



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# **Aplicación web para operar con contratos digitales a través de blockchain**

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Francisco Javier Gozalvo Cervera

**Tutor:** Javier Esparza Peidró

2022-2023



# Agradecimientos

---

A mi familia y pareja, quienes me han brindado todo su apoyo, cariño y ánimos durante todo el periodo académico.

Agradecer a mi tutor Javier Esparza Peidró, por integrarse y ayudarme en el nuevo mundo de la programación a través de sus clases.

Tanto como a Andrea Carozzo, mi tutor de prácticas, por brindarme la oportunidad de profesionalizarme en el sector de la programación.

## Resumen

---

El proyecto aborda la creación de una plataforma web que permita a sus usuarios crear, enviar, modificar y gestionar sus contratos de una forma simple y segura. Además la aplicación permitirá tecnologías propias de la Web 3.0, tales como conexiones a la blockchain y transacciones a la misma, otorgando un punto extra de descentralización, inmutabilidad y transparencia en cuanto a los contratos entre los usuarios.

## Palabras clave

---

Contratos, Aplicación Web, Blockchain, Web 3.0, Smart Contracts

## Resum

---

El projecte aborda la creació d'una plataforma web que permeti als seus usuaris crear, enviar, modificar i gestionar els seus contractes d'una forma simple i segura. A més l'aplicació permetrà tecnologies pròpies de la Web 3.0, com ara connexions a la \*blockchain i transaccions a aquesta, atorgant un punt extra de descentralització, inmutabilitat i transparència quant als contractes entre els usuaris.

## Paraules clau

---

Contractes, Aplicació Web, Blockchain, Web 3.0, Smart Contracts

## Abstract

---

The project addresses the creation of a web platform that allows its users to create, send, modify and manage their contracts in a simple and secure way. In addition, the application will allow technologies typical of Web 3.0, such as connections to the blockchain and transactions to it, granting an extra point of decentralization, immutability and transparency in terms of contracts between users.

## Keywords

---

Contracts, Web Application, Blockchain, Web 3.0, Smart Contracts



# Tabla de contenidos

---

1.	Introducción.....	7
1.1.	Motivaciones.....	8
1.2.	Estado del arte.....	8
1.3.	Objetivos.....	9
1.4.	Metodologías.....	10
1.4.1.	KanBAN.....	10
1.4.2.	Diagrama de Gantt.....	11
1.5.	Alcance.....	12
1.6.	Impacto esperado.....	12
1.7.	Estructura de la obra.....	12
2.	Definición del problema.....	13
2.1.	Análisis.....	13
2.1.1.	Análisis de requisitos funcionales.....	13
2.1.2.	Análisis de requisitos no funcionales.....	15
2.1.3.	Análisis de requisitos informativos.....	16
2.2.	Casos de uso.....	17

# 1. Introducción

---

La gestión de datos se ha vuelto una necesidad del día a día. Todo el mundo necesita tener un control organizado de sus emails, una lista de contactos con los que interactuar y hablar diariamente, o incluso una pila de tareas del día, semana, e incluso mes.

Esta gestión se vuelve más importante cuando los datos, archivos o contactos sobre los que necesitamos tener un firme control, podrían perjudicarnos legalmente al perderlos, olvidarnos de donde los dejamos por última vez, o incluso si los desechemos o borramos de nuestra computadora.

Hablamos de los contratos. Si en algún momento de tu vida has llegado a un acuerdo con otra persona mediante un contrato, lo más normal, después de haber firmado las dos o demás partes, es dejarlo estático en algún rincón de nuestro disco duro o perderlo por algún cajón o archivador de nuestra casa. Cuando realmente, este tipo de archivos deberían de ser fácilmente accesibles, inmutables y persistentes. (mas info sobre antecedentes??)

De la misma forma que utilizamos gestores de correo como Gmail, Outlook, Yahoo, las personas que interactúan diariamente con los contratos deberían tener control total sobre ellos, una plataforma donde subir, gestionar, enviar, y mantener un registro de las acciones con sus contratos, con la ayuda de la blockchain.

Por una parte, con la aplicación web tendríamos un buen control y manejo de nuestros contratos, podrías subirlos y guardarlos, modificarlos, cambiar su estado y eliminarlos. Pero sumando la blockchain a la ecuación, podríamos mantener un control de estado de cada contrato, guardando de forma inmutable, persistente, incorruptible y segura, todos los cambios relacionados a un contrato, como por ejemplo, quién lo ha creado y cuando, que partes se ven involucradas en el contrato, si alguien pidió o realizó una modificación antes de su aceptación o incluso registrar en la blockchain si alguien borró un contrato que tenía abierto con otra persona.

Las acciones a grabar dentro de la blockchain pueden ser registradas gracias a un contrato inteligente. Un contrato inteligente no es más que un trozo de código almacenado en la blockchain que se ejecuta cuando se cumplen determinadas condiciones. De esta forma podemos automatizar que se lea la información de un contrato en el momento que este sea creado, guardando, por ejemplo, el título y el creador del contrato y realizando una transacción a la blockchain con la información de este.

Por todo ello, se ha llegado a la conclusión de que crear un proyecto como este va a ser todo un reto ya que, el alumno disponía inicialmente del conocimiento necesario en cuanto a desarrollo web para poder abordar el inicio del proyecto, pero, por otro lado, se carecía de información técnica y práctica sobre la implantación de la arquitectura blockchain al proyecto, aprender el lenguaje y herramientas necesarias para poder desplegar esta nueva arquitectura emergente e integrarla con la aplicación web.

## 1.1. Motivaciones

---

La motivación real de este proyecto viene por mi interés creciente sobre la tecnología blockchain desde hace unos años atrás. Cuando la descubrí gracias a estudiar y aprender sobre Bitcoin, quedé fascinado ante este tipo de arquitectura. Después conocí Ethereum, la segunda blockchain más importante de nuestros tiempos. Esta nueva red tiene la capacidad de implementar Smart Contracts o Contratos Inteligentes en la propia blockchain.

