

Relatório técnico - Especificações do Projeto

YesFinance

Unidade Curricular: IES - Introdução à Engenharia de Software

Data: Aveiro, 2 de dezembro de 2021

Estudantes: 98474 : João António Assis Reis
97631 : Nuno Santos Falha
97673 : Pedro Daniel Lopes Duarte
98388 : Ricardo Manuel Batista Rodriguez

Abstrato do projeto: YesFinace é uma aplicação web com o intuito de tornar o mundo de investimentos em criptomoedas mais intuitivo. Para tal, enquadra-se como um simulador de compra e venda de moedas virtuais.

Tabela de conteúdos:

[1 Introdução](#)

[2 Conceito do Produto](#)

[Visão do produto](#)

[Personas](#)

[Cenários Principais](#)

[3 Caderno de arquitetura](#)

[Requisitos chave e limitações](#)

[Vista arquitetónica](#)

[Interações entre módulos](#)

[4 Informação perspetiva](#)

[5 Referências e Recursos](#)

1 Introdução

Nesta primeira iteração são analisados os requisitos do cliente para o desenvolvimento da aplicação.

A aplicação, YesFinance, será então um aplicação web com o objetivo de acompanhar, comparar e investir na evolução dos valores de mercado das várias moedas virtuais existentes.

O conceito da aplicação será explorado através da criação de user stories e da análise de use cases. No final, é abordada uma visão da arquitetura e da estrutura da base de dados planeadas para o desenvolvimento deste projeto.

2 Conceito do Produto

Visão do produto

A YesFinance visa ser um simulador de compra e venda de moedas virtuais, recorrendo a um acompanhamento rigoroso da evolução dos valores das mesmas em tempo real.

Cada utilizador pode ter vários portfólios. Estes funcionam como uma carteira de moedas, onde o utilizador terá acesso a estatísticas sobre as criptomoedas que lhe interessam, por exemplo, gráficos das variações dos valores de mercado. Poderá então comprar ou vender as suas moedas virtuais, mediante o seu interesse e estudo sobre os seus valores, e, então, obter lucro com a transação.

Nesta aplicação será possível adicionar extensões programáveis, onde indicar-se-á quando se quer investir, por exemplo, cada vez que o gráfico ultrapassar x valor, o cliente vende as suas criptomoedas automaticamente.

O impacto pretendido para este sistema é facilitar a entrada no investimento neste mercado às pessoas mais inexperientes, tornando-a mais intuitiva, não sendo necessário um conhecimento profundo nesta área para iniciar uma vida de investimentos. No entanto, a aplicação não é exclusivamente pensada para os amadores, está também desenvolvida para oferecer uma experiência mais organizada e personalizável aos utilizadores que já dominam a área, através de portfólios e extensões.

Esta aplicação destina-se, portanto, a todos aqueles que pretendem fazer uma análise mais profunda e investir no mercado das criptomoedas, ou até, simplesmente, acompanhar os valores das moedas virtuais.

Personas

1. Duarte Gafas



No auge da sua carreira como investidor em ações e análise de mercado, aos 43 anos Gafas decide investigar sobre investimentos em criptomoeda devido ao seu crescente nível de popularidade. Está atualmente a fazer sondagens e a recolher informação sobre este mercado de forma a ter mais conhecimento para se juntar aos outros investidores em criptomoedas.

MOTIVAÇÃO: Acompanhar e evolução do mercado de criptomoeda para quando se sentir confiante investir

2. Teresa Regal



Como experiente programadora e investidora em todo o tipo de mercados, Teresa trabalha como gestora dos investimentos dos seus clientes, tendo como objetivo, obviamente, retornar o máximo de lucro. Apesar de usar plataformas e até mesmo moedas diferentes, para fins diferentes, um reservado grupo dos seus clientes faz questão de usar YesFinance como meio de investimento.

O processo de investimento para múltiplas pessoas torna-se bastante confuso para alguém com as tarefas da Teresa, pois tem de acompanhar todos os processos em simultâneo.

