|  |
| --- |
| **Prueba de Concepto:**  **Conversión a Token el dinero fiduciario** |
|  |
|  |

CONTROL DE MODIFICACIONES POR CAMBIO DE VERSIONES

|  |  |  |  |
| --- | --- | --- | --- |
| VERSIÓN | PARTES QUE  CAMBIAN | DESCRIPCIÓN DEL CAMBIO | FECHA DE CAMBIO |
| 1.0 | N/A | Documento original | 13/08/2021 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

ÍNDICE

[1. INTRODUCCIÓN 4](#_Toc79750366)

[1.1. Objeto del documento 4](#_Toc79750367)

[1.2. Alcance DEL CAMBIO 4](#_Toc79750368)

[2. Alcance del cambio 4](#_Toc79750369)

[2.1. CAMBIOS EN LA PLATAFORMA 4](#_Toc79750370)

[2.2. ELEMENTOS AFECTADOS 7](#_Toc79750371)

[2.3. ESTIMACIÓN JORNADAS 7](#_Toc79750372)

# INTRODUCCIÓN

## Objeto del documento

El presente documento recoge el alcance funcional, los aspectos y requerimiento básicos para el equipo, así como la estimación de tiempo.

## Alcance DEL CAMBIO

Se necesita crear un software que permita tokenizar el dinero fiduciario de los usuarios a través de una aplicación móvil. Este software debe cumplir con los estándares de calidad, seguridad, escalabilidad, así como tener un riguroso cumplimiento de la normativa legal, establecida hasta el momento.

# Alcance del cambio

## CAMBIOS EN LA PLATAFORMA

## FRONT END:

* **Interfaz de Usuario: Se solicita una la aplicación para dispositivos móviles soportada, tanto en Android como iOS:**
  + Se estudia la viabilidad de la interfaz móvil con los siguientes Framework: Angular y REAC.

- Finalmente se decide REAC por los siguientes puntos:

* + - Trabaja con datos inmutables: la aplicación en Back End va a estar enfocada en Smart Contract, por lo tanto, se entiende que se ajusta con la arquitectura
    - Permite MVVM: Lo que permitirá, separar la interfaz de la lógica de la aplicación de la interfaz de usuario, por lo que cumpliría con el requerimiento sobre que el software que se crease debe ser lo mas reutilizable posible.
    - Es válido tanto para Android como para iOS.
    - Gracias a que tiene licencia MIT, rebaja los costes del software.
  + Ventanas:
    - Entre las ventanas a desarrollar se encuentran: la ventana de login, creación de usuario, ventanas operaciones y ventanas necesarias para el KYC

## BACK END:

* **Arquitectura y lógica de negocio: Los requisitos es que sean usable, escalable, segura, auditable y reutilizable.** 
  + La **arquitectura** que se va a utilizar es una arquitectura de **microservicios**:
    - Permite escalabilidad
    - Los microservicios pueden ser reutilizados por otros softwares de la empresa.
    - Es una arquitectura versátil en cuanto al diferentes usos de tecnologías.
    - Mejor control de las tareas de mantenimiento
  + La **tecnología** elegida es **Blockchain**, es una tecnología robusta para trabajar con las librerías de Etherium, así mismo, con componentes externos necesarios para llevar a cabo el proceso de conversión a Token la moneda fiduciaria. El lenguaje de programación elegido será **Solidity.**
  + En cuanto a los componentes externos se debe crear la integración necesaria para trabajar con los datos necesarios de **INVERSIS** o con los **procesos de identificación de identidad gubernamental**.
  + Lógica de negocio:
    - Será necesario desarrollar diferentes microservicios:
      * Servicio de identidades: funcionalidad de negocio basadas en el altas o modificaciones relativas a los usuarios. Así como bajas en el sistema.
      * Servicio KYC: Lógica necesaria para que a través de los documentos acreditativos que introduzca el desarrollo, se validen contra una unidad externa evaluadora de documentos. Así como los datos pertinentes necesarios, por ejemplo: Validez de la identificación.
      * Servicio de operaciones: Operaciones que puede realizar el operador como recarga, transferencia, retirada, compra, etc.
      * Servicio Token: Operaciones necesarias para la creación del TOKEN. Se encuentra la conexión con INVERSIS, ya que será necesario la obtención de la cantidad de dinero fiduciario para su posterior creación del token.
      * Conexión externa con Alastria: A través de Alastria se podrán realizar las operaciones monedero a través del token creado en ERC20.
  + **Pruebas y despliegue:** Como se ha explicado anteriormente se va a utilizar la tecnología Blockchain, y el lenguaje de programación Solidity, por lo que se ha elegido para la realización de pruebas, así como su despliegue, el Framework de Truffle. Nos permite la ventaja de poder probar los Smart Contract, así como automatizar los despliegues de los servicios, minimizando los riegos del factor humano.
  + La **Seguridad** como se ha mencionado anteriormente estará ligada con la validación del KYC. Además, el servicio de TOKEN, deberá tener la funcionalidad necesaria que valide en todo momento al propietario del token.

## EQUIPO:

* **Jefe de Proyecto / Arquitecto:** Será necesario la figura de la Jefatura de proyecto, sería muy recomendable que, ya que es un proyecto de corta duración, 4 meses, pudiese ser el arquitecto del aplicativo o en caso de que no, sería un recurso compartido con otros proyectos símiles.
* **Desarrolladores (Analistas programadores)** 
  + **Desarrollador Full Stack:** Deberá conocer el front, en cuanto a usabilidad y además en accesibilidad. Ya que se puede y se debe pensar en la accesibilidad de los usuarios. Así mismo, apoyará en la parte de back end.
  + **Desarrollador Back End:** Sus funciones serán el desarrollo de la lógica de negocio como las integraciones.

**Ambos deben realizar pruebas unitarias, de integración y funcionales.**

* **Seguridad y Calidad:** De forma transversal necesitaremos contar con la ayuda de un equipo / personas de seguridad y calidad. Para que auditen el proceso de desarrollo y contemplen los riesgos y soluciones del aplicativo.

## CULTURA:

* **Metodología de Proyecto:** Se ha elegido para este proyecto SCRUM.
  + Se ajusta a que el proyecto es un proyecto con un equipo pequeño
  + Las fechas de entregas son fechas acotadas: 4 meses, por lo que se pueden establecer iteraciones de 2 semanas, valorable ampliarlas a 4 semanas, pero al comienzo deben establecerse bien las bases.
  + El jefe de proyecto será el scrum master.
  + Al ser un proyecto basado en microservicios, las personas que trabajen en él pueden trabajar con bastante autonomía y adquirir en los desarrollos que realice la responsabilidad necesaria.
  + Herramientas: Se puede utilizar Trello, es gratuito y puede contener diferentes tableros con los sprints y funcionalidades.
* **Cultura:** Se debe pensar que todos somos un reloj, y que el reloj mueve sus manillas de forma sincronizada, porque todas las partes tienen el mismo objetivo. No hay ninguna pieza que sea mas o menos importante, porque si una se rompe, sea cual sea, el reloj deja de funcionar.