**INDICE DE ANEXOS**

[Anexo 1: Funcionamiento de las Billeteras HD 74](#_Toc517793063)

[Anexo 2: Cronograma del Proyecto 75](#_Toc517793064)

[Anexo 3: Modelos de Objetos Almacenados en Firebase Realtime Database 76](#_Toc517793065)

[3.1. Modelo de Datos User 76](#_Toc517793069)

[3.2. Modelo de Datos Usuario (Firebase) 78](#_Toc517793070)

[3.3. Modelo de Datos User (List) 80](#_Toc517793071)

[3.4. Modelo de Datos AddressBook 81](#_Toc517793072)

[3.5. Modelo de Datos Wallet (User) 82](#_Toc517793073)

[3.6. Modelo de Datos Wallet (List) 83](#_Toc517793074)

[3.7. Modelo de Datos Keys 84](#_Toc517793075)

[3.8. Modelo de Datos Crypto 85](#_Toc517793076)

[3.9. Modelo de Datos Activities 86](#_Toc517793077)

[3.10. Modelo de Datos Token 87](#_Toc517793078)

[Anexo 4: Ejemplo de la estructura de datos JSON en Realtime Database 88](#_Toc517793079)

[Anexo 5: Descripción de las tareas de Kanban 89](#_Toc517793080)

[Anexo 6: Tablero Kanban Semana por Semana 102](#_Toc517793081)

[Anexo 7: Clases y Servicios creados durante la adaptación de las vistas del FrontEnd 107](#_Toc517793082)

[Anexo 8: Manejo de Errores dentro de la aplicación móvil 110](#_Toc517793083)

[Anexo 9: Estructuras de Datos de la aplicación móvil: Modelos 111](#_Toc517793084)

[Anexo 10: Estructuras de Datos de la aplicación móvil: Interfaces 113](#_Toc517793085)

[Anexo 11: Generación de la información criptográfica de una Billetera HD 116](#_Toc517793086)

[Anexo 12: Proceso de creación y envió de una transacción de criptomonedas. 117](#_Toc517793087)

[Anexo 13: Diagrama de Secuencia de la Activación del Segundo Factor de Autentificación 118](#_Toc517793088)

[Anexo 14: Pantallas de la Aplicación Móvil 119](#_Toc517793089)

[Anexo 15: Reglas de Seguridad establecidas en Firebase Realtime Database 123](#_Toc517793090)

**Anexo 1: Funcionamiento de las Billeteras HD**

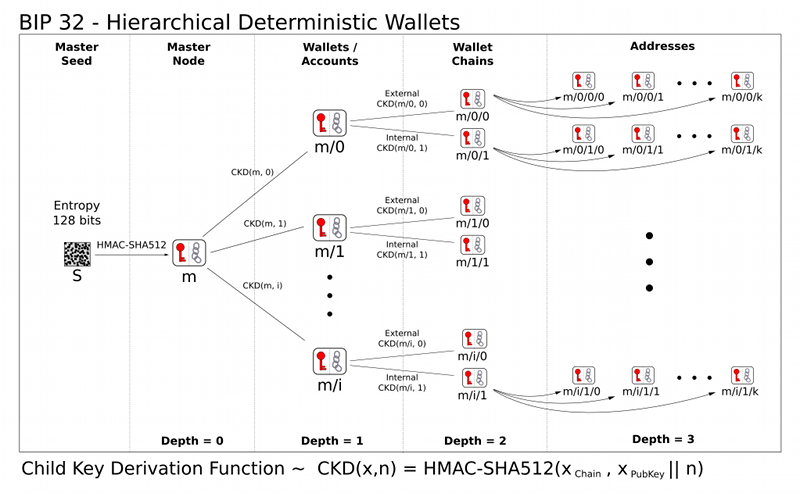
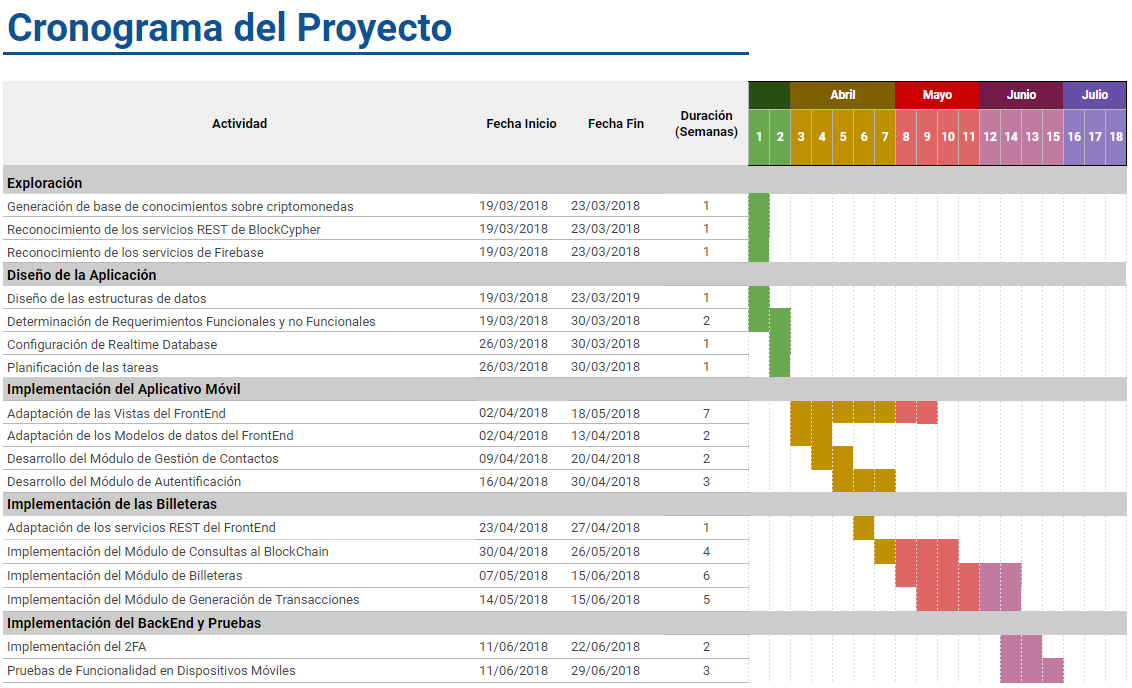


Imagen extraida de <https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki>

**Anexo 2: Cronograma del Proyecto**



**Anexo 3:** **Modelos de Objetos Almacenados en Firebase Realtime Database**

3. 1. Modelo de Datos User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objeto | User | | | |
| Descripción  Objeto que almacena información relevante de un usuario de la aplicación, incluyendo preferencias de usuario, datos de cada billetera y su libreta de contactos. | | | | |
| Atributos | | | | |
| Nombre | **Tipo** | **Descripción** | **Requerido** | **Único** |
| userEmail | String | Correo electrónico del usuario. | Si | Si |
| Img | String | Dirección a la imagen de perfil del usuario. Si el proveedor es Google, este es su imagen del correo electrónico. En caso contrario, es una imagen almacenada en Firebase Storage. | Si | No |
| currency | String | Abreviación de la moneda local que el usuario utiliza. Permite generar una referencia al valor de una criptomoneda o del saldo de una billetera. Valor por defecto es USD. | Si | No |
| token | Token | Cada usuario tiene asociado un objeto de tipo Token, el cual permite manejar la información referente al segundo factor de autentificación | Si | Si |
| wallet | Árbol de Billeteras | Lista de todas las billeteras asociadas al usuario. Cada vez que se inserta una billetera nueva, esta se encuentra debajo de un id único. | No | No |
| addressBook | Árbol de Contactos | Lista de todos los contactos de la libreta de contactos del usuario. Cada vez que se inserta un contacto nuevo, esta se encuentra debajo de un id único. | No | No |

* 1. Modelo de Datos Usuario (Firebase)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Interfaz | Usuario (Firebase Atuh) | | | |
| Descripción  Interfaz de la clase usuario de Firebase Auth. Posee información relevante a un usuario registrado en la aplicación. | | | | |
| Atributos | | | | |
| Nombre | Tipo | Descripción | Requerido | Único |
| uid | String | Cadena de caracteres única generada por Firebase Auth que identifica al usuario. | Si | Si |
| email | String | Correo electrónico del usuario. | Si | Si |
| emailVerified | Boolean | Indica si el usuario confirmo su correo electrónico a través de Firebase Auth. Si el usuario inicia sesión a través de Google, esto es True. | Si | No |
| photoURL | String | Dirección a la imagen de perfil del usuario. Si el proveedor es Google, este es su imagen del correo electrónico. En caso contrario, es una imagen almacenada en Firebase Storage | Si | Si |
| displayName | String | El nombre o alias del usuario. Si este accede a través de Google, es el nombre asociado al correo electrónico. En caso contrario es el correo electrónico. | No | No |
| providerData | Objeto | Información sobre el proveedor del usuario | No | No |
| providerID | String | ID del proveedor del usuario. | Si | No |
| phoneNumber | String | Número de teléfono asociado al usuario. | No | No |
| metadata | Objeto | Metadata sobre el usuario, tal como último acceso, si es un usuario nuevo y coordenadas, | No | No |
| isAnonymous | Boolean | Indica si el usuario se encuentra anónimo | Si | No |

* 1. Modelo de Datos User (List)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objeto | User (List) | | | |
| Descripción  Objeto que almacena información relevante de un usuario de la aplicación, incluyendo su imagen de para consulta por parte de otros usuarios. | | | | |
| Atributos | | | | |
| Nombre | Tipo | Descripción | Requerido | Único |
| Email | String | Correo electrónico del usuario. | Si | Si |
| Img | String | Dirección a la imagen de perfil del usuario. Si el proveedor es Google, este es su imagen del correo electrónico. En caso contrario, es una imagen almacenada en Firebase Storage | Si | No |
| Email | String | Abreviación de la moneda local que el usuario utiliza. Permite generar una referencia al valor de una criptomoneda o del saldo de una billetera. Valor por defecto es USD. | Si | No |

* 1. Modelo de Datos AddressBook

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objeto | AddressBook | | | |
| Descripción:  Almacena una lista de contactos que el usuario haya guardado en su libreta de contactos. Los contactos son otros usuarios de la aplicación. | | | | |
| Atributos | | | | |
| Nombre | Tipo | Descripción | Requerido | Único |
| Alias | String | Nombre que el usuario coloca al contacto. | Si | No |
| Email | String | Correo electrónico del contacto. | Si | Si |
| Img | String | Dirección a la imagen de perfil del contacto. Si el proveedor es Google, este es su imagen del correo electrónico. En caso contrario, es una imagen almacenada en Firebase Storage. | No | Si |
| Uid | String | Cadena de caracteres única generada por Firebase Auth que identifica al contacto. | Si | No |

