

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
FACULDADE DE TECNOLOGIA DE CAMPINAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS

VINÍCIUS SOLON SILVA

**DESENVOLVIMENTO DO BACK-END DE UM APLICATIVO
MOBILE PARA CONTROLE DE FINANÇAS PESSOAIS**

CAMPINAS/SP
2022

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
FACULDADE DE TECNOLOGIA DE CAMPINAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS

VINÍCIUS SOLON SILVA

**DESENVOLVIMENTO DO BACK-END DE UM APLICATIVO
MOBILE PARA CONTROLE DE FINANÇAS PESSOAIS**

Trabalho de Graduação apresentado por Vinícius Solon Silva, como pré-requisito para a conclusão do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Faculdade de Tecnologia de Campinas, elaborado sob a orientação do Prof. Anderson Luiz Barbosa.

CAMPINAS/SP
2022

FICHA CATALOGRÁFICA
CEETEPS - FATEC Campinas - Biblioteca

S586d

SILVA, Vinicius Solon

Desenvolvimento do Back-end de um aplicativo para o controle de
finanças pessoais. Vinicius Solon Silva. Campinas, 2022.

40 p.; 30 cm.

Trabalho de Graduação do Curso de Análise e Desenvolvimento de
Sistema – Faculdade de Tecnologia de Campinas, 2022.

Orientador: Prof. Me. Anderson Luiz Barbosa.

1. Planejamento. 2. Finanças. 3. Tecnologia. 4. Aplicativos. 5. Interface.
I. Autor. II. Faculdade de Tecnologia de Campinas. III. Título.

CDD 005.26

Catálogo-na-fonte: Bibliotecária: Aparecida Stradiotto Mendes – CRB8/6553

TG ADS 22.2

Vinicius Solon Silva

Desenvolvimento do Back-End de um aplicativo para o controle de finanças pessoais

Trabalho de Graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pelo CEETEPS / Faculdade de Tecnologia – Fatec Campinas.

Campinas, 02 de dezembro de 2022.

BANCA EXAMINADORA

Anderson Luiz Barbosa

Prof. Anderson Luiz Barbosa
Fatec Campinas

Cláudio Luis Vieira Oliveira

Prof. Claudio Luis Vieira Oliveira
Fatec Campinas

Sandro Pinheiro de Assis Cosso

Prof. Sandro Pinheiro de Assis Cosso
Fatec Campinas

AGRADECIMENTOS

A minha mãe, Denise Soares Solon, por zelar meu espírito e mente, durante toda a minha jornada acadêmica. Meu eterno amor e gratidão por este ser abençoado.

Aos meus professores e colegas de classe, pela parceria, amizade e conhecimento compartilhado por todo este período desafiador.

RESUMO

O número de pessoas endividadas no Brasil continua batendo recordes ao longo do tempo, por isso, fazer o planejamento financeiro acaba se tornando a principal ferramenta na hora de equilibrar as finanças. Um problema que parece ser uma tarefa simples, mas manter a consistência nesse planejamento, principalmente quando falamos de longo prazo, pode se tornar muito complicado. Principalmente para pessoas que têm a famosa “*investment-phobia*”, ou seja, pessoas que têm dificuldade em falar sobre dinheiro ou até mesmo sobre a falta dele. Para esclarecer esse problema, este projeto abordou de forma introdutória sobre educação financeira, além de mostrar os possíveis benefícios da tecnologia para mapeamento de gastos e ganhos na vida financeira das pessoas, visto que a popularização e acessibilidade de celulares e dispositivos móveis vem crescendo no Brasil e no mundo. Também foi realizada uma pesquisa de mercado, mostrando os principais aplicativos móveis no ambiente financeiro, buscando entender suas fragilidades e onde são necessárias melhorias. Com isso, foi realizado o desenvolvimento um aplicativo com *back-end* simples, com desempenho otimizado e gratuito, que se mostrou muito viável e de importante valia no combate ao endividamento da população.

Palavras-chave: planejamento, finanças, tecnologia, aplicativos e interface.

ABSTRACT

The number of people in debt in Brazil continues to break records over time, so financial planning ends up becoming the main tool when it comes to balancing finances. A problem that seems to be a simple task, but maintaining consistency in this planning, especially when we are talking about the long term, can become very complicated. Mainly for people who have the famous “investment-phobia”, that is, people who have difficulty talking about money or even the lack of it. To clarify this problem, this project addressed financial education in an introductory way, in addition to showing the possible benefits of technology for mapping expenses and gains in people's financial lives, since the popularization and accessibility of cell phones and mobile devices has been growing in Brazil and in the world. A market survey was also carried out, showing the main mobile applications in the financial environment, seeking to understand their weaknesses and where improvements are needed. With this, an application with a simple back-end was developed, with optimized performance and free of charge, which proved to be very viable and of important value in combating the population's indebtedness.

Keywords: *planning, finance, technology, applications and interface.*

LISTA DE FIGURAS

Figura 1 - Diagrama de casos de uso	24
Figura 2 - Diagrama de classes do banco de dados	29
Figura 3 - Interface do Azure SQL Database	30
Figura 4 - Tabela de registro dos cartões.....	30
Figura 5 - Tabela de categorias	30
Figura 6 - Tabela de registro de objetivos.....	31
Figura 7 - Tabela da receita.....	31
Figura 8 - Tabela de registro de usuários.....	31
Figura 9 - Estimativas de custo do banco de dados Azure SQL Database.....	32
Figura 10 – Reclamação da interface e usabilidade no aplicativo.	33
Figura 11 - Reclamação de interface.....	33
Figura 12 - Reclamação de interface.....	34
Figura 13 – Reclamação do aplicativo ao sair do gratuito para versão paga	34
Figura 14 - Primeira pergunta	35
Figura 15 - Segunda pergunta	35
Figura 16 - Terceira pergunta.....	36
Figura 17 - Quarta pergunta	36
Figura 18 - Quinta e última pergunta	37

LISTA DE QUADROS

Quadro 1 - Versão revisada das heurísticas de Nielsen	18
--	----

LISTA DE TABELAS

Tabela 1 - Descrição de requisitos do Back-End.....	22
Tabela 2 - Descrição do diagrama de caso de uso.....	25

SUMÁRIO

1	INTRODUÇÃO	12
1.1	CONTEXTUALIZAÇÃO.....	12
1.2	JUSTIFICATIVA/PROBLEMÁTICA	13
1.3	OBJETIVOS	14
2	REVISÃO BIBLIOGRÁFICA.....	15
3	MATERIAIS E MÉTODOS	18
3.1	MATERIAIS.....	18
3.1.1	Descrição dos materiais	19
3.2	MÉTODOS.....	21
3.3	Modelagem	21
3.3.1	Requisitos Funcionais e Não Funcionais	22
3.3.2	UML – Diagrama de Caso de Uso	23
3.3.3	Descrição do Diagrama de Caso de Uso	25
3.3.4	UML – Diagrama de Classes	28
3.3.5	Banco de Dados na nuvem <i>Azure</i>	29
3.3.6	Estimativas de custo do banco de dados	31
4	RESULTADOS E DISCUSSÃO	33
5	CONSIDERAÇÕES FINAIS.....	38
	REFERÊNCIAS BIBLIOGRÁFICAS	40

1 INTRODUÇÃO

Finanças pessoais sempre foram um desafio para o brasileiro, desde contabilizar as moedas para ir comprar itens na mercearia, até ao pagamento de boletos e contas em um banco em uma lotérica. A administração financeira sempre foi algo em que as pessoas deveriam se informar através de dicas de amigos, parentes ou até mesmo (num contexto mais atual) em cursos via Internet.

Também chamada de a “matéria esquecida” do ensino público, fez com que que grande parte dos indivíduos que possuem o conhecimento desta de forma aprimorada, vem geralmente de famílias com poderes aquisitivos significativos ou já com um histórico e hábitos familiares que proporcionam este aprendizado desde berço.

Como esclarecido por Calixto et al (2022, p. 2), “O assunto dinheiro ainda é um tabu muito grande no Brasil. A falta de conversa sobre o tema em casa e nas escolas acarretam vários problemas na vida adulta do ser humano como a inadimplência e problemas psicológicos”.

Com este cenário em vista, e com a atual situação financeira das pessoas no Brasil, é de interesse coletivo uma forma simples, eficaz e dinâmica de gerenciamento de suas finanças pessoais. Portanto, este trabalho de graduação irá tratar do desenvolvimento do *back-end* de um aplicativo *mobile* para celulares que irá realizar toda a parte administrativa e de monitoramento de finanças para todo e qualquer indivíduo.