MOTIVAÇÃO: Organizar os investimentos de cada cliente em portfólios e automatizar os processos de compra/venda, à medida de cada cliente

3. Helena Ramos



Em paralelo à sua profissão de fotógrafo para uma revista local, Helena procura aumentar o seu vencimento mensal. Para isso, escolheu investimento em criptomoeda, através de um elevado número de transações de baixo valor. Assim, vai acompanhando as flutuações do mercado e simultaneamente obtendo uma margem de lucro. Tem como objetivo seguir carreira no estrangeiro, para atingir esse objetivo ela achou por bem investir de forma a ter dinheiro suficiente de lucro para se estabelecer e recomeçar a sua vida noutro país.

MOTIVAÇÃO: Comprar criptomoeda e eventualmente vender quando atingir a sua margem de lucro desejada

Cenários Principais

- Helena começa por adquirir uma certa quantidade de moeda:
 - abre a app, faz login, acede à tab de mercados, escolhe o mercado desejado, escolhe o que a aprouver, cria um pedido, escolhe a quantidade da moeda e o preço a que quer comprar
- Helena quer vender uma certa quantidade de moeda:
 - abre a app, faz login, acede à tab de mercados, escolhe o mercado desejado, escolhe o que a aprouver, cria um pedido, escolhe a quantidade da moeda e o preço a que quer vender
- Teresa verifica o estado do investimento de um dado cliente:
 - abre a app, faz login, escolhe o portfólio correspondente ao cliente, analisa os dados
- Duarte analisa os mercados para trocas de uma moeda:
 - acede ao menu de moedas, escolhe a moeda, escolhe um dos mercados onde ela se encontra, analisa os dados
- Teresa cria uma extensão que compra e vende automaticamente a moeda:
 - abre a app, faz login, acede ao menu de registo de extensões, efetua o registo, cria um portfólio, instala a extensão nesse portfólio

3 Caderno de arquitetura

Requisitos chave e limitações

Existem alguns requisitos-chave e limitações do sistema que têm uma influência significativa na arquitetura. Eles são:

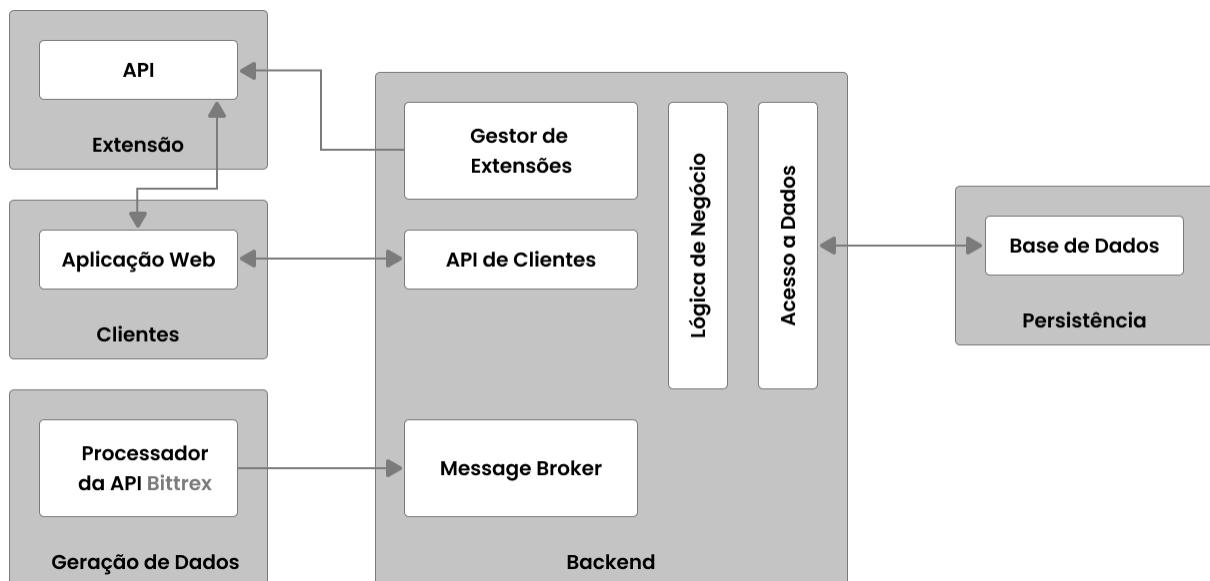
- Os utilizadores devem conseguir aceder à aplicação web através de um computador, dispositivo móvel ou outro aparelho tecnológico com acesso à internet.
- A aplicação web deve estar constantemente disponível e responsiva, uma vez que é um ponto crítico para os utilizadores.
- O sistema deve ser escalável horizontalmente de forma a suportar vários utilizadores simultâneos
- A aplicação deve ser capaz de receber um grande fluxo de dados a cada segundo, vindos da API Bittrex
- Um utilizador sem login só tem acesso à leitura de informações sobre uma moeda, o mercado da mesma e os seus detalhes
- Um utilizador com login consegue ler informações sobre uma moeda, o mercado da mesma e os seus detalhes, criar/atualizar/remover ordens, ver ordens de compra e venda, pode criar/atualizar/remover portfólios.
- Os dados confidenciais do utilizador devem ser encriptados, para proteção do mesmo.
- Um utilizador com login pode optar por utilizar uma extensão providenciada pela aplicação web, bem como cancelar o uso da mesma a qualquer altura
- Os dados recebidos pela Bittrex API devem ser rapidamente introduzidos na base de dados
- Os dados apresentados na aplicação web devem ser rapidamente atualizados sempre que sejam inseridos novos dados na base de dados
- A realização das transações entre ordens deve ser rápida

