 

Caio Larroza

Leonardo Costa

Giovanni Armane

Matheus Santos

**Projeto Jarbas:**

**Controle de Finanças Pessoais**

São Caetano do Sul / SP

2017

Caio Larroza

Leonardo Costa

Giovanni Armane

Matheus Santos

**Projeto Jarbas:**

**Controle de Finanças Pessoais**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Caetano do Sul, sob a orientação do Prof. Humberto Luiz de Toledo, como requisito parcial para a obtenção do diploma de Graduação no Curso de Análise e Desenvolvimento de Sistemas.

São Caetano do Sul / SP

2017

**Agradecimentos**

**Resumo**

LARROZA, Caio; COSTA, Leonardo; ARMANE, Giovanni; SANTOS, Matheus. **Projeto Jarbas: Controle de Finanças Pessoais**. Trabalho de Conclusão de Curso – Faculdade de Tecnologia de São Caetano do Sul – Antônio Russo, São Caetano do Sul, 2017. Projeto de sistema mobile híbrido que auxilia o usuário a controlar suas finanças pessoais e a projetar investimentos para o futuro. O projeto tem a proposta de desenvolver uma aplicação responsiva, intuitiva e moderna, capaz de auxiliar qualquer usuário na tarefa de controlar suas contas contábeis e projetar investimentos para o futuro. É uma aplicação baseada em tecnologias *web*, tornando possível o acesso as informações de qualquer dispositivo que esteja conectado na *internet*.

**Palavras-chave:** finanças; mobile; web; responsivo; moderno; intuitivo.

**Abstract**

**Sumário**

[Introdução 7](#_Toc491893856)

[Objetivos Gerais 7](#_Toc491893857)

[Objetivos Específicos 7](#_Toc491893858)

[Problema 7](#_Toc491893859)

[Justificativa 7](#_Toc491893860)

[Metodologia 7](#_Toc491893861)

[Tecnologias Usadas 7](#_Toc491893862)

[1.1. HTML 8](#_Toc491893863)

[1.2. CSS 8](#_Toc491893864)

[1.3. JavaScript 8](#_Toc491893865)

[1.4. AngularJS 9](#_Toc491893866)

[1.5. Ionic 9](#_Toc491893867)

[1.6. C# (C Sharp) 9](#_Toc491893868)

[1.7. .NET Core e ASP.NET Core Framework 10](#_Toc491893869)

[1.8. Entity Framework Core 10](#_Toc491893870)

[1.9. Identity Core 11](#_Toc491893871)

[1.10. Python 11](#_Toc491893872)

[1.11. PostgreSQL 11](#_Toc491893873)

# **Introdução**

# **Objetivos Gerais**

Desenvolver uma ferramenta para controle financeiro do usuário:

* + Controle de gastos e movimentações
  + Poupar (para eventualidades e compras de bens)
  + Acompanhar investimentos
  + Projetar gastos de acordo com o padrão
  + Fluxo de caixa (receita e despesa)

# **Objetivos Específicos**

Tornar prático, acessível e confiável o controle de finanças. Superar as deficiências encontradas em aplicativos semelhantes. Apresentar uma ferramenta que estaria disponível em qualquer plataforma com suporte a tecnologias web.

# **Problema**

Como auxiliar o usuário a administrar/gerenciar as suas contas pessoais?

# **Justificativa**

O sistema se justifica porque um grande problema atualmente é o controle financeiro pessoal. As pessoas têm muita dificuldade em controlar seus gastos e, sendo assim, um sistema que será capaz não só de controlar gastos já feitos, mas também estimar projeções futuras pode ser de grande utilidade para aqueles que se preocupam em manter as suas finanças organizadas.

O desenvolvimento deste tipo de aplicativo também se justifica por ser uma oportunidade para o grupo desenvolvedor de utilizar conhecimentos que foram sendo apresentados durante o curso e que agora poderão ser colocados em prática, tais como HTML, CSS, JavaScript, Engenharia de Software, Banco de Dados, Contabilidade, Economia, Estatística e Estrutura de Dados.

# **Metodologia**

O sistema será desenvolvido em modelo cliente-servidor com banco de dados PostgreSQL. No lado cliente, serão utilizadas tecnologias *web*, sendo estas HTML, CSS, JavaScript e frameworks especializadas. Do lado servidor, será utilizado de tecnologias do ambiente .NET, da Microsoft, com banco de dados relacional PostgreSQL e scripts Python para tarefas especializadas.

