del cual se proponen nuevas iniciativas. Como se aprecia en la figura 4.7, las iniciativas tienen una etiqueta distinta en función de su estado y relación con el usuario: "Propietario", "Apuntado", "Tramitando", "Rechazada", "Cancelada" o "Llena".

Cuando se presiona el botón de añadir, se abre un popup con un formulario en el que añadir los datos de la iniciativa (figura 4.11).

Si lo que se presiona es el cuerpo de una iniciativa, se abrirán los detalles de esta y los botones de abajo se adaptarán automáticamente a las acciones que el usuario puede realizar. Por ejemplo, en la figura 4.12 el usuario, al ser administrado y estar apuntado, puede desapuntarse, finalizar la iniciativa, cancelarla o editarla. En la versión móvil, los botones cambian un poco para que entren correctamente en la pantalla, como se puede apreciar en la figura 4.13

**Añadir Iniciativa**

Título  
Entrevista con Carlos Sainz Jr.

Descripción de la iniciativa  
El piloto español nos proporciona una entrevista en exclusiva, tiene po

Fecha de la iniciativa  
23/09/2021 14:27

Piadas  
20

¿Es una actividad recompensada?

Coste en tokens  
100

**SUBIR IMAGEN**

CANCELAR AÑADIR

Figure 4.11: Añadir Iniciativa

#### 4.4.4 Tienda

La interfaz de la tienda es muy similar a la de las iniciativas, solo que sin tabla, se presentan los objetos disponibles directamente, como se muestra en la figura 4.14. Los productos que se hayan acabado, se mostrarán únicamente a los administradores y en blanco y negro, para que estos puedan aumentar el stock si lo creen conveniente.

Cualquier objeto puede ser borrado por un administrador con el botón con icono de papelera, en cuyo caso desaparecerá de la tienda.

Si un administrador pulsa el botón de añadir (que es exclusivo de este rol), saldrá un popup similar al de las iniciativas para añadir el nuevo producto, como se ve en la figura 4.11

Los adminsitradores también podrán editar los productos existentes utilizando el icono de editar, al lado del de borrar, que mostraría un popup como el de la figura ??

#### 4.4.5 Página mis compras

Esta página es donde un usuario puede consultar el historial de sus compras y el estado de las mismas, que puede ser: "Aprobada", "Rechazada", "Tramitando". Figura 4.18