* 1. Modelo de Datos Wallet (User)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objeto | Wallet (User) | | | |
| Descripción  Objeto que almacena la información de una billetera. Incluye los datos de las llaves públicas y privadas, el nombre de la criptomoneda y preferencias del usuario. | | | | |
| Atributos | | | | |
| Nombre | Tipo | Descripción | Requerido | Único |
| Name | String | Nombre de la billetera. Este valor es utilizado por BlockCypher para identificar cada billetera dentro de un Blockchain. | Si | Si |
| Keys | Keys | Objeto que almacena la información criptográfica de la billetera, necesario para realizar transacciones y generar nuevas direcciones | Si | Si |
| Crypto | Crypto | Objeto que almacena la información asociada a la criptomoneda seleccionada y las preferencias del usuario respecto a las unidades. | Si | No |
| address | String | Dirección de envió y recibo de criptomonedas. Solo si la criptomoneda de la billetera es Ethereum. Es generado a través del Keys de la billetera. | No | Si |

* 1. Modelo de Datos Wallet (List)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objeto | Wallet (List) | | | |
| Descripción  Lista que almacena información no sensible. Utilizada por la aplicación para poder manejar ciertas funciones dentro de la aplicación. | | | | |
| Atributos | | | | |
| Nombre | Tipo | Descripción | Requerido | Único |
| Name | String | Nombre de la billetera. Este valor es utilizado por BlockCypher para identificar cada billetera dentro de un Blockchain. | Si | Si |
| User | String | Correo electrónico del usuario que es dueño de la billetera. | Si | No |
| Crypto | String | Abreviación del nombre de la criptomoneda y si es cadena principal o de prueba. Valores: bcy (BlockCypher Testnet), btc (Bitcoin), tes (Bitcoin Testnet), dog (Dogecoin), ltc (Litecoin), eth (Ethereum), tet (Ethereum Testnet), das (Dash). | Si | No |
| address | String | Dirección de envió y recibo de criptomonedas. Solo si la criptomoneda de la billetera es Ethereum. Es generado a través del Keys de la billetera. | No | Si |

* 1. Modelo de Datos Keys

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objeto | Keys | | | |
| Descripción  Información criptográfica de una billetera. | | | | |
| Atributos | | | | |
| Nombre | Tipo | Descripción | Requerido | Único |
| mnemonics | String | Frase de recuperación de la Billetera, generada a través de BIP 39. | Si | Si |
| passphrase | String | Clave de encriptación utilizada al momento de generar la semilla de la frase de recuperación, es un segundo factor de seguridad. Necesario junto con la clave de recuperación para poder importar una billetera. | Si | No |
| seed | String | Valor de la semilla hexadecimal utilizada para generar la llave privada extendida. Es la clave de recuperación más la clave de encriptación (si esta existe) | Si | Si |
| xpriv | String | Llave privada extendida de la billetera, o llave privada en caso de que la criptomoneda sea Ethereum. | Si | Si |
| xpub | String | Llave pública extendida de la billetera, o llave publica en caso de que la criptomoneda sea Ethereum. | Si | Si |

* 1. Modelo de Datos Crypto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objeto | Crypto | | | |
| Descripción  Objeto que almacena información sobre la criptomoneda de una billetera y las preferencias de usuario de la misma. | | | | |
| Atributos | | | | |
| Nombre | Tipo | Descripción | Requerido | Único |
| value | String | Abreviación del nombre de la criptomoneda y si es cadena principal o de prueba. Valores: bcy (BlockCypher Testnet), btc (Bitcoin), tes (Bitcoin Testnet), dog (Dogecoin), ltc (Litecoin), eth (Ethereum), tet (Ethereum Testnet), das (Dash). | Si | No |
| coin | String | Abreviación del nombre de la criptomoneda. Valores: BTC, DOGE, LTC, DASH. | Si | No |
| difference | number | Número que representa la diferencia entre una criptomoneda y su unidad más pequeña no divisible. (Ej.: 1 BTC = 100,000,000 satoshis) | Si | No |
| name | String | Nombre completo de la moneda. | Si | No |
| units | Objeto | Objeto que guarda la preferencia del usuario, con el nombre de la unidad elegida y su diferencia con la unidad más pequeña no divisible | Si | No |

* 1. Modelo de Datos Activities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objeto | Activities | | | |
| Descripción  Almacena una lista de notificaciones referente a actividades que ha realizado el usuario. | | | | |
| Atributos | | | | |
| Nombre | Tipo | Descripción | Requerido | Único |
| date | String | Fecha en la que ocurrió la actividad. | Si | No |
| description | String | Cadena de caracteres con información adicional sobre una | No | No |
| name | String | Nombre de la actividad. | Si | No |

* 1. Modelo de Datos Token

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objeto | Token | | | |
| Descripción  Almacena la información única del segundo facto de autenticación del usuario. | | | | |
| Atributos | | | | |
| Nombre | Tipo | Descripción | Requerido | Único |
| activated | boolean | Indica que la opción de segundo factor de autenticación fue activada y verificada. | Si | No |
| enabled | boolean | Indica que la opción de segundo factor de autenticación fue activada, mas no necesariamente verificada. | Si | No |
| otpURL | String | Descripción del url que provee de información a la aplicación de autentificación | No | Si |
| dataURL | String | Valor en Base32 del código QR utilizado por la aplicación de autenticación | No | Si |
| secret | String | Identificador único del usuario, utilizado para autenticarlo | No | Si |

**Anexo 4: Ejemplo de la estructura de datos JSON en Realtime Database**

**Anexo 5: Descripción de las tareas de Kanban**

5.1. Descripción de las tareas de Adaptación de las vistas del FrontEnd

|  |  |  |  |
| --- | --- | --- | --- |
| División  Adaptación de las vistas del FrontEnd | | | Color |
| Tareas asociadas a la actualización del FrontEnd para mostrar información dinámica, manejo de errores y cambios de estilo | | |
| N | Nombre | Descripción | Prioridad |
| F-1 | Desarrollo Servicio Global Compartido AlertService | Creación del servicio e implementación de los métodos encargados de mostrar errores y mensajes informativos al usuario a través del uso del componente Alert de Ionic 3. | Baja |
| F-2 | Desarrollo Servicio Global Compartido LoaderService | Creación del servicio e implementación de los métodos encargados de mostrar pantallas de espera a través del uso del componente Loader de Ionic 3. | Baja |
| F-3 | Desarrollo Interceptor HttpErrorInterceptor | Creación de un interceptor de errores mediante el uso de la clase HttpClient de Angular, con el fin de manejar los errores retornados por los servidores REST y las APIs. | Media |
| F-4 | Traducción de la Aplicación Móvil | Uso de la librería de npm ngx-translate y archivos JSON para traducir la aplicación mediante el uso de un servicio clave – valor. | Baja |
| F-5 | Implementación de las vistas del módulo de Gestión de Contactos | Modificación de los componentes de Ionic – Angular que permiten al usuario interactuar con el módulo de Gestión de Contactos: libreta de contactos, agregar contacto y editar contacto. | Media |
| F-6 | Implementación de las vistas del módulo de Autenticación | Generación y modificación de los componentes de Ionic – Angular que permiten al usuario interactuar con el módulo de Autenticación: inicio de sesión, registro, configuración de cuenta y confirmar correo. | Alta |
| F-7 | Implementación de las vistas del módulo de Consultas al Blockchain | Modificación del componente de Ionic – Angular que permiten al usuario interactuar con el módulo de Consultas al Blockchain: explorador del Blockchain. | Baja |
| F-8 | Implementación de las vistas del módulo de Gestión de Billeteras | Generación y modificación de los componentes de Ionic – Angular que permiten al usuario interactuar con el módulo de Consultas al Gestión de Billeteras: inicio de sesión, ajustes de billeteras, creación de billeteras y mostrar frase de recuperación. | Alta |
| F-9 | Implementación de las vistas del módulo de Generación de Transacciones | Generación y modificación de los componentes de Ionic – Angular que permiten al usuario interactuar con el módulo de Generación de Transacciones: recibir, enviar, confirmación de envió transacciones y detalles de transacción. | Alta |
| F-10 | Cambios a vistas para agregar elementos de 2FA. | Modificación y generación de los componentes de Ionic – Angular que permiten al usuario interactuar con las opciones de segundo factor de autenticación: enviar y ajustes de seguridad. | Alta |
| F-11 | Manejo de errores HTTP | Mostrar errores de manera adecuada utilizando AlertService, LoaderService, HttpErrorInterceptor y ngx-translator. | Medio |
| F-12 | Implementación de clase global ExchangeService | Creación de la clase ExchangeService y métodos necesarios para mostrar el valor actualizado del balance de una billetera de criptomonedas en otras divisas. | Bajo |
| F-13 | Aplicación de estilo final a las vistas del aplicativo. | Aplicación de hojas de estilo y recursos visuales a los componentes de Ionic según el diseño determinado por la empresa. | Medio |

5.2. Descripción de las tareas de Adaptación de la capa lógica del FrontEnd

|  |  |  |  |
| --- | --- | --- | --- |
| División  Adaptación de la capa lógica del FrontEnd | | | Color |
| Tareas asociadas a la actualización del FrontEnd previo para ajustarse a los servicios REST a utilizarse y las nuevas estructuras de datos. | | |
| N | Nombre | Descripción | Prioridad |
| L-1 | Configuración de Reglas de Firebase | Creación y configuración desde la consola de Firebase para los servicios utilizados dentro del aplicativo. | Baja |
| L-2 | Cambio a las estructuras de Datos de la aplicación móvil | Creación de nuevos modelos de datos y actualización de los preexistentes según lo establecido en la sección de Anexo 9 y Anexo 10 dentro del aplicativo móvil. | Alta |
| L-3 | Creación Servicio Global Compartido SharedService | Generación y construcción de la clase SharedService a ser utilizado de manera global por los demás componentes y servicios dentro del aplicativo móvil. | Alta |
| L-4 | Creación Proveedor Compartido FirebaseProvider | Generación y construcción de la clase FirebaseProvider, encargada de la realización de las llamadas asíncronas a Firebase Realtime Database con el fin de realizar operaciones CRUD en los modelos de datos establecidos en Anexo 3. | Alta |
| L-5 | Creación Servicio Global Compartido RestService | Generación y construcción de la clase RestService, encargada de las llamadas asíncronas a la API de BlockCypher. | Alta |

5.3. Descripción de las tareas del Módulo de Autenticación

|  |  |  |  |
| --- | --- | --- | --- |
| División  Módulo de Autenticación | | | Color |
| Tareas asociadas a la autentificación, manejo de sesión y registro de los usuarios de la aplicación móvil. | | |
| N | Nombre | Descripción | Prioridad |
| A-1 | Creación de clase AuthService | Generación de la clase AuthService donde se implementan los métodos y asociados al módulo de autenticación. | Alta |
| A-2 | Registro de usuarios | Registro de nuevos usuarios dentro del sistema utilizando correo electrónico y contraseña. | Media |
| A-3 | Autentificación del usuario. | Proceso asociado a verificar al usuario registrado dentro de la aplicación móvil. | Alta |
| A-4 | Inicio de Sesión | Inicio de sesión de usuarios ya registrados dentro del sistema utilizando correo electrónico y contraseña. | Media |
| A-5 | Verificación del Correo Electrónico | Verifica que el correo electrónico que un usuario utilizo para registrarse en el sistema es de su propiedad al enviar un mensaje de verificación. | Baja |
| A-6 | Recuperación de Contraseña | Envía un mensaje al correo electrónico asociado a un usuario registrado, para restablecer la contraseña utilizada para iniciar sesión | Baja |
| A-7 | Manejo de Estado de Sesión | Manejo del estado de sesión del usuario dentro del aplicativo, incluyendo cierre de sesión y acciones asociadas al inicio de sesión | Alta |
| A-8 | Autenticación de usuario (Proveedor Google) | Registro e inicio de sesión de usuarios al sistema utilizando sus cuentas Google. Manejo de redireccionamiento y credenciales asociados. | Media |