Os requisitos serão levantados juntamente com a coorientadora do projeto Prof. Me. Sônia Cristina Menoce com objetivo de melhor entender as melhores técnicas de controle financeiro, projeções e fontes de pesquisa. Em seguida, serão realizadas algumas pesquisas de campo com o objetivo de melhor entender as deficiências de aplicativos similares já no mercado e como melhor solucionar os problemas apresentados.

# **Tecnologias Usadas**

## **HTML**

Criada por Tim Berners Lee na década de 90, HTML é a mais popular linguagem de marcação utilizada na *web*. Se baseia em *tags* escritas de forma estruturada e hierárquica, interpretada pelos browsers modernos. É de fácil aprendizado, não requer programas específicos para seu desenvolvimento, podendo ser escrito com apenas um editor de texto.

O HTML é a linguagem responsável por dar estrutura a uma página *web*. Está atualmente em sua versão HTML5, contando com suporte a *tags* multimídia e melhor especificação de separações de seções na página. As especificações da linguagem são controladas pela W3C (World Wide Web Consortium), um consórcio internacional responsável por estabelecer os padrões de conteúdo para a internet.

No projeto Jarbas, o HTML é responsável pela estruturação do conteúdo na aplicação front-end. É utilizado juntamente com a tecnologia Ionic (vide abaixo).

## **CSS**

Cascading Style Sheets, ou CSS, é uma linguagem *web* com finalidade de adicionar estilos em páginas HTML. É um conjunto de regras composta por seletores e blocos de seleção, por qual são declarados uma variedade de estilos definidos. Com CSS é possível controlar a posição de elementos HTML na página, definir *backgrounds*, fontes, alinhamentos, etc.

Suas especificações são definidas pela W3C (World Wide Web Consortium) e está atualmente em sua terceira versão, CSS3.

No projeto Jarbas, o CSS é responsável pela estilização do conteúdo na aplicação front-end. É utilizado juntamente com a tecnologia Ionic (vide abaixo).

## **JavaScript**

JavaScript é uma linguagem interpretada, de tipagem fraca e dinâmica, com suporte a diversos paradigmas de programação - orientação a objetos, protótipo funcional, entre outras. Implementada por todos navegadores *web* modernos, considerado a *língua franca* do desenvolvimento *cliente-side*. É baseada na especificação ECMAScript, padronizada pela Ecma International.

Desenvolvida inicialmente na Sun Microsystems por Brendan Eich, é hoje uma das linguagens de programação mais utilizadas no mundo, com suporte de corporações como Google (desenvolvedora da engine V8 para JavaScript) e Fundação Mozilla.

É responsável por dar funcionalidade e dinamismo às páginas *web*. No projeto Jarbas, o JavaScript é utilizado em sua forma TypeScript (um *subset* JavaScript desenvolvido pela Microsoft) em conjunto com a framework AngularJS (vide abaixo) para realizar requisições HTTP ao servidor, trabalhar com dados e controlar aspectos da interface gráfica.

## **AngularJS**

O AngularJS é um framework open-source, mantido pela Google, que auxilia no desenvolvimento de *single-page applications* e *web apps*. É construído seguindo o padrão MVVC (model-view-view-model). É utilizado com a linguagem HTML e JavaScript para dar dinamismo as páginas.

Um dos mais conhecidos e utilizados *frameworks front-end*, o Angular encontra-se atualmente na versão Angular 4, entretanto ainda contando com suporte ás versões legadas Angular 1.

No projeto Jarbas, a versão utilizada é o AngularJS 1 – por razões de familiaridade da equipe de desenvolvimento e estabilidade do framework. É responsável pela camada de exibição e controle das *views* da aplicação cliente.

## **Ionic**

O Ionic é um *framework* destinado ao desenvolvimento de aplicações mobile híbridas. Construído em cima das plataformas AngularJS e Apache Cordova, a framework prove diversos componentes e serviços para a construção de aplicações mobile que utilizam tecnologias *web*, como HTML5 e CSS. Criada pela Drfty Co. em 2013.

No projeto Jarbas, Ionic é responsável pelo auxílio e construção das interfaces da aplicação cliente em conjunto com AngularJS.

