

BITLEVEL

1.0 - Introducción	1
1.1 - Descripción propuesta/problema.	1
1.2 - Diagrama inicial del conjunto del proyecto	4
2 - Metodología Incremental	4
3.0 - Requisitos / Análisis	5
3.1 - Partes Interesadas.	5
3.2 - Requisitos funcionales.	5
3.3 - Requisitos no funcionales (opcional. pero recomendable).	6
3.4 - Requisitos sistema (opcional, pero recomendable).	6
3.5 - Diagrama de casos de uso.	7
3.6 - Documento detallado de requisitos.	8
Secuencia normal	8
Secuencia Alternativa	9
3.7 - Matriz Requisitos/Tecnologías.	11
3.8 - Modelo Entidad-Relación.	12
4.0 Diseño	13
4.1 Diagrama de paquetes (opcional).	13
4.2 Diagrama de clases.	14
4.3 Diagrama de secuencia	15
4.4 Paso a relacional y normalización	16
Modelo Relacional	16

1.0 - Introducción

1.1 - Descripción propuesta/problema.

[Nombre del proyecto]

BitLevel

[Nombre integrantes]

Kevin Coaquira

Heber M.Rodriguez

Rafa Perez

[Fecha inicio proyecto]

21/03/2022

[Descripción proyecto]

En base a un acuerdo e intereses en común de los integrantes del proyecto, el tema que engloba la propuesta son las criptomonedas y tecnología blockchain, inspirados en sitios web ya existentes de visualización, custodia y gestión de estos activos digitales como [Coingecko.com](https://www.coingecko.com), [Coinmarketcap.com](https://www.coinmarketcap.com), [Coinpaprika.com](https://www.coinpaprika.com), y más alternativas, nace el concepto de **BitLevel** una plataforma web como centro de información confiable sobre activos criptográficos.

Los usuarios en función de su nivel de registro/inscripción se catalogan en **Usuario no registrado**, **Usuario default** y **Usuario premium**.

Todos los usuarios cuentan con una serie de funcionalidades comunes como son: realizar una búsqueda en el listado de criptomonedas aplicando una serie de filtros, consultar su apartado de portfolio donde se listaran los activos que posee el usuario, consultar la sección de noticias relacionadas con el ecosistema blockchain y aplicar un cambio la apariencia del sitio web, tema oscuro y tema claro (default).

El **Usuario no registrado** cuenta, además de las funcionalidades básicas descritas anteriormente, con la posibilidad de probar durante un tiempo limitado y con saldo ficticio una cuenta *demo* con el objetivo de desbloquear más funcionalidades e invitar al usuario a registrarse.

Un **Usuario registrado**, realiza un *Login* y *Logout*, tiene acceso a un objeto de vital importancia en la plataforma, su **wallet** / billetera, permitiendo así desbloquear las funciones de compra-venta de criptomonedas, almacén de las mismas, acceso al historial de

transacciones realizadas y un método de ahorro/bloqueo de activos gestionado por el usuario con recompensa de intereses denominado *staking*, del cual se darán más detalles en posteriores versiones de la documentación.

Relacionado a la gestión de la cuenta e incentivos diarios, un usuario ya registrado puede cambiar su contraseña actual, subir imagen a su avatar, recibir recompensa diaria por *check in* en la plataforma como método de afiliación, acceso al conversor de monedas y con la iniciativa de crear una comunidad ligada a la plataforma, una serie de vistas donde se detalle documentación relacionada al mundo de las criptomonedas cubriendo desde conceptos básicos para los menos familiarizados a términos concretos en *BitLevel Academy* y un apartado de preguntas frecuentes (FAQ).

Un **Usuario Premium**, dispone de una serie de características especiales así como la posibilidad de configurar alertas personalizadas para el aviso de cambio de precio de un activo, descuentos en la compra-venta de activos, participación en sorteos y solicitar una tarjeta de crédito con una serie de ventajas que se documentará posteriormente.

La figura del **Admin** representa una línea de seguridad para los usuarios, dentro de las funciones que puede desempeñar el administrador es recibir alertas de inicio de sesión fraudulentas y el envío de correo phishing a los usuarios, la creación de usuarios para realizar pruebas (testing) y *backup* de seguridad para asegurar la integridad de los datos.