Esto nos permite automatizar transacciones a la blockchain, teniendo en cuenta si se realizan o no ciertas acciones en la parte del cliente, el Smart Contract lo detectará y enviará cierta información a la blockchain.

En este punto, es donde surge la idea de implementar blockchain y Smart Contracts a una aplicación web donde puedas gestionar tus contratos personales.

No obstante, la solución por la que se ha optado es la de realizar el proyecto desde el comienzo, sin utilizar código, plantillas o modelos previamente disponibles. Este proyecto ha sido planteado para ser realizado únicamente con los conocimientos previos obtenidos a través de las asignaturas de la titulación. Asignaturas como Desarrollo Web, Tecnología de Bases de Datos, Ingeniería del Software, han sido claves para obtener el conocimiento necesario para poner en marcha el motor de creación del trabajo final de grado, esperando así obtener un resultado con una calidad deseada.

La idea inicial comienza con una propuesta del alumno. La propuesta inicial es modificada y mejorada entre alumno y tutor, y una vez encaja dentro de los intereses de ambas partes, el trabajo entra en marcha con una motivación elevada por parte del alumno.

## 1.2. Estado del arte

---

Actualmente, la tecnología blockchain se encuentra en pañales, se trata de una tecnología emergente y es normal que las grandes empresas tarden en analizar y estudiar si es conveniente incorporarla en sus proyectos o servicios.

Bien es cierto que, dentro del mundo blockchain, muchas empresas nacen ya preconcebidas como blockchain-nativas, es decir, ya incorporan o tienen pensado incorporar esta tecnología desde un inicio. Al existir, también, múltiples cadenas de bloques en el ecosistema muchas empresas se especializan alrededor de diferentes blockchains, las más importantes, Bitcoin, Ethereum, Cardano, Binance Smart Chain, Polkadot... Cada una contiene su propia blockchain y su propio conjunto de empresas o start-ups que construyen sus servicios con tales blockchains como base.

En cuanto a grandes empresas que estén incorporando blockchain a sus servicios, nos podemos encontrar con que distintos sectores, como el automovilístico, logístico y tecnológico estarían altamente interesados en incorporar esta tecnología.



Por la parte automovilística nos encontraríamos a la FORD, y TOYOTA, que planean implementar esta tecnología para mejorar los sistemas de movilidad así como de conducción autónoma.

En el caso de las logística, tendríamos a ALIBABA y FedEx, donde buscarían rastrear productos de lujo en sus plataformas de comercio electrónico y buscar una mejor alternativa para la resolución de disputas entre clientes.

Por último entre las empresas tecnológicas, tenemos a gigantes como Apple, Facebook, Samsung o Google, entre los cuales, sus principales motivos para integrar la cadena de bloques son, mejorar la seguridad de datos en la nube, mejorar la privacidad de los datos de sus usuarios e incluso rumores de que Apple podría estar planeando patentar su propia blockchain para manejar datos de marca temporal.

## 1.3. Objetivos

---

El objetivo principal del proyecto es crear una plataforma atractiva y funcional para aquellas personas que traten con contratos en su día a día y requieran de una plataforma donde guardar y almacenar de forma segura la información sobre sus contratos.

Para la realización del proyecto se han planteado una serie de objetivos, dadas las necesidades encontradas para el correcto uso de la solución propuesta.

En primer lugar, elegir en qué lenguaje, framework y tecnologías vamos a depender en la creación del programa, así como elegir qué metodología de trabajo realizaremos para tener un seguimiento del trabajo.

Seguidamente, determinar cuántos modelos de datos van a existir en la aplicación, que características o atributos va a tener cada modelo y cómo van a estar relacionados entre ellos. Crear las correspondientes vistas y controladores para cada una de ellas.

Una vez la aplicación sea funcional, se puedan dar de alta y baja usuarios, y tener las funciones CRUD definidas y funcionales sobre los contratos, añadirle una buena experiencia de usuario utilizando Bootstrap y CSS.

Deberá tener una interfaz sencilla y entendible para que no aparezca ningún tipo de dificultad de uso ni confusiones en las opciones.

Finalmente, integrar blockchain en el proyecto creando un sistema agradable e intuitivo que acompañe de la mano al usuario en todas las acciones al interactuar con esta tecnología y que no dificulte el uso básico de la aplicación.

## 1.4. Metodologías

### 1.4.1. KanBan

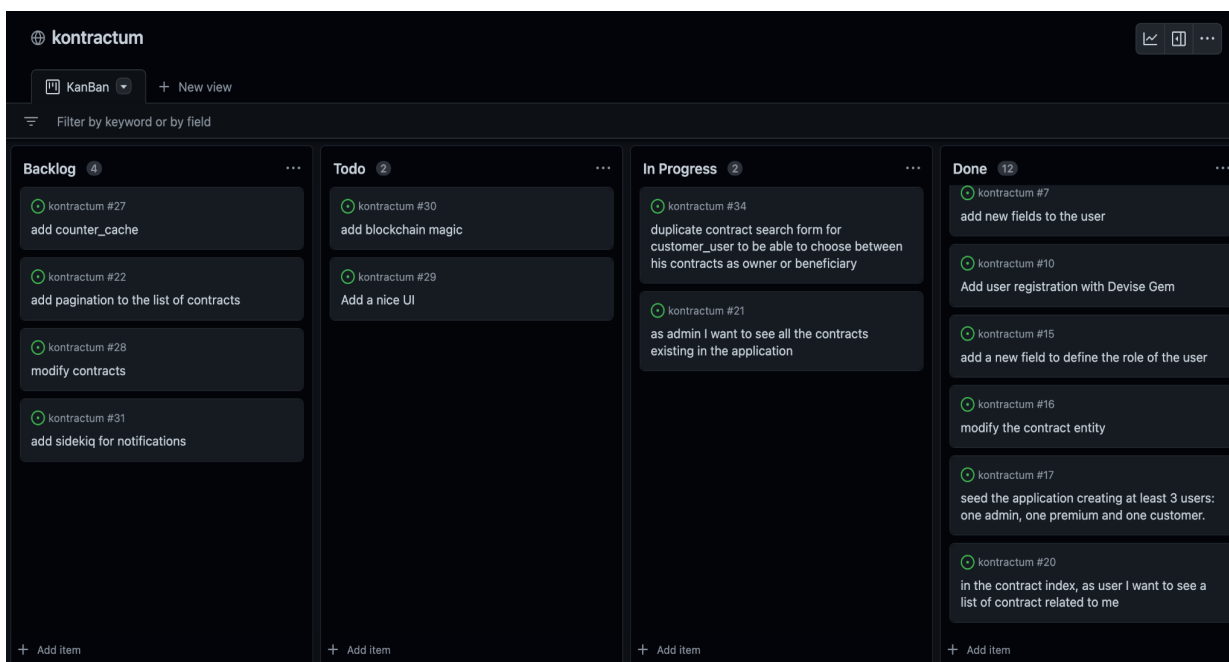
El desarrollo del proyecto ha sido marcado siguiendo la metodología KanBan.

