

Revista Científica

FACULDADE ATENAS- PARACATU-MG

Ano 2023, V.16, N.1



FACULDADE
ATENAS

www.atenas.edu.br

38 3672-3737

DESENVOLVIMENTO DE UM SISTEMA PARA GESTÃO FINANCEIRA DE REPÚBLICAS UNIVERSITÁRIAS: software web destinado a gestão financeira

Dieverson Dos Santos Rocha
Romério Ribeiro Da Silva
Andressa Cristina De Souza Almeida
Francielle Alves Marra
Thais Dias dos Santos
Cristiano André Peixoto

RESUMO

Com o crescente avanço da tecnologia, cresceu exponencialmente a quantidade de soluções tecnológicas que auxiliam no dia a dia das pessoas. O presente trabalho tem como objetivo propor mais uma dessas soluções tecnológicas. Com foco em estudantes universitários residentes em repúblicas universitárias, este trabalho propõe uma solução para auxiliar no controle dos gastos financeiros visando um melhor controle destes gastos que normalmente não são feitos, ou se exercido são elaborados de forma manual com a utilização de anotações ou até mesmo planilhas. O sistema será sustentado através de entrada de dados por parte do usuário tais como despesas, usuários integrantes da república, novas repúblicas. Estes dados são processados pelo sistema e retornados de forma compartilhada entre os usuários integrantes da república cadastrada e feito a divisão de despesas compartilhadas entre os integrantes da república.

Palavras-chave: Sistemas; Controle Financeiro; Despesas; Repúblicas Universitárias

ABSTRACT

With the growing advancement of technology, the number of technological solutions that help people's daily lives grow exponentially. The present work aims to propose one more of these technological solutions. With a focus on university students residing in university republics, this work proposes a solution to assist in the control of financial expenses, a better control of these expenditures that are not normally made, or if exercised, they are prepared manually with the use of annotations or even spreadsheets. The system will be supported through data entry by the user such as expenses, users belonging to the republic, new republics. These data are processed by the system and returned in a shared way among the users who are members of the registered republic and the division of expenses shared between the members of the republic is made.

Keywords: Systems; Financial control; Expenses; University Republics.

1 INTRODUÇÃO

As Tecnologias da Informação e Comunicação (TIC) no século 21 está sendo um dos se não o grande marco da humanidade, levando praticidade e acessibilidade para a população, mesmo que indiretamente. Serviços e atividades do cotidiano vêm mudando para melhor com o avanço das TIC's, tudo isso graças as tecnologias digitais como: computadores, notebooks, celulares e tablet's.

Segundo uma pesquisa realizada pelo IBGE em 2018, destaca que a utilização da internet é a maior entre as pessoas de 18 a 29 anos. Parte desses jovens estão hoje em uma universidade, e segundo censo do Data Fala! 30% desses estudantes ao ingressar no ensino superior passam a morar sozinhos ou com amigos em repúblicas. Tendo assim que passar a ter responsabilidades que muito das vezes até então esses estudantes não tinham como contratação de serviços, aluguéis, contas de água, energia, gás, despesas com comida e entre outros. Com tantas responsabilidades em mãos e juntando o fator de inexperiência na vida adulta, no caso das repúblicas, ocorrem alguns problemas e desavenças na questão financeira ao ter que cumprir essas responsabilidades.

Tendo em vistas estes problemas que são gerados em repúblicas universitárias decorridos da falta de gestão financeira por parte de seus integrantes, será desenvolvido um sistema onde os integrantes da república poderão gerenciar e organizar essas responsabilidades mensais com gastos dentro da república, evitando assim problemas e desavenças financeiras dentro da república.

1 REVISÃO BIBLIOGRAFICA

Neste capítulo será apresentado a revisão bibliográfica acerca do tema do presente trabalho, onde será dividido em três partes, sendo elas: O surgimento das repúblicas universitárias e identificação do público-alvo que fará a utilização do sistema. A revisão bibliográfica acerca das ferramentas tecnológicas que serão utilizadas no desenvolvimento do sistema, e pôr fim a importância do controle financeiro pessoal.

a. REPÚBLICAS UNIVERSITÁRIAS

O termo república deriva do termo latim *res publica* que tem como significado “coisa do povo”, “coisa pública” que se refere a algo que é mantida por mais de uma pessoa.