1.2 - Diagrama inicial del conjunto del proyecto

2 - Metodología Incremental

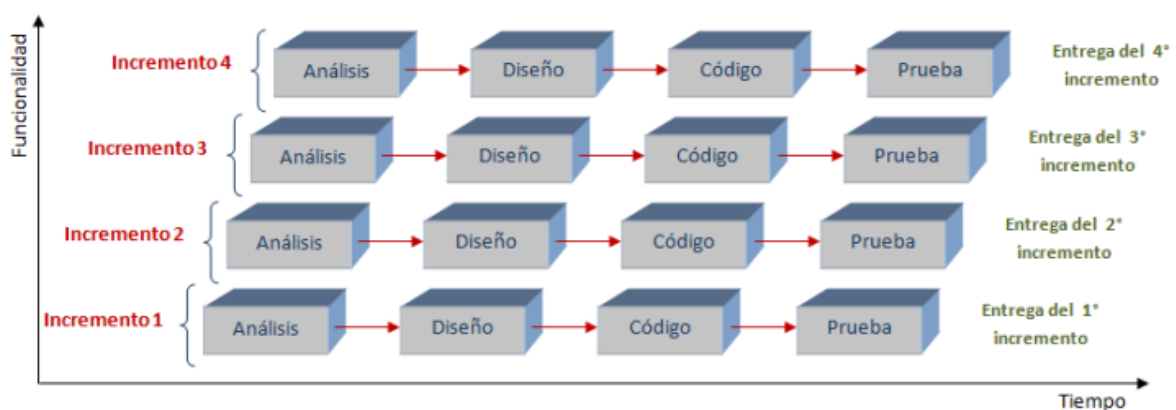


Figura 1: El Modelo Incremental

El modelo incremental combina elementos del modelo en 'Cascada', con la filosofía interactiva del modelo 'Prototipado' que veremos más adelante. Como se muestra en la figura anterior, el modelo incremental aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software. El primer incremento generalmente es un producto esencial denominado núcleo.

Como su nombre lo indica esta metodología consta de ir aumentando las funcionalidades de un programa a medida que se hacen iteraciones o surgen nuevas necesidades del programa. Cada incremento es solo una fracción del producto final del programa e implica una entrega del proyecto cada vez que se hace un incremento.

3.0 - Requisitos / Análisis

3.1 - Partes Interesadas.

USUARIO (NO REGISTRADO)
USUARIO (REGISTRADO)
USUARIO (PREMIUM)
ADMIN

3.2 - Requisitos funcionales.

Requisitos primarios

[Requisitos secundarios](#)

- Usuario no registrado :
 - RF01-Demo
 - RF02-Filtrar Crypto
 - RF03-Cambiar tema web
 - RF04-Registrarse
 - RF05-Visitar Portafolios
 - RF06-Consultar Noticias
- Usuario "default" :
 - RF02-Filtrar Crypto
 - RF03-Cambiar tema web
 - RF05-Visitar Portafolios
 - RF06-Consultar Noticias
 - RF07-Recompensas diarias
 - RF08-Comprar Crypto
 - RF09-Vender Crypto
 - RF10-Resetear contraseñas / qr 2fa
 - RF11-Consultar registro de transacciones

- RF12-Login
- RF13-Logout
- [RF17-Conversiones de Crypto](#)
- [RF18-BitLevel Academy](#)
- [RF19-Preguntas frecuentes](#)
- [RF20-Stacking](#)
- Usuario “premium” :
 - RF02-Filtrar Crypto
 - RF03-Cambiar tema web
 - RF05-Visitar Portafolios
 - RF06-Consultar Noticias
 - RF07-Recompensas diarias
 - RF08-Comprar Crypto
 - RF09-Vender Crypto
 - RF10-Resetear contraseñas / qr 2fa
 - RF11-Consultar registro de transacciones
 - RF12-Login
 - RF13-Logout
 - RF14-Alertas de Precio (A poder ser un aviso cada 24h o X tiempo)
 - RF15-Descuento comisiones
 - RF16-Participar en sorteos
 - [RF17-Conversiones de Crypto](#)
 - [RF18-BitLevel Academy](#)
 - [RF19-Preguntas frecuentes](#)
 - [RF20-Stacking](#)
 - [RF21-Virtual Card](#)
- Admin :
 - RF22-Recibir alertas de robo sin confirmar “phishing” “fuerza bruta”
 - RF23-Registro de transacciones
 - RF24-Crear usuarios (para testing)
 - RF25-Backup

3.3 - Requisitos no funcionales (opcional. pero recomendable).

RnF01: La aplicación debe ser transparente para el usuario.

RnF02: Los usuarios deben ser verificados para cumplir con la normativa KYC (Know Your Customer).