1.1 CONTEXTUALIZAÇÃO

No Brasil, quase 78% da população está endividada, segundo uma pesquisa feita em março de 2022 pela confederação Nacional do Comércio de Bens. Os números são impactantes, e evidenciam um assunto que a grande maioria dos brasileiros possui dificuldade ou até mesmo não dão devida atenção... O planejamento financeiro.

Muitas vezes, as pessoas acabam se perdendo nas próprias finanças e quando menos esperam estão com a conta “no vermelho”. O planejamento financeiro é um item essencial para que não haja surpresas ou até mesmo necessidades durante o mês. Porém anotar toda a movimentação monetária de uma pessoa, em um caderno ou folha de papel não parece ser uma

tarefa fácil ou segura. Por conta disso, utilizar a tecnologia para auxiliar neste problema se tornou uma opção viável, prática e acessível, já que no Brasil a cada um habitante existem dois dispositivos digitais como celulares ou computadores.

1.2 JUSTIFICATIVA/PROBLEMÁTICA

O planejamento financeiro de uma pessoa pode ser feito de diversas formas, seja ela através de meios tradicionais como fazer as anotações em um caderno, por exemplo. Até mesmo utilizando um aplicativo, site ou planilha que visam descomplicar todo o processo.

Contudo, apesar destas diversas alternativas tecnológicas, muitas delas podem parecer complexas de mais para usuários que não possuem uma afinidade tão grande com um celular ou computador. Fazendo com que as pessoas abandonem o processo mesmo antes de iniciar seus planejamentos e metas. Além destas lacunas que geram uma oportunidade do projeto, existe uma grande motivação do autor, que mesmo possuindo facilidade com a tecnologia, passou por diversas dificuldades utilizando aplicativos e sites existentes.

Neste contexto, foi desenvolvido um aplicativo mobile com um *back-end* simples e performativo, que visa otimizar a performance do aplicativo a ser criado, aprimorando a organização financeira de seus usuários e incentivando os mesmos a introduzir esta nova rotina de planejar o uso do dinheiro em seu cotidiano.

A implementação e integração do *back-end* com o *front-end* e interface deste aplicativo, se deu pela utilização e união de ferramentas como *Azure SQL Database*, linguagem de programação *Python*, a biblioteca de mapeamento objeto-relacional (ORM) *SQLAlchemy*, o *Framework KivyMD* e o ambiente de desenvolvimento de código *Visual Studio Code*. Todos os dados dos usuários (como nome, idade e renda bruta por exemplo) foram armazenados no *Azure SQL Database*, que por sua vez pode disponibilizar os dados para visualização através da implementação com o framework *KivyMD*, no qual foi utilizado para desenvolver toda a interface necessária para o *front-end*, como gráficos, tabelas e métricas da renda do usuário.

O trabalho teve uma grande influência das 10 heurísticas de Nielsen, com a finalidade de atribuir um visual funcional e agradável para a aplicação. Com essas características o

aplicativo é um grande aliado para a diminuição das pessoas endividadas no Brasil e também ser útil para as pessoas que estão iniciando sua vida financeira.

1.3 OBJETIVOS

A) Analisar o cenário da educação financeira e aspectos sociológicos econômicos de maneira breve, com caráter introdutório.

B) Desenvolver o *back-end* de um aplicativo para finanças pessoais.

C) Implantar as melhores práticas da Engenharia de *Software*, bem como as Heurísticas de Nielsen e os principais requisitos e normas para o desenvolvimento do *back-end* para aplicativos *mobile*.

2 REVISÃO BIBLIOGRÁFICA

A dificuldade de planejamento financeiro do brasileiro, não é nenhuma novidade nos tempos atuais, uma problemática tão antiga e presente no dia a dia da população, carece de alternativas simples e eficazes para a resolução desta adversidade. Como consequência deste grave problema, o Brasil segue aumentando de uma maneira exponencial a porcentagem dos devedores. Segundo pesquisa publicada em março de 2022 pelo G1: “77,5% das famílias brasileiras fecharam o mês com alguma dívida, segundo levantamento da Confederação Nacional do Comércio de Bens, Serviços e Turismo (CNC).”

O planejamento da utilização do capital é um do forte candidato para evitar que a falta de dinheiro não surpreenda no decorrer do mês. Na teoria a regra “gaste menos do que você ganha” parece ser simples, porém na prática, aparecem as dificuldades, principalmente no momento de mapear todos os gastos, que possibilita entender como reduzir as despesas ou até mesmo onde aumentar a aplicação do capital.

O orçamento doméstico é o principal instrumento para fazer o planejamento financeiro para hoje, amanhã e dias futuros. É utilizado como ferramenta para planejar o equilíbrio entre receitas e despesas das contas pessoais, mas para isso o indivíduo deverá se comprometer com o projeto de estruturação do orçamento pessoal e disposto a colaborar para que o resultado final seja atingido (MOTA; CARLA *et al.*, p.8,2015).

Dito isso, uma justificativa plausível para tal dificuldade, seria o medo de lidar com a vida financeira. Tal medo também pode ser conhecido como “*Investment Phobia*”, título de uma pesquisa realizada pela universidade de Cambridge em 2003, onde afirma que o temor de lidar com a falta de dinheiro e falar sobre o assunto, são o terceiro maior medo das pessoas no Reino Unido, onde uma a cada cinco pessoas possuem esta fobia.

Tendo em vista tais dificuldades, um fator que poderá contribuir e facilitar o planejamento financeiro, é a tecnologia, já que grande parte das pessoas possuem um dispositivo móvel como tablets ou celulares em suas casas. Através destes aparelhos, é possível criar aplicativos para que esta gestão de capital seja facilmente acessada e armazenada. Além de auxiliar na quebra deste tipo de fobia, através de interfaces lúdicas e amigáveis, além da introdução de novos hábitos na rotina dos utilizadores.

Os aplicativos, através dos dispositivos móveis não portáteis e não acessíveis, têm facilitado a vida das pessoas, ampliando o impacto da tecnologia na sociedade, permitindo a expansão da portabilidade e flexibilidade para vários tipos de relações, como um meio de automatização de diversas tarefas, até então realizadas manualmente (MAIA e MARIN, p.2, 2021).

Contudo, apesar de existirem diversas plataformas onde é possível fazer o planejamento através do celular ou computador, sistemas com funcionalidades gratuitas e simples de serem utilizadas são escassas no mercado. Muitas vezes os usuários com pouca familiaridade com tecnologia, possuem extrema dificuldade em operar os aplicativos e não possuem o devido apoio, como tutoriais por exemplo. Já que uma interface complexa para usuários iniciantes, pode acabar afastando ao invés de atrair seu usufruidor. Segundo Nielsen e Loranger (2006), a usabilidade permite que pessoas com baixo grau de instrução possam ter empregos significativos, ela também permite conectar idosos com a comunidade e até mesmo fornecer informações e serviços a usuários portadores de deficiência física da mesma maneira que fornecemos a todas as outras pessoas. Para eles, a usabilidade beneficia os negócios e beneficia a humanidade.

Em vista destes fatores, foi desenvolvido um aplicativo para dispositivos mobile em conjunto com o aluno Matheus Bosco Martins da Silva, que desenvolveu a interface e o *front-end* da aplicação. O aplicativo conta com uma interface simples, intuitiva e acessível para pessoas com deficiências, onde possibilita que o utilizador possa lidar com seu próprio dinheiro, podendo fazer planejamentos para os mais diferentes objetivos de uma forma ágil. Sobretudo o objetivo é contribuir para a diminuição da população endividada no país.

A área aplicação deste tipo de aplicativo se daria para uso do cidadão comum brasileiro, este possui uma renda mensal que se desdobra em contas a pagar, lazer, alimentação, manutenção doméstica e imprevistos ocasionais. O usuário em questão irá inserir inicialmente o valor de sua renda mensal e a partir disso o aplicativo irá solicitar a divisão desta para as despesas já aqui citadas.

Para desenvolver o *back-end* da aplicação, foi utilizada a linguagem de programação *Python*, tendo em vista que é uma linguagem poderosa e de fácil implementação. “O *Python* tem uma sintaxe clara e concisa que favorece a legibilidade do código-fonte, tornando a linguagem mais produtiva” (Borges, p.14, 2014). Esta linguagem possui suporte a uma biblioteca de mapeamento objeto-relacional chamada *SQLAlchemy*, com ela estabelecemos a conexão e realizar consultas no banco de dados que foi provisionado na nuvem *Azure*, o *Azure SQL Database*. Com este pudemos armazenar de forma persistente os dados necessários para cadastro e criação dos usuários, bem como suas rendas e consultas referentes a estas.