5.4. Descripción de las tareas del Módulo de Gestión de Contactos

|  |  |  |  |
| --- | --- | --- | --- |
| División  Módulo de Gestión de Contactos | | | Color |
| Tareas asociadas a el manejo de la libreta de contactos de el aplicativo. | | |
| N | Nombre | Descripción | Prioridad |
| C-1 | Agregar Contactos | Permite a un usuario con sesión activa, agregar nuevos registros dentro de la libreta de contactos de su aplicación. | Media |
| C-2 | Eliminar Contactos | Permite a un usuario con sesión activa, eliminar nuevos registros dentro de la libreta de contactos de su aplicación. | Baja |
| C-3 | Consultar Contactos | Recupera la información de la libreta de contactos de un usuario con sesión activa dentro de la aplicación. | Media |
| C-4 | Editar Contactos | Permite editar la información de un registro almacenado en la libreta de contactos de un usuario con sesión activa dentro de la aplicación. | Baja |
| C-5 | Buscar Contactos por Correo Electrónico | Verifica que un contacto que se vaya a registrar dentro de la libreta de contactos se encuentre registrado en el sistema. | Media |

5.5. Descripción de las tareas del Módulo de Consultas al BlockChain

|  |  |  |  |
| --- | --- | --- | --- |
| División  Módulo de Consultas al BlockChain | | | Color |
| Tareas asociadas a las consultas de información relacionada con las distintas cadenas de bloques, direcciones y bloques. | | |
| N | Nombre | Descripción | Prioridad |
| B-1 | Consultas de Blockchains | Permite consultar el estado mas reciente de un Blockchain especifico dentro de la aplicación móvil. | Bajo |
| B-2 | Consulta de Bloques | Permite consultar un bloque en específico a partir de su hash. | Bajo |
| B-3 | Consulta de Balances de Direcciones | Consulta el balance de una dirección de criptomonedas. | Media |
| B-4 | Consultas de Transacciones | Consulta la información de una transacción en especifico a partir de su hash. | Media |

5.6. Descripción de las tareas del Módulo de Gestión de Billeteras

|  |  |  |  |
| --- | --- | --- | --- |
| División  Módulo de Gestión de Billeteras | | | Color |
| Tareas asociadas con la creación y gestión de las billeteras de criptomonedas del usuario. | | |
| N | Nombre | Descripción | Prioridad |
| W-1 | Desarrollo de clase global compartida KeyService | Creación de la clase KeyService, que permite manejar la información criptográfica de una billetera y realizar el firmado de las transacciones. | Alta |
| W-2 | Creación de Billeteras HD | Implementación de las funciones asociadas a la creación y almacenamiento de una billetera HD y su registro en BlockCypher. | Alta |
| W-3 | Creación de Direcciones Ethereum | Implementación de las funciones asociadas a la creación y almacenamiento de una dirección de Ethereum y Ethereum Testnet. | Alta |
| W-4 | Consulta de Balances de Billeteras HD | Permite consultar el balance, incluyendo número de transacciones, de una billetera HD. | Alta |
| W-5 | Consulta de Balances de Direcciones Ethereum | Permite consultar el balance, incluyendo número de transacciones de una dirección de Ethereum y Ethereum Testnet. | Alta |
| W-6 | Generación Direcciones de Recibo de Criptomonedas | Genera y muestra, una dirección para el recibo de criptomonedas. | Alta |
| W-7 | Mostrar Frase de Recuperación | Muestra la frase de recuperación de una billetera HD dentro de la interfaz de la aplicación móvil. | Baja |
| W-8 | Importar Billeteras HD | Permite al usuario importar billeteras HD a la aplicación móvil a partir de una clave de recuperación. | Baja |
| W-9 | Cambiar preferencias de las Billeteras | Permite al usuario configurar las opciones de unidades de criptomonedas | Baja |

5.7. Descripción de las tareas del Módulo de Generación de Transacciones

|  |  |  |  |
| --- | --- | --- | --- |
| División  Módulo de Generación de Transacciones | | | Color |
| Tareas asociadas con la generación de transacciones de criptomonedas. | | |
| N | Nombre | Descripción | Prioridad |
| T-1 | Generación de Transacciones | Implementación de los métodos necesarios para permitir que un usuario genere y modifique la información de una transacción de criptomonedas. | Alta |
| T-2 | Firmado de Transacciones de Billeteras HD | Implementación de las funciones necesarias para poder realizar el firmado de las transacciones de billeteras HD. | Alta |
| T-3 | Firmado de Transacciones de Direcciones Ethereum | Implementación de las funciones necesarias para poder realizar el firmado de las transacciones de direcciones Ethereum y Ethereum Testnet. | Alta |
| T-4 | Envio de Transacciones | Implementación de las funciones necesarias que permiten enviar transacciones firmadas a BlockCypher. | Alta |

5.8. Descripción de las tareas de Implementación del servidor Backend de 2FA

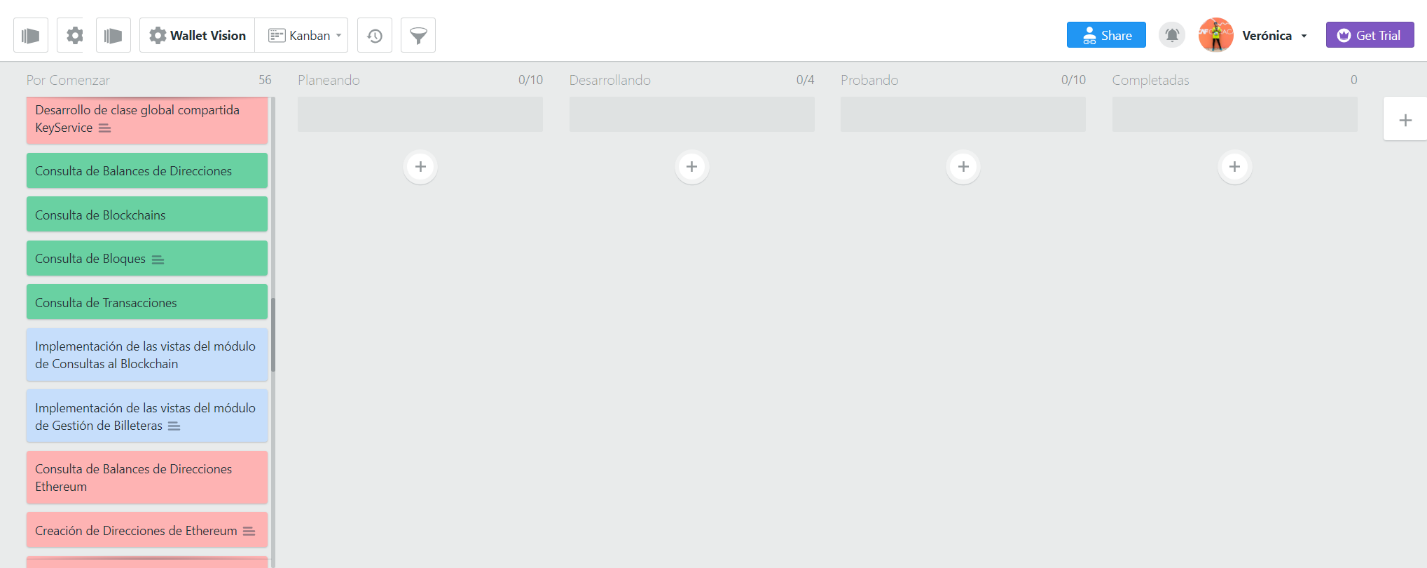
|  |  |  |  |
| --- | --- | --- | --- |
| División  Implementación del servidor Backend de 2FA | | | Color |
| Tareas asociadas con la creación y gestión de las billeteras de criptomonedas del usuario. | | |
| N | Nombre | Descripción | Prioridad |
| N-1 | Creación y configuración de servidor 2FA | Creación y configuración inicial de un servidor NodeJS para recibir solicitudes por parte de la aplicación móvil. | Alta |
| N-2 | Implementación de ruta para activar opción de 2FA en el servidor. | Desarrollo de la ruta para poder activar la opción de 2FA en el servidor, incluyendo almacenr el token del usuario y envió de mensaje al correo del usuario con instrucciones. | Alta |
| N-3 | Implementación de ruta para desactivar opción de 2FA en el servidor. | Desarrollo de la ruta que permite a un usuario desactivar la opción de 2FA en el servidor, incluyendo la actualización de los datos del token del usuario. | Media |
| N-4 | Implementación de ruta para verificar identidad del usuario en el servidor. | Desarrollo de la ruta que permite a un usuario ser autenticado siempre y cuando posea la opción de 2FA activada en el servidor e introduzca la clave de un solo uso o OTP. | Alta |
| N-5 | Implementación de clase global TwoFactorService en aplicación móvil. | Generación e implementación de un servicio de Angular que permite a la aplicación móvil manejar las solicitudes al servidor de 2FA. | Alta |