RnF03: La aplicación debe ofrecer escalabilidad y confianza al usuario.

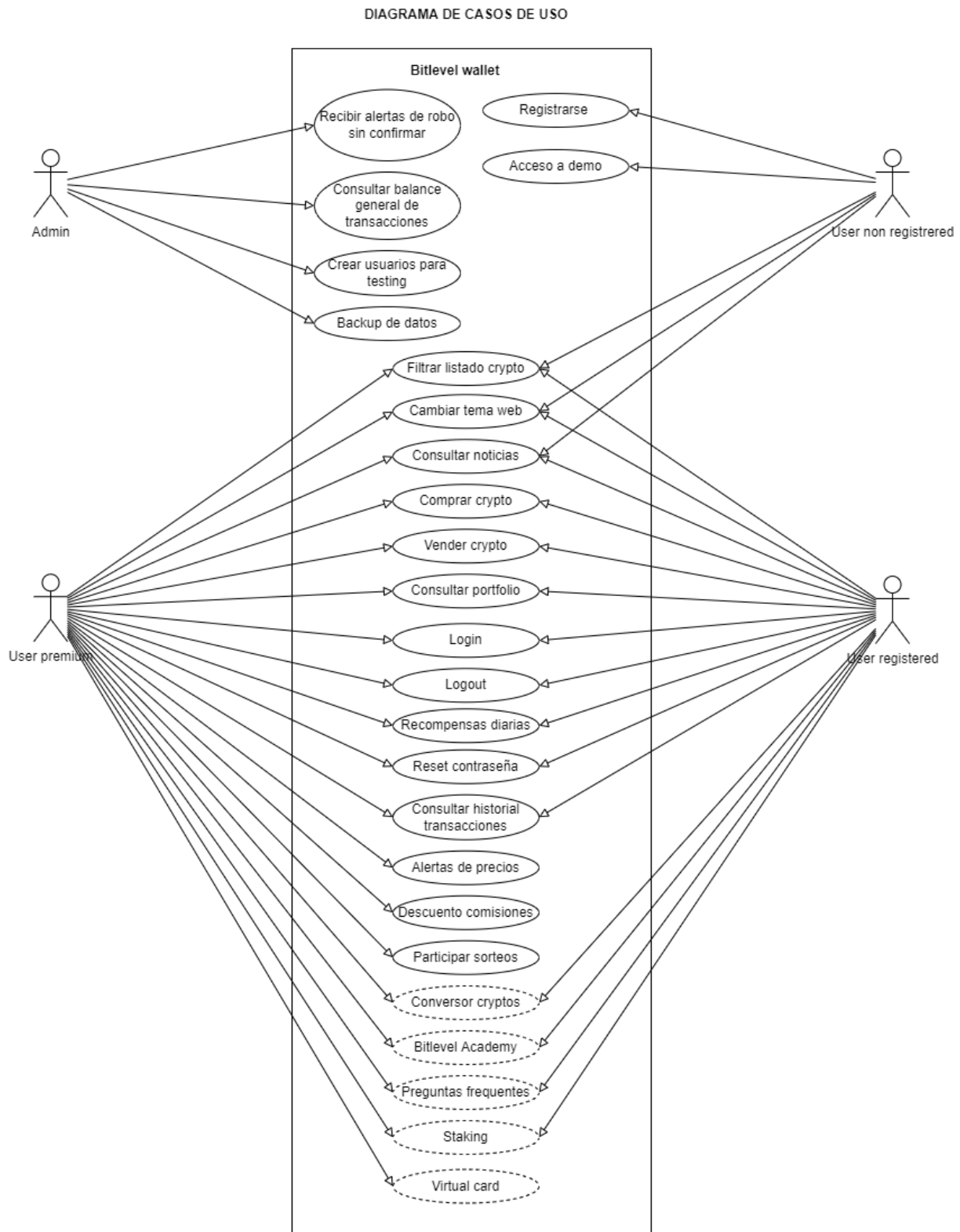
RnF04: La aplicación debe ofrecer confianza al usuario al depositar capital.

RnF05: La aplicación debe tener un servicio de respaldo por si el api consumido da algún tipo de error

3.4 - Requisitos sistema (opcional, pero recomendable).

RS01: Los servicios de la plataforma deben estar disponibles las 24h.

3.5 - Diagrama de casos de uso.



3.6 - Documento detallado de requisitos.

