**APLICATIVO DE GERENCIAMENTO FINANCEIRO PESSOAL**

***Debora Pereira dos Santos -*** [***deborapesantos@gmail.com***](mailto:deborapesantos@gmail.com)

Prof.

**Avaliação da Banca**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ORIENTADOR**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROFESSOR 1**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROFESSOR 2**

**ETEP CENTRO UNIVERSITÁRIO**

**Junho/2020**

***Resumo -***

***Palavras-chave: Controle Financeiro; Educação Financeira; Financeiro;***

***Abstract -***

***Keywords:***

# INTRODUÇÃO

Justificativa do projeto

O projeto tem por finalidade atender as necessidades das pessoas em gerenciar seus gastos e ter o controle sobre suas contas, a fim de auxiliar na educação financeira de cada usuário.

Gráfico de endividados no Brasil

# FUNDAMENTAÇÃO TEORICA

*2.1 - Linguagens de Programação*

*2.1.1 - Javascript*

O Javascript é uma linguagem de programação voltada para aplicações Web. Tem como características principais: Mult-paradigma e tipagem fraca. Em versões mais recentes é possível utilizar orienação a objetos. O que torna mais fácil o desenvolvimento de aplicações complexas. É baseada em ECMAScript, uma padronização de linguagens script criada pela empresa ECMA Internacional.

É muito utilizada para dinamizar e aumentar a interatividade dos usuários nas páginas Web. Além disso, é possível a criação de conteúdos dinâmicos, controle multimídia, animação de objetos.

*2.1.2 - Typescript*

Criado pela Microsoft, o TypeScript é uma linguagem/superset do JS, tem o objetivo de se aproximar linguagens como Java e C# ao Javascript, quando escrito para a criação do front-end. O Typescript, possui algumas características que estão presentes nas linguagens orientadas a objetos, o que fornece uma facilidade de escrita e é possível fazer reuso de código.

*2.1.3 – React*

É uma biblioteca Javascript para construção de interfaces tendo como conceito principal a criação de componentes para a divisão da interface em diversas partes independentes que podem ser reutilizadas. Facilita muito a manutenção do código, pois pode ser tratada de forma separada sem afetar outras áreas da aplicação. O React utiliza o conceito JSX, que tem o significado JavaScript eXtension, ou seja, uma extensão de sintaxe para o JavaScript onde permite juntar o HTML ao JavaScript, apesar de ser diferente, facilita muito o desenvolvimento de componentes. Assim como o TypeScript, é preciso usar um transpiler para converter JSX em JavaScript.

*2.1.4 – React Native*

O React Native é um framework baseado em React, desenvolvido pela equipe do Facebook, que possibilita o desenvolvimento de aplicações nativas para multiplataforma utilizando apenas um código fonte escrito em JavaScript. Esse tipo de framework está sendo muito utilizado por ter um desenvolvimento rápido e por ter um custo menor, mas diferente de outros frameworks que empacotam a aplicação web em um “browser”, o react native compila uma aplicação nativa, ou seja, a aplicação é convertida para a linguagem nativa do sistema operacional.

*2.1.5 - C# .NET Framework.*

O C# (leia-se C-Sharp), é uma linguagem de programação orientada a objeto fortemente tipada. Foi desenvolvida pela Microsoft e faz parte da plataforma .NET. É considerada uma linguagem de altíssima complexidade e permite desenvolver aplicações robustas e seguras executados no .NET Framework. É possível criar aplicações desktops, aplicações Web, serviços Web XML, componentes distribuídos, aplicativos de banco de dados e entre outros.

A sintaxe C# simplifica as complexidades da linguagem C++ e fornece alguns recursos poderosos, como tipos anulados, delegados, expressões lambda, enumerações, e acesso direto à memória. As expressões LINQ (Language-Integrated Query) tornam a consulta mais simples e de fácil entendimento. Por ser uma linguagem orientada a objeto, o C# oferece suporte aos conceitos de encapsulamento, herança e claro, o polimorfismo.

*2.1.6 - Serenity*

O Serenity é uma plataforma de aplicativos ASP.NET Core/MVC/TypeScript, desenvolvida com base em tecnologias de código aberto. O objetivo principal do Serenity é facilitar o desenvolvimento de aplicações Web e reduzir os cursos com manutenção e repetição de tarefas, como implementação de CRUD (em inglês, Create, Read, Update e Delete), as quatro operações básicas (criação, consulta, edição e exclusão) utilizadas em bases de dados relacionais. Serenity utiliza as melhores práticas do design patterns e facilita o desenvolvimento de sistemas simples a complexos.