Com o objetivo de desenvolver a aplicação dentro do prazo esperado e com as funcionalidades completas, foi utilizada a metodologia ágil *Scrum*. Segundo Michael (2018),

apesar de as implementações das metodologias ágeis serem recentes no universo do desenvolvimento de sistemas, o resultado de imediato já é muito satisfatório.

Primeiramente ocorreu um pré-planejamento, onde foram definidos os objetivos das *sprints* iniciais juntamente com alocação de recursos que estarão disponíveis para o desenvolvimento do aplicativo. No momento de desenvolvimento, ocorreram reuniões diárias com um pequeno período de tempo, para que sejam esclarecidos todos os pontos que foram trabalhados e suas dificuldades. Um dos benefícios do *scrum* para o projeto, foi a flexibilização da mudança de direção do projeto, ou seja, em casos que foram necessárias algumas alterações de requisitos ou de tecnologias no decorrer do desenvolvimento, estes tiveram a possibilidade de serem aceitos.

3 MATERIAIS E MÉTODOS

O material escolhido foi a linguagem *Python*, que é uma linguagem de interpretação para a tratativa do sistema no *back-end* da aplicação. O banco de dados utilizado na aplicação foi o *Azure SQL Database*, pois é caracterizado por ser um banco de dados na nuvem *Azure*, garantindo a persistência dos dados dos usuários, bem como uma boa responsividade a consultas. Para estabelecer a conexão com o banco bem como realizar consultas nele, foi utilizada a biblioteca de mapeamento objeto-relacional chamada *SQLAlchemy*. Já no *front-end*, foi escolhido o framework *open-source KivyMD*. Todo o aplicativo foi desenvolvido no ambiente de desenvolvimento do *Visual Studio Code*. O software desenvolvido, é capaz de ser executado em *Android*.

Para atingir os objetivos propostos neste trabalho foi utilizado o teste de usabilidade muito conhecido e aproveitado para esses tipos de testes. O método Avaliação Heurística proposta por Jakob Nielsen, também denominado de Heurísticas de Nielsen.

3.1 MATERIAIS

Para desenvolvimento do *back-end* do projeto, foi utilizada a linguagem de programação *Python*.

Para o armazenamento dos dados dos usuários, persistência dos dados, bem como a geração de cadastro e usabilidade multifatorial dentro do aplicativo, como ter um fluxo de caixa personalizado para cada usuário, foi utilizado o banco de dados *Azure SQL Database*.

Para estabelecer a conexão com o banco bem como realizar consultas nele, foi utilizada a biblioteca de mapeamento objeto-relacional chamada *SQLAlchemy*.

A plataforma escolhida para ser executado os testes foi o ambiente do *Android*.

O desenvolvimento do aplicativo foi feito no ambiente do *Visual Studio Code*.

O repositório em nuvem *Github* foi usado como ferramenta para controle e versionamento de código.

Foram utilizadas as Heurísticas de Nielsen para a implementação e validação das usabilidades do aplicativo.

3.1.1 Descrição dos materiais

No processamento de dados e informações da interface, bem como dados do usuário, foi utilizada a linguagem de programação *Python*. Que é uma linguagem de alto nível interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte.

Para o armazenamento dos dados fornecidos e processados dos usuários, foi utilizado o banco de dados *Azure SQL Database*, um sistema de gerenciamento de banco de dados (SGBD) na nuvem. Nele podemos criar tabelas, realizar consultas e armazenar funções para otimização de performance e obter persistência de dados.

O ambiente de execução escolhido para o aplicativo foi o *Android*, um sistema operacional projetado principalmente para dispositivos móveis com tela sensível ao toque como *smartphones* e *tablets*.

Neste projeto, foram implementadas as Heurísticas de Nielsen, na quais consistem em um método no qual é conduzido por um avaliador ou um conjunto de avaliadores que examinam o ambiente gráfico (interface gráfica) e julgam suas propriedades com o auxílio de princípios de usabilidade, denominados heurísticas (Quadro 1). (ROCHA; BARANAUSKAS, 2000).

Quadro 1 – Versão revisada das heurísticas de Nielsen

<p>1. Visibilidade do status do sistema</p> <ul style="list-style-type: none"> • sistema precisa manter os usuários informados sobre o que está acontecendo, fornecendo um feedback adequado dentro de um tempo razoável.
<p>2. Compatibilidade do sistema com o mundo real</p> <ul style="list-style-type: none"> • sistema precisa falar a linguagem do usuário, com palavras, frases e conceitos familiares ao usuário, ao invés de termos orientados ao sistema. Seguir convenções do mundo real, fazendo com que a informação apareça numa ordem natural e lógica.
<p>3. Controle do usuário e liberdade</p> <ul style="list-style-type: none"> • usuários frequentemente escolhem por engano funções do sistema e precisam ter claras saídas de emergência para sair do estado indesejado sem ter que percorrer um extenso diálogo. Prover funções <i>undo</i> e <i>redo</i>.
<p>4. Consistência e padrões</p>

<ul style="list-style-type: none"> usuários não precisam adivinhar que diferentes palavras, situações ou ações significam a mesma coisa. Seguir convenções de plataforma computacional.
<p>5. Prevenção de erros</p> <ul style="list-style-type: none"> melhor que uma boa mensagem de erro é um design cuidadoso o qual previne o erro antes dele acontecer.
<p>6. Reconhecimento ao invés de relembração</p> <ul style="list-style-type: none"> tornar objetos, ações e opções visíveis. O usuário não deve ter que lembrar informação de uma para outra parte do diálogo. Instruções para uso do sistema devem estar visíveis e facilmente recuperáveis quando necessário.
<p>7. Flexibilidade e eficiência de uso</p> <ul style="list-style-type: none"> usuários novatos se tornam peritos com o uso. Prover aceleradores de formar a aumentar a velocidade da interação. Permitir a usuários experientes "cortar caminho" em ações frequentes.
<p>8. Estética e design minimalista</p> <ul style="list-style-type: none"> diálogos não devem conter informação irrelevante ou raramente necessária. Qualquer unidade de informação extra no diálogo irá competir com unidades relevantes de informação e diminuir sua visibilidade relativa.
<p>9. Ajudar os usuários a reconhecer, diagnosticar e corrigir erros</p> <ul style="list-style-type: none"> mensagens de erro devem ser expressas em linguagem clara (sem códigos) indicando precisamente o problema e construtivamente sugerindo uma solução.
<p>10. Help e documentação</p> <ul style="list-style-type: none"> embora seja melhor um sistema que possa ser usado sem documentação, é necessário prover <i>help</i> e documentação. Essas informações devem ser fáceis de encontrar, focalizadas na tarefa do usuário e não muito extensas.

Fonte: ROCHA; BARANAUSKAS (2003)

3.2 MÉTODOS

Para o desenvolvimento do código escrito em *Python*, foi utilizado editor de código-fonte *Visual Studio Code*, pois conta com um suporte para depuração de código, versionamento *Git*, realce de sintaxe, e uma vasta biblioteca de extensões que auxiliaram durante o desenvolvimento do aplicativo.

O repositório em nuvem *Github* foi usado como ferramenta para controle e versionamento de código. Ele permitiu o desenvolvimento contribuído e dinâmico do aplicativo.

O código escrito foi integrado com o *SQLAlchemy* para estabelecer a conexão com o banco *SQL* da *Azure*, este código também teve em sua estrutura trechos e funções que fizeram a divisão de renda escolhida pelo usuário durante o uso. Esta divisão pode ser customizada posteriormente conforme o desejado.

O banco *SQL* da *Azure* é o responsável por estabelecer a persistência de dados do usuário, como a renda, as divisões desejadas, bem como os dados pessoais deste.

3.3 Modelagem

Para o desenvolvimento do projeto, foram utilizados os conhecimentos demonstrados nas aulas da disciplina engenharia de software no decorrer do curso. Foi necessário iniciar o projeto em um estágio conceitual, onde foram desenvolvidos os requisitos e toda a modelagem através de diagramas UML feitos através da plataforma *Lucidchart*.

O projeto foi desenvolvido com a metodologia ágil Scrum, através desta metodologia ágil, foi possível uma melhor organização, estabelecimento dos prazos e objetivos, para que as sprints pudessem ser concluídas paralelamente de uma forma organizada.