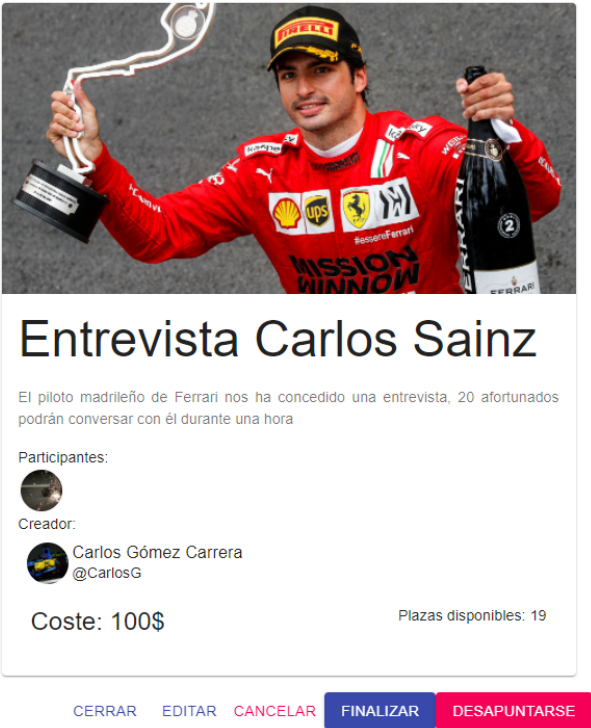


Figure 4.12: Detalles Iniciativa

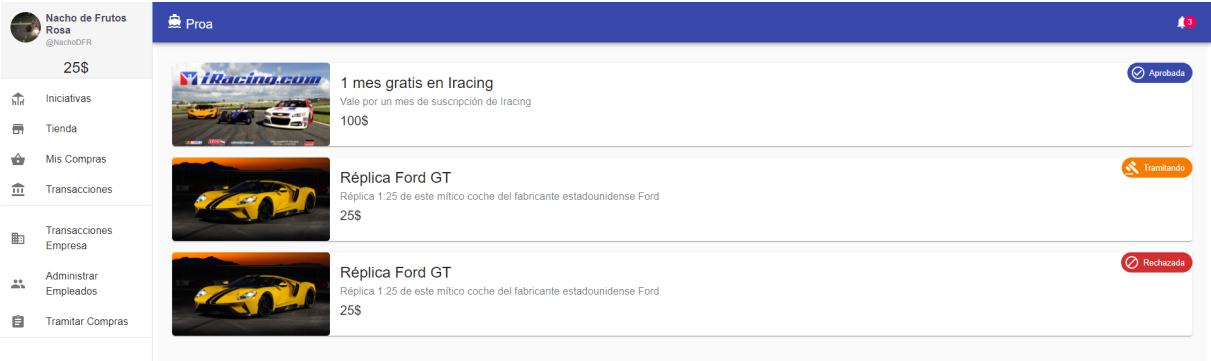


Figure 4.18: Página Mis Compras



Figure 4.13: Detalles Iniciativa Móvil

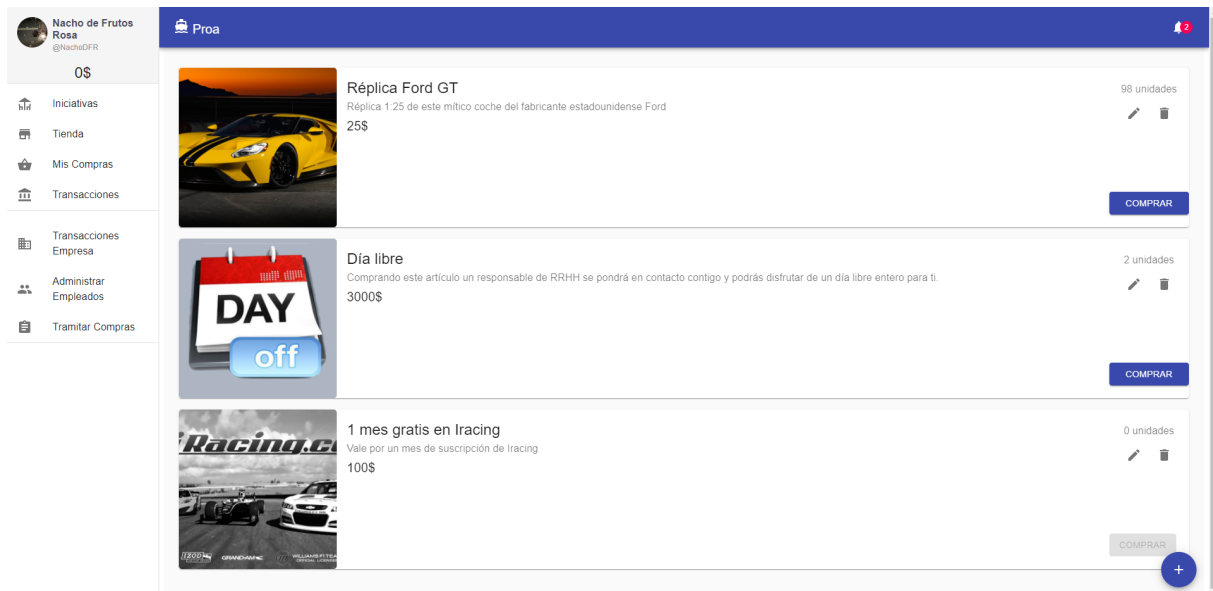


Figure 4.14: Tienda

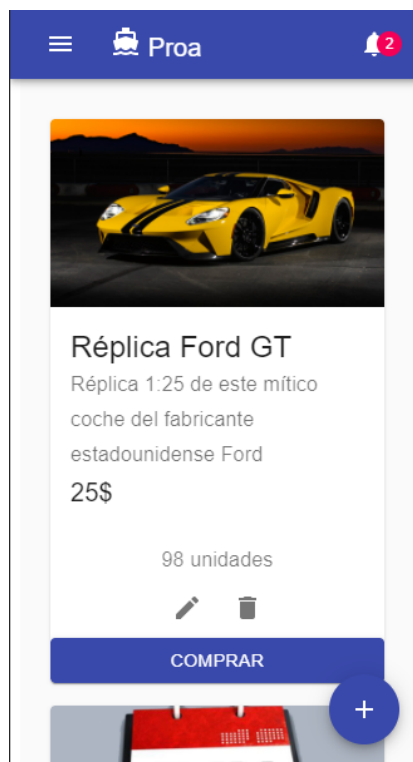


Figure 4.15: Tienda versión móvil

#### 4.4.6 Página Tramitar Compras

Esta página exclusiva de los administradores es donde se gestionan las compras realizadas en la tienda. Cuando un administrador tramita una compra tiene que gestionar cómo conseguir el bien comprado al usuario comprador y si lo consigue aprobar la compra. Si esto no es posible, la puede rechazar. Figura 4.19



### Añadir producto a la tienda

Nombre  
Producto

---

Descripción  
descripción del producto

---


Precio  
10

---

Cantidad  
10

---

SUBIR IMAGEN




CANCELAR   GUARDAR

Figure 4.16: Añadir producto a la tienda

Nacho de Frutos Rosa @NachoDFR
Proa 🔔 2

0\$


- 🏠 Iniciativas
- 🛒 Tienda
- 👛 Mis Compras
- 🏠 Transacciones
- 📅 Transacciones Empresa
- 👤 Administrar Empleados
- 📄 Tramitar Compras