O surgimento das repúblicas no Brasil deu se início em 1897, quando a capital do Estado de Minas Gerais transferiu-se para Belo Horizonte, que levou ao abandono de diversas casas devido a um esvaziamento econômico na cidade de Ouro Preto (Antiga capital do estado de Minas Gerais). Então estudantes das Escolas de Minas e de Farmácia ocuparam essas casas abandonadas e a conservaram-nas (SARDI, 2000).

Segundo uma pesquisa do Instituto de Pesquisa Econômica Aplicada (Ipea) o acesso ao ensino superior no Brasil teve um número elevado de matrículas onde em 1991 era de 1,5 milhão e já em 2012 alcançou 7,1 milhões, alcançando um aumento da taxa bruta de matrícula de cerca de 30%.

O público-alvo deste trabalho são esses estudantes que com a crescente procura pelos jovens de ingressar ao ensino superior, e a falta de instituições de ensino em suas cidades, acabam optando pelo ingresso em repúblicas universitárias em cidades que tem instituições de ensino para assim ingressar no ensino superior. E que ao fazer parte de uma república universitária percebe a necessidade de adaptar-se e passar pela transição de jovem para adultos, assumindo assim novas responsabilidades e interdependência no contexto interrelacional de viver-com-viver (SILVA, 2012)

b. FERRAMENTAS TECNOLÓGICAS

Neste tópico serão abordadas as ferramentas tecnológicas necessárias para o desenvolvimento do sistema, tais como linguagens, arquitetura, framework e banco de dados.

2.2.1 PADRÃO MVC

Segundo Massari (2017), Model View Controller (MVC) é um padrão de arquitetura de softwares que separa as várias camadas da aplicação, que passam à assumir responsabilidades individuais na interação com o usuário. Deste modo, as Models tornam-se as responsáveis por tratar os dados da aplicação, como constantes, entidades, e regras de negócios. As Views assumem o papel de exibição da parte visual ao usuário, que são representadas normalmente por telas dos sistemas. E por fim, temos os Controllers, que são os controladores da nossa aplicação, ficando responsável por um intermédio entre as Views e Models, recebendo e retornando informações entre si. A arquitetura MVC é muito utilizada por possibilitar uma melhor reusabilidade e manutenção do código do sistema.

No presente trabalho, a utilização da arquitetura MVC torna-se necessária para obter uma separação das camadas bem definidas. Onde as models conterá apenas definições de dados. Os controllers responsáveis pelas rotas e requisições HTTP, intermediando a comunicação dos dados nas models e retornando as informações para as views. E por último a camada de visualização (Views) que receberá e enviará dados fornecidos pelo usuário.

2.2.2 LINGUAGEM DE PROGRAMAÇÃO PHP

Conforme a própria documentação da linguagem, o PHP é uma linguagem de programação com foco em scripts do lado do servidor (server-side), que possui uma ampla

compatibilidade, que pode ser utilizado na maioria dos sistemas operacionais como Windows, Mac, Linux e as variantes do Unix.

Com o PHP é possível não só ter a liberdade de escolha entre os diversos sistemas operacionais, como também o seu paradigma, podendo escolher entre o paradigma de programação estruturada, ou programação orientada a objetos (POO), ou até mesmo os dois paradigmas. (TURINI, 2015)

Segundo Bento Evaldo em seu livro “Desenvolvimento web com PHP e MySQL. Uma das principais características do PHP é a sua ampla variedade de suporte a banco de dados, como por exemplo o MySQL, ou utilizando uma camada de abstração com o PDO ou conectar a qualquer banco de dados que suporte o padrão “Open Database Connection” usando a extensão ODBC.

Por ser uma linguagem com uma baixa curva de aprendizagem em comparação a outras linguagens como Java, C#, C, C++ e entre outras. Por sua grande compatibilidade com outros sistemas, e por sua liberdade de utilização, podendo escolher entre POO e programação orientada a objetos. E como a junção do PHP e de um bom framework se é possível construir grandes sistemas com alta escalabilidade, o PHP foi escolhido para a construção do presente trabalho

2.2.3 FRAMEWORK LARAVEL

Segundo Soegoto, Laravel é um poderoso framework backend com arquitetura MVC que facilita o desenvolvimento de aplicações web e às torna mais rápida através de suas funcionalidades já prontas. Laravel também é um framework que utiliza a linguagem de programação PHP e tem seu código aberto, o que faz com que a framework seja construída através de contribuições da própria comunidade.