Documento detallado de caos de uso

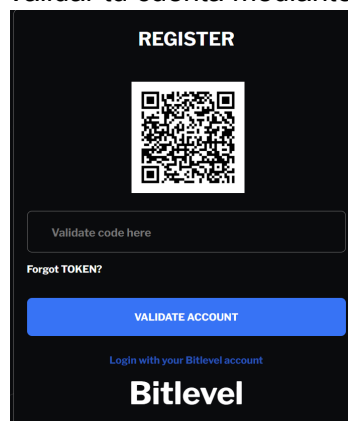
Identificador	01
Nombre	Register
Descripción	Registro de usuario
Actores	Usuario_no_registrado
Precondiciones	No estar registrado

Secuencia normal

Paso 1	<p>Acceder a http://register.bitlevel.es</p> 
Paso 2	<p>Introducir los datos requeridos (mail,username,password)</p> 
Paso 3	<p>Marcar la casilla de certificación de edad</p> 

Paso 4

Validar tu cuenta mediante un qr o mail otp



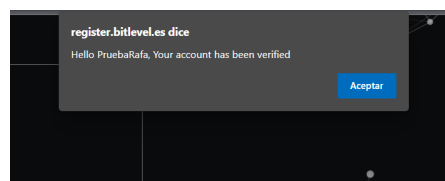
OTP VERIFICATION CODE Recibidos x

cryptoappld@gmail.com
para mí ▼

Your OTP CODE IS e0e375e3

Responder

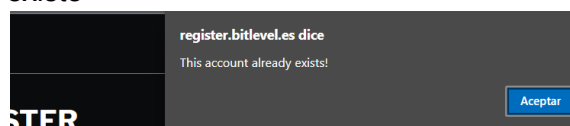
Reenviar



Secuencia Alternativa

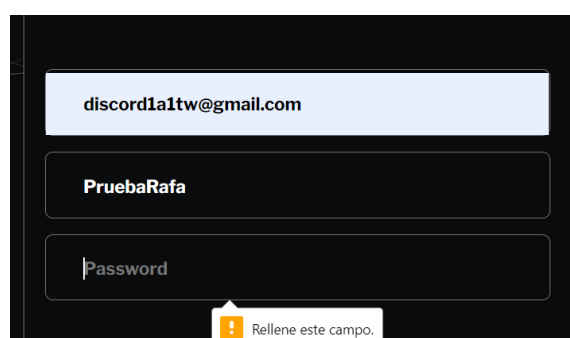
E1

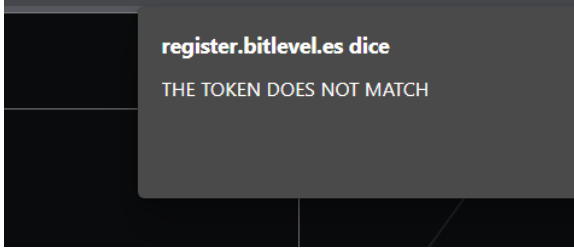
Si a la hora de registrarse el username o el correo ya están en uso, saltará una alerta para avisar al usuario de que esa cuenta ya existe



E2

Si a la hora de darle al botón de creación de cuenta, faltan campos por rellenar, saltará una alerta