Na grande maioria dos processos de engenharia de software, o processo inicial que auxilia na clareza de ideias, é a modelagem. Através da modelagem é possível enxergar o sistema como um todo e planejar onde será o início do desenvolvimento do programa. Segundo Sommerville (2011), os melhores modelos são aqueles que são mais simples, que deixam de lado os detalhes, para assim extrair as características mais salientes do projeto, tendo em mente que um modelo é uma abstração do sistema final.

Portanto no projeto de desenvolvimento do *back-end* do aplicativo Myfinanz, a modelagem foi desenvolvida em três etapas: levantamento de requisitos funcionais e não funcionais, diagrama de caso de uso, descrição do diagrama de caso de uso e diagrama de classes.

Por fim, foi confeccionado o banco de dados físico na plataforma do *Azure SQL Database* bem como um levantamento de seus custos envolvendo um provisionamento para produção da aplicação. Através deste, foi possível implementar as características e necessidades levantadas durante a modelagem.

3.3.1 Requisitos Funcionais e Não Funcionais

Primordialmente, foram estabelecidos todos os requisitos que o sistema teria e quais necessidades o aplicativo iria suprir. Foi de suma importância a definição destes pontos logo de início, pois auxiliou a consolidar as ideias e direcionar a prioridade do desenvolvimento.

“Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições ao seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações” (Sommerville, p.71, 2011).

Dito isso, os requisitos funcionais e não funcionais, estão descritos na tabela abaixo:

Tabela 1- Descrição de requisitos do Back-End

Requisitos Funcionais		Requisitos não Funcionais	
RF01	Fazer login	RNF01	Manutenção do aplicativo
RF02	Fazer Cadastro	RNF02	<i>Back-End</i> em <i>Python</i>
RF03	Adicionar despesas	RNF03	Consultas responsivas e rápidas.
RF04	Categorizar despesas	RNF04	Interface simples e intuitiva.
RF05	Indicar se a despesa já foi paga	RNF05	Acessível para pessoas com menos habilidade com tecnologia ou em planejamentos financeiros.
RF06	Adicionar receita	RNF06	Utilização do banco de dados na nuvem <i>Azure</i> .
RF07	Categorizar receitas		
RF08	Adicionar cartões de crédito		

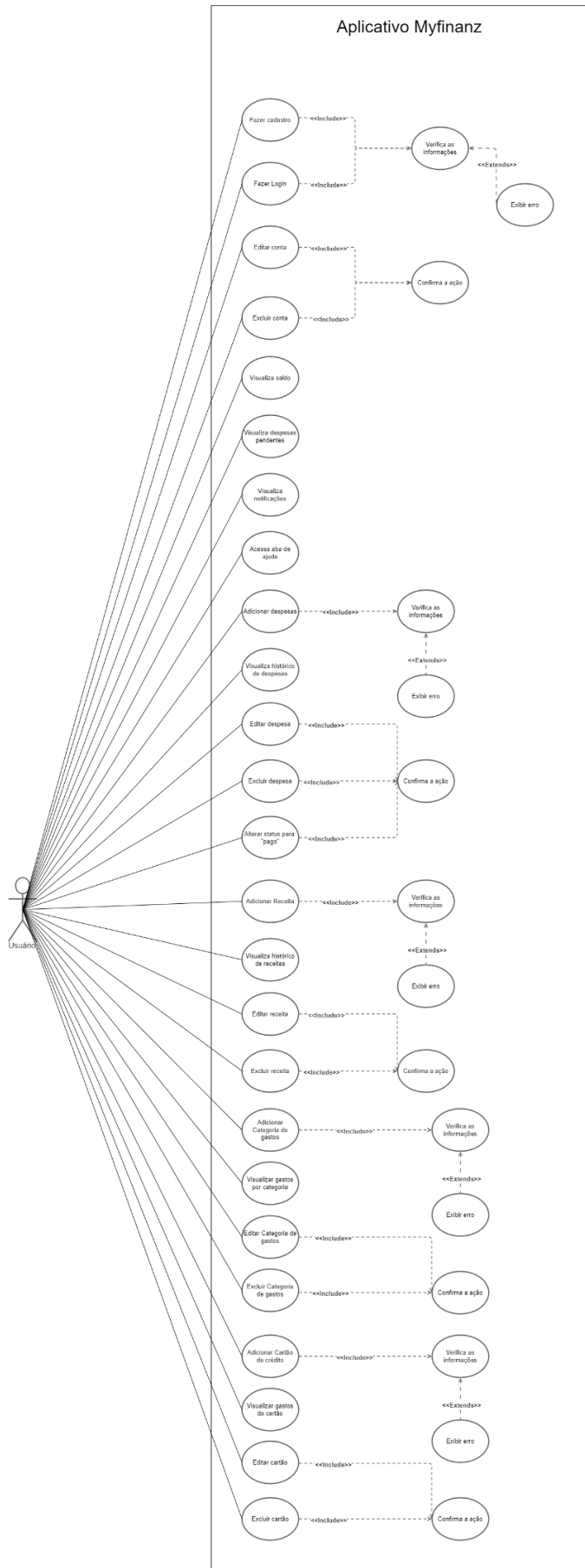
RF09	Visualizar o total a pagar do cartão de crédito		
RF10	Visualizar saldo atual da conta		
RF11	Visualizar despesas pendentes		
RF12	Visualizar histórico de despesas por categoria		
RF13	Visualizar histórico de receitas		
RF14	Adicionar novas categorias de gasto		
RF15	Traçar objetivos		
RF16	Estabelecer limites para cada categoria de gasto		
RF17	Visualização de notificações		
RF18	Editar conta		
RF19	Excluir conta		
RF20	Visualização de tutoriais para usar o aplicativo		
RF21	Excluir despesa		
RF22	Excluir receita		
RF23	Editar despesa		
RF24	Editar receita		
RF25	Editar cartão		
RF26	Excluir cartão		
RF27	Editar objetivo		
RF28	Excluir objetivo		

3.3.2 UML – Diagrama de Caso de Uso

Segundo Booch et al (p.13, 2005), a Unified Modeloing Language (UML) é uma linguagem padrão para a visualização, especificação, construção e documentação da estrutura de projetos de software. Uma das vantagens de construir um diagrama deste modelo, foi a possibilidade de construir um modelo que permitisse de fato como a aplicação final iria ser programada.

No modelo construído foi incluído um ator, cujo nome é “usuário”. O diagrama visa exemplificar as ações possíveis de um utilizador dentro do aplicativo Myfinanz.

Figura 1 - Diagrama de casos de uso



3.3.3 Descrição do Diagrama de Caso de Uso

Após a confecção do diagrama de Casos de Uso, foram levantadas as condições e o fluxo básico de navegação da aplicação através de uma tabela descritiva abaixo.

Esta tabela, também chamada de Descrição de Casos de Uso, é necessária para a compreensão geral das condições em que cada requisito funcional do projeto estará sujeito ao entrar em contato com uma ação do usuário.

Tabela 2 - Descrição do diagrama de caso de uso

Caso de uso	Referência	Pré-condições	Fluxo básico
Fazer cadastro	RF02	Aplicativo baixado no dispositivo e conexão com a internet	1 – Usuário entra na tela de cadastro 2 – Preenche os campos corretamente 3 – Pressiona o botão “cadastrar”
Fazer Login	RF01	Aplicativo baixado no dispositivo, conexão com a internet e conta previamente criada	1 – Usuário entra na tela de login 2 – Preenche os campos corretamente 3 – Pressiona o botão “entrar”
Editar conta	RF18	Aplicativo baixado no dispositivo, conexão com a internet e conta previamente criada	1 – Usuário faz login 2 – Entra na tela de ajustes 3 – Edita a informação que deseja 4 – Confirma a alteração
Excluir conta	RF19	Aplicativo baixado no dispositivo, conexão com a internet e conta previamente criada	1 – Usuário faz login 2 – Entra na tela de ajustes 3 – Pressiona o botão “Excluir conta” 4 – Confirma a alteração
Visualizar saldo	RF10	Aplicativo baixado no dispositivo, conexão com a internet e conta previamente criada	1 – Usuário faz login 2 – Visualiza na tela principal
Visualizar despesas pendentes	RF11	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada e adicionar as despesas previamente	1 – Usuário faz login 2 – Visualiza na tela principal
Visualiza notificações	RF17	Aplicativo baixado no dispositivo, conexão com a	1 – Usuário faz login 2 – Pressiona o ícone de notificações