5.9. Descripciones de las tareas de Pruebas y Trabajos finales del aplicativo

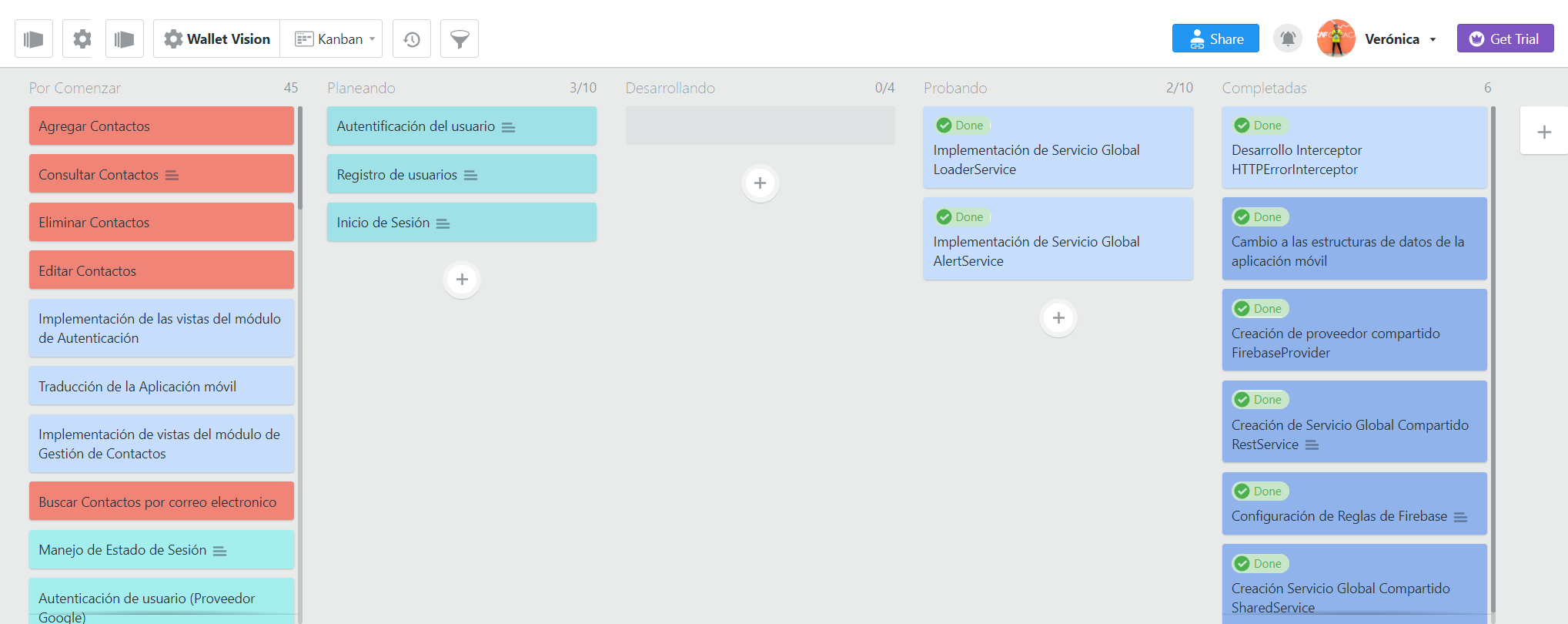
|  |  |  |  |
| --- | --- | --- | --- |
| División  Pruebas y Trabajos finales del aplicativo | | | Color |
| Tareas asociadas con las pruebas de funcionalidad en dispositivos móviles y requerimientos finales. | | |
| N | Nombre | Descripción | Prioridad |
| M-1 | Integración con Cordova. | Agregar a la aplicación móvil todas las extensiones que sea requeridas para agregar soporte nativo en dispositivos móviles. | Media |
| M-2 | Pruebas en Dispositivos Móviles | Pruebas del aplicativo móvil y servidor 2AF en dispositivos reales, verificando que cumple con los requerimientos establecidos. | Media |
| M-3 | Generación de archivos de configuración. | Generación de archivos de configuración para facilitar el proceso de instanciación del aplicativo móvil. | Baja |
| S-1 | Proceso de Gestión de Riesgos de Seguridad de la Aplicación | Análisis de los factores vulnerables de la aplicación móvil, incluyendo estructuras de datos, procesos internos del aplicativo, firmado de las transacciones y seguridad general. | Alta |
| S-2 | Comparación entre GulpJS y WebPack | Realizar una comparación entre dos herramientas para automatización de tareas en aplicaciones web NodeJs para realizar instancias del producto. | Alta |