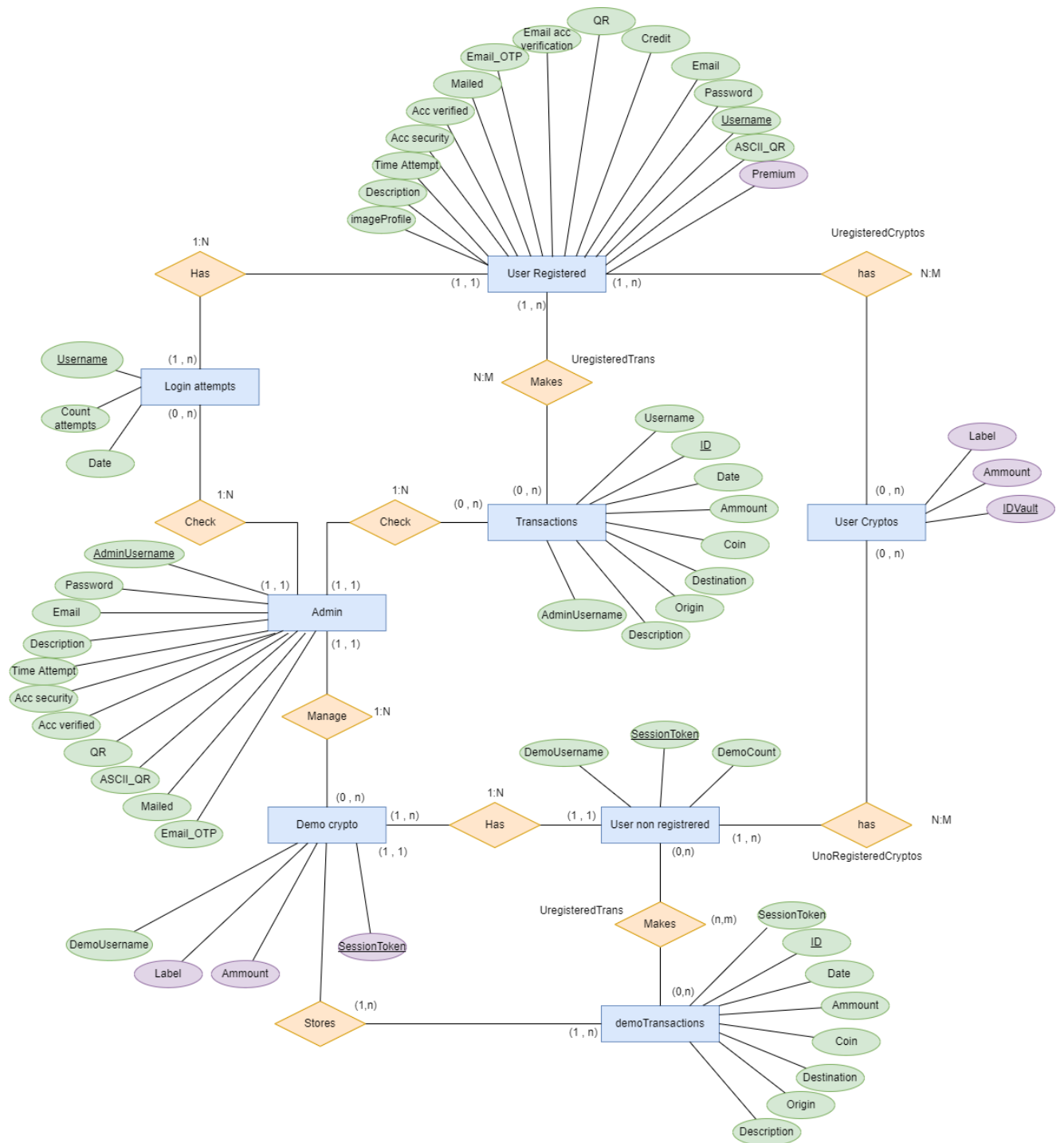


E3	<p>Si a la hora de verificar el qr / mail otp , el usuario introduce un dato incorrecto, saltará una alerta</p> 
Postcondicion	<p>A la hora de realizar el login, el usuario necesitará tener acceso al mail o tener escaneado el qr para verificar su identidad</p>

3.7 - Matriz Requisitos/Tecnologías.