Laravel possui diversas funcionalidades já prontas que serão utilizadas no presente trabalho, como por exemplo a utilização do composer para instalar pacotes ao projeto, um sistema completo e seguro de autenticação, mapeamentos de objetos com o Eloquent ORM incluindo classes do Laravel para o banco de dados, que facilita o acesso e manipulação dos dados. O Artisan como command-line interface (CLI) com dezenas de comandos.

2.2.4 HTML

HTML é a primeira camada de um website e uma importante ferramenta deste projeto pois é através dela que os navegadores se utilizando de protocolos http vão conseguir ler e interpretar as páginas do sistema. Sem ela não seria possível o navegador carregar e exibir o conteúdo das páginas.

HTML é uma linguagem de marcação padrão da web que é mantida e supervisionada pela W3C (The World Wide Web Consortium). Pode ser

considerada o esqueleto de todo website, e que juntamente com outras linguagens como CSS e JavaScript temos a formação completa do corpo de um website. (YANDEX, 2019)

2.2.5 CSS

CSS é outra importante ferramenta que com a utilização dela é possível adicionar estilizações ao HTML. Com isso é possível aplicar estilizações de espaçamentos, cores, posicionamentos, fontes bordas e entre outros efeitos visuais que dão “vida” ao documento HTML. (MARQUES, 2019)

CSS também é mantida pela W3C (The World Wide Web Consortium) que é a mesma mantenedora do HTML. Atualmente a versão mais recente do CSS é o CSS3 que é o que vai ser utilizado no presente trabalho e faz parte da segunda camada de um website.

2.2.6 JAVASCRIPT

JavaScript é uma linguagem de programação que faz parte da terceira e última camada de um website, e é responsável por dar movimentos ao website e tornar os seus elementos dinâmicos. Qualquer ação feita dentro de um website que resulte em uma alteração no website, é executado por um script. (MARQUES, 2019)

Com a junção das três camadas de um website, com HTML, CSS e JavaScript, é possível formar uma página completa de um website, páginas que no presente trabalho fazem parte da camada View da arquitetura MVC (Model View Controller) que é onde será exibida toda a parte visual do sistema.

2.2.7 BANCO DE DADOS

O banco de dados escolhido para este projeto é o MySQL que como segundo Richard Boyett em seu artigo “What is MySQL: MySQL Explained For Beginners” cita que em 1994 foi criado o MySQL por uma empresa sueca, e posteriormente em 2008 vendido para a empresa Microsystem e em 2010 a empresa Microsystem foi comprada pela gigante Oracle, que hoje possui os direitos sobre o MySQL.

Em uma definição geral segundo Kofler M., MySQL é um banco de dados relacional. Para poder exemplificar melhor o que é um banco de dados, pense no ato de setirar uma selfie: Quando o botão de tirar a foto é pressionado, a foto retirada e armazenada em sua galeria. Sua foto é o dado e a galeria é o banco de dados. Um banco de dados é um lugar onde cada dado é armazenado e organizado. A palavra “relacional” significa que os dados são organizados em tabelas que se relacionam entre si através de chaves primarias.

Para este projeto, foi escolhido o banco de dados MySQL por ser um

dos bancos mais utilizados atualmente, por ser flexível e fácil de se utilizar, pela sua alta performance, sua segurança no armazenamento dos dados. E por ser de código aberto, isso significa que qualquer um pode baixar o código e modificar como quiser, o que facilita a modificação para melhor atender o projeto em que se é utilizado. (BOYETT RICHARD, 2021)

c. FINANÇAS PESSOAIS

Conforme Cherobim e Espejo (2010), finanças pessoais é a ciência que estuda a aplicação de conceitos financeiros nas decisões financeiras de uma pessoa, família ou até mesmo de um grupo de integrantes de uma república de estudantes por se tratar de uma estrutura parecida como a de uma família em uma residência.

Finanças pessoais é um assunto extremamente sério, e de relevância na vida de todas as pessoas. Pode não agradar as pessoas por se tratar de um tema um pouco maçante, mas tem total importância e deve ser feito por todos. E como dito por Cherobim e Espejo (2010) é considerado finanças pessoais os aspectos financeiros individuais, já que deve ser analisado a situação de cada indivíduo para assim poder ser tomada as mais variadas decisões necessárias para se manter uma boa saúde financeira.

Ainda segundo Cherobim e Espejo (2010), algumas das tarefas relacionadas as finanças pessoais é o controle domésticos que tem como objetivo melhorar a qualidade de vida dos integrantes de um lar e individual. Controle doméstico que engloba alguns dos aspectos como: alimentação, moradia, despesas fixas públicas essenciais como água e energia, saúde, vestuários etc.