Por contextualizar, KanBan fué desarrollado por Taiichi Ohno, ingeniero japonés de Toyota a finales de la época de los 40. De hecho, “Kanban” es una combinación de dos palabras japonesas: 看 (Kàn), que significa “signo” o “señal visual”, y 板 (B ã n), que significa “tablero”.

KanBan utiliza tableros, llamados tableros KanBan, que son una forma visual de gestionar tu proyecto en diferentes etapas o estados, son muy populares entre equipos de ingeniería de productos y desarrollo software.

Con esta metodología, encontramos un equilibrio entre las tareas que se deben realizar y la disponibilidad de cada miembro del equipo. En el caso de este proyecto de final de carrera, el equipo está formado sólo por una persona, el alumno, por lo que esta metodología se centrará en la mejora continua, donde las tareas a realizar se “extraen” de una tabla de acciones pendientes en un flujo de trabajo constante.

La metodología se implementa a través de los anteriormente mencionados, tableros KanBan. Cada columna representa una etapa de trabajo. De forma simple, podríamos dividir el trabajo en los siguientes tableros: Trabajo pendiente, progreso y terminado. Las tareas individuales se posicionarian en su tablero correspondiente y, dependiendo de su orden en la lista, estarían ordenadas dependiendo de cuán importante sea cada tarea.



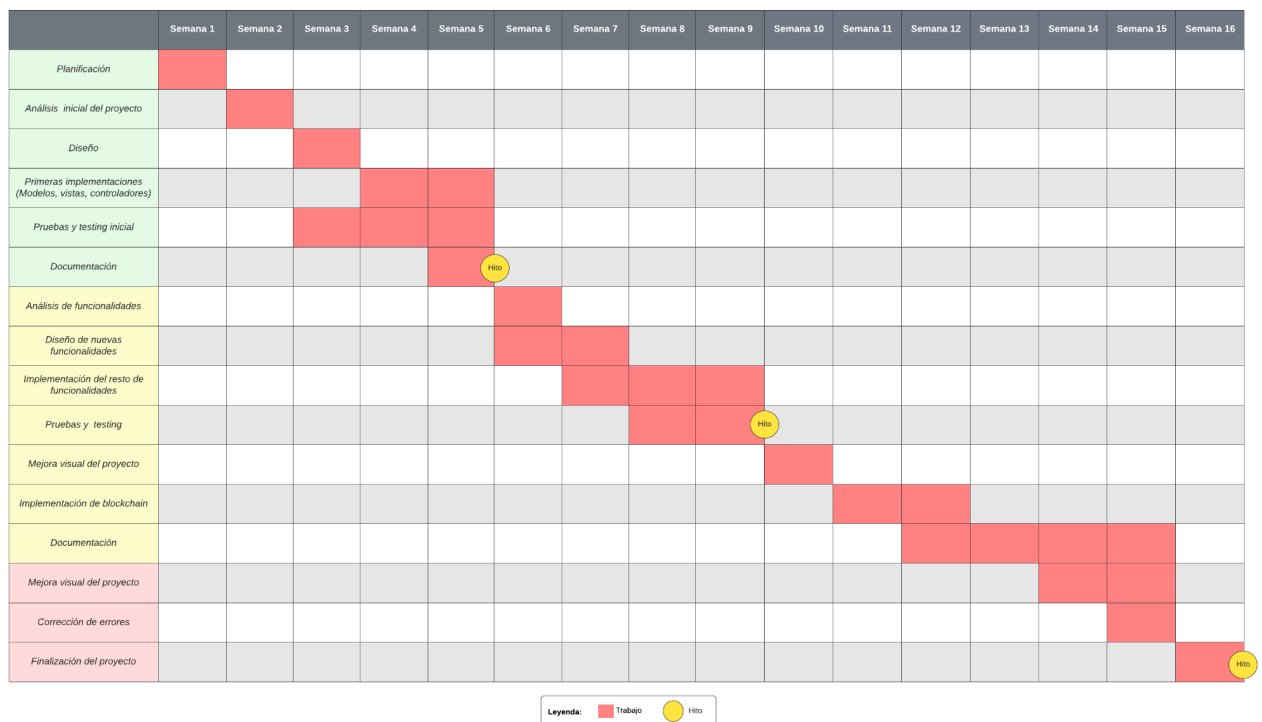
## 1.4.2. Diagrama de Gantt

A continuación, se muestra el diagrama de Gantt en el cual hemos de apuntar dos cosas, en primera instancia los fines de semana no están dentro de nuestro calendario.

Lo segundo es que no se han incluido los recursos puesto que el proyecto lo ha llevado a cabo una sola persona y por lo tanto en cada uno de los puntos los recursos para esa persona alcanzarán el 100% y no habrá recursos sobre asignados.