## **C# (C Sharp)**

C# (lê-se CSharp) é uma linguagem de programação de alto-nível, compilada, de tipagem estática e/ou dinâmica forte, parcialmente inferida e com suporte multi-paradigmas de programação – orientação a objetos, funcional, baseada em eventos, estruturada e concorrente.

É objetivamente a linguagem principal do .NET Framework, criado pela Microsoft para desenvolvimento Windows e *web*. Encontra-se atualmente em sua sétima versão - C#7. Uma das linguagens mais maduras e utilizadas no mundo, provê capacidades genéricas para desenvolvimento de qualquer aplicação.

É baseada em um tempo de execução em máquina virtual. Um código C# é compilado e otimizado para um código de máquina especificado, chamado *Common Language Infrastructure*, ou CLI abreviado, que é então executado em uma implementação da máquina virtual *Common Language Runtime*, ou CLR abreviado.

A linguagem é utilizada no projeto Jarbas como base linguagem base das tecnologias do servidor – back-end.

## **.NET Core e ASP.NET Core Framework**

.NET Core é a mais nova framework de código aberto livre da Microsoft e comunidade .NET. É uma completa reescrita e reestruturação da antiga .NET Framework. Modular e multiplataforma, atualizada para as novas tendências do mercado de desenvolvimento *web*, *mobile*, *IoT*, etc.

O ASP.NET Core é o framework da plataforma .NET Core para desenvolvimento *web*. Originalmente chamado de ASP.NET MVC 5, a Microsoft decidiu alterar seu nome para não confundir o novo framework com uma simples atualização do ASP.NET MVC 4.

No projeto Jarbas o ASP.NET Core é responsável pelo funcionamento do servidor back-end da aplicação. A escolha pela plataforma se deve ao fato de permitir o desenvolvimento e deploy da aplicação em múltiplas plataformas, performance excepcional e suporte extenso por parte da comunidade e Microsoft.

## **Entity Framework Core**

Componente do ASP.NET Core, o Entity Framework permite ao desenvolvedor conectar sua aplicação com banco de dados de forma simples e fácil. O EF, sigla do framework, trabalha como ORM (*object-relational mapping*), possibilitando que um desenvolvedor programe sua aplicação pensando apenas nas relações entre classes da programação, deixando todo fardo de tratamento, mapeamento e transcrição de dados de objetos para tabelas relacionais de bancos de dados.

O Entity Framework foi escolhido para o projeto não apenas por ser considerado o framework padrão da plataforma .NET – de fato, existem diversos outros frameworks como o Entity, mas também pela agilidade que conseguiu trazer para os desenvolvedores do projeto acoplado com sua leve curva de aprendizado.

## **Identity Core**

Identity Core é um componente do ASP.NET Core responsável por tratar da autenticação e segurança das aplicações na plataforma.

O Identity Core foi escolhido por ser o mais simples e eficiente framework para o desenvolvimento da camada de autenticação para o projeto Jarbas.

## **Python**

Python é uma linguagem de programação de alto nível, interpretada, de tipagem dinâmica e forte, e com suporte multi-paradigmas de programação – orientação a objetos, funcional, imperativa, estruturada e concorrente.

Criada por Guido van Rossum, possui a filosofia de prezar por legibilidade de código e simplicidade em alta e baixa escala. Sua principal implementação, o interpretador CPython, é mantido pela comunidade Python e pela Python Software Foundation, uma organização sem fins lucrativos. É adotada por diversas empresas de software pelo mundo, notavelmente Dropbox, Google, Instagram e Spotify.

Python foi escolhida para o projeto Jarbas como complementar a tarefas simples onde não fazia sentido executar toda uma aplicação .NET para sua realização ou como suplementar a casos de uso fora do escopo da plataforma principal do servidor.

## **PostgreSQL**

PostgreSQL, ou simplemente Postgres, é um sistema de gerenciamento de banco de dados relacional de código aberto livre, com um grande foco em extensibilidade e acordo com padrões de mercado. É desenvolvido pelo PostgreSQL Global Development Group, um órgão internacional composto por diversas empresas da indústria de software.

O Postgres foi escolhido como gerenciador de banco de dados relacional do projeto Jarbas pela sua facilidade de uso, por ser software livre e multiplataforma.