*2.2 - Banco de Dados*

Banco de dados é uma coleção de dados inter-relacionados que representam uma informação.

*2.2.1 - SQL*

*2.3 - Ambientes de Desenvolvimento*

*2.2.1 - Visual Studio 2019*

O Visual Studio é uma IDE (Ambiente de Desenvolvimento Integrado).

*2.2.2 - Visual Studio Code*

O Visual Studio é uma IDE (Ambiente de Desenvolvimento Integrado). Assim como você usa o Microsoft Word para escrever documentos, você usará um IDE para criar aplicativos.....

*2.3 – Sistema de Gerenciamento de banco de banco de dados (SGDB)*

*2.3.1 - SQL Server 2014*

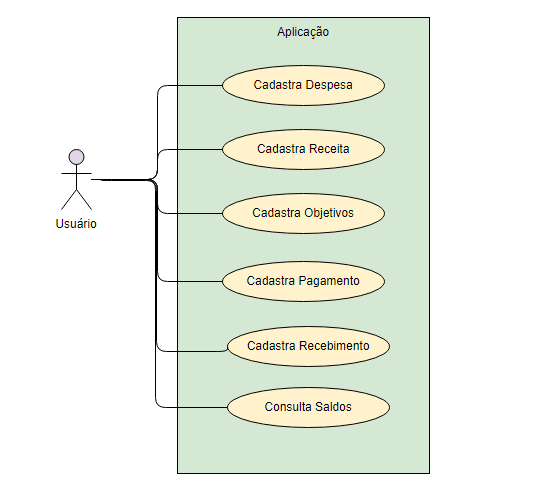
O Visual Studio é uma IDE (Ambiente de Desenvolvimento Integrado). Assim como você usa o Microsoft Word para escrever documentos, você usará um IDE para criar aplicativos.......

*2.2 – Conceitos financeiros básicos*

# METODOLOGIA

*3.1 - Cenário Atual*

*3.2 – Análise de Requisitos*

*3.2 – Caso de uso*

*3.3 – Mockup da Aplicação*

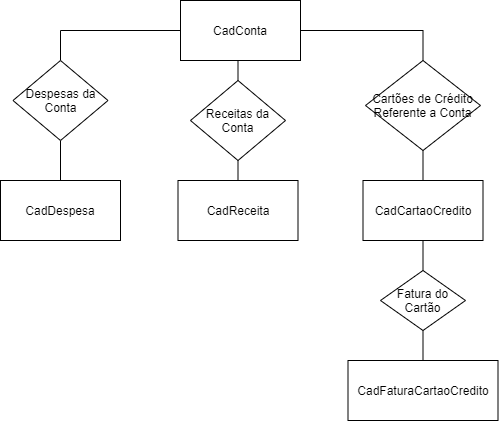
*3.4 – Banco de Dados*

A aplicação foi desenvolvida de forma a ser escalonável, desta forma foi escolhido utilizar banco de dados relacional, pois dado a complexidade, esse tipo de banco é o melhor que se encaixaria.

As entidades foram criadas com base no protótipo inicial. (protótipo) A princípio a aplicação teria funcionalidades principais e secundarias.

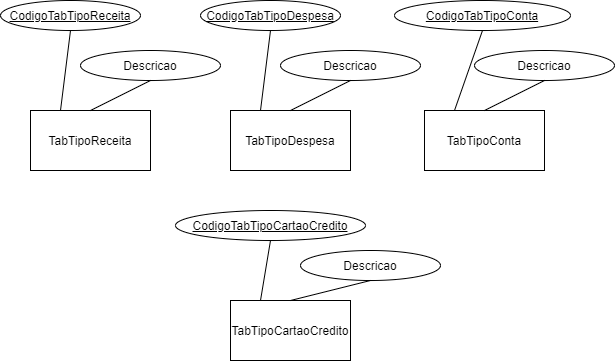
As entidades principais, onde são feito "CRUD" foram separadas no grupo "CAD", são elas:

CadReceita, CadDespesa, CadConta, CadCartaoCredito,CadFaturaCartaoCredito



As entidades secundarias, onde são padrões do sistema, foram separadas no grupo "TAB", são elas:

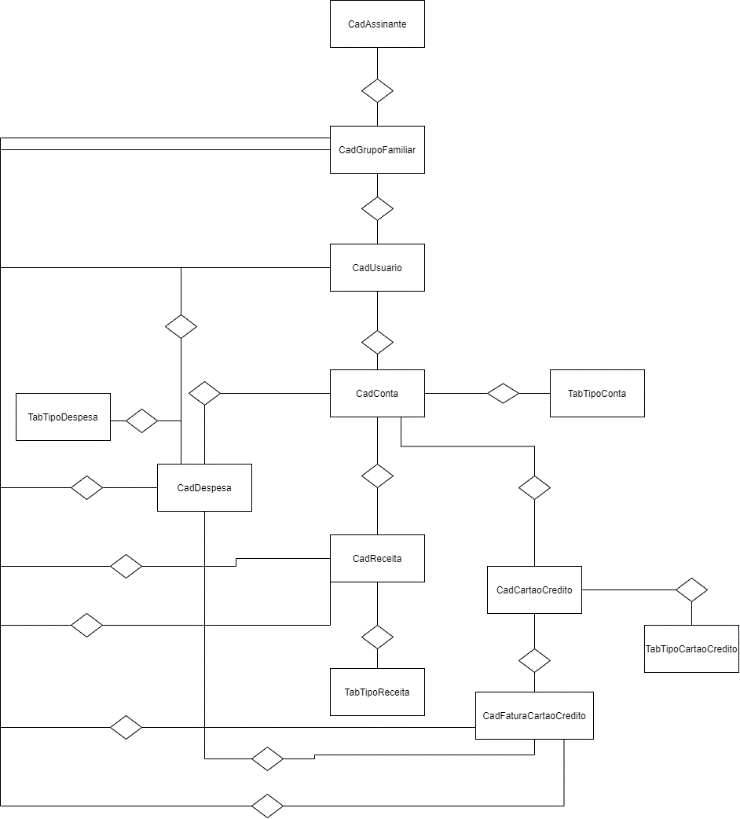
TabTipoReceita, TabTipoDespesa, TabTipoConta, TabTipoCartaoCredito



Para formar a relação de dados, foi analisado qual seria a entidade principal que ligaria todas as tabelas ao usuário, e assim poderia compartilhar informações de usuário para usuário,- funcionalidade diferencial da aplicação.

Foi pensado em uma piramide, onde teria uma herarquia de visualização de conteúdo, desta forma:

CadAssinante > CadGrupoFamiliar > CadUsuario



Quem cria um login no aplicativo pode incluir usuarios já logados ou criar usuarios para o seu grupo familiar.

Desta forma, é possivel compartilhar informações de despesas e receitas de um grupo familiar.

Por exemplo, um casal pode ter uma conta em conjunto, cada um continua tendo suas

proprias contas e pode compartilhar contas que são do casal, como despesas de água, energia, TV a cabo e etc..

As outras tabelas do sistema fazem parte da autenticação do usuario na aplicação, são tabelas padrões do Serenity,

são elas:

*3.3 - Funcionamento*

*3.4 – Sistema Mobile*

*3.5 - Sistema Web*

*3.6 - Integração dos Sistemas*

API

3.7 - *Testes*

*3.8 - Conformidade Final*

# RESULTADOS

# CONCLUSÃO

# TRABALHOS FUTUROS

# REFERÊNCIAS BIBLIOGRÁFICAS

MORE: Mecanismo online para referências, versão 2.0. Florianópolis: UFSC Rexlab, 2013. Disponível em: http://www.more.ufsc.br/. Acesso em: 03/04/2020.

TypeScript: o Javascript moderno para criação de aplicações

Autor(es)Luís Abreu

Editora FCA – Editora de Informática, Lda.

Primeiros Passos com React

Construindo aplicações web

Autor: Stoyan Stefanov

ISBN: 978-85-7522-520-2

Ano: 2016

# REFERÊNCIAS CONSULTADAS

<https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/O_que_e_JavaScript>

<https://reactnative.dev/docs/out-of-tree-platforms>

<https://www.hostinger.com.br/tutoriais/o-que-e-react-javascript>

<https://www.w3schools.com/REACT/react_intro.asp>

<https://pt-br.reactjs.org/tutorial/tutorial.html>

# ANEXO I – BANCO DE DADOS

# ANEXO II – PROGRAMAÇÃO