Por contextualizar, la realización del proyecto se ha realizado en 3 fases, las cuales podríamos agrupar como:

- Fase **Inicial**
- Fase **Funcional**
- Fase **Final**



## 1.5. Alcance

---

Este proyecto va destinado a aquellos usuarios que requieran un cierto nivel en cuanto a gestión de sus contratos, ya sea porque necesitan una mayor centralización de estos para tener un mayor control y seguridad sobre ellos, o porque tienen muchos tipos de contratos de diferentes temáticas y necesitan un espacio donde poder almacenarlos de forma organizada y ordenada.

## 1.6. Impacto esperado

---

Con la realización de este proyecto, se busca añadir valor y funcionalidad, a una parte de la gestión de archivos que nunca se había valorado mejorar.

De hecho, este proyecto va un paso más allá e integra blockchain, permitiendo que los usuarios tengan un acercamiento inicial al operar con esta nueva tecnología de forma totalmente guiada y autónoma.

## 1.7. Estructura de la obra

---

-

# 2. Definición del problema

---

El estado del arte nos lleva de forma natural a plantear un análisis para contestar a esta propuesta. En primer lugar, se debería sistematizar un buen análisis de requisitos:

## 2.1. Análisis

---

## 2.1.1. Análisis de requisitos funcionales

---

Un requisito funcional puede ser definido como una breve descripción de lo que un sistema debería poder hacer. La aplicación debe de ser capaz de realizar estos requisitos para poder cumplir su funcionalidad.

<b>RF-001</b>	<b>Dar de alta a usuario</b>
Autor	Fco Javier Gozalvo Cervera
Dependencias	-
Descripción	La aplicación deberá de poder dar de alta a usuarios con sus datos correctamente validados

*Tabla 1. Requisito funcional: Dar de alta usuario*

<b>RF-002</b>	<b>Recuperar contraseña usuario</b>
Autor	Fco Javier Gozalvo Cervera
Dependencias	RF-001 [Dar de alta a usuario]
Descripción	La aplicación deberá facilitar via email una recuperación de la contraseña

*Tabla 2. Requisito funcional: Recuperar contraseña usuario*

<b>RF-003</b>	<b>Entrar a aplicación</b>
Autor	Fco Javier Gozalvo Cervera
Dependencias	RF-001 [Dar de alta a usuario]
Descripción	La aplicación deberá permitir el acceso a sus funciones a aquellos usuarios registrados

*Tabla 3. Requisito funcional: Entrar a aplicación*

<b>RF-004</b>	<b>Dar de baja a usuario</b>
Autor	Fco Javier Gozalvo Cervera

Dependencias	-
Descripción	Si el usuario lo requiere, la aplicación deberá de borrar todos sus datos

Tabla 4. Requisito funcional: Dar de baja a usuario

<b>RF-005</b>	<b>Crear contratos</b>
Autor	Fco Javier Gozalvo Cervera
Dependencias	-
Descripción	La aplicación permitirá a usuarios, si su rol se lo permite, crear contratos

Tabla 5. Requisito funcional: Crear contratos

<b>RF-006</b>	<b>Editar contrato</b>
Autor	Fco Javier Gozalvo Cervera
Dependencias	RF-005 [Crear contratos]
Descripción	La aplicación permitirá editar los contratos siempre y cuando el usuario sea su creador

Tabla 6. Requisito funcional: Editar contrato

<b>RF-007</b>	<b>Eliminar contratos</b>
Autor	Fco Javier Gozalvo Cervera
Dependencias	RF-005 [Crear contratos]
Descripción	La aplicación permitirá eliminar contratos si el usuario es su creador

Tabla 7. Requisito funcional: Eliminar contratos

<b>RF-008</b>	<b>Listar contratos</b>
---------------	-------------------------

Autor	Fco Javier Gozalvo Cervera
Dependencias	-
Descripción	La aplicación listará todos aquellos contratos donde el usuario sea creador o sea beneficiario.

Tabla 8. Requisito funcional: Listar contratos

<b>RF-009</b>	<b>Editar perfil</b>
Autor	Fco Javier Gozalvo Cervera
Dependencias	-
Descripción	La aplicación permitirá editar la información personal del usuario

Tabla 9. Requisito funcional: Editar perfil

<b>RF-010</b>	<b>Editar contraseña</b>
Autor	Fco Javier Gozalvo Cervera
Dependencias	-
Descripción	La aplicación permitirá editar la contraseña de acceso al usuario

Tabla 10. Requisito funcional: Editar contraseña

## 2.1.2. Análisis de requisitos no funcionales

Un requisito no funcional, en referencia a ingeniería de software, un requisito que detalla y especifica criterios que pueden ser relevantes en el momento de juzgar las operaciones en vez del comportamiento específico.

Los requisitos no funcionales se encuentran listados a continuación:

<b>RNF-001</b>	<b>Navegador Web</b>
Autor	Fco Javier Gozalvo Cervera
Descripción	La aplicación deberá funcionar en los

	navegadores más utilizados del mercado
--	--

Tabla 11. Requisito no funcional: Navegador web

RNF-002	Diseño responsive
Autor	Fco Javier Gozalvo Cervera
Descripción	La aplicación deberá adaptarse a las dimensiones de pantalla en las que se esté visualizando

Tabla 12. Requisito no funcional: Diseño responsive

RNF-003	Aprendizaje
Autor	Fco Javier Gozalvo Cervera
Descripción	La aplicación deberá permitir que un usuario promedio sea capaz de manejar las funcionalidades de la WebApp

Tabla 13. Requisito no funcional: Aprendizaje

### 2.1.3. Análisis de requisitos informativos

Los requisitos de información son una forma más específica de los requisitos funcionales, indican el tipo de información que la aplicación almacena, en este caso, tanto de usuarios como de contratos.