**1 mes gratis en Iracing**

Vale por un mes de suscripción de Iracing


100\$



**Réplica Ford GT**

Réplica 1.25 de este mítico coche del fabricante estadounidense Ford


25\$



**Réplica Ford GT**

Réplica 1.25 de este mítico coche del fabricante estadounidense Ford


25\$



Nacho de Frutos Rosa @NachoDFR

RECHAZAR


APROBAR



Nacho de Frutos Rosa @NachoDFR

RECHAZAR

APROBAR



Nacho de Frutos Rosa @NachoDFR

RECHAZAR

APROBAR

Figure 4.19: Tramitar compras

Editar 1 mes gratis en Iracing

Editar Foto

SUBIR IMAGEN

Cambiar Nombre

Nuevo Nombre  
Nuevo Nombre

---

Cambiar Descripción

Nuevo Descripción  
Nueva descripción

---

Cambiar Cantidad

Nueva Cantidad

---

Cambiar Tokens

Nuevo numero de Tokens

---

[CANCELAR](#) [GUARDAR](#)

Figure 4.17: Editar producto de la tienda

### 4.4.7 Transacciones y Transacciones Empresa

Esta página muestra una tabla con todas las transacciones del usuario (o de la empresa en el caso de esa página). Ambas páginas son iguales estéticamente con la única diferencia de que en Transacciones Empresa se muestra el saldo de la empresa. Figura 4.20. Si el usuario hace click en el botón de enviar, se abrirá el popup de la figura 4.21 para introducir los datos de la nueva transacción.

The screenshot shows the Proa interface. At the top, the user profile 'Nacho de Frutos Rosa' is visible with a balance of 25\$. The main header displays 'Saldo de la empresa: 9825\$'. Below this is a table titled 'Transacciones de la Empresa' with columns for Origen, Destinatario, Concepto, Fecha, and Cantidad. The table contains seven rows of transaction data. A sidebar on the left lists navigation options like 'Iniciativas', 'Tienda', 'Mis Compras', and 'Transacciones'. A bottom navigation bar includes a home icon and a right-pointing arrow.

Origen	Destinatario	Concepto	Fecha	Cantidad
Empresa	NachoDFR	Reembolso por compra rechazada	20/9/2021	25
NachoDFR	Empresa	Compra producto	20/9/2021	25
NachoDFR	Empresa	Compra producto	20/9/2021	25
NachoDFR	Empresa	Compra producto	20/9/2021	100
Empresa	CarlosG	Bienvenido!	20/9/2021	100
Empresa	NachoDFR	TRANSFERENCIA	20/9/2021	100
Empresa	NachoDFR	Bienvenido!	20/9/2021	100

Figure 4.20: Transacciones Empresa

### Enviar Tokens

Destinatario

CarlosG

Cantidad

100

Concepto

Trasferencia

CANCEL ENVIAR

Figure 4.21: Nueva Transferencia

## 4.4.8 Administrar Empleados

Esta sección exclusiva de los administradores es donde se gestionan todos los empleados. El contenido principal es una tabla con todos los empleados del sistema, como se aprecia en la figura 4.22. Dentro de esta tabla, se puede hacer click en cualquier empleado para ver todos sus datos, mostrados como en la figura 4.23, además si el usuario quiere, se pueden editar todos los datos de un empleado menos el nombre de usuario y la dirección del wallet. Figura 4.24