Segundo o artigo Gestão financeira pessoal: uma aplicação prática. E a partir do estudo feito dentro deste artigo, pode-se constatar que 75% dos pesquisados, não realizam nenhum controle financeiro mensal. E os outros 25% que fazem algum controle financeiro 17% utilizam planilhas como controle de receitas e despesas onde anotam os gastos e receitas diárias. 17% Possuem cadernos com anotações de despesas. 33% também utilizam planilhas, porém só anotam os principais gastos do mês. 17 % através de extratos bancários e os outros 33% fazem cálculos aproximados no início do mês de quanto irá gastar e controla essas despesas a partir desses parâmetros.

Com esse estudo do artigo citado, podemos concluir que grande parte da população se quer realizam controle financeiro. E os que o fazem, se utilizam de ferramentas não tão apropriadas com inserções manuais e passíveis de erros. O que viabiliza o desenvolvimento de um sistema que possa facilitar este controle financeiro, e que leve saúde financeira para os utilizadores do sistema.

2 DESENVOLVIMENTO

Neste capítulo serão abordados os métodos utilizados no desenvolvimento do presente trabalho como: modelos gráficos e textuais que deram base para a conclusão do sistema proposto.

a. MÉTODO

Com base em experiências pessoais deste hiato acadêmico e colegas de repúblicas, foi feito um levantamento dos requisitos funcionais e não funcionais do sistema a fim de ter um sistema onde se possa utilizar em uma república para a gestão financeira de suas despesas.

Em um segundo momento foram definidas as tecnologias que seriam utilizadas e a plataforma de hospedagem em nuvem. As tecnologias definidas para o desenvolvimento foram: linguagem de programação PHP, linguagem de folhas de estilo CSS, linguagem de programação Javascript, framework Laravel, padrão MVC e serviço de nuvem da Microsoft Azure para a hospedagem.

Após a finalização do desenvolvimento do sistema, foi hospedado na nuvem utilizando a plataforma Azure da Microsoft

b. ATOR

Para o desenvolvimento do sistema, foi preciso definir o ator do sistema que fará a utilização do sistema.

Usuário Residente: Este é o único ator do sistema que vai fazer uso do sistema criando república, convidando usuários para a república, registrando contas a pagar e analisar demonstrativos.

c. REQUISITOS

Os requisitos são divididos em requisitos funcionais e não funcionais e que juntos compõem e definem as funcionalidades e características do sistema. Requisito funcional é tudo aquilo que o sistema deverá fazer de acordo com ações tomadas pelo usuário ator. Já os requisitos não-funcionais estão mais relacionados a desempenho, usabilidade, segurança, utilização de tecnologias específicas, disponibilidade e entre outros. (Bezerra, 2007)

3.3.1 REQUISITOS FUNCIONAIS

- **Cadastrar novos usuários (RF1):** Disponibilizar uma funcionalidade no sistema onde novos usuários possam se cadastrar utilizando-se de seus dados e registrar suas contas de usuários no sistema.

- **Autenticação antes de acessar o sistema (RF2):** Possibilitar que os usuários que após terem efetuado o cadastro de novo usuário possam efetuar o acesso ao sistema utilizando suas credenciais de e-mail e senha.
- **Criar república (RF3):** Permitir que os usuários do sistema possam criar república utilizando de informações da república como nome, endereço, foto etc.
- **Criar código de convite para outros usuários poder ingressar na república (RF4):** Permitir que usuários que estejam em uma república dentro do sistema, possam criar um código de convite para outros usuários poder ingressar na república.
- **Ingressar em república a partir de código de convite (RF5):** Viabilizar ingressar em uma república já existente no sistema através de um código de convite gerado por um usuário que esteja na república.
- **Registrar despesas a pagar (RF6):** O sistema deve possibilitar a inserção de novas despesas a pagar da república e dividi-la em partes iguais para cada integrante dentro da república.
- **Pagar despesas a pagar (RF7):** Após a inserção de uma despesa a pagar e ser dividida em partes iguais para os integrantes, o sistema deve possibilitar para cada integrante o pagamento dessas partes das despesas via PIX.
- **Visualizar demonstrativo da república (RF8):** O sistema deve fornecer uma visualização do demonstrativo da república contendo informações de quantidades como total pago de despesas, despesas a pagar, total de despesas pago, total de moradores, gráfico de quanto cada morador já pagou em despesas.

3.3.2 REQUISITOS NÃO FUNCIONAIS