		internet e conta previamente criada	
Acessa aba de ajuda	RF20	Aplicativo baixado no dispositivo, conexão com a internet e conta previamente criada	1 – Usuário faz login 2 – Pressiona o ícone de ajustes
Adicionar despesas	RF03, RF05	Aplicativo baixado no dispositivo, conexão com a internet e conta previamente criada	1 – Usuário faz login 2 – Pressiona o botão “Adicionar despesas” 3 – Preenche os campos corretamente 4 – Pressiona o botão “Adicionar”
Visualizar histórico de despesas	RF11, RF12	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada e adicionar as despesas previamente	1 – Usuário faz login 2 – Pressiona o card de despesas
Editar despesa	RF05, RF23	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada e adicionar as despesas previamente	1 – Usuário faz login 2 – Pressiona o card de despesas 3 – Identifica a despesa 4 – Aperta o ícone de editar 5 – Confirma a alteração
Excluir Despesa	RF21	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada e adicionar as despesas previamente	1 – Usuário faz login 2 – Pressiona o card de despesas 3 – Identifica a despesa 4 – Aperta o ícone de excluir 5 – Confirma a alteração
Alterar status para: “pago”	RF05, RF23	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada e adicionar as despesas previamente com status de não pago.	1 – Usuário faz login 2 – Pressiona o card de despesas 3 – Identifica a despesa 4 – Aperta o ícone de editar 5 – Confirma a alteração
Adicionar receita	RF06	Aplicativo baixado no dispositivo, conexão com a internet e conta previamente criada	1 – Usuário faz login 2 – Pressiona o botão “Adicionar receitas” 3 – Preenche os campos corretamente 4 – Pressiona o botão “Adicionar”
Visualizar histórico de receita	RF07, RF13	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada e adicionar as receitas previamente	1 – Usuário faz login 2 – Pressiona o card de receitas
Editar receita	RF24	Aplicativo baixado no dispositivo, conexão com a	1 – Usuário faz login

		internet, conta previamente criada e adicionar as receitas previamente	2 – Pressiona o card de receita 3 – Identifica a receita 4 – Aperta o ícone de editar 5 – Confirma a alteração
Excluir receita	RF22	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada e adicionar as receitas previamente	1 – Usuário faz login 2 – Pressiona o card de receitas 3 – Identifica a receita 4 – Aperta o ícone de excluir 5 – Confirma a alteração
Adicionar categoria de gastos	RF04, RF14	Aplicativo baixado no dispositivo, conexão com a internet e conta previamente criada	1 – Usuário faz login 2 – Pressiona o botão “categorias” 3 – Preenche os campos corretamente 4 – Pressiona o botão “Adicionar”
Visualizar gastos por categoria	RF12	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada, adicionar as despesas previamente e adicionar categorias previamente	1 – Usuário faz login 2 – Pressiona o card de gastos por categoria
Editar categoria de gastos	RF14, RF16	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada, adicionar as despesas previamente e adicionar categorias previamente	1 – Usuário faz login 2 – Pressiona o card de categorias 3 – Identifica a categoria 4 – Aperta o ícone de editar 5 – Confirma a alteração
Excluir categoria de gastos	RF14, RF16	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada, adicionar as despesas previamente e adicionar categorias previamente	1 – Usuário faz login 2 – Pressiona o card de categorias 3 – Identifica a categoria 4 – Aperta o ícone de excluir 5 – Confirma a alteração
Adicionar Cartão de crédito	RF08	Aplicativo baixado no dispositivo, conexão com a internet e conta previamente criada	1 – Usuário faz login 2 – Pressiona o botão “cartões” 3 – Preenche os campos corretamente 4 – Pressiona o botão “Adicionar”
Visualizar gastos de cartão	RF09	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada, cartão previamente criado e adicionar as despesas previamente na categoria cartão	1 – Usuário faz login 2 – Pressiona o botão “cartões”

Editar cartão	RF25	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada e cartão previamente criado	1 – Usuário faz login 2 – Pressiona o botão “cartões” 3 – Identifica o cartão 4 – Aperta o ícone de editar 5 – Confirma a alteração
Excluir cartão	RF26	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada e cartão previamente criado	1 – Usuário faz login 2 – Pressiona o botão “cartões” 3 – Identifica o cartão 4 – Aperta o ícone de excluir 5 – Confirma a alteração
Adicionar Objetivo	RF15	Aplicativo baixado no dispositivo, conexão com a internet e conta previamente criada	1 – Usuário faz login 2 – Pressiona o botão “objetivos” 3 – Preenche os campos corretamente 4 – Pressiona o botão “Adicionar”
Editar objetivo	RF27	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada e objetivo criado	1 – Usuário faz login 2 – Pressiona o botão “objetivos” 3 – Identifica o objetivo 4 – Aperta o ícone de editar 5 – Confirma a alteração
Excluir objetivo	RF28	Aplicativo baixado no dispositivo, conexão com a internet, conta previamente criada e objetivo criado	1 – Usuário faz login 2 – Pressiona o botão “objetivos” 3 – Identifica o objetivo 4 – Aperta o ícone de excluir 5 – Confirma a alteração

3.3.4 UML – Diagrama de Classes

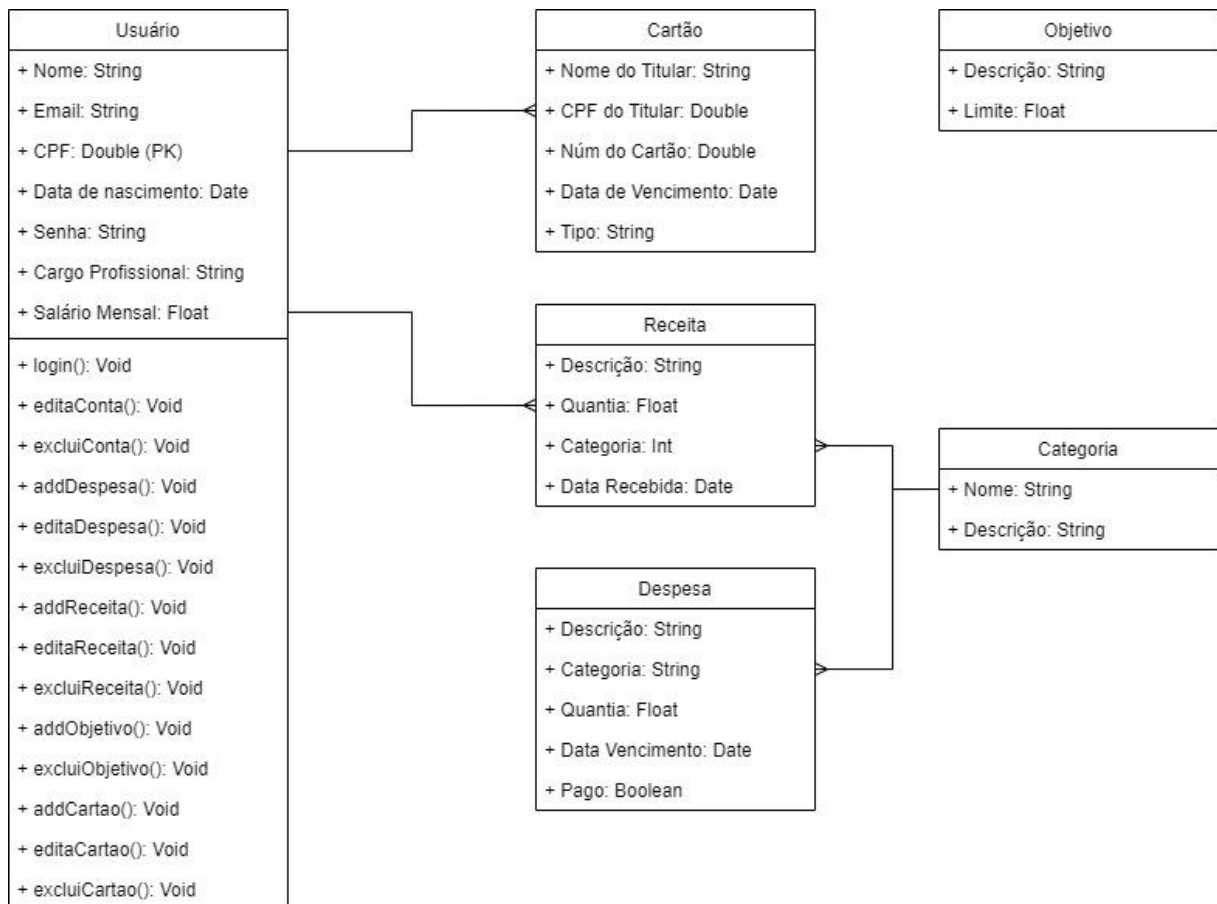
O Diagrama de Classes é um diagrama UML que descreve e representa graficamente como será a estrutura da aplicação e dos objetos que a compõem.