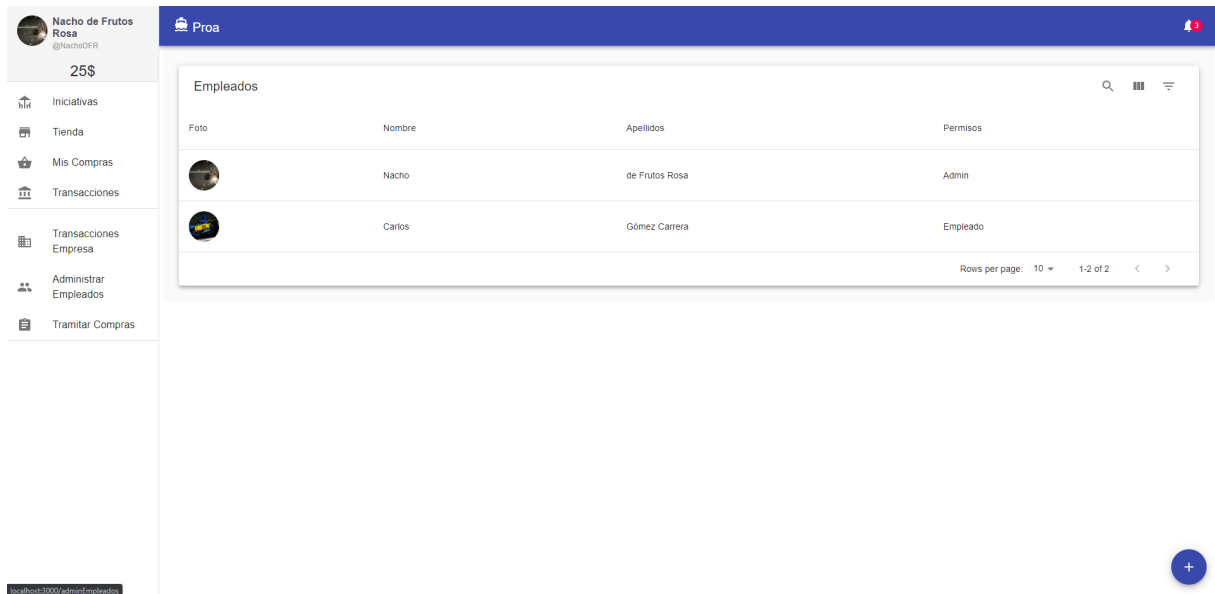


Figure 4.22: Administrar Empleados



Figure 4.23: Datos Empleado

Editar perfil de Nacho de Frutos Rosa

Editar Foto

SUBIR IMAGEN

Cambiar Puesto

Nuevo Puesto

---

Permisos de Admin

CANCELAR    GUARDAR

Figure 4.24: Editar Empleado

## 4.5 Pruebas

### 4.5.1 Pruebas de los smart contract

Debido a la naturaleza de los smart contracts, todas las pruebas que se les han hecho han sido de caja negra, es decir, solo se analizaba que, dado un input, el output fuese el deseado.

Para la realización de estas pruebas se ha utilizado el IDE remix, que permite interactuar con los contratos sin la necesidad de llamarlos desde una aplicación externa propia. Esto se puede apreciar en la figura 4.25

Cuando se terminaba de implementar cada función, el procedimiento era establecer ciertos valores de input y probar que el output fuese el correcto.

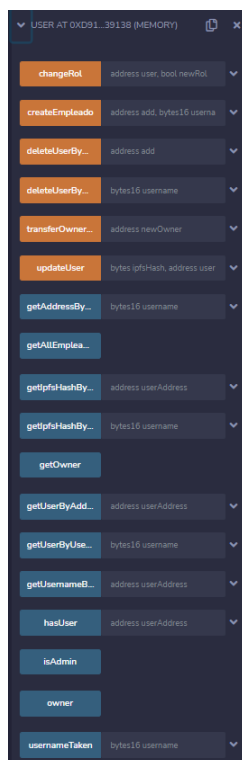


Figure 4.25: Interfaz de Remix para probar contratos

Algunos ejemplos de pruebas realizadas:

<b>ID prueba y contrato</b>	PU-01. User
<b>Método</b>	addUser
<b>Input</b>	<ol style="list-style-type: none"> <li>1. 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2</li> <li>2. "username"</li> <li>3. 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2</li> <li>4. true</li> </ol>
<b>Output esperado</b>	Error
<b>Output</b>	Error
<b>Éxito</b>	Sí
<b>Explicación</b>	El input 2 debe de ser bytes16, un string da error

Table 4.1: PU-01. User

<b>ID prueba y contrato</b>	PI-01. Iniciativa
<b>Información previa</b>	La iniciativa 2 tiene un coste y ha sido creada por un empleado
<b>Método</b>	cerrarIniciativa
<b>Input</b>	<ol style="list-style-type: none"> <li>1. 2</li> <li>2. {0xf91329a3b7d491977c7c430f3308Cd3a663E674B, 0x54e608356ee999dC1734CD81869147885098b6fb}</li> </ol>
<b>Output esperado</b>	true
<b>Output</b>	true
<b>Éxito</b>	Sí
<b>Explicación</b>	Se han cobrado los tokens establecidos a ambos participantes y se han transferido a la cuenta del creador

Table 4.2: PU-01. User

### 4.5.2 Pruebas de los componentes de React

Las pruebas de los componentes de React se han hecho a la vez que la implementación, probando todos los posibles casos de error y solventándolos durante la implementación.





# Chapter 5

## Despliegue

### 5.1 Despliegue real de la aplicación

Lo descrito en este proyecto es todo una versión de desarrollo. Para un despliegue real haría falta seguir unos pasos más:

1. Cambiar el proveedor de ethereum de ganache a una red de pruebas que se asemeje más a la realidad, por ejemplo Rinkeby.
2. Probar los contratos en esta red y ver si la experiencia de usuario obtenida es correcta.
3. Volver a cambiar el proveedor de Ethereum a la red principal, esto ya tiene coste económico
4. Crear una build de producción de react.
5. Desplegar la build de producción en un servidor web.
6. Crear los primeros administradores con la cuenta de la empresa (la que ha desplegado los contratos).