Los requisitos de información obtenidos en este proyecto son:

RI-001	Usuario
Autor	Fco Javier Gozalvo Cervera
Datos específicos	<ul style="list-style-type: none"> <li>- ID del usuario</li> <li>- Nombre</li> <li>- Apellidos</li> <li>- Número de identidad</li> <li>- Tipo de documento identidad</li> <li>- Email</li> <li>- Fecha nacimiento</li> <li>- Dirección</li> <li>- Código Postal</li> </ul>



	<ul style="list-style-type: none"> <li>- Ciudad</li> <li>- Rol en el sistema</li> <li>- Fecha de creación</li> <li>- Fecha de actualización</li> <li>- Contraseña encriptada</li> </ul>
--	---

Tabla 15. Requisito informativo: Usuario

RI-002	Contratos
Autor	Fco Javier Gozalvo Cervera
Datos específicos	<ul style="list-style-type: none"> <li>- Título</li> <li>- Descripción</li> <li>- Tipo</li> <li>- Estado</li> <li>- Fecha creación</li> <li>- Fecha actualización</li> <li>- Propietario (usuario)</li> <li>- Beneficiario (usuario)</li> <li>- Pagabilidad</li> </ul>

Tabla 15. Requisito informativo: Contratos

## 2.2. Casos de uso

En UML, un diagrama de casos de uso, es una forma de expresar un diagrama de comportamiento UML mejorado. Expresa mediante notaciones gráficas, representaciones de casos de uso.

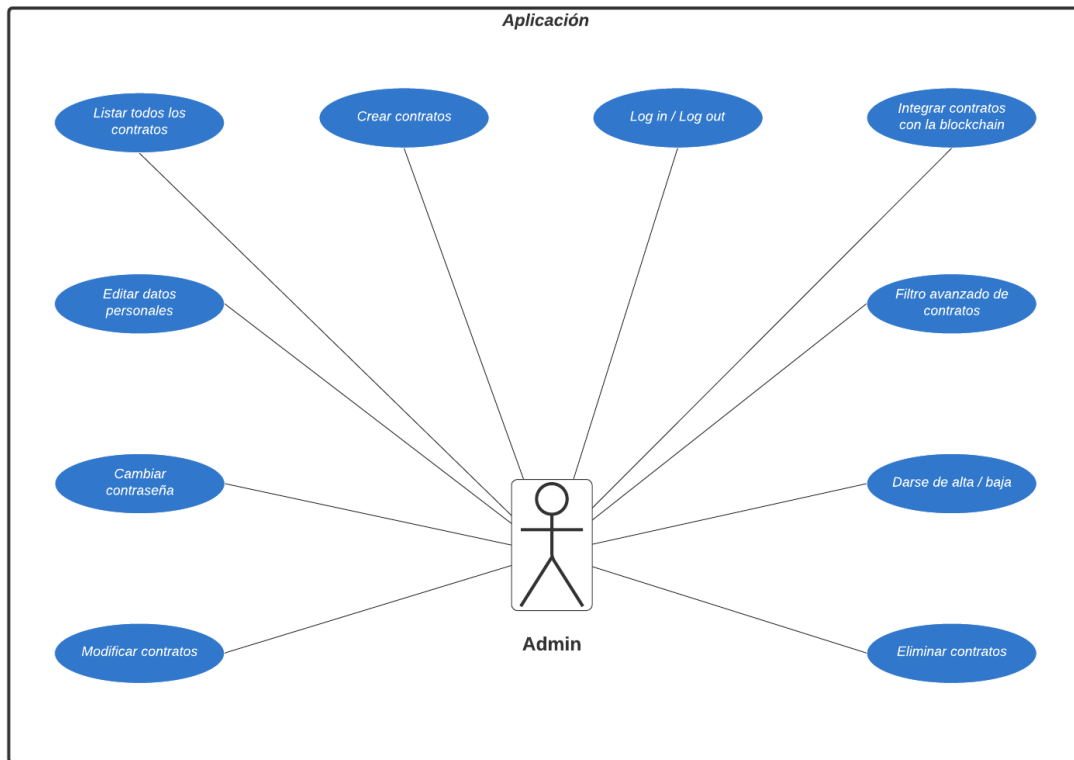
En estas notaciones mostramos las distintas funcionalidades que nos proporciona la aplicación, habiendo analizado antes, requisitos, y actores que interactúan con el sistema.

En este caso, tendremos distintos casos de uso dependiendo del rol asignado al usuario, podrá tener más, o menos acceso a funcionalidades de la aplicación.

Recordar que, los distintos roles, listados de mayor a menor acceso a funciones, serían los siguientes:

- Admin
- Premium Customer

- Customer
- Banned
- Blocked



*Ilustración . Diagrama de casos de uso - Admin*

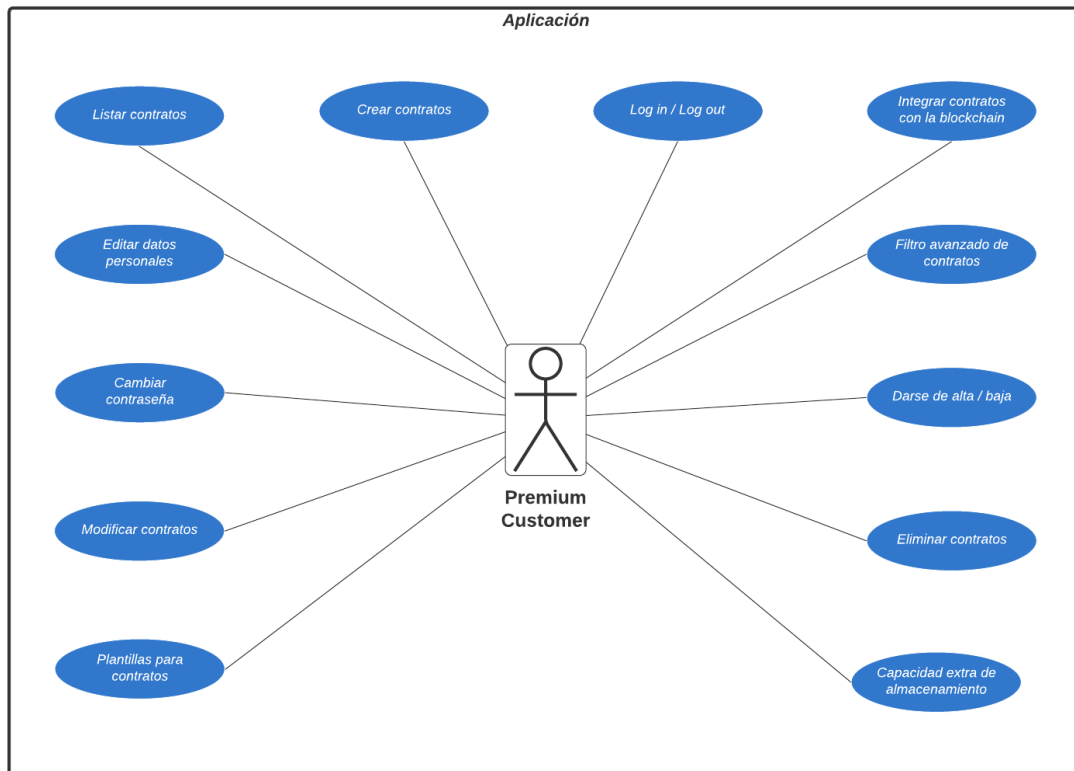


Ilustración . Diagrama de casos de uso - Premium Customer

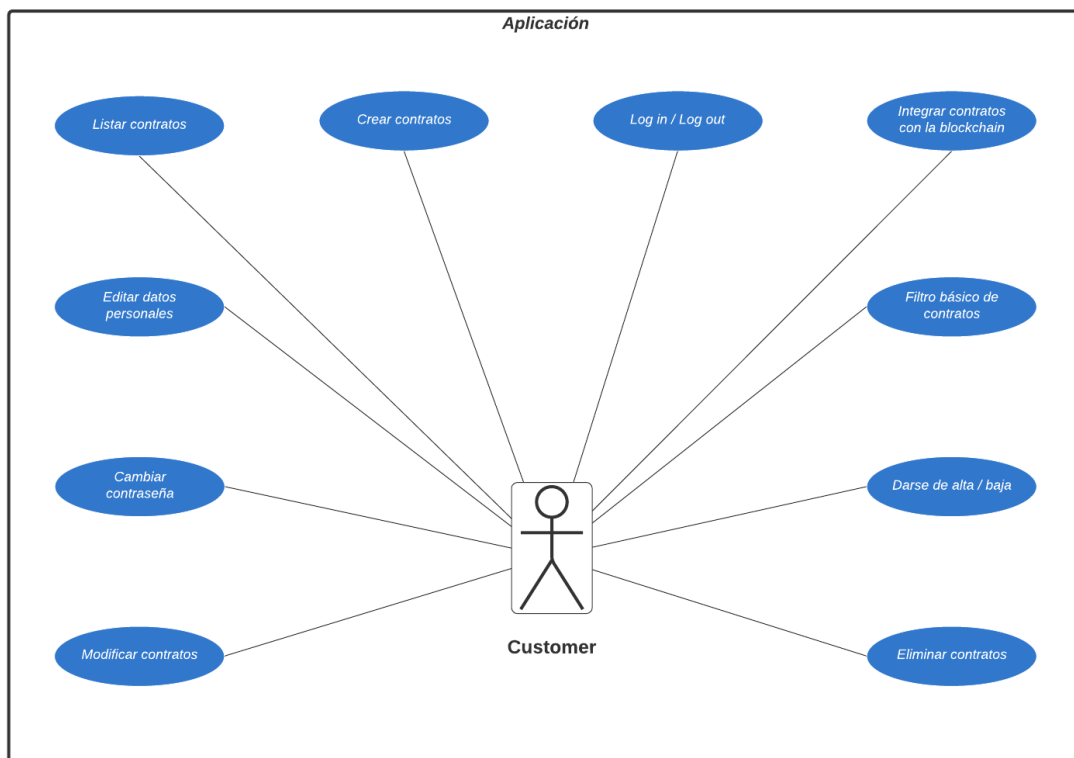
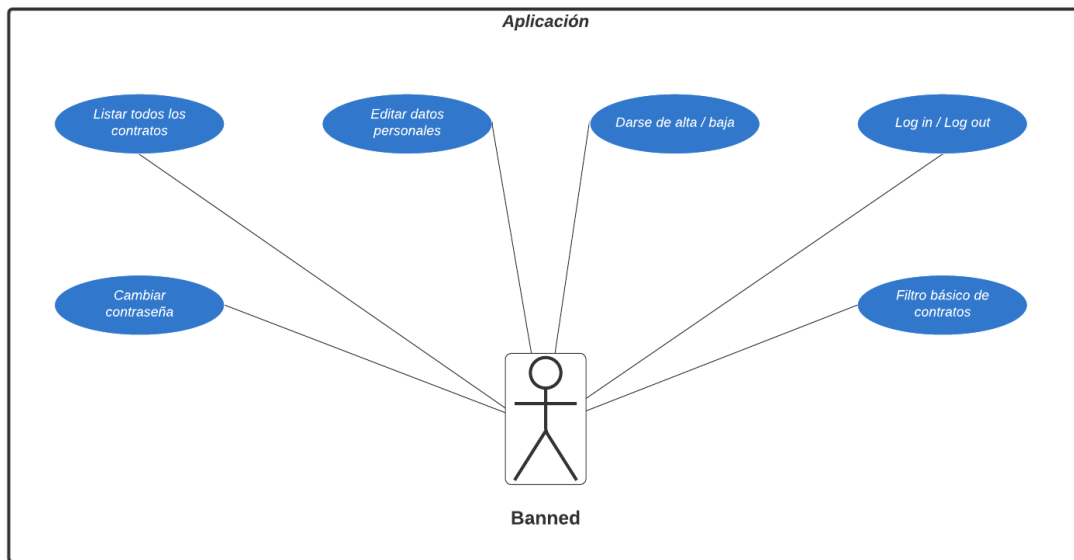
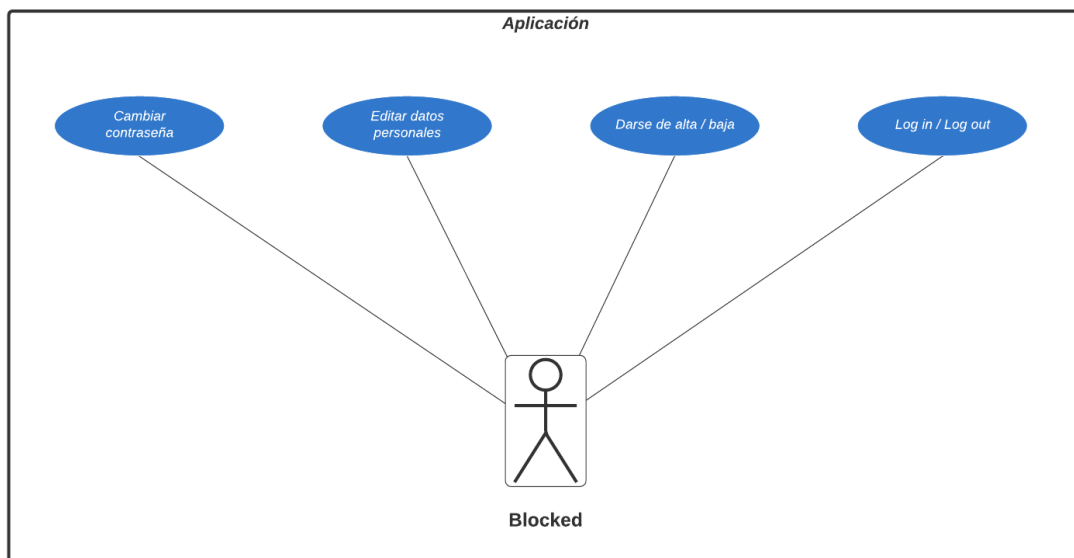