Vista arquitetónica

O sistema é composto por 5 componentes:

- A **fonte de dados** é um programa simples que receberá dados da API [Bittrex](#) com websockets e os vai enviar para o message broker do backend;
- O **backend** é composto por:
 - uma **API** que vai ser desenvolvida em Spring Boot, é responsável por fornecer dados aos clientes;
 - um **Message Broker**, RabbitMQ, que vai guardar os dados da fonte de dados até que sejam consumidos pela aplicação, estes serão posteriormente

- guardados ou atualizados na base de dados eos clientes e extensões interessadas são notificados;
- um **gestor de extensões** é uma interface para gerir as extensões existentes, a informação que precisam, quando a precisam e os utilizadores que a estão a utilizar. Permite também fazer pedidos ás extensões (REST APIs) com atualizações de informação de forma a receber instruções de atuação.
- As **extensões externas** são REST APIs criadas pela comunidade que podem ser instaladas por utilizadores e automatizar tarefas ou acrescentar/modificar funcionalidades;
 - A **persistência** será uma base de dados MySQL que irá guardar os dados críticos;
 - A **aplicação web** é a interface que o utilizador vai usar, esta irá conter gráficos com informação da fonte de dados que requerem atualização constante e podem ainda conter elementos associados às extensões instaladas.



Interações entre módulos

A típica interação entre os diferentes módulos consiste no acesso à aplicação web através do login do utilizador.

A interação entre Geração de Dados e o Message Broker funciona da seguinte forma: o primeiro obtém dados da API externa, faz algum processamento e envia-os para o Message Broker, onde os vários módulos do backend podem ler de forma a atualizar informação nas aplicações web, notificar extensões e atualizar a base de dados.

Em relação à interação entre o Backend e a Persistência, serve apenas para realizar as operações comuns de acesso, criação e atualização de dados.