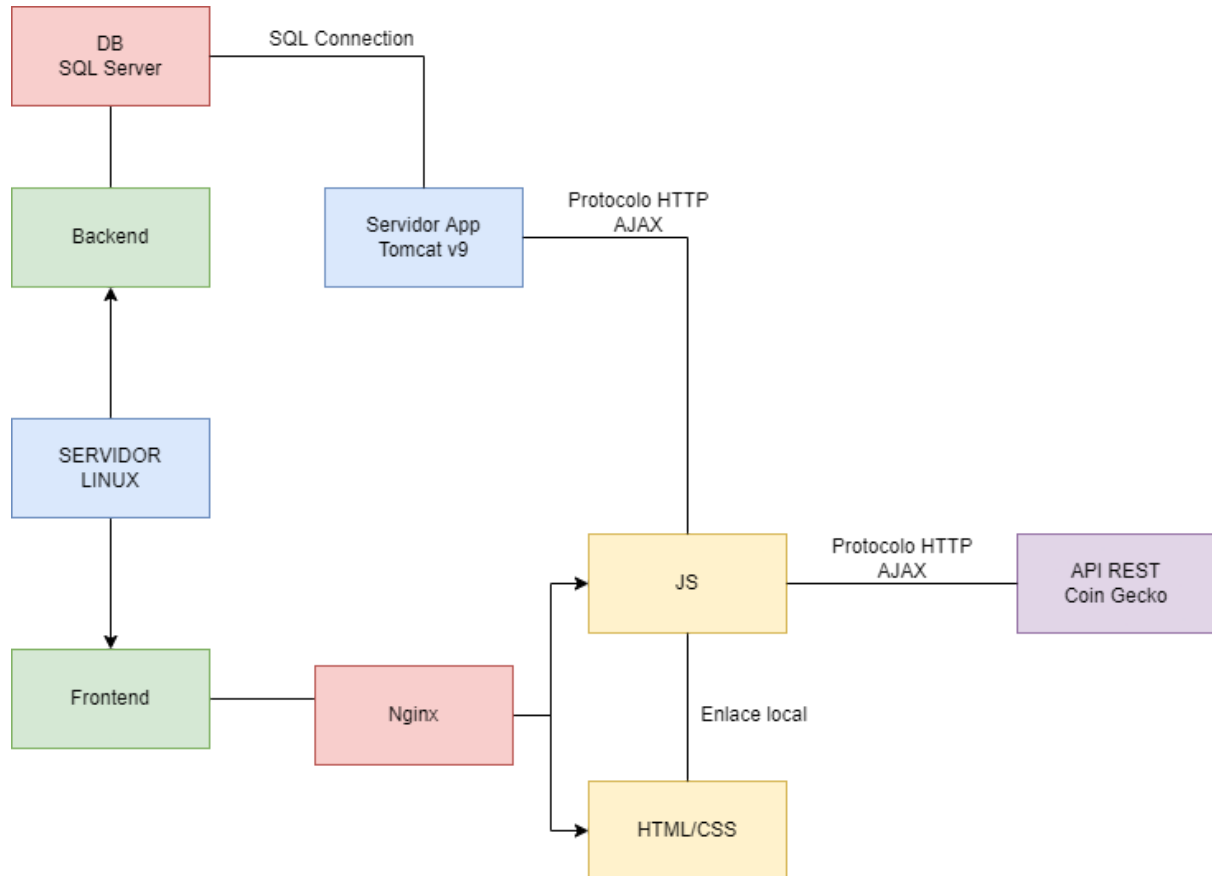
Matriz					
Requisitos / Tecnologías	SQL Server	Nginx(HTML/CSS)	Nginx(JS)	API REST	Tomcat v9 (Java)
RF01-Demo	x	x	x	x	x
RF02-Filtrar Crypto		x	x	x	
RF03-Cambiar tema web		x	x		
RF04-Registrarse	x	x	x		x
RF05-Visitar Portafolios	x	x	x		x
RF06-Consultar Noticias		x	x		
RF07-Recompensas diarias	x	x	x		x
RF08-Comprar Crypto	x	x	x		x
RF09-Vender Crypto	x	x	x		x
RF10-Resetear contraseñas / qr 2fa	x	x	x		x
RF11-Consultar registro de transacciones	x	x	x		x
RF12-Login	x	x	x		x
RF13-Logout	x	x	x		x
RF17-Conversiones de Crypto		x	x		x
RF18-BitLevel Academy		x	x		
RF19-Preguntas frecuentes		x	x		
RF20-Stacking	x	x	x		x
RF21-Virtual Card	x	x	x		x
RF22-Recibir alertas de robo sin confirmar "phishing" "fuerza bruta"		x	x		x
RF23-Registro de transacciones	x	x	x		x
RF24-Crear usuarios (para testing)	x	x	x		x
RF25-Backup	x	x	x		x

3.8 - Modelo Entidad-Relación.