Ilustración . Diagrama de casos de uso - Customer



*Ilustración . Diagrama de casos de uso - Banned*



*Ilustración . Diagrama de casos de uso - Blocked*

A continuación, se pueden ver los casos de uso en detalle, mostrando la funcionalidad de la aplicación.

<b>CU-001</b>	<b>Listar Contratos</b>
Autor	Fco Javier Gozalvo Cervera
Precondición	-
Secuencia normal	<ol style="list-style-type: none"> <li>1. El usuario se encuentra en la página principal</li> <li>2. Usuario hace click en botón Listar Contratos</li> <li>3. Se carga la página con los contratos listados</li> </ol>
Postcondición	El caso se ha ejecutado correctamente y el usuario es capaz de visualizar los contratos listados
Flujos alternativos	<ol style="list-style-type: none"> <li>4. No existen contratos creados, se muestra una página con un mensaje indicando si quiere crear el primer contrato</li> </ol>

Tabla NUMERO . Caso de uso: Listar contratos

<b>CU-002</b>	<b>Crear Contrato</b>
Autor	Fco Javier Gozalvo Cervera
Precondición	El rol del usuario le permite crear contratos
Secuencia normal	<ol style="list-style-type: none"> <li>1. El usuario se encuentra en la página principal</li> <li>2. Hace click en el botón crear contrato</li> <li>3. Se carga la página de creación de contrato con el formulario a rellenar</li> <li>4. El usuario rellena todos los campos correctamente y hace click en guardar contrato</li> <li>5. El usuario es redirigido a los contratos listados para poder ver su contrato creado</li> </ol>
Postcondición	El usuario ha creado correctamente el contrato y puede visualizarlo en la página de listado
Flujos alternativos	<ol style="list-style-type: none"> <li>6. El usuario no ha introducido datos válidos y es informado a través de mensajes de error</li> </ol>

Tabla NUMERO . Caso de uso: Crear contratos

<b>CU-003</b>	<b>Listar todos los contratos</b>
Autor	Fco Javier Gozalvo Cervera
Precondición	El usuario debe tener el siguiente rol: - Admin
Secuencia normal	<ol style="list-style-type: none"> <li>1. El usuario se encuentra en la página principal</li> <li>2. Usuario hace click en botón Listar Contratos</li> <li>3. Se carga la página con todos los contratos listados</li> </ol>
Postcondición	El caso se ha ejecutado correctamente y el usuario es capaz de visualizar los contratos listados
Flujos alternativos	<ol style="list-style-type: none"> <li>4. No existen contratos creados, se muestra una página con un mensaje indicando si quiere crear el primer contrato</li> </ol>

Tabla NUMERO . Caso de uso: Listar todos los contratos

<b>CU-004</b>	<b>Cambiar contraseña</b>
Autor	Fco Javier Gozalvo Cervera
Precondición	-
Secuencia normal	<ol style="list-style-type: none"> <li>1. El usuario hace click en el botón del header “Editar Contraseña”</li> <li>2. Se carga la página de cambio de contraseña con el formulario de cambio</li> <li>3. Rellena el formulario y al guardar es redirigido a la pantalla principal con un mensaje de confirmación</li> </ol>
Postcondición	El caso se ha ejecutado correctamente y el usuario ha editado su contraseña
Flujos alternativos	<ol style="list-style-type: none"> <li>4. El usuario no ha introducido una contraseña válida y se le indica por un mensaje de error</li> </ol>

Tabla NUMERO . Caso de uso: Cambiar contraseña

<b>CU-005</b>	<b>Modificar contratos</b>
Autor	Fco Javier Gozalvo Cervera
Precondición	El contrato debe existir y el usuario ser su creador
Secuencia normal	<ol style="list-style-type: none"> <li>1. El usuario entra en el listado de contratos</li> <li>2. Busca el contrato a modificar y hace click en “Editar”</li> <li>3. Modifica los campos que necesite del contrato</li> <li>4. Al guardarlo, es redirigido de nuevo al listado de contratos</li> </ol>
Postcondición	El caso se ha ejecutado correctamente, el contrato ha sido modificado y el usuario lo verifica en la lista de contratos
Flujos alternativos	<ol style="list-style-type: none"> <li>5. El usuario ha introducido datos no válidos el contrato y la aplicación muestra mensajes de error</li> </ol>