O diagrama confeccionado estabelece todos os atributos e métodos contidos dentro do banco de dados do sistema. Nele, cada um dos objetos possui atributos e métodos que são utilizados durante o uso do aplicativo.

A principal entidade neste esquemático é a de Usuário, pois é ele quem vai realizar todas as inserções, edições ou remoções das informações necessárias dentro do aplicativo. Outras

funções como a geração de gráficos e tabelas para o planejamento financeiro do usuário, não são intrínsecas do diagrama de classes, estas sendo confeccionadas via código *Python* em conjunto da biblioteca *ORM SQLAlchemy*.

Figura 2 - Diagrama de classes do banco de dados



3.3.5 Banco de Dados na nuvem *Azure*

Com o levantamento de requisitos concluídos, bem como o diagrama de caso de uso, suas descrições e o desenvolvimento do diagrama de classe, foi então criado um banco de dados SQL na nuvem do *Azure*.

Todas as etapas aqui previamente citadas foram cruciais para o desenvolvimento deste banco. As etapas que dizem respeito aos requisitos, casos de uso e principalmente o diagrama de classes, são uma peça-chave para o desenvolvimento de um banco conciso, coerente e eficaz.

Cada tabela teve seu propósito e função descritos abaixo.

Figura 3 - Interface do Azure SQL Database

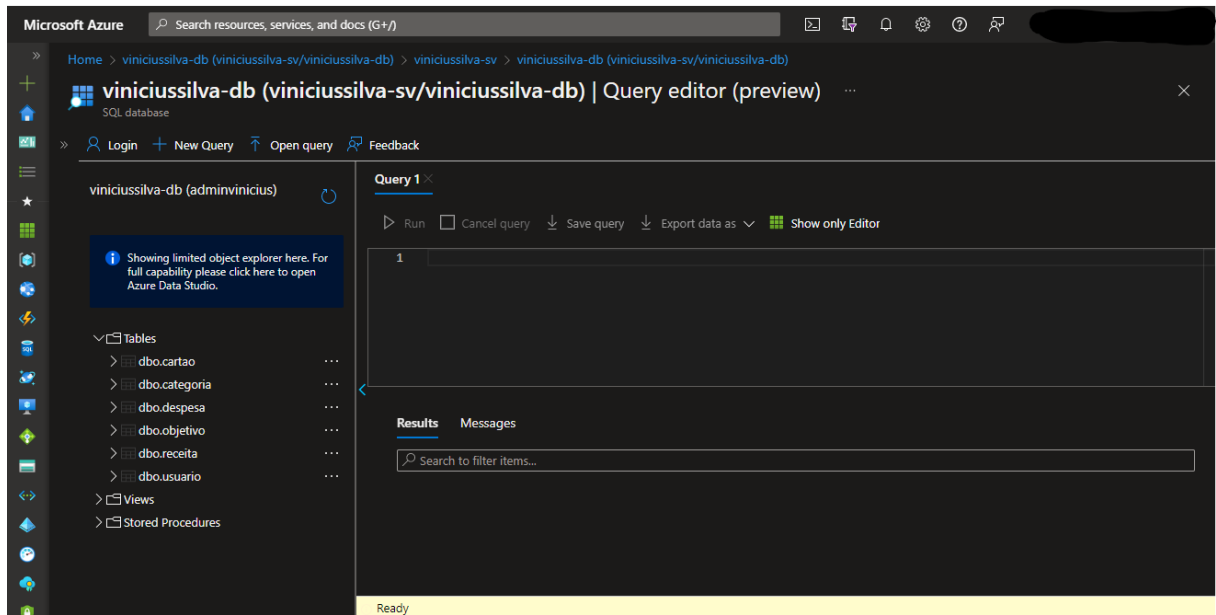


Figura 4 - Tabela de registro dos cartões

dbo.cartao	
Nome_Titular	(varchar, null)
Email	(varchar, null)
CPF_Titular	(bigint, not null)
Numero_Cartao	(bigint, null)
Data_Vencimento	(varchar, null)
Tipo	(varchar, null)

Figura 5 - Tabela de categorias

dbo.categoria	
id	(PK, int, not null)
Nome	(varchar, null)
Descricao	(varchar, null)

Figura 6 - Tabela de registro de objetivos

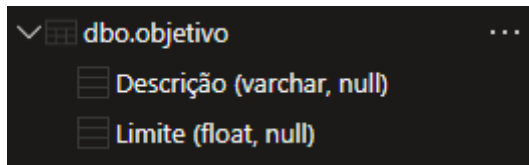


Figura 7 - Tabela da receita

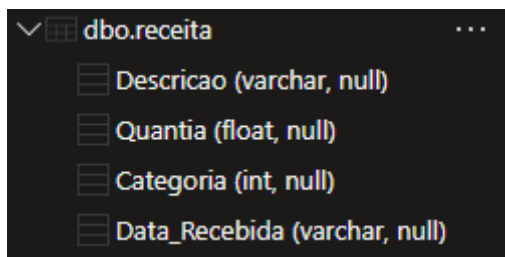
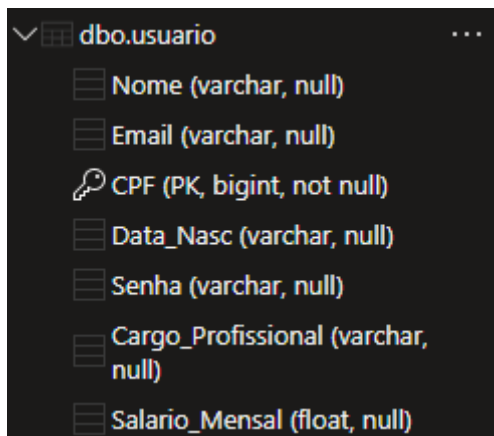


Figura 8 - Tabela de registro de usuários



3.3.6 Estimativas de custo do banco de dados

Utilizando a Calculadora de Preços do Azure, foi possível estimar os custos mensais envolvendo o banco de dados *SQL* provisionado.

Figura 9 - Estimativas de custo do banco de dados Azure SQL Database

Microsoft Azure Estimate				
Banco remoto MyFinanz				
Service category	Service type	Region	Description	Estimated monthly cost
Bancos de dados	Azure SQL Database	Brazil South	Banco de dados único, vCore, Armazenamento de Backup LRS, Propósito geral, Provisionado, Série Standard (Gen 5), Localmente Redundante, 1 - 2 vCore Banco(s) de dado x 31 Dias, Armazenamento de 32 GB, 0 GB de armazenamento de backup	R\$3.113,83
		Licensing Program	Microsoft Customer Agreement (MCA)	
		Billing Account		
		Billing Profile		
		Total		R\$ 3.113,83

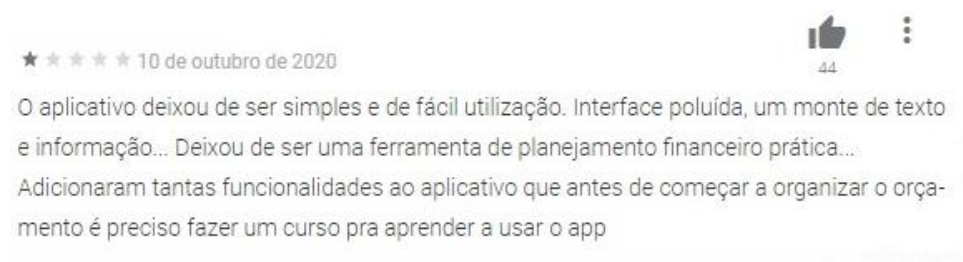
A estimativa levou em conta a quantidade inicial de usuários do aplicativo, que inicialmente seriam poucos antes de entrar em produção. Foi decidido a não utilização de backup, visto que isso traria um custo adicional na mensalidade, fora o fato de o banco e seus dados estarem protegidos devido à redundância do recurso em outro datacenter da *Azure*.

A redundância é a cópia do recurso, neste caso o *SQL Database*, em um outro datacenter do mesmo provedor, fazendo com que o recurso esteja sempre disponível, sem interrupções ou perdas de dados.