**Anexo 6: Tablero Kanban Semana por Semana**

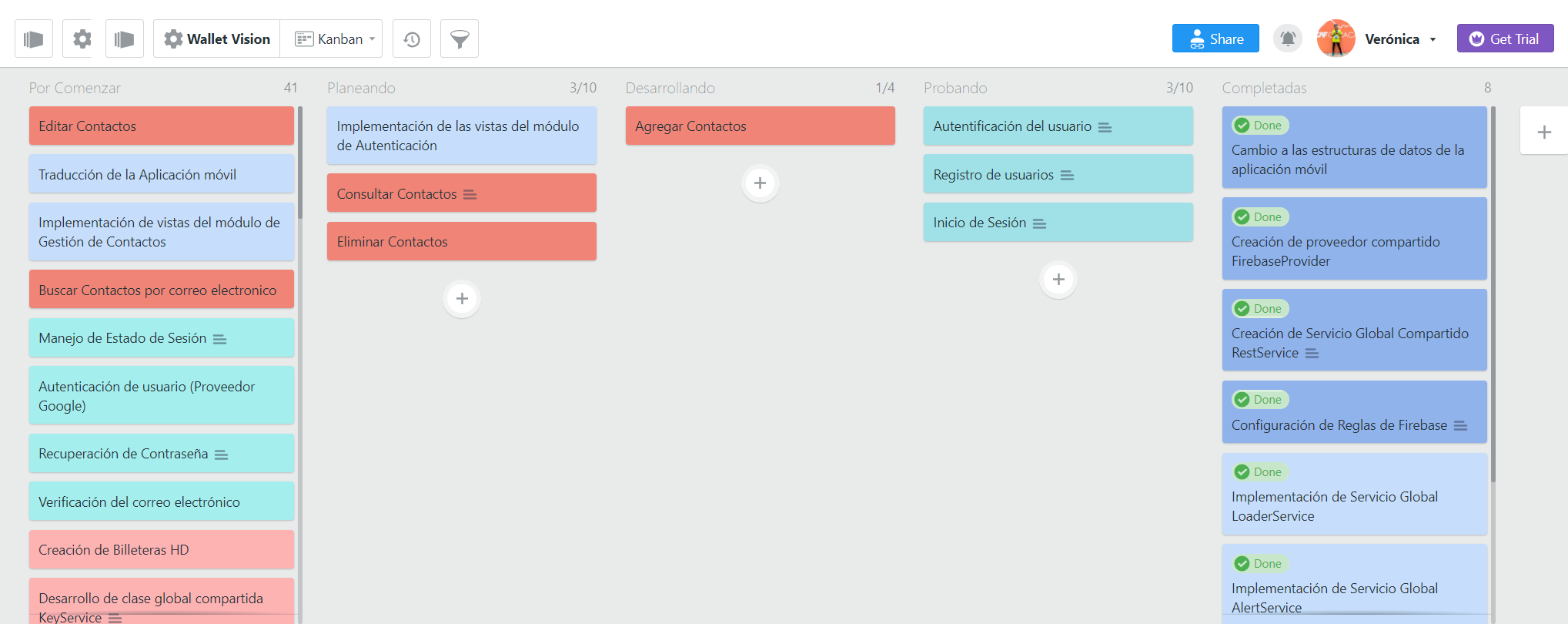
Semana 1



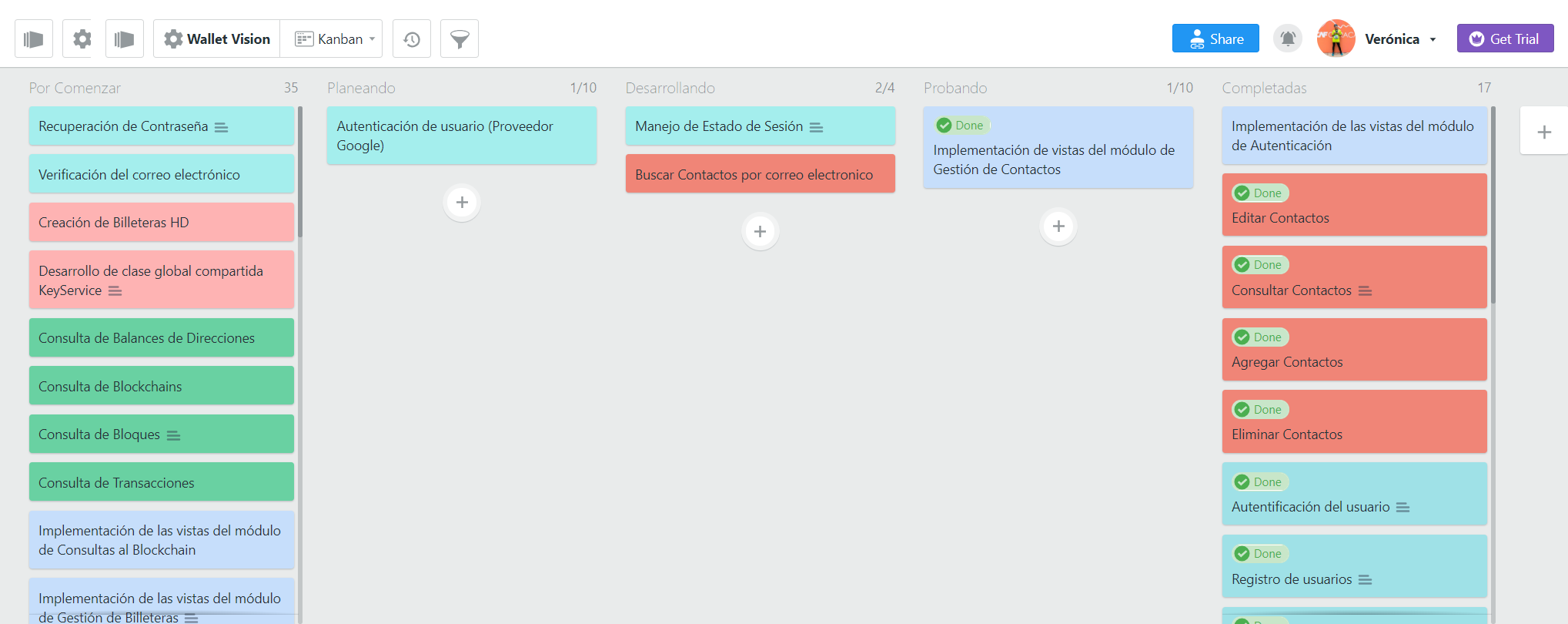
Semana 2



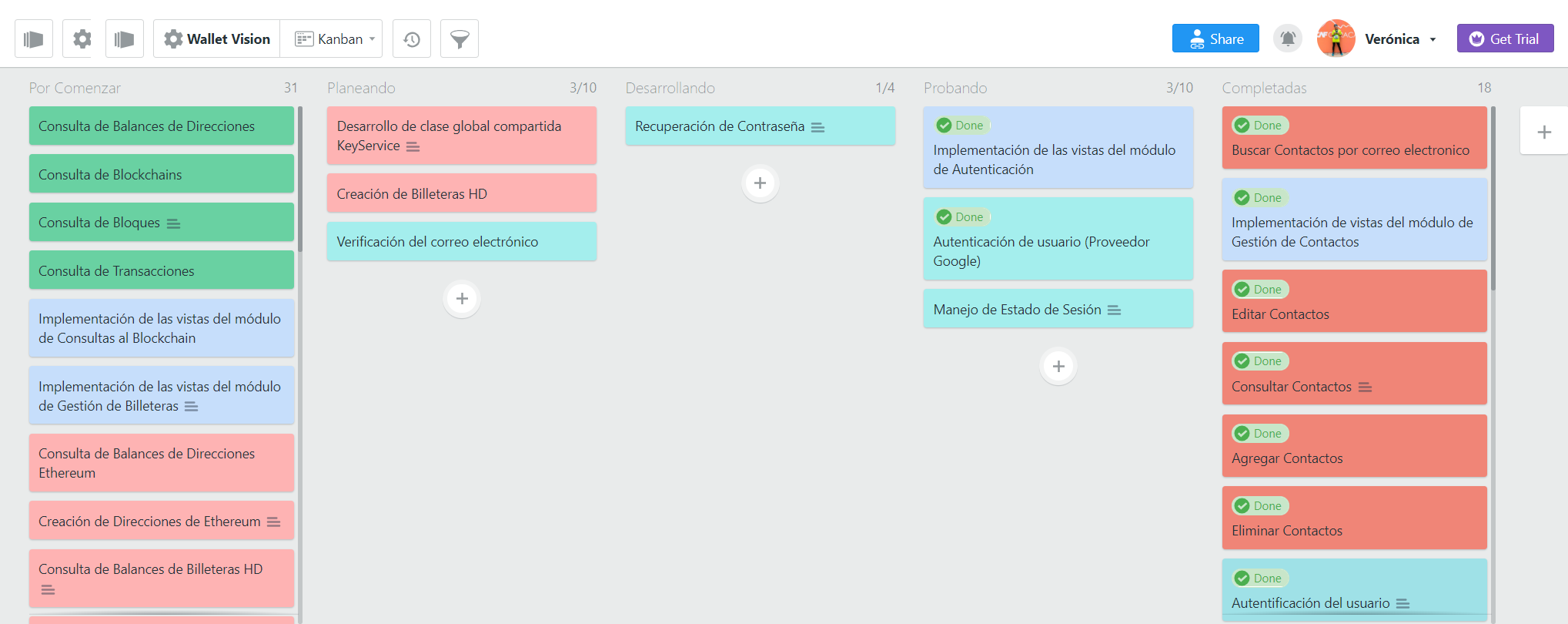
Semana 3



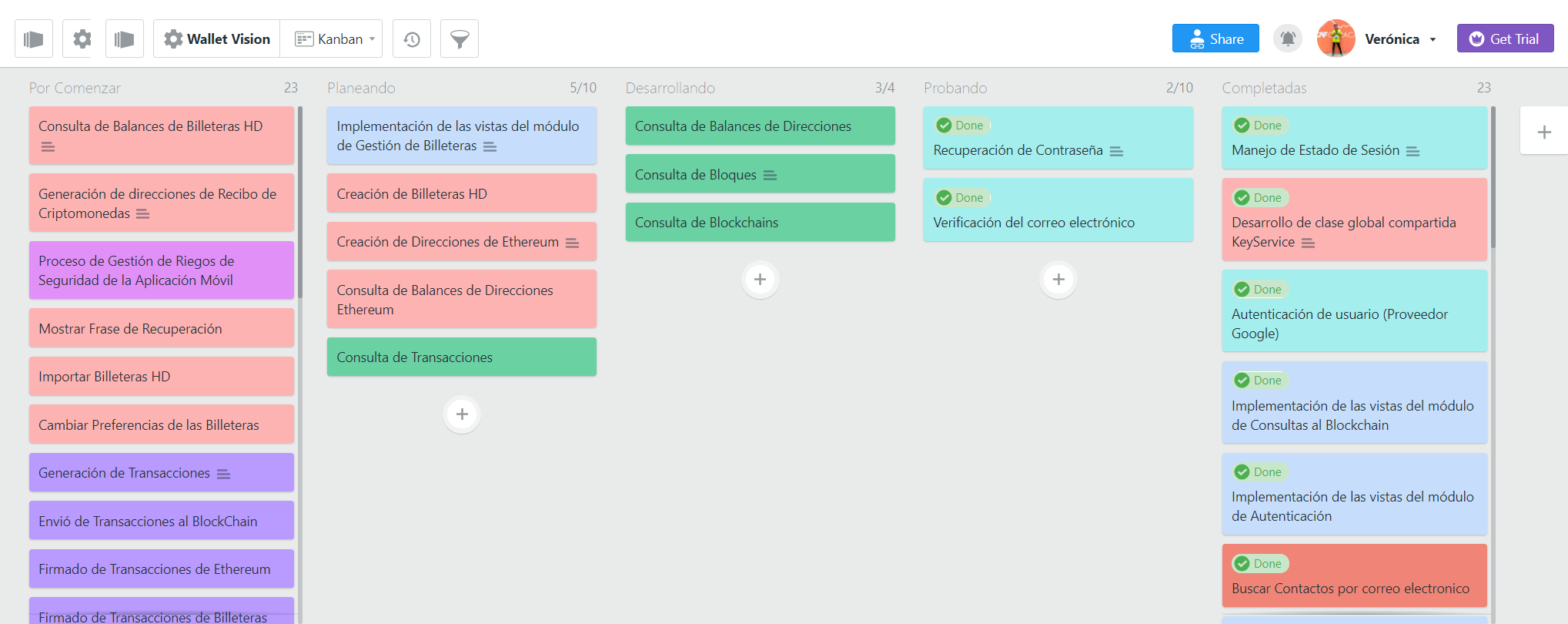
Semana 4



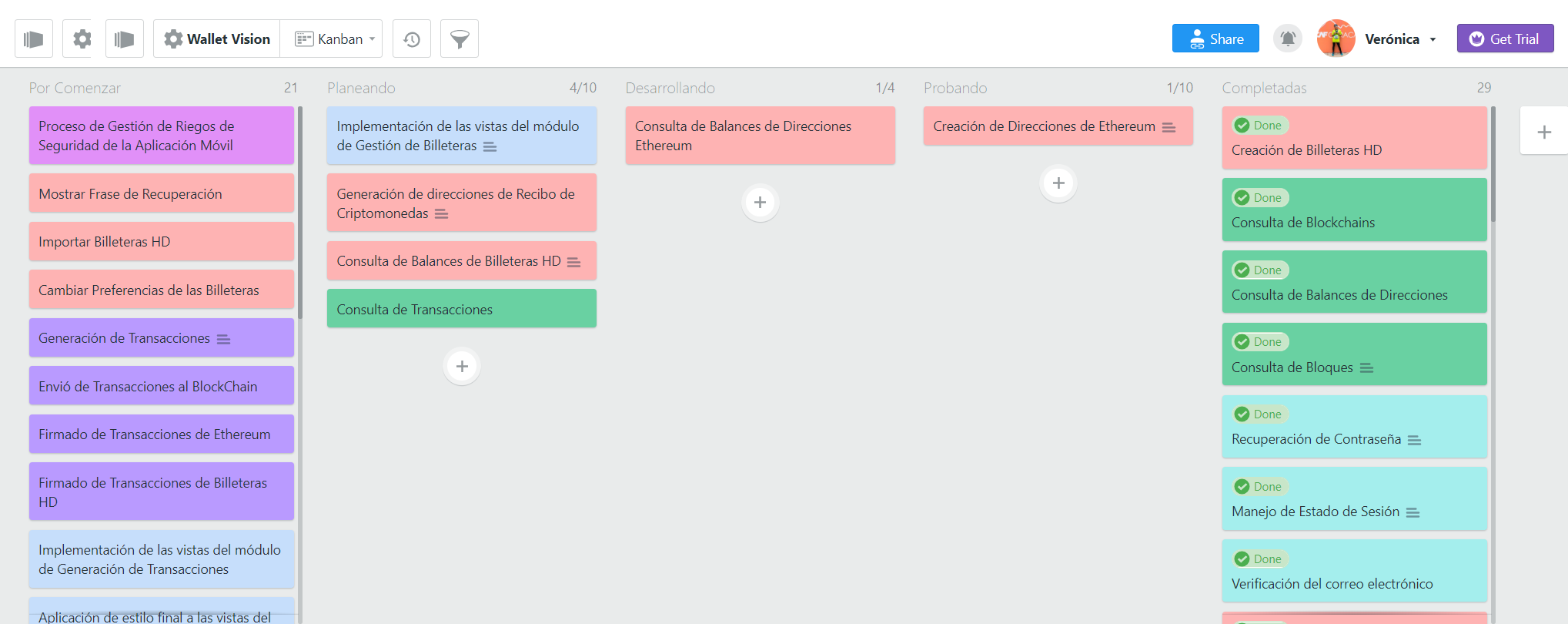
Semana 5



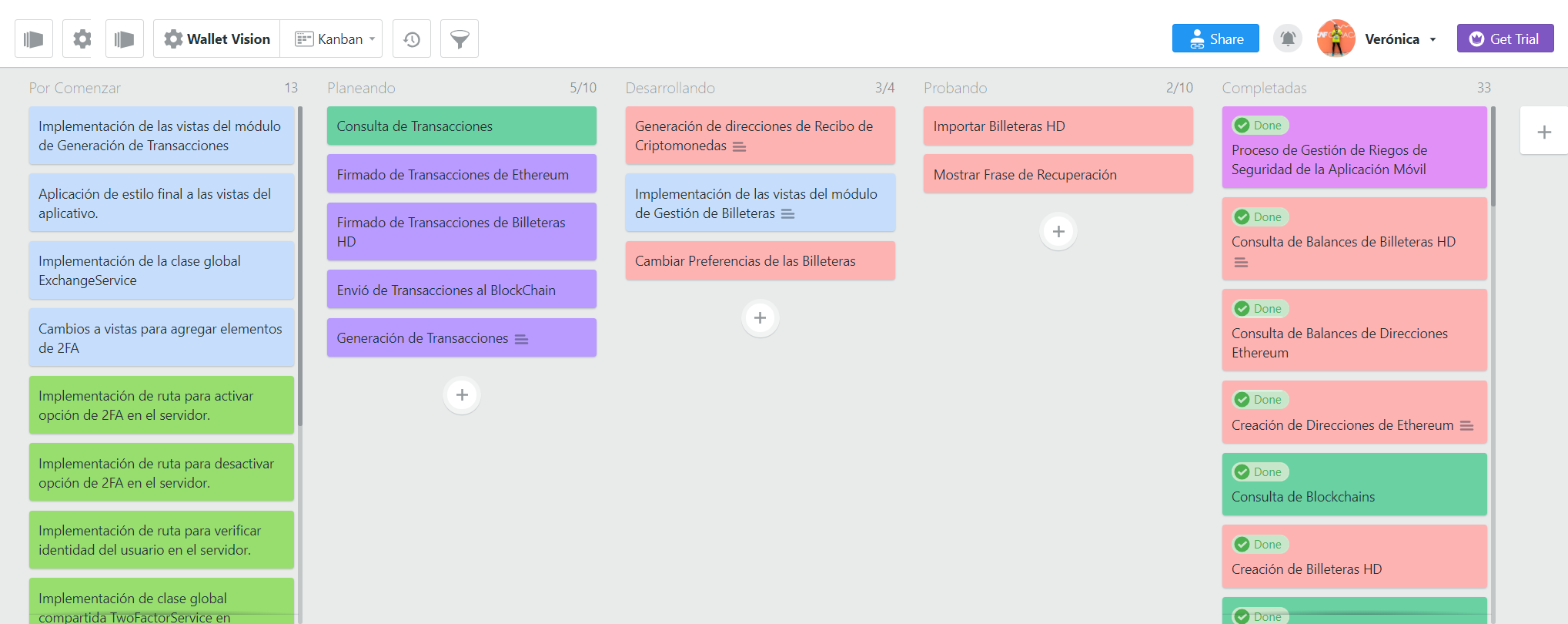
Semana 6



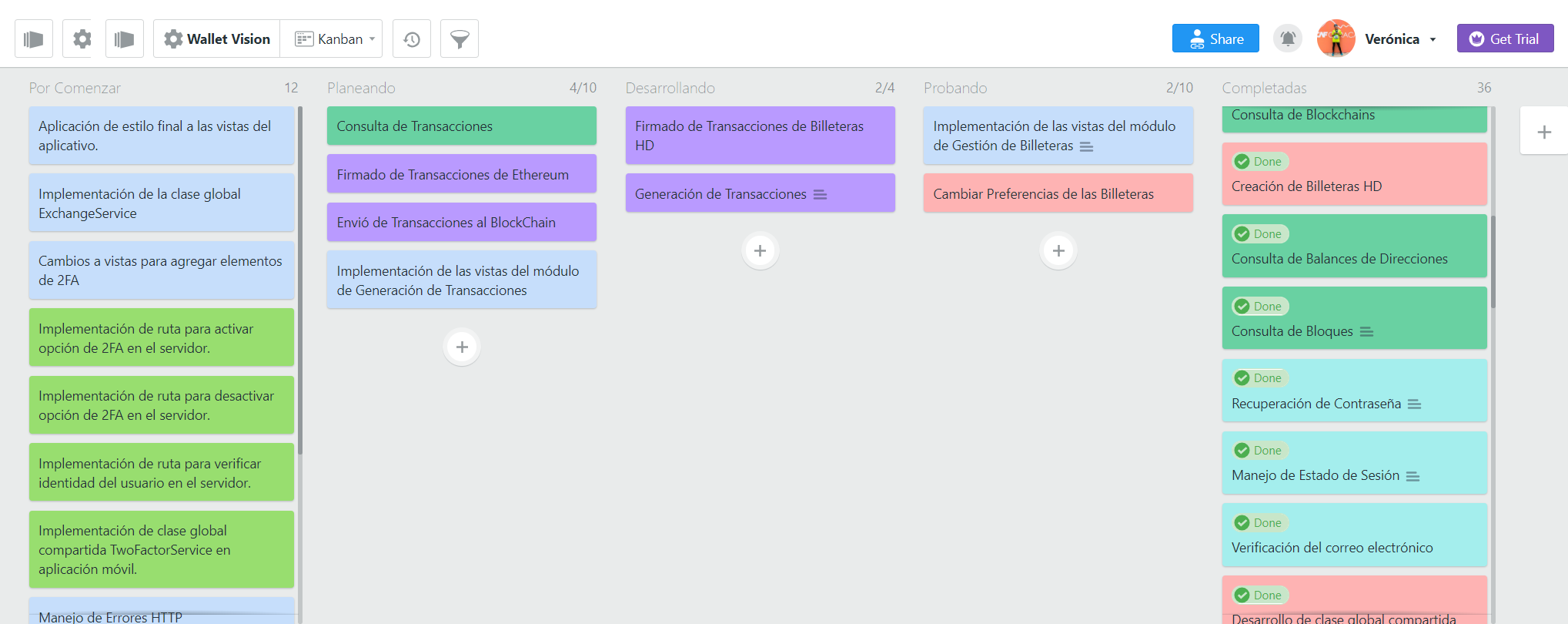
Semana 7



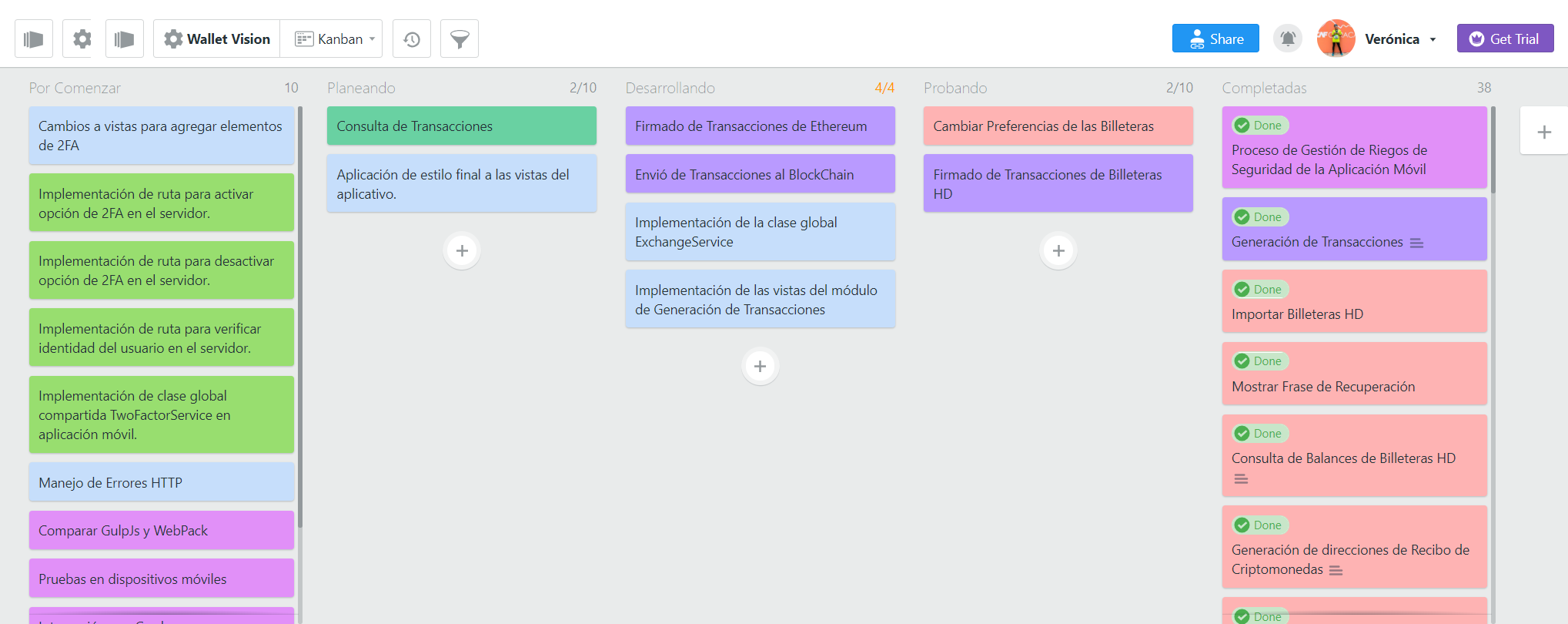
Semana 8



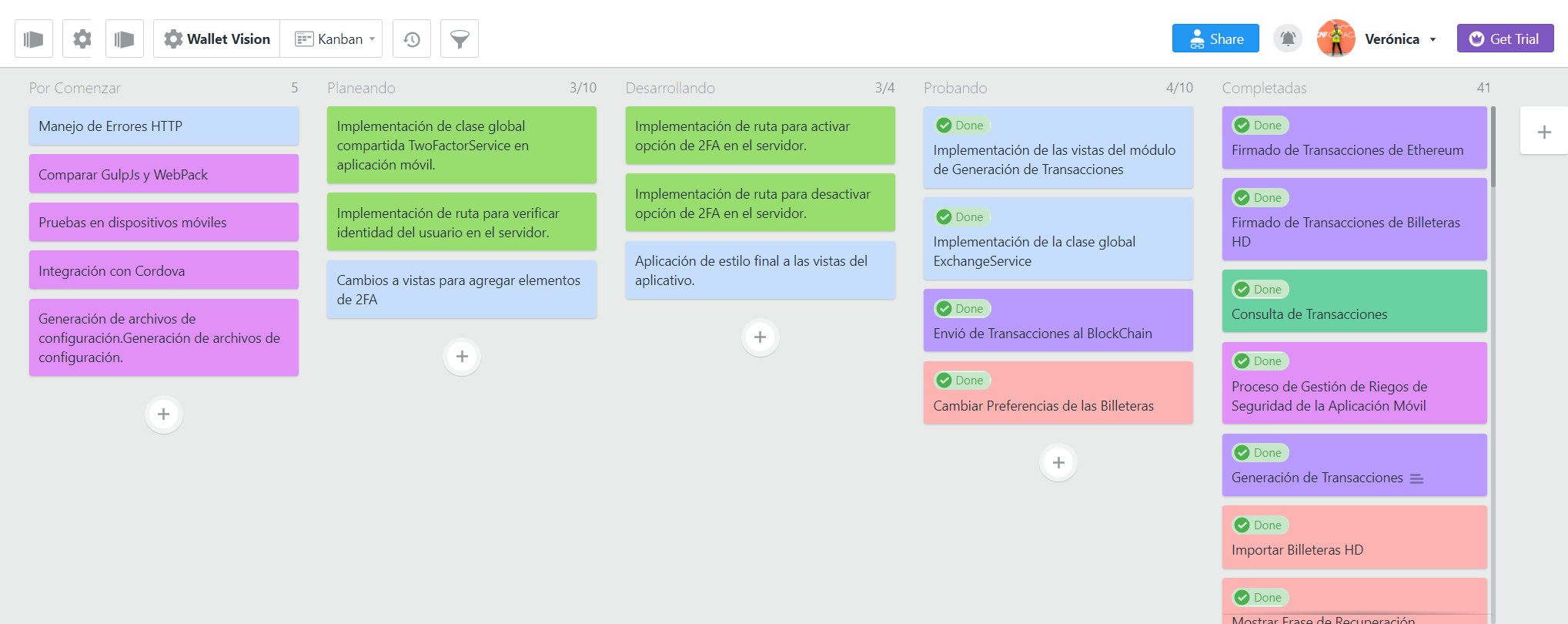
Semana 9



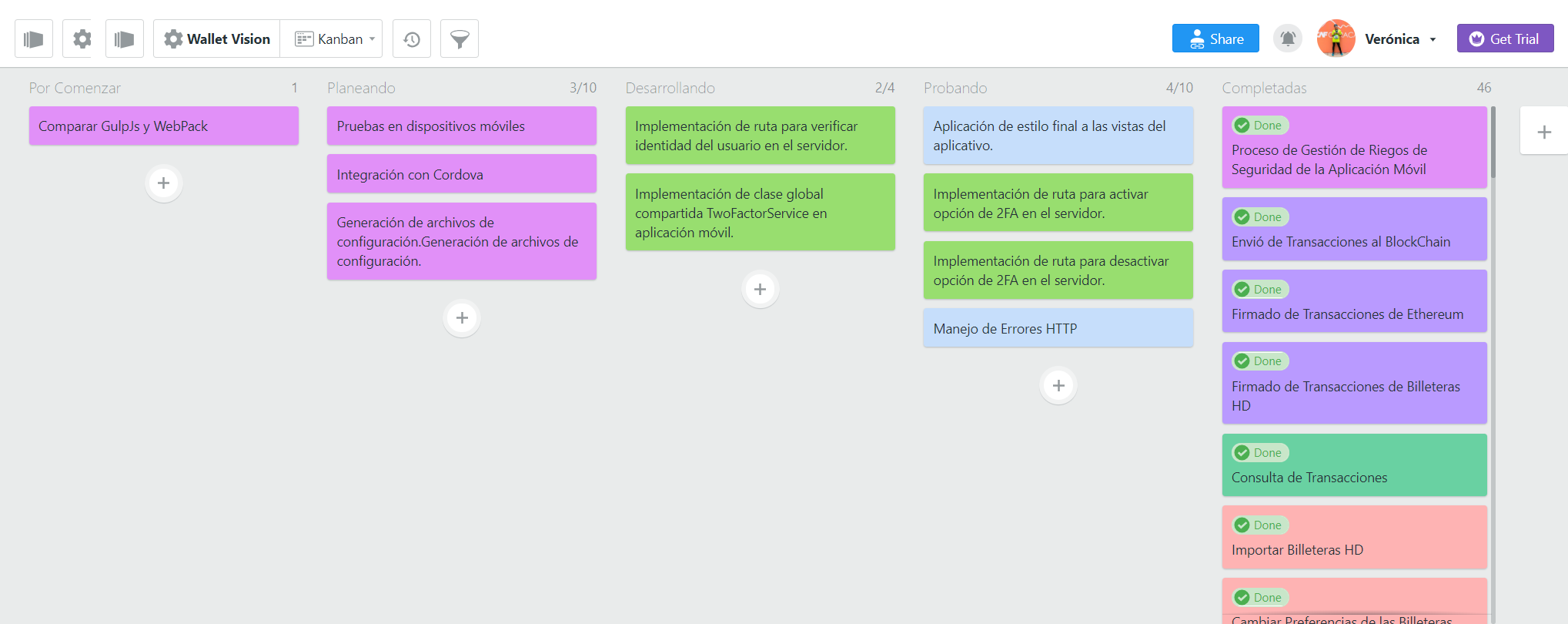
Semana 10



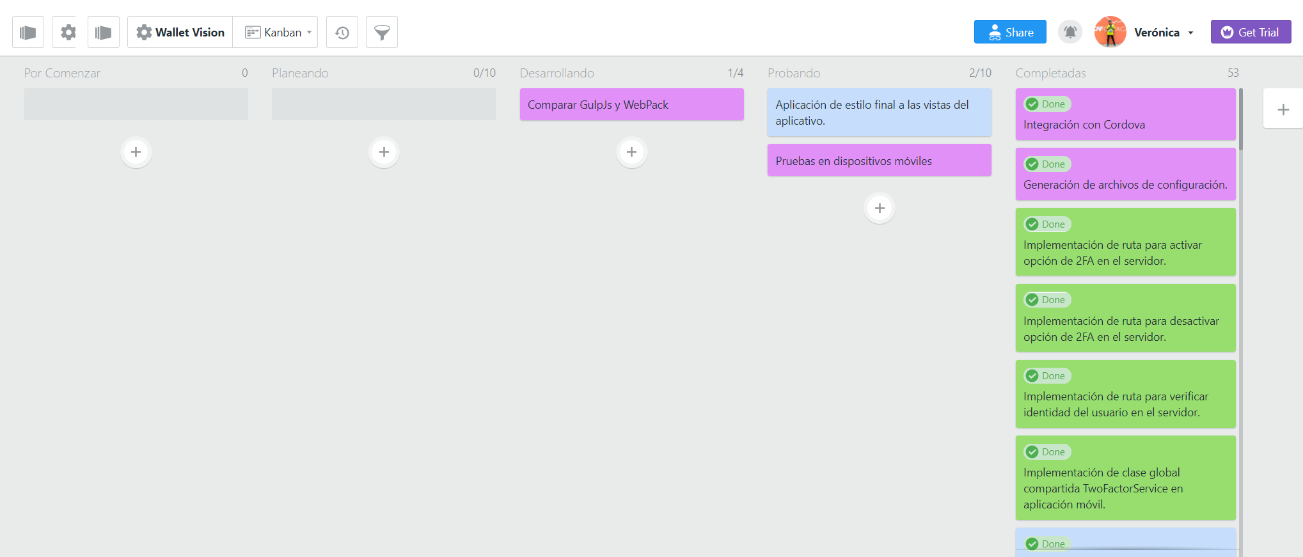
Semana 11



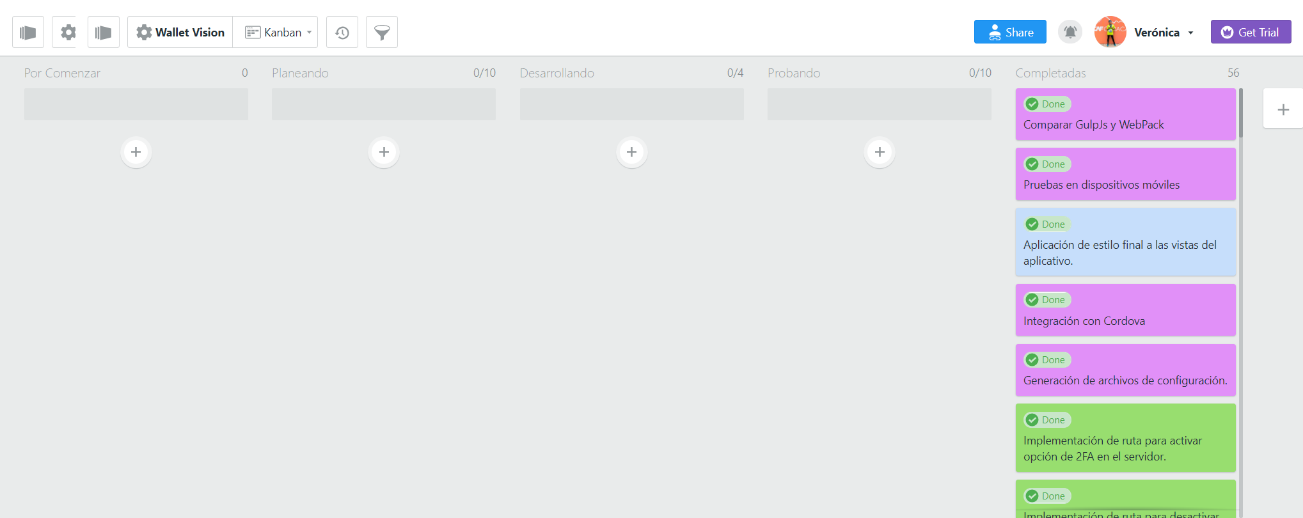
Semana 12



Semana 13



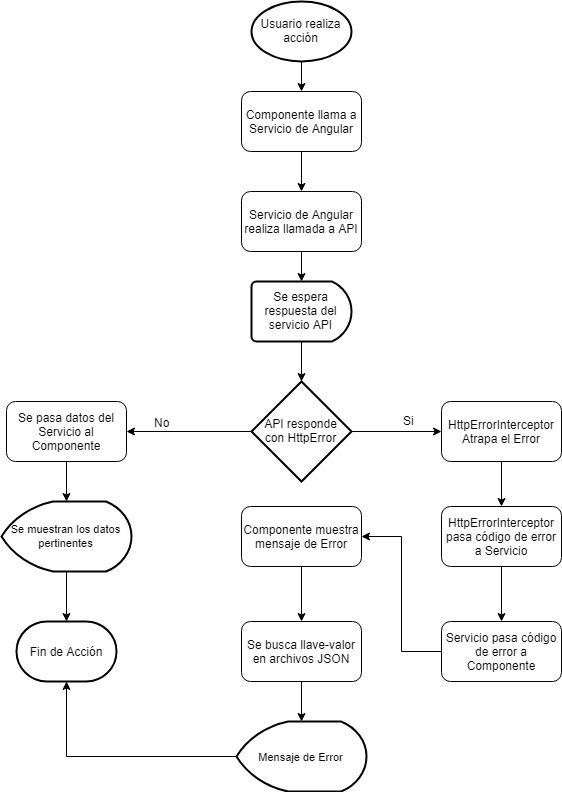
Semana 14



**Anexo 7: Clases y Servicios creados durante la adaptación de las vistas del FrontEnd**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | **Métodos** |
| **HttpError**  **Interceptor** | Clase que intercepta las solicitudes y respuestas a las APIS que utiliza la aplicación, con el fin de manejar adecuadamente los códigos de errores que se puedan recibir, sea por solicitudes incorrectas