Tabla NUMERO . Caso de uso: Modificar contratos

<b>CU-006</b>	<b>Log in / inicio de sesión</b>
Autor	Fco Javier Gozalvo Cervera
Precondición	Tener una cuenta de usuario existente
Secuencia normal	<ol style="list-style-type: none"> <li>1. El usuario rellena el formulario de inicio de sesión</li> <li>2. La aplicación comprueba en la base de datos que el usuario existe y lo redirige a la pantalla principal</li> </ol>
Postcondición	El caso se ejecuta correctamente y el usuario es capaz de entrar a la aplicación
Flujos alternativos	<ol style="list-style-type: none"> <li>3. El usuario ha introducido incorrectamente sus credenciales y un mensaje de error se lo indica</li> <li>4. El usuario no recuerda la contraseña e inicia el proceso de recuperación de contraseña</li> </ol>

Tabla NUMERO . Caso de uso: Log in / Inicio de sesión

<b>CU-007</b>	<b>Log out / Cerrar sesión</b>
Autor	Fco Javier Gozalvo Cervera
Precondición	Estar con la sesión abierta dentro de la aplicación
Secuencia normal	<ol style="list-style-type: none"> <li>1. El usuario hace click en el botón del header “Log Out”</li> <li>2. La aplicación detecta el cierre de sesión y redirige al usuario a la pantalla de inicio de sesión</li> </ol>
Postcondición	El caso se ejecuta correctamente y el usuario cierra su sesión de la aplicación
Flujos alternativos	-

Tabla NUMERO . Caso de uso: Log out / Cerrar sesión

<b>CU-008</b>	<b>Filtrar contratos</b>
Autor	Fco Javier Gozalvo Cervera
Precondición	Que existan contratos creados para poder filtrarlos
Secuencia normal	<ol style="list-style-type: none"> <li>1. El usuario, estando en la pantalla principal, hace click en el botón “Ver contratos”</li> <li>2. La aplicación carga la lista de contratos, en la parte superior aparecerá un recuadro de búsqueda</li> <li>3. El usuario elige entre los diferentes filtros del buscador y hace click en “Buscar”</li> <li>4. En la lista, aparecerán aquellos contratos que coincidan con los criterios de búsqueda introducidos</li> </ol>
Postcondición	El caso se ejecuta correctamente y el usuario es capaz de ver los contratos filtrados
Flujos alternativos	<ol style="list-style-type: none"> <li>5. Los filtros aplicados no coinciden con ningún contrato y el usuario ve una lista vacía de contratos</li> </ol>

Tabla NUMERO . Caso de uso: Filtrar contratos



<b>CU-009</b>	<b>Darse de alta / Registro</b>
Autor	Fco Javier Gozalvo Cervera
Precondición	-
Secuencia normal	<ol style="list-style-type: none"> <li>1. El usuario no dispone de cuenta para entrar en la aplicación</li> <li>2. Hace click en "Sign up"</li> <li>3. Se carga un formulario de registro</li> <li>4. El usuario rellena el formulario con datos válidos y envía el formulario</li> <li>5. Una vez crea su usuario, es redirigido a su página principal dentro de la aplicación</li> </ol>
Postcondición	El caso se ejecuta correctamente, el usuario crea su perfil en la aplicación y ya puede utilizar sus funciones
Flujos alternativos	<ol style="list-style-type: none"> <li>6. El usuario rellena el formulario de registro con datos no válidos y la aplicación se lo informa</li> </ol>

Tabla NUMERO . Caso de uso: Darse de alta / Registro

<b>CU-010</b>	<b>Darse de baja / Borrar cuenta</b>
Autor	Fco Javier Gozalvo Cervera
Precondición	Tener una cuenta existente
Secuencia normal	<ol style="list-style-type: none"> <li>1. El usuario hace click en el botón "Cambiar contraseña"</li> <li>2. En la parte inferior de la pantalla, se le pregunta al usuario si quiere borrar su cuenta</li> <li>3. El usuario hace click en el botón "Cancelar cuenta"</li> <li>4. La aplicación recibe la petición, borra los datos del usuario y lo redirige a la pantalla de inicio de sesión</li> </ol>
Postcondición	El caso se ejecuta correctamente, borrando la cuenta y los datos del usuario
Flujos alternativos	-

Tabla NUMERO . Caso de uso: Darse de baja / Borrar cuenta

<b>CU-011</b>	<b>Borrar contratos</b>
Autor	Fco Javier Gozalvo Cervera
Precondición	El contrato debe existir
Secuencia normal	<ol style="list-style-type: none"> <li>1. El usuario hace click en “Ver contratos”</li> <li>2. La aplicación carga la lista de contratos</li> <li>3. El usuario escoge el contrato a borrar y hace click en al botón “Eliminar”</li> <li>4. El contrato se elimina de la base de datos del usuario</li> </ol>
Postcondición	El caso se ejecuta correctamente, borrando el contrato que el usuario ha indicado
Flujos alternativos	-

Tabla NUMERO . Caso de uso: Borrar contratos

<b>CU-012</b>	<b>Integrar contratos con la blockchain</b>
Autor	Fco Javier Gozalvo Cervera
Precondición	-
Secuencia normal	<ol style="list-style-type: none"> <li>1. En la pantalla principal, el usuario hace click en el botón “Enable Ethereum”</li> <li>2. Si el usuario tiene instalado en el navegador la extensión de Metamask, saldrá un pop-up de la extensión preguntándole si quiere conectar su billetera al sitio web.</li> <li>3. El usuario escoge una de sus cuentas para conectarse</li> <li>4. Una vez conectado, en la aplicación se podrá leer que cuenta está conectada con Metamask</li> <li>5. El usuario ya podrá operar con contratos en la blockchain</li> </ol>
Postcondición	El caso se ejecuta correctamente y el usuario realiza indirectamente transacciones a la blockchain mientras realiza tareas con sus contratos
Flujos alternativos	<ol style="list-style-type: none"> <li>6. El usuario no tiene descargada la</li> </ol>

	extensión y no puede conectar su billetera a la web
--	---

*Tabla NUMERO . Caso de uso: Integrar contratos con la blockchain*