PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

IIC2113 - Diseño Detallado de Software (2020-2)

# Proyecto Semestral - Rúbrica Entrega 1

## Objetivos

- Analizar un problema medianamente complejo enfocado a la creación de un producto tecnológico
- Modelar la solución del problema identificado y utilizar diagramas para comunicar la solución planteada
- Aterrizar requerimientos funcionales a una primera aproximación de la modelación

## Plazo de entrega: 4 de Septiembre a las 23:59.

## Descripción

En los grupos asignados con el tema asignado, deberá realizar un informe o documentación de la solución que seguirá.

El equipo deberá describir la solución en cuanto a:

- Su arquitectura, se recomienda seguir una diagramación UML. Su solución debe estar basada principalmente en una aplicación web en Ruby on Rails.
- Su estructura y componentes. Para lo anterior debe utilizar la notación de un diagrama UML de componentes o de clases (a libre elección cuál diagrama usar).
- Al menos dos wireframes, enfocado al usuario cliente de la aplicación.

Además deberá comunicar cualquier otra decisión que vayan a tomar.

**Entregable**:

- Informe con los diagramas correspondientes en la carpeta docs de su repositorio. La carpeta docs se encuentra en la siguiente ubicación (basándonos en el repo-base): https://github.com/IIC2113-2020-2/repo-base/tree/master/docs

**Rúbrica:**

| RUBRICA E1 | NIVELES DE DESEMPEÑO |
|---|---|
| El informe entregado resuelve la problemática del tema asignado. | 3 puntos si la solución propuesta resuelve el problema<br>1 puntos si la solución deja fuera algunas funcionalidades del problema<br>0 puntos si su solución no resuelve el problema principal |
| El apoyo visual entregado aporta a la solución descrita. | 7 puntos si el diseño de la arquitectura, los componentes/objetos y los wireframes son legibles y hacen sentido con la solución<br>5 puntos si uno de los diseños no se entiende<br>3 puntos si dos de los diseños no se entienden<br>0 puntos si ninguno de los diseños se entiende |
| El informe tiene un lenguaje apropiado y está lógicamente ordenado tal que facilita la lectura. | 1 punto si cumple<br>0 puntos si no cumple |

Rúbrica entrega 1.

# ANEXO - TEMAS

## Tema 1: Exchange de Bitfake

El Bitfake (BTF[1]) es una *criptomoneda* que ha ganado popularidad en el último tiempo. Debido a su popularidad, todos quieren tener un exchange de BTF-CLP.

Funcionalidades esperadas para Entrega 2:

- Debe permitir el registro y login de usuarios.
- Un usuario debe tener una cuenta.
    - Debe ver cuánto BTF y CLP tiene.
    - Puede ver sus transacciones hechas (compra y venta de BTF).
- Por simplicidad en esta entrega, todos los usuarios parten con un valor aleatorio entre los 10.000 CLP y los 100.000 CLP.
- Por simplicidad el valor inicial del BTF (o cualquier moneda) y la cantidad será definida por el ayudante al inicio del proyecto.
- Los usuarios podrán transar BTF, esto es:
    - Vender BTF si tienen en su cuenta. Al ejecutar una orden de venta de BTF, el usuario ingresa "cuánto BTF quiere vender" y el precio viene dado por el "precio de venta" del mercado.
    - Comprar BTF si es que tienen CLP. El ejecutar una orden de compra de BTF, el usuario ingresa "cuánto BTF quiere comprar" y el precio viene dado por el "precio de compra" del mercado.
    - Los precios de compra y venta serán rangos aleatorios cuyos mínimos y máximos iniciales serán distintos para cada grupo. Luego de cada transacción, el valor de compra y venta se debe actualizar con la siguiente lógica:
        - Si la transacción es venta, el precio del BTF de compra disminuye en 0.03 sobre el porcentaje de su valor.
        - Si la transacción es compra, el precio del BTF de venta aumenta en 0.03 sobre el porcentaje de su valor.
    - Por simplicidad, las transacciones no tienen delay.
    - Por cada transacción se debe enviar un correo al usuario.
    - Por simplicidad la cantidad de BTF es constante (ver bonus).

[1] No confundir con Bitcoin Faith.