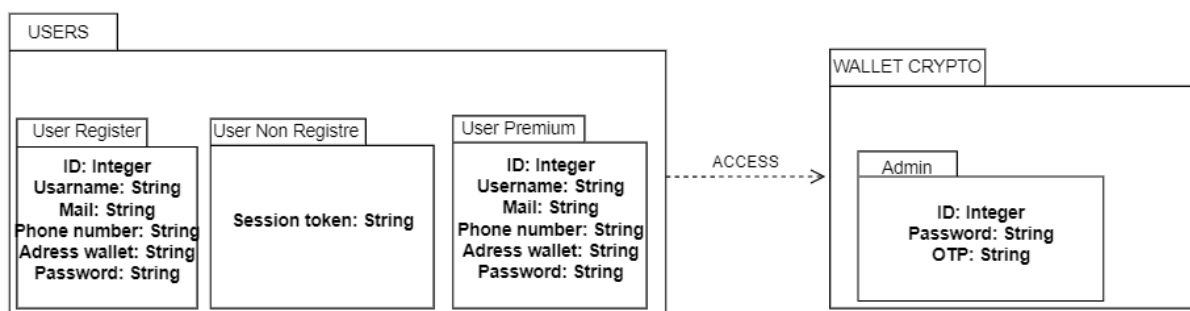


4.0 Diseño

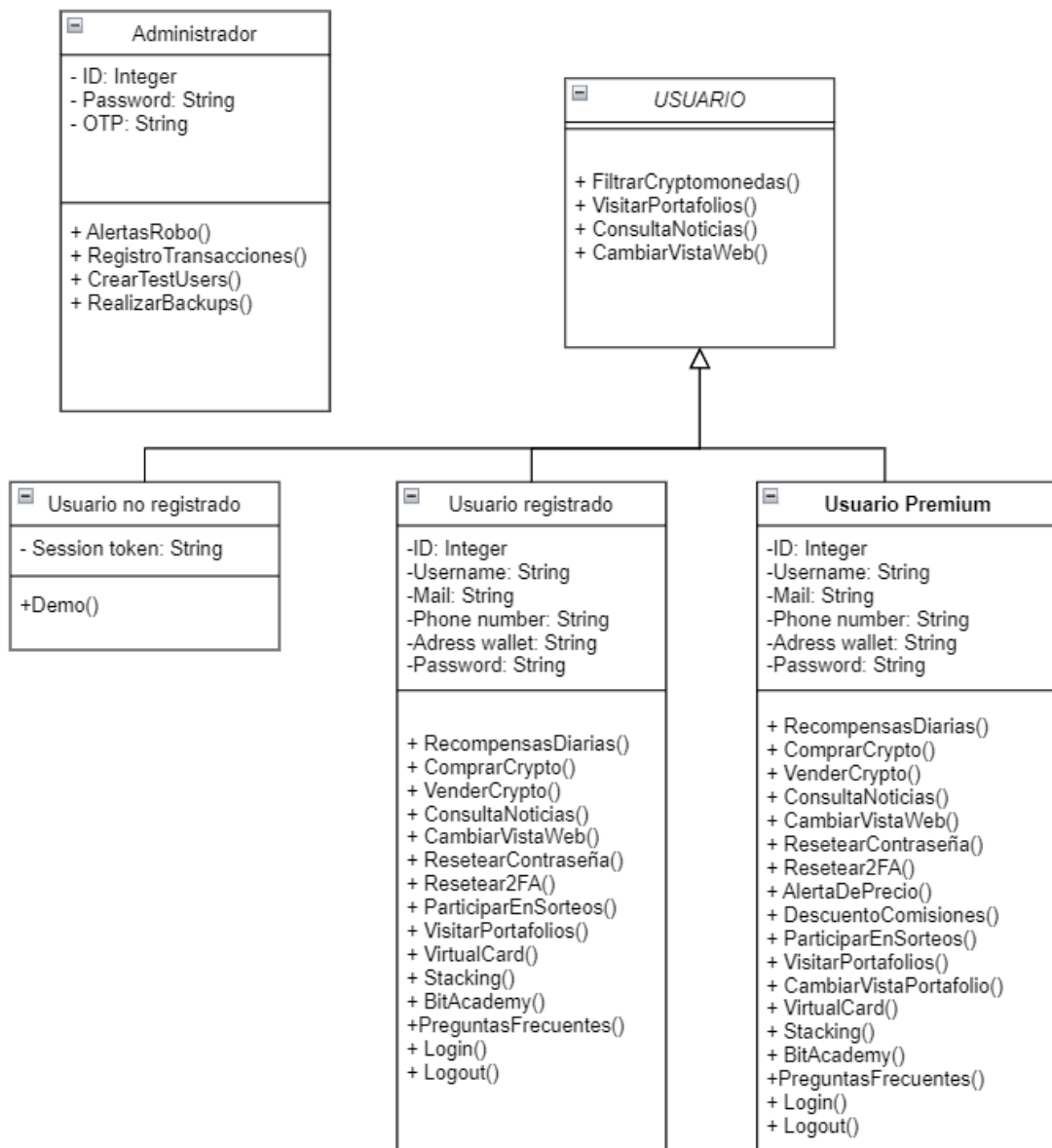
4.1 - Diagrama inicial del conjunto del proyecto