o factores externos (problemas con la conexión) | * intercept: función que se encarga de modificar las solicitudes a servicios REST, modificarlas si es necesario, y redirigir las respuestas del servidor que sean de tipo HttpErrorResponse al método handleError * handleError: se encarga de asignar los códigos específicos de cada API a un código interno de la aplicación, con el fin de que este se maneje adecuadamente en otros servicios o componentes. |
| **LoaderService** | Servicio de Angular inyectable, que permite a través de una serie de métodos, presentar en la interfaz un “loader” o mensaje de carga, con el fin de indicarle al usuario que la aplicación está recuperando datos o realizando una acción en el fondo. Utiliza como base el componente de carga nativo de Ionic 3. | * showLoader: muestra un componente de carga péqueña. Se pasa como parámetro el código de un mensaje a mostrar. * showSpinner: muestra un componente de carga pequeña con un “spinner” o icono giratorio. * showFullLoader: muestra un componente de pantalla completa. Se pasa como aprametro el código de un mensaje a mostrar. * dismissLoader: eliminar la instancia del componente que este activa en una vista. |
| **AlertService** | Servicio que muestra al usuario mensajes de tipo alertas, los cuales poseen un mensaje de retroalimentación a una acción que el usuario desea realizar o que indican que un error ha ocurrido. Suelen ser acompañaos de cambios en la navegación de la aplicación. Utiliza como base el componente de alertas nativo de Ionic 3. | * showError: muestra una alerta con un mensaje de error y un botón para continuar. Acepta como parámetro un código del error a mostrar. * showAlert: muestra una alerta con un título y un mensaje, y un botón para continuar. Acepta como parámetros un código para el título y uno del mensaje a mostrar. * showFullError: equivalente al método showErorr, pero oculta toda la pantalla. * showFullAlert: equivalente al método showAlert, pero oculta toda la pantalla. |
| **ActivityService** | Servicio que maneja mostrar notificaciones al usuario relacionada con actividad registrada en la aplicación. Este servicio se encuentra desarrollado para aceptar objetos de tipo Activityby mostrar la información de manera atractiva. Las notificaciones |  |