- Debe permitir el tipo de cuenta “partner”. Este tipo de cuenta puede vender o comprar BTF sin límite (tener saldos negativos) ya que al ser partner, hacen ajustes de cuentas de forma externa (funcionalidad pensada para simplicidad en la entrega 3).
- Debe exponer una API. Cada usuario puede generar su “token” para usar la api. Los usuarios a través de la API podrán:
    - Obtener los precios actuales de compra y venta en el exchange.
    - Vender BTF (ejecutar una orden de compra).
    - Comprar BTF (ejecutar una orden de venta).
    - Conocer el balance actual de mi cuenta (cuanto BTH y CLP tengo).

**Bonus**: Registrar una segunda moneda: Badtherium (BTH[2]), que permite compra y venta de BTH-CLP.

Funcionalidades esperadas para Entrega 3:

- Los usuarios parten con 0 CLP.
- Para cargar CLP, los usuarios deben hacer una transferencia a una de las cuentas de Mercado Bitfake en los Bancos de Inversiones asignados.
    - Deben conectarse a la api de cada banco y revisar su cuenta empresa para ver si hay nuevas transacciones asociadas al RUT de ese usuario. En caso de existir una transacción (no registrada con anterioridad) debe contabilizar el depósito, aumentando así la cantidad de CLP en su cuenta.
- Cada 3 horas, el precio de venta o compra se debe disparar o caer de forma aleatoria.

Ejemplo de una api de un exchange: https://api.buda.com/

[2] No confundir con Bithereum.

## Tema 2: Banco de Inversiones

Se te pide crear la aplicación de un banco que además posee un fondo de inversión riesgoso.

Funcionalidades esperadas para Entrega 2:

- Debe permitir el registro y login de usuarios.
- Un usuario debe tener una cuenta.
    - Puede ver cuánto CLP tiene en su cuenta y cuánto tiene en Ahorros.
    - Puede ver las transacciones hechas (transferencias, depósitos o ahorros).
- Por simplicidad en esta entrega, todos los usuarios parten con un valor aleatorio entre los 10.000 CLP y los 100.000 CLP.
- Los usuarios podrán transferir dinero entre cuentas del mismo banco. Para transferir dinero debe conocer el número de cuenta del receptor. Así mismo al hacer una transferencia, el otro usuario recibirá un depósito.
    - Para poder efectuar la transferencia, la página debe validar que el usuario ingrese el mismo código que le llegará a su correo al momento de solicitar transferir.
    - Mientras no ingrese el código, la transferencia no se realizará.
- El banco tiene un módulo de inversiones. El módulo de inversiones es un fondo de inversión riesgoso que el banco usa para invertir y así generar ganancias (o pérdidas) proporcionales para todos sus usuarios dependiendo del monto que ha ahorrado.
    - Un usuario puede enviar dinero desde su cuenta al fondo de inversiones. Se registra este valor como ahorro.
    - El banco usa este fondo para invertir. Para efectos de esta entrega, y por simplicidad, este “invertirá 60 minutos”. Esto se traduce en que el fondo crecerá o disminuirá en un factor aleatorio entre [-0.03, +0,03] porcentaje del valor.
    - Si el fondo gana o pierde, se deberá ver reflejado no solo en el fondo total, si no que también en las cuentas de ahorro individuales de cada usuario.
- Debe exponer una API. Cada usuario puede generar su “token” para usar la api. Los usuarios a través de la API podrán:
    - Obtener las transacciones de la cuenta del usuario.
        - Todas.
        - En un período de tiempo.

**Bonus**: Registrar un segundo fondo de inversión. Los usuarios pueden destinar sus ahorros a un segundo fondo un poco más conservador, cuya lógica de inversión implica invertir solo la mitad del fondo ahorro (la otra mitad se mantiene sin invertir).

Funcionalidades esperadas para Entrega 3:

- Cada 60 minutos el fondo del banco no debe crecer ni disminuir aleatoriamente. En su lugar, debe comprar o vender BTF en un exchange asignado.
    - El banco debe tener una cuenta en el exchange.
    - Si desde la api del exchange existe la posibilidad de arbitrar, el banco debe generar una orden de compra o venta en el exchange.
    - Por simplicidad el banco podrá tener CLP negativo en su cuenta de exchange para poder comprar o vender BTF (es una cuenta partner en el exchange).
- Para comprar debe usar el dinero que se encuentre en su fondo de inversión riesgoso.

Ejemplo de una api bancaría: https://fintoc.com/docs#introduccion