4 RESULTADOS E DISCUSSÃO

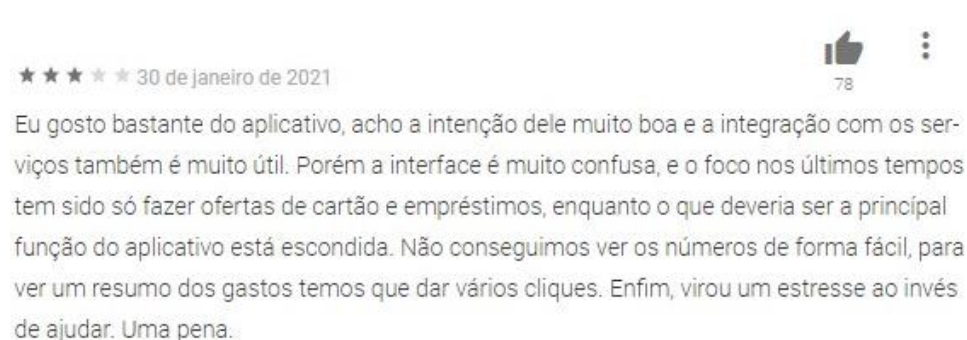
Como visto anteriormente, os índices de pessoas endividadas no Brasil são grandes e em constante crescimento, para auxiliar no entendimento desta problemática, no desenvolvimento deste projeto foi necessário analisar o cenário da educação financeira e aspectos sociológicos econômicos de maneira breve para que, no futuro seja possível a criação de um aplicativo, com o intuito de ajudar no planejamento financeiro dos usuários. Apesar do grande número de sistemas existentes com tal finalidade, o mercado carece de soluções com interfaces simples, intuitivas, acessíveis e gratuitas, como mostram as figuras 10, 11, 12 e 13 listadas abaixo, onde foram coletadas avaliações de alguns dos mais famosos sistemas mobile no ramo de planejamento financeiro:

Figura 10 – Reclamação da interface e usabilidade no aplicativo.



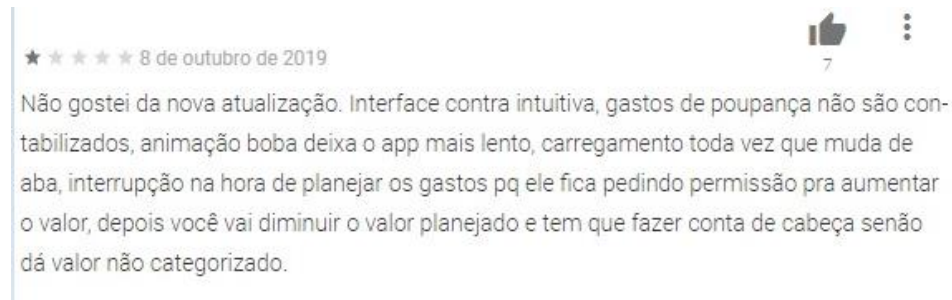
Fonte: Google play

Figura 11 - Reclamação de interface



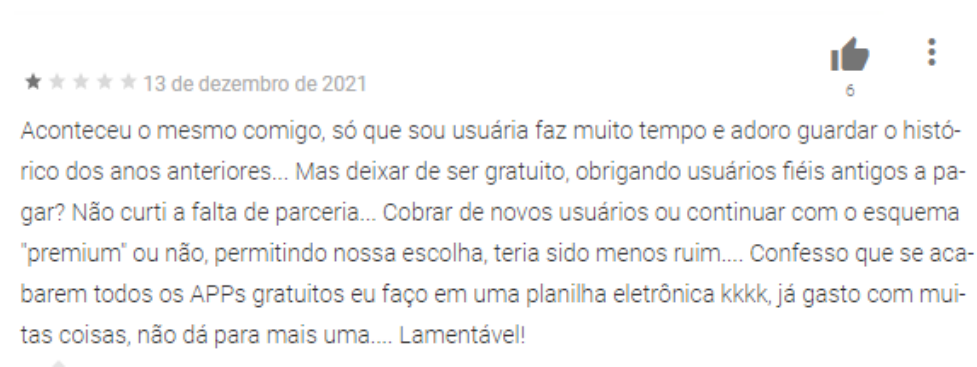
Fonte: Google play

Figura 12- Reclamação de interface



Fonte: Google play

Figura 13 – Reclamação do aplicativo ao sair do gratuito para versão paga



Fonte: Google play

Deste modo, com tais avaliações e com os estudos feitos anteriormente, a ideia de criação de um aplicativo com estes objetivos se mostrou viável, promissora e benéfica para a população, já que esta incentiva a superação do receio de tratar sobre suas finanças pessoais.

Levando em conta tais avaliações negativas dos concorrentes, o aplicativo MyFinanz se mostrou extremamente útil para as necessidades dos usuários. Todo o *back-end* passou pelas principais confecções e análise de utilização de recurso de performance, tais como: Diagrama de caso de uso, diagrama de classe, escolha e confecção de um banco de dados remoto, além da seleção de alguns participantes de diferentes idades para o uso da plataforma, e na grande maioria das ocasiões a performance do aplicativo foi muito bem aceita. Para os usuários convidados, foi criado um formulário com algumas perguntas simples, para receber as impressões dos utilizadores, os resultados estão presentes nas figuras 14, 15, 16, 17 e 18 abaixo.

Figura 14 - Primeira pergunta

Você possui algum tipo de dívida?

6 respostas

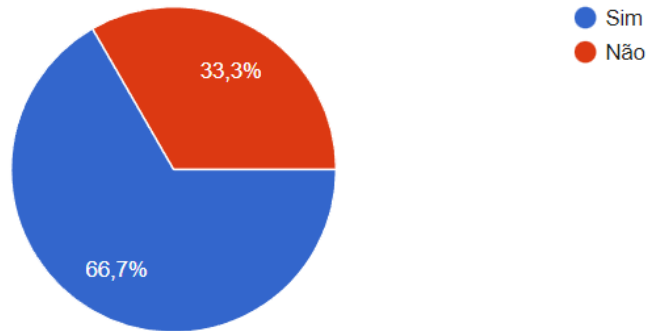


Figura 15 - Segunda pergunta

Já tentou utilizar algum aplicativo e obteve problemas com performance?

6 respostas

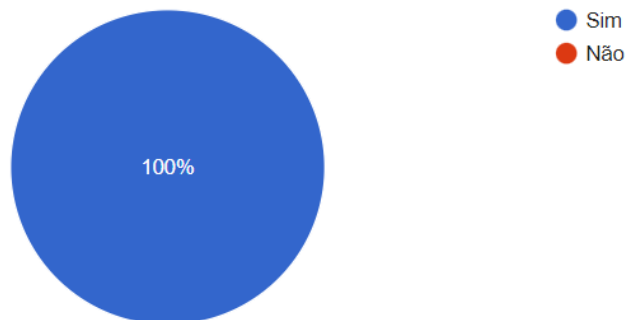


Figura 16 - Terceira pergunta

Após o teste do aplicativo MyFinanz, você o utilizaria novamente?

6 respostas

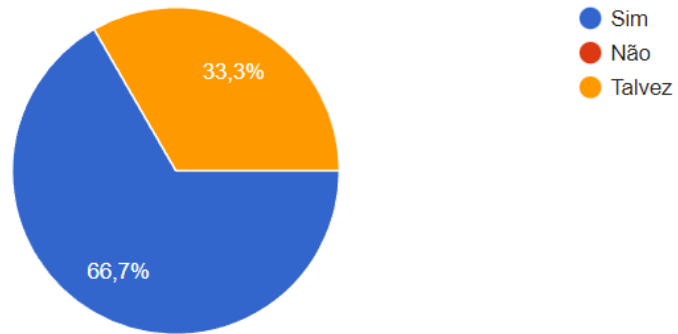


Figura 17 - Quarta pergunta

Você sentiu que a responsividade (performance/velocidade) do aplicativo estava dentro das suas expectativas?