**Anexo 8: Manejo de Errores dentro de la aplicación móvil**



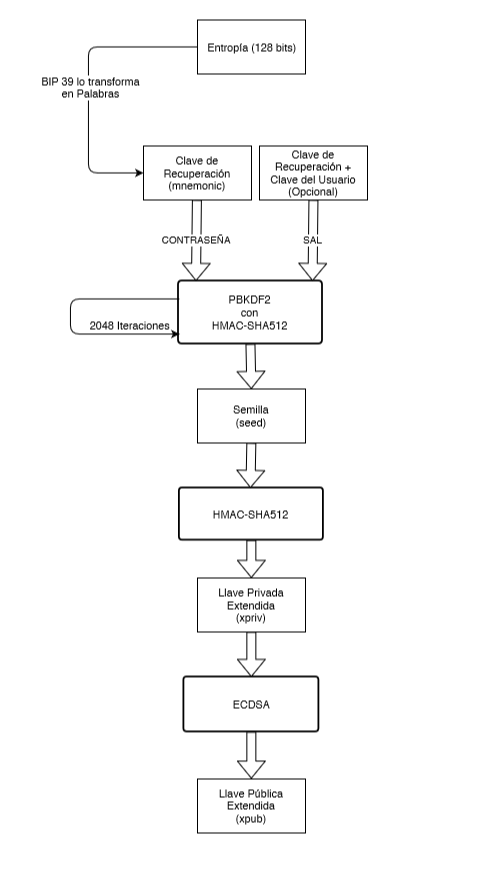
**Anexo 9: Estructuras de Datos de la aplicación móvil: Modelos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Atributos** | **Opcional** | **Métodos** |
| **Activity** | Modelo de datos que almacena las actividades del usuario | * key: String * name: String * date: String * description: Any (Objeto o String) | * Si * No * No * Si | * constructor * setName * getDate |
| **Address** | Modelo de datos que almacena los contactos de la libreta de direcciones. | * key: String * email: String * alias: String * uid: String * img: String | * Si * No * No * Si * Si | * constructor * setUID |
| **Crypto** | Modelo de datos que almacena la información de la criptomoneda de una billetera, y la preferencia del usuario | * name: String * value: String * difference: Number * coin: String * units: Objeto:   { name: String,  exchange: Number } | * No * No * No * No * Si | * constructor * setUnits * getUnits |
| **Keys** | Modelo de datos que almacena la información criptográfica de una billetera | * mnemonics: String * seed: String * xpub: String * xpriv: String * passphrase: String | * No * No * No * No * Si | * constructor * setPassphrase |
| **Token** | Modelo de datos que almacena las preferencias del usuario respecto a | * activated: boolean * enabled: boolean | * No * No | * Constructor |
| **User** | Modelo de datos que almacena los datos del usuario que posee la sesión activa dentro de la billetera | * uid: String * email: String * emailVerified: boolean * photoURL: String * displayName: String * token: Token | * No * No * No * Si * Si * Si | * constructor * setPhotoURL * getDisplayName * createToken * getToken |
| **Wallet** | Modelo de datos que almacena los datos de una billetera perteneciente al usuario. | * key: String * name: String * keys: Keys * crypto: Crypto * address: String | * Si * No * No * No * Si | * constructor * setKeys * changeUnits |

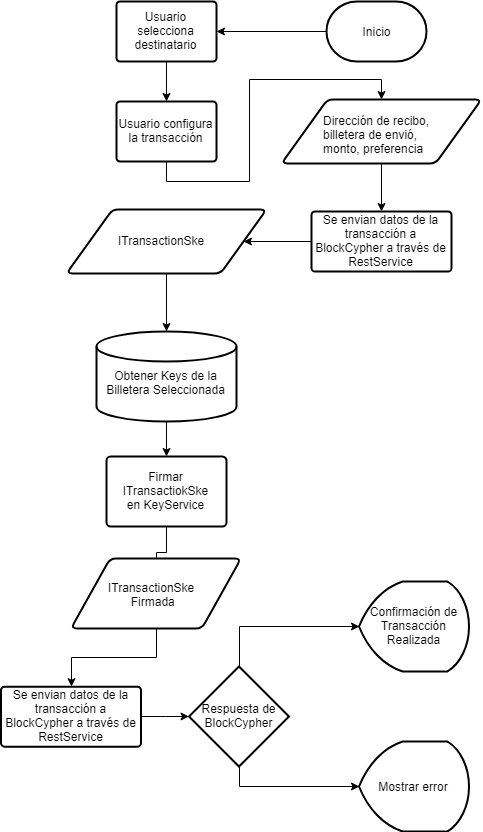
**Anexo 10: Estructuras de Datos de la aplicación móvil: Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Recurso** | **Método** |
| **IAddress** | Representa una dirección pública de un Blockchain especifico, incluye información sobre su balance y transacciones realizadas. | https://api.blockcypher.com/v1/$MONEDA/$CADENA/addrs/$ADDRESS  https://api.blockcypher.com/v1/$MONEDA/$CADENA/addrs/$ADDRESS/full | GET  GET |
| **IBalance** | Versión alternativa de IAddress con menos atributos, ya que no incluye datos de transacción. | https://api.blockcypher.com/v1/$MONEDA/$CADENA/addrs/$ADDRESS/balance | GET |
| **IBlock** | Objeto que muestra los datos de un bloque particular del Blockchain, tal como el número de transacciones y el volumen en criptomonedas. | https://api.blockcypher.com/v1/$MONEDA/$CADENA/blocks/$HASH  https://api.blockcypher.com/v1/$MONEDA/$CADENA/blocks/$BLOCK\_HEIGHT | GET  GET |
| **IBlockchain** | Objeto que representa el estado más actualizado de un Blockchain particular. Muestra información general, referente a el tamaño, última actualización y el hash del bloque más nuevo. | https://api.blockcypher.com/v1/$MONEDA/$CADENA/ | GET |
| **IExchange** | Objeto utilizado para obtener la tasa de cambio entre criptomonedas y otras monedas. | min-api.cryptocompare.com/data/  price?fsys=$CRYPTOCOINLIST&tsym=$CURRENCYLIST | GET |
| **IHDAddress** | Estructura que almacena la información de una dirección especifica de una Billetera HD, incluyendo su profundidad. | https://api.blockcypher.com/v1/$MONEDA/$CADENA/wallets/hd/$NAME/addresses/derive | POST |
| **IHDChain** | Estructura que almacena la información de las cadenas y sus respectivas direcciones de una Billetera HD. Por defecto se utiliza una sola cadena. | Objeto anidado dentro de IHDWallet. |  |
| **IHDWallet** | Objeto que almacena la información sobre una billetera HD, incluyendo la profundidad actual de la cadena de derivación. | https://api.blockcypher.com/v1/$MONEDA/$CADENA/wallets  /hd  https://api.blockcypher.com/v1/$MONEDA/$CADENA/wallets  /hd/$NAME  https://api.blockcypher.com/v1/$MONEDA/$CADENA/wallets  /hd/$NAME/addresses | GET / POST  GET /  GET |
| **ITInput** | Una entrada consumida dentro de una transacción, es decir, información referente a la dirección de la cual originan los fondos. | Objeto típicamente anidado dentro de otras respuestas, particularmente ITransaction y ITransactionSke. | GET |
| **ITOutput** | Representa la salida creada por una transacción, incluyendo monto en criptomonedas y direcciones de recibo. | Objeto típicamente anidado dentro de otras respuestas, particularmente ITransaction y ITransactionSke. | GET |
| **ITransaction** | Objeto con la información correspondiente a una transacción, independientemente del estado de confirmación. | https://api.blockcypher.com/v1/$MONEDA/$CADENA/txs  https://api.blockcypher.com/v1/$MONEDA/$CADENA/txs/$TX\_HASH | GET  GET |
| **ITransaction**  **Ske** | Estructura utilizada para la creación y envió de nuevas transacciones, incluyendo la data a ser firmada, la cual se encuentra almacenada en un atributo llamado tosign. | https://api.blockcypher.com/v1/$MONEDA/$CADENA/txs/new  https://api.blockcypher.com/v1/$MONEDA/$CADENA/txs/send | POST  POST |

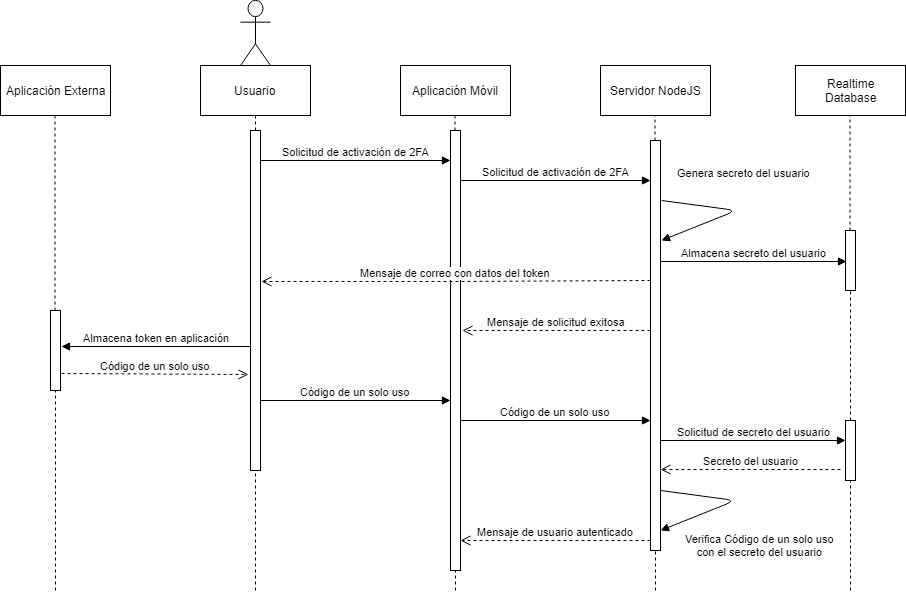
**Anexo 11: Generación de la información criptográfica de una Billetera HD**



**Anexo 12: Proceso de creación y envió de una transacción de criptomonedas.**



**Anexo 13: Diagrama de Secuencia de la Activación del Segundo Factor de Autentificación**



**Anexo 14: Pantallas de la Aplicación Móvil**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| *Figura 14.1 Pantalla de Inicio de Sesión* | *Figura 14.2 Pantalla de Registro* | *Figura 14.3 Pantalla de Confirmar Correo* |
|  |  |  |
| *Figura 14.4 Pantalla de Inicio* | *Figura 14.5 Pantalla de Menú Lateral* | *Pantalla 14.6 Pantalla de Configuración de la Cuenta* |
|  |  |  |
| *Pantalla 14.7 Configuración de Seguridad* | *Pantalla 14.8 Pantalla de Configuración de Billeteras* | *Pantalla 14.9 Pantalla de Libreta de Contactos* |
|  |  |  |
| *Pantalla 14.10 Pantalla de Agregar Contacto* | *Pantalla 14.11 Pantalla de Transacciones* | *Pantalla 14.12 Pantalla de Detalles de Transacción* |
|  |  |  |
| *Pantalla 14.13 Pantalla de Actividad* | *Pantalla 14.14 Pantalla de Explorador* | *Pantalla 14.15 Pantalla de Recibo* |
|  |  |  |
| *Pantalla 14.16 Pantalla de Envió* | *Pantalla 14.17 Pantalla de Confirmación de Envió* | *Pantalla 14.18 Pantalla de Creación de Billeteras* |
|  |  |  |
| *Pantalla 14.19 Pantalla para mostrar Frase de Recuperación* |  |  |

**Anexo 15:** **Reglas de Seguridad establecidas en Firebase Realtime Database**