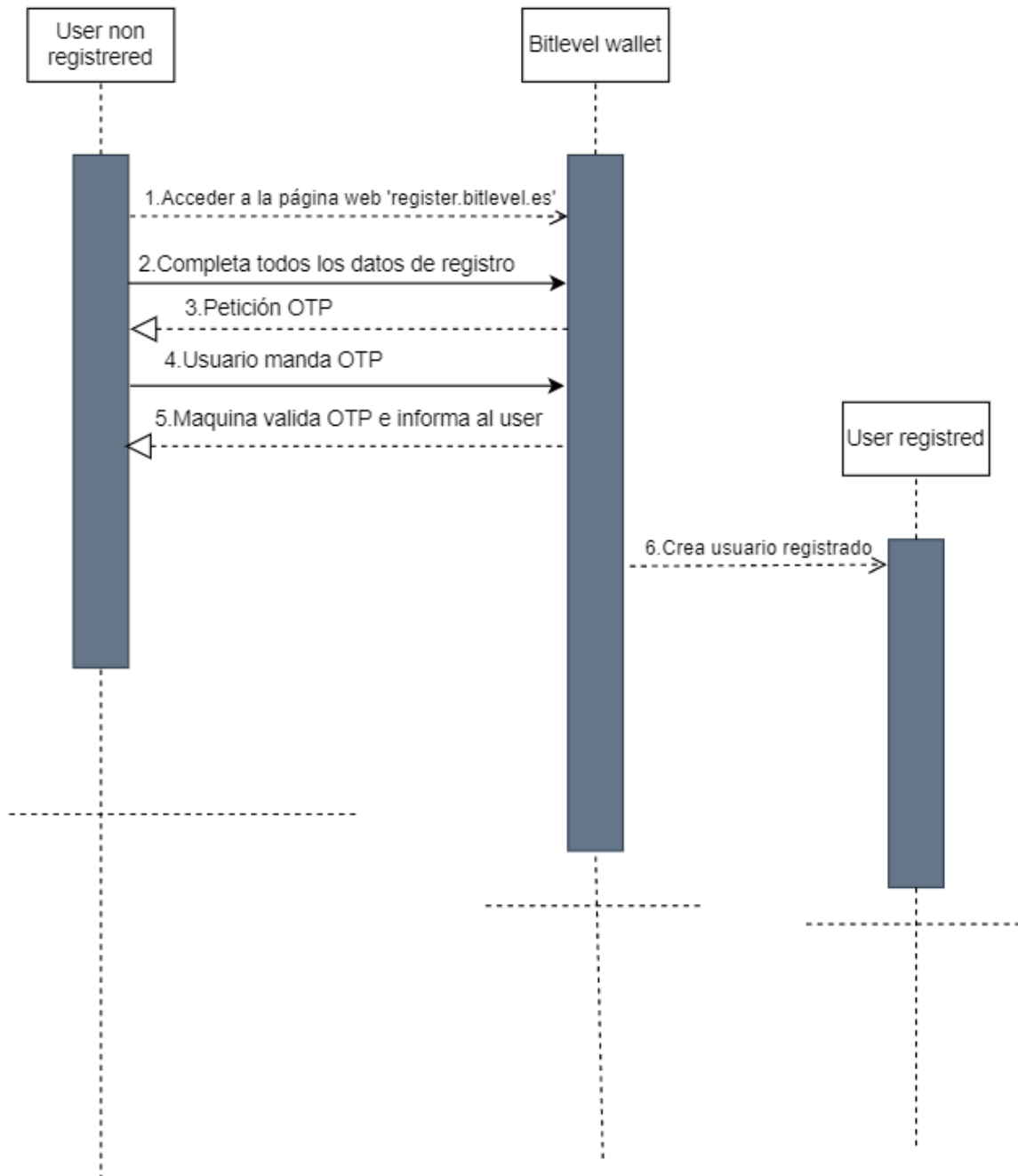
4.2 Diagrama de paquetes (opcional).



4.3 Diagrama de clases.



4.4 Diagrama de secuencia



4.5 Paso a relacional y normalización

Modelo Relacional

UserRegistered (Username, Premium, Email, Password, ASCII_QR, Credit, QR, EmailVerified, EmailOTP, Mailed, AccVerified, AccSecurity, TimeAttempt, Description, ImageProfile)

UserCryptos (IDVault, Label, Ammount)

Transactions (ID, UserRegisteredFK, Date, Ammount, Coin, Destination, Origin, Description, Premium)

Admin (AdminUsername, Password, Email, Description, TimeAttempt, AccSecurity, AccVerified, ASCII_QR, QR, Mailed, EmailOTP)

UserNonRegistered (SessionToken, DemoUsername, DemoCount, Balance)

DemoTransactions (ID, UserNoRegFK, Username, Date, Ammount, Coin, Destination, Origin, Description, Demo, Premium)

DemoUserCryptos (IDVault, Label, Ammount)