6 respostas

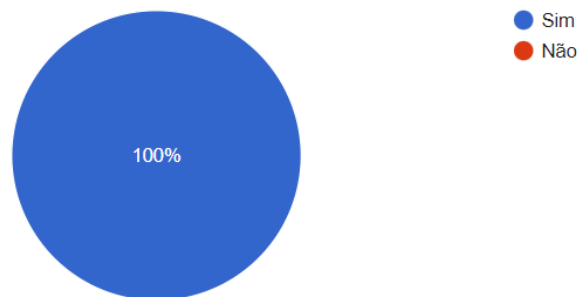
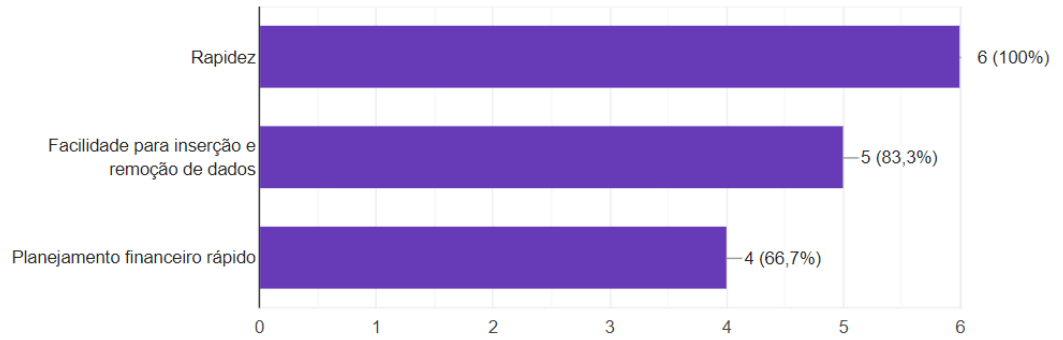


Figura 18 - Quinta e última pergunta

Após o uso do aplicativo, quais foram as questões envolvendo a parte performática que mais ficaram evidenciadas?

6 respostas



5 CONSIDERAÇÕES FINAIS

O objetivo deste trabalho, foi mostrar a viabilidade da construção do *back-end* de um aplicativo *mobile*, para o controle de finanças pessoais. Para alcançar tal objetivo, foi necessário evidenciar os números das pessoas endividadas no Brasil, além de mostrar de maneira breve os pontos positivos do planejamento financeiro e como a tecnologia pode facilitar esta tarefa. Também foi realizada uma pesquisa de mercado procurando os principais pontos de melhoria em aplicativos no segmento de finanças pessoais.

Dados estes objetivos, foi comprovado que de fato, um dos pilares para a saúde financeira da população é o planejamento de capital pessoal, onde tem o papel de auxiliar o planejador a entender onde estão seus acertos e erros no pagamento das suas contas no decorrer do mês. O planejamento quando feito com seriedade e compromisso, possibilita a quitação de dívidas, liberdade financeira, entre diversos outros objetivos.

Ainda por cima, a tecnologia foi apresentada como fator facilitador destas tarefas de mapeamento de finanças, já que celulares e dispositivos móveis vieram para automatizar tarefas manuais. A popularização e acessibilidade de dispositivos móveis cresceram de maneira exponencial no mundo todo, no Brasil não foi diferente, inclusive dados mostram que existe a proporção de 2 aparelhos para um habitante, segundos dados da Fundação Getúlio Vargas (FGV).

Visto que a acessibilidade de dispositivos móveis não é um problema que inviabilize o desenvolvimento do sistema, foram procuradas avaliações de aplicativos famosos de controle de finanças, com o objetivo de buscar pontos de melhorias nos mesmos. Com isso foram identificadas diversas reclamações sobre a interação com as interfaces, a complexidade dos layouts, além da escassez de recursos gratuitos e completos.

Dessa forma, foi evidenciado que existe a viabilidade do desenvolvimento de um novo sistema, que possui em seu *back-end*, uma performance que visa otimizar a experiência de usabilidade do aplicativo a ser criado. Tudo isso de uma forma gratuita e baseadas nas heurísticas de Nielsen, com o objetivo principal de ajudar a população, principalmente aqueles que possuem baixa afinidade com tecnologia, com os problemas de planejamento financeiro.

Em relação ao desenvolvimento do aplicativo MyFinanz, no qual foi realizado com absoluto êxito, ficou evidente a utilização de tecnologias alinhadas com o mercado atual para

obtermos uma boa experiência de usuário, tecnologias tais como o *Azure SQL Database* e o sistema de versionamento de código remoto *GitHub* foram essenciais para um bom aproveitamento e extração de todo potencial que o aplicativo pôde ter para ser criado.

Dito isto, a performance de tal aplicativo também foi diretamente afetada por estas decisões de utilização de recurso, visto que tais decisões foram cruciais para obter uma base performática eficaz e de alta responsividade.

REFERÊNCIAS BIBLIOGRÁFICAS

BORGES, Luiz Eduardo. **Python para desenvolvedores**: Aborda Python 3.3. Rio de Janeiro: Novatec, 2014

CALIXTO, L. A.; SOUZA, D. S.; SANTOS, F. K. G. PLANEJAMENTO DE FINANÇAS PESSOAIS: UM ENSAIO PARA UMA VIDA ECONÔMICA SEGURA. **Caderno de Graduação - Ciências Humanas e Sociais - UNIT - SERGIPE**, [S. l.], v. 7, n. 2, p. 83, 2022. Disponível em: <https://periodicos.set.edu.br/cadernohumanas/article/view/8256>. Acesso em: 11 maio. 2022.

Globo.com. Disponível em: <https://g1.globo.com/economia/noticia/2022/03/31/endividamento-bate-recorde-em-marco-puxado-pelo-cartao-de-credito-diz-cnc.ghtml>. Acessado em 14 abril de 2022.

LEAL, Cicero; NASCIMENTO, José. Planejamento Financeiro Pessoal. **Revista de Ciências Gerenciais**. Brasília, v. 15, n. 22, jul. 2015.

MAIA, JS; MARIN, HF. Aplicativos móveis para as sociedades menos favorecidas. **Acta Paul Enferm.** 2021;34:eAPE002214

MOTA, Carla; CHIMELO, Géssica; BENDER, Cíntia; NESPOLO, Daniele; BORELLI, Verena; FACHINELLI, Ana. Organização Financeira Pessoal: Análise dos fatores que influenciam no endividamento e inadimplência dos jovens. **Revista Eletrônica de Ciências Sociais Aplicadas - RECSA** . Garibaldi, RS, Brasil. v.4, n.1, jan/jun, 2015.

NIELSEN, Jakob. 10 Usability Heuristics for User Interface Design. 1995.

Disponível em: <http://www.nngroup.com/articles/ten-usability-heuristics/>. Acessado em: 20 de Abril de 2022.

NIELSEN, Jakob; BUDIU, Raluca. Usabilidade Móvel. Rio de Janeiro: Elsevier, 2014.

NIELSEN, Jakob. How to Conduct a Heuristic Evaluation. 1995. Disponível em:

<<http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>>. Acessado em: 20 de abril de 2022.

NIELSEN, Jakob; LORANGER, Hoa. Usabilidade na Web: Projetando Websites com qualidade. Rio de Janeiro: Elsevier, 2006.

NIELSEN, Jakob. Usability 101: Introduction to Usability. 2012. Disponível em <<http://www.nngroup.com/articles/usability-101-introduction-to-usability/>>. Acessado em: 20 de abril de 2022.

NIELSEN, Jakob. Usability Engineering. San Diego (CA): Academic Press, 1993.

PICCINI, R.A.; PINZETTA, G. Planejamento Financeiro Pessoal e Familiar. **Unoesc & Ciência - ACSA**, Joaçaba, v. 5, n. 1, p. 95-102, jan./jun. 2014.

Radio Agência Nacional. Disponível em: <<https://agenciabrasil.ebc.com.br/radioagencia-nacional/economia/audio/2022-03/quase-78-da-populacao-esta-endividada-no-brasil>>. Acessado em 13 de abril de 2022.

RIBEIRO, Rodrigo; LARA, Ricardo. O endividamento da classe trabalhadora no Brasil e o capitalismo manipulatório, **Serv. Soc. Soc.**, São Paulo, n. 126, p. 340-359, maio/ago. 2016.

ROCHA, H. V. da; BARANAUSKAS, M. C. C. **Design e Avaliação de Interfaces Humano-Computador**. São Paulo: NIED/Unicamp, 2003. E-book. Disponível em: <https://www.nied.unicamp.br/biblioteca/design-e-avaliacao-de-interfaces-humano-computador/>. Acesso em: 17 maio. 2022.

SOARES, Michael. Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software. **Revista Eletrônica de Sistemas de Informação**. Jan-Abr 2018, v.17, n.1

BRASIL. Confederação Nacional do Comércio de Bens, Serviços e Turismo (CNC). **Pesquisa de Endividamento e Inadimplência do Consumidor (Peic)**. Abril de 2022. 3 p.

SOMMERVILLE, Ian. **Engenharia de software**. 9 ed. São Paulo: Person Education, 2011.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML Guia do usuário**. 2 ed. São Paulo: Elsevier Editora Ltda, 2005.