Con esto ya tendríamos una aplicación totalmente funcional y lista para usar.

### 5.2 Coste de despliegue y operación

Todas las transacciones en la blockchain de Ethereum tienen un coste, es el llamado gas, por eso hay que tener cuidado de solo usar la blockchain para lo imprescindible y no hacer escrituras en exceso. Con el coste de gas (0.000000050) y el precio de Ethereum (2923€) actuales, el despliegue de los 4 contratos sería de 1,702.75€ .

Además de esto, cada vez que se escribe en la blockchain hay que hacer otro pago. Tras 20 minutos utilizando la aplicación de forma exhaustiva, he llegado a gastar 200€ en comisiones.

Además de el coste de desplegar y utilizar los contratos, hay que sumar el coste de operación del servidor web que proporcionará la aplicación de React. Esto se puede desplegar en un hosting web si la empresa no se quiere hacer cargo del mantenimiento de este servidor. Un servicio así costaría unos 100€/año [8]

Por último, el servicio que nos proporciona infura, el nodo de IPFS, tiene un coste gratuito hasta los 5GB de almacenamiento, a partir de ahí cobran \$0.08 por GB almacenado y \$0.12 por GB transferido.[11]

# Chapter 6

## Ampliaciones y Conclusión

### 6.1 Posibles ampliaciones y trabajo futuro

Como es lógico, este proyecto todavía tiene margen de ampliación y mejora. Por ello, en esta sección se describen tres funcionalidades extra que me parecen interesantes pero que ha sido posible implementar en este proyecto:

- Un foro sería el complemento perfecto para esta aplicación. Participando en el foro, los usuarios podrían conseguir tokens por ayudar a sus compañeros y comentar las iniciativas futuras o activas.
- Reconocimientos: los empleados podrían pedir el reconocimiento de títulos u otras acciones externas para que a cambio se les otorgasen tokens
- Mensajería: un servicio de mensajería entre los empleados sería ideal para que estos pudiesen hablar sobre el intercambio de tokens, para organizar mejor las iniciativas y para tener un contacto más sencillo con los administradores del sistema. Esta funcionalidad no ha sido implementada, entre otras cosas, porque una aplicación distribuida de mensajería instantánea no es posible. Las transacciones tardan de media 3 minutos en ser validadas por la red de Ethereum, por lo que ese es el tiempo que habría que esperar para recibir los mensajes

### 6.2 Propuestas de mejora

Uno de los problemas de esta implementación, es el coste de operación (ver siguiente sección). Centralizar algunas partes de la aplicación como las notificaciones o el almacenamiento de las iniciativas y productos de la tienda acabaría siendo más barato y aportaría una mejor experiencia de usuario. Los smart contracts tienen muchas ventajas, pero en este caso concreto lo más óptimo hubiese sido un modelo híbrido.

Otro de los problemas es que el usuario debe de tener ethereum en su wallet, que gastará en comisiones al utilizar Proa. Lo ideal sería que la empresa pagase por estas

transacciones, pero para ello también es necesario un backend. Hasta entonces habría que utilizar el allowance de Ethereum para dar saldo a los usuarios o directamente hacerles una transferencia de Ethereum para que lo gasten en comisiones.

## 6.3 Conclusión

Este ha sido con diferencia el proyecto más largo y complicado al que me he enfrentado hasta la fecha y, aunque a veces ha sido desesperante, estoy muy contento con el resultado final. Han sido muchas horas de trabajo y, sobre todo, muchas horas de aprendizaje. Elegí este tema porque era la excusa para aprender tecnologías nuevas y vaya si ha sido excusa... Estoy seguro de que más de la mitad del tiempo que le he dedicado al proyecto ha sido formación, puede que incluso haya habido demasiada. Los primeros dos meses era muy complicado avanzar ya que no sabía ni donde me metía, sin embargo, cuanto más aprendía más rápido avanzaba; problemas que los primeros días tardaba días en resolver al final del proyecto los resolvía en minutos.

Este proyecto vale para saber dónde está el límite del aprendizaje autodidacta, y he llegado a la conclusión de que no tiene límite. Internet es un mundo inmenso con respuestas para todo, solo hace falta saber buscar y dedicarle tiempo.

También he aprendido que si ya la planificación de un proyecto es complicada, hacerlo sobre un proyecto del que no sabes ni cómo empezar es tarea imposible. Ahora, tras 4 meses trabajando con estas tecnologías es cuando quizá podría anticipar lo que voy a tardar en hacer algo, intentar estimar tiempos y esfuerzo cuando uno no ha trabajado nunca con algo es dar palos de ciego.