Já na interação Cliente - Backend, há uma ligação bidirecional uma vez que o

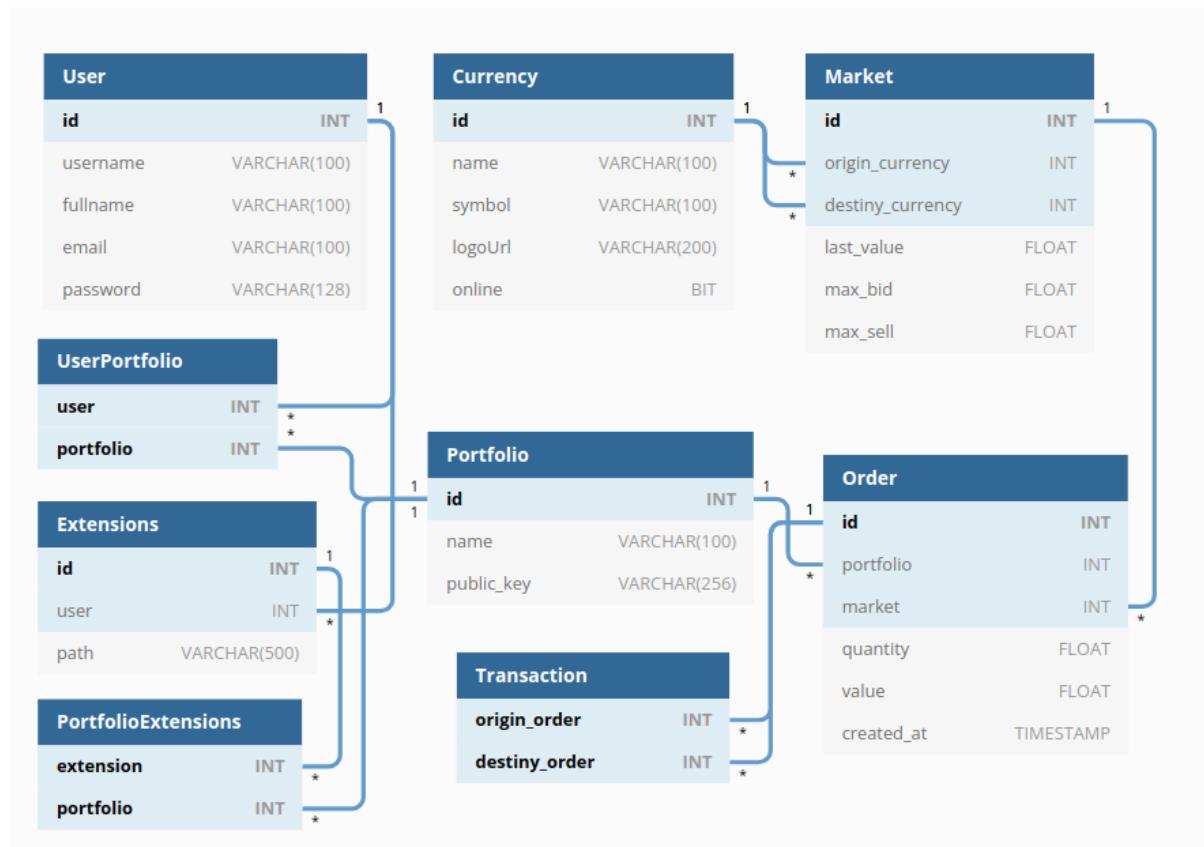
cliente, para além de pedir informação sobre as moedas virtuais, os mercados e suas oscilações ao backend, irá ainda receber do mesmo as atualizações de valores, através de websockets.

Entre a API Externa e o Backend, é este último o responsável por pedir informações sobre a extensão e enviar os dados requisitados quando existam atualizações de forma a receber instruções de atuação.

O módulo dos Clientes e a Extensão estão conectados pois poderá haver extensões que modificam o frontend, alterando ou adicionando funcionalidades. É dado como exemplo uma extensão que adiciona um feed de notícias sobre as moedas em que o utilizador investe.

4 Informação perspetiva

O seguinte diagrama relacional descreve toda a informação relativa ao domínio da aplicação, apresentando todas as entidades e as relações entre elas.



5 Referências e Recursos

[Most Trusted Cryptocurrency Market Data API](#)

[Bitrex Developers](#)

[Gitflow Workflow](#)

[Workflow overview](#)