Otra cosa que me ha enseñado este proyecto, es que hay que ser flexible y no tener miedo a volver para atrás. Cuando ya llevaba tiempo con el proyecto iniciado, tuve un problema irreparable. Había comenzado a desarrollar el frontend en Angular y este era incompatible con IPFS... Me costó días de pelearme con archivos de configuración darme cuenta de que había que eliminar uno de los dos para que esto siguiese adelante. Con mucho dolor tiré todo lo que había avanzado con Angular y me puse a aprender React.

Por último y, no menos importante, este proyecto me ha recordado todo lo que me gusta de la informática y que, como es algo que me apasiona, soy capaz de dedicarle todo el tiempo del mundo sin ningún problema. Estar todo el día intentando solucionar un problema y no levantarme hasta que lo consigo y, cuando finalmente lo consigo, que la alegría me invada por completo. Por esto, ha sido un proyecto perfecto para cerrar esta etapa, un proyecto duro, con gran aprendizaje y que me ha recordado por qué estudié esta carrera.

## Part III

# Manuales de la Aplicación



# Chapter 7

## Manual de Instalación

Actualmente, la aplicación solo se puede ejecutar en un entorno de desarrollo, por lo que necesitaremos las siguientes aplicaciones:

- SO: la aplicación en principio es compatible con cualquier SO, aunque solo se ha probado su correcto funcionamiento en Windows10
- NodeJS: versión 14.16.1 o posterior
- npm: versión 6.14.12
- Truffle: 5.3.11
- Un editor de texto
- Navegador (preferiblemente chrome)
- Extensión de Metamask instalada en el navegador

Una vez tengamos estas herramientas instaladas, lo único que tenemos que hacer es situarnos con windows powershell en el directorio principal del proyecto (/tfginterfaz). A partir de aquí ejecutaremos en orden los siguientes comandos:

1. `cd ./src/Blockchain`
2. `truffle develop`
3. `compile`
4. `deploy --reset`

Cuando hayamos ejecutado el comando 2, nos saldrán las cuentas que podremos usar en metamask y sus claves privadas. Es importante no perder esto porque son las cuentas que hay en la red. Concretamente, la primera cuenta que aparezca será la de la empresa, que es la única que podrá acceder de primeras.

Aunque hayamos hecho los pasos anteriores correctamente, la aplicación aún no funcionará. Necesitamos editar el fichero `/tfginterfaz/src/App.jsx`. En la línea 59, está la declaración del objeto "direccionesContratos", con cuatro campos (user, token, iniciativas y tienda). En cada uno de esos campos hay que introducir la dirección de los contratos que acabamos de desplegar. Para ello, nos fijamos en la terminal powershell. Cuando hemos ejecutado el tercer comando de la lista anterior, han ido apareciendo datos para cada contrato, como los de la figura 7.1, tenemos que copiar el contract address (señalado en rojo en la figura 7.1) y objeto "direccionesContratos". En el ejemplo de la figura 7.1 nos quedaría algo como la figura 7.2

Ahora que ya tenemos toda la parte de la blockchain preparada pasamos a iniciar el servidor de NodeJS, que ejecutará la aplicación de React. Abrimos otra ventana de powershell en el directorio `/tfginterfaz` y ejecutamos el siguiente comando:

```
npm start
```

Ahora, si todo ha ido bien, se nos abrirá el navegador automáticamente en `http://localhost:3000/` y podremos comenzar a utilizar la aplicación con las cuentas obtenidas del comando 2.



```
2_deploy_contracts.js
-----
Deploying 'User'
-----
> transaction hash: 0x305887a3452e63fbeda741490a0eafb4ffbc46ed263b1240bf03aaba14d33c9
> Blocks: 0 Seconds: 0
> contract address: 0xaCC84FDAf5893D154bfAE0De4b8Dddd2415De537
> block number: 826
> block timestamp: 1632156939
> account: 0xc20c319e7a2813d80dbc92BC35C9E30f006A4564
> balance: 92.66979552
> gas used: 2783685 (0x2a79c5)
> gas price: 50 gwei
> value sent: 0 ETH
> total cost: 0.13918425 ETH

Deploying 'Token'
-----
> transaction hash: 0x0e8fb5b0c02114e546ab71b4925369376dd348b4b98c9f4efdb1cd3fca4d11cf
> Blocks: 0 Seconds: 0
> contract address: 0x12C620992aaF378Ab14d230F7AD67C5cf02a6511
> block number: 827
> block timestamp: 1632156943
> account: 0xc20c319e7a2813d80dbc92BC35C9E30f006A4564
> balance: 92.55175197
> gas used: 2360871 (0x240627)
> gas price: 50 gwei
> value sent: 0 ETH
> total cost: 0.11804355 ETH

Deploying 'Iniciativas'
-----
Deploying 'Tienda'
-----
> transaction hash: 0x0b35ec02d9ebd969ae9c0190840adac07908e12d693b7089c6053a0e3347a003
> Blocks: 0 Seconds: 0 > transaction hash: 0x5131d29899613fb1093804b9143bf7ee7
> Blocks: 0 Seconds: 0
> contract address: 0xDacB04eAd2aFc1D745563CE2c41CA98873b557E7
> block number: 828
> block timestamp: 1632156946
> account: 0xc20c319e7a2813d80dbc92BC35C9E30f006A4564
> balance: 92.19504207
> gas used: 4874419 (0x4a60b3)
> gas price: 50 gwei
> value sent: 0 ETH
> total cost: 0.24372095 ETH

> Blocks: 0 Seconds: 0 > Blocks: 0 Seconds: 0
> contract address: 0xFea7Cf80bd8A84E06ee5154D23A277ba4dA8a004
> block number: 829
> block timestamp: 1632156946
> account: 0xc20c319e7a2813d80dbc92BC35C9E30f006A4564
> balance: 92.19504207
> gas used: 2259779 (0x227b43)
> gas price: 50 gwei
> value sent: 0 ETH
> total cost: 0.11298895 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost: 0.6139377 ETH
```

Figure 7.1: Despliegue Contratos

```
57  --//Contratos
58
59  ✓ --const direccionesContratos = {
60    --  user: "0xaCC84FDaf589D154bfAE0De4b8Dddd2415De537",
61    --  token: "0x12C620992aaF378Ab14d230F7AD67C5cf02a6511",
62    --  iniciativas: "0xDaCB04eAd2aFc1D745563CE2Cc41CA98873b557E7",
63    --  tienda: "0xFea7cf80bd8A84E06ee5154D23A277ba4adA8a004",
64  }
```

Figure 7.2: /tfginterfaz/src/App.jsx:59

# Chapter 8

## Manual de Usuario

### 8.1 Manual de Usuario

La primera vez que el se inicie la aplicación, se nos pedirá iniciar sesión en metamask, con un popup como el de la figura 8.1. Metemos ahí nuestras credenciales de acceso y automáticamente se nos llevará a la pantalla de iniciativas (4.7).

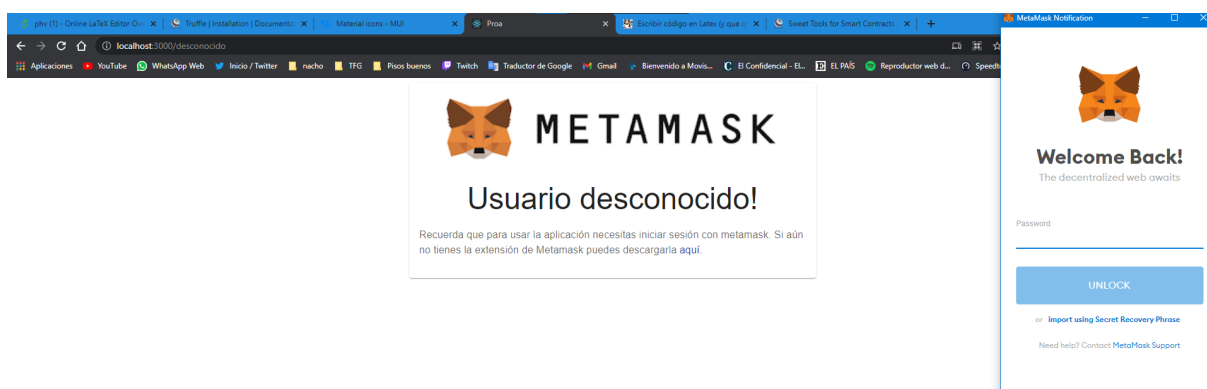


Figure 8.1: Página usuario desconocido

Arriba a la derecha tendremos el icono de notificaciones, si tenemos alguna notificación, se marcará en rojo y con un número. Clickando podremos ver todas nuestras notificaciones pendientes y, si queremos, eliminarlas. (4.10)

A la izquierda, está la información sobre nuestro usuario y la navegación hacia otras páginas.

### 8.1.1 Página iniciativas

En esta página, lo primero que vemos es una tabla con todas las iniciativas disponibles. En la parte superior de esta tabla, encontraremos los filtros, que nos permitirán reducir el número de iniciativas que se nos muestran en función de lo que necesitemos. Hay dos filtros; el primero filtra por estado de la iniciativa y el segundo permite separar entre aquellas que ya han ocurrido y las que todavía no han tenido lugar.

Si hacemos click en una iniciativa, se abrirá un popup con los detalles de la misma 4.12 y con las acciones que podemos realizar.

Si en lugar de hacer click sobre una iniciativa lo hacemos sobre el botón circular de abajo a la derecha, se nos abrirá un formulario con el que podremos proponer una iniciativa, tal y como se muestra en la figura 4.11. Una vez propuesta la iniciativa, tendremos que esperar a que un administrador la tramite, pero no te preocupes, serás notificado cuando esto ocurra.

### 8.1.2 Tienda

En esta página se muestran todos los productos disponibles para comprar, si alguno te llama la atención y tienes tokens suficientes pulsa en comprar y se enviará una petición de compra a los administradores. Cuando estos rechacen o aprueben la compra se te notificará. (figura 4.14)

### 8.1.3 Mis Compras

En esta página se muestran todas tus compras y su estado: tramitando, aprobada o rechazada (figura 4.18)

### 8.1.4 Transacciones

En esta página se muestran todas tus transacciones. En la parte superior de la tabla puedes ver que podrás filtrar o buscar la transacción que te interese.

Además, si pulsas el botón inferior derecho, se abrirá un popup en el que podrás realizar nuevas transacciones 4.21

## 8.2 Manual de Administración

El rol de administrador puede acceder a todas las funciones descritas en el manual de usuario, pero además tiene otras funciones extra.

### **8.2.1 Tienda**

En esta página, el administrador tiene 2 botones en la parte superior de cada producto, uno para borrarlo y otro para editarlo. Además se te mostrarán los productos fuera de stock.

Si clickas en el botón inferior derecho, te saldrá un formulario 4.16 a través del cual podrás añadir un nuevo producto a la tienda

### **8.2.2 Transacciones Empresa**

Funciona igual que la página de transacciones del usuario solo que lo que se maneja es la cuenta de la empresa

### **8.2.3 Administrar Empleados**

En esta página (figura 4.22) se muestran todos los empleados del sistema, de los cuales se pueden ver los datos detallados haciendo click en ellos (Figura 4.23).

Además, si se pulsa en el botón inferior derecho, se puede añadir un nuevo empleado al sistema.

### **8.2.4 Tramitar Compras**

En esta página (figura 4.19) se muestran todas las compras por tramitar del usuario, su deber es aprobarlas o rechazarlas. Aprobará una compra cuando consiga el producto y se lo haya dado al comprador, se rechazará cuando esto no haya sido posible.



# Bibliography

- [1] URL: <https://trends.google.es/trends/explore?date=all&geo=ES&q=blockchain,%2Fm%2F05p0rrx>.
- [2] *BBVA Campus Wallet promueve que el empleado asuma de forma proactiva su desarrollo.* Gref. URL: [https://www.gref.org/nuevo/articulos/art\\_180718.pdf](https://www.gref.org/nuevo/articulos/art_180718.pdf).
- [3] *Beneficios de las dinámicas de trabajo en equipo.* OBS Business School. URL: <https://www.obsbusiness.school/blog/como-benefician-las-dinamicas-de-trabajo-en-equipo-tu-empresa>.
- [4] *Empirical Evaluation of Blockchain Smart Contracts.* Islamic University of Gaza. Febrero 2020. URL: [https://www.researchgate.net/publication/339248221\\_Empirical\\_Evaluation\\_of\\_Blockchain\\_Smart\\_Contracts](https://www.researchgate.net/publication/339248221_Empirical_Evaluation_of_Blockchain_Smart_Contracts).
- [5] *Introduction to Dapps.* Ethereum.org. URL: <https://ethereum.org/en/developers/docs/dapps/>.
- [6] *Introduction to Smart Contracts.* Ethereum.org. URL: <https://ethereum.org/en/developers/docs/smart-contracts/>.
- [7] *Karma.* Telefónica. URL: <https://blockchain.telefonica.com/casos-de-exito/karma/>.
- [8] *Los mejores Hostings Web: el mejor, el barato y el gratuito.* Aula CM. URL: <https://aulacm.com/mejores-hosting-web-alojamiento/>.
- [9] *Material Design.* Wikipedia. URL: [https://en.wikipedia.org/wiki/Material\\_Design](https://en.wikipedia.org/wiki/Material_Design).
- [10] *Meeds.* URL: <https://www.meeds.io/>.
- [11] *Pricing.* Infura.io. URL: <https://infura.io/pricing>.
- [12] *¿Cuánto gana un Junior Software Developer?* Glassdoor. URL: [https://www.glassdoor.es/Sueldos/junior-software-developer-sueldo-SRCH\\_K00,25.htm](https://www.glassdoor.es/Sueldos/junior-software-developer-sueldo-SRCH_K00,25.htm).
- [13] *¿Las empresas necesitan competencia o colaboración en sus equipos?* Triangle Solutions. URL: <https://www.trianglerrrh.es/las-empresas-necesitan-competencia-o-colaboracion-en-sus-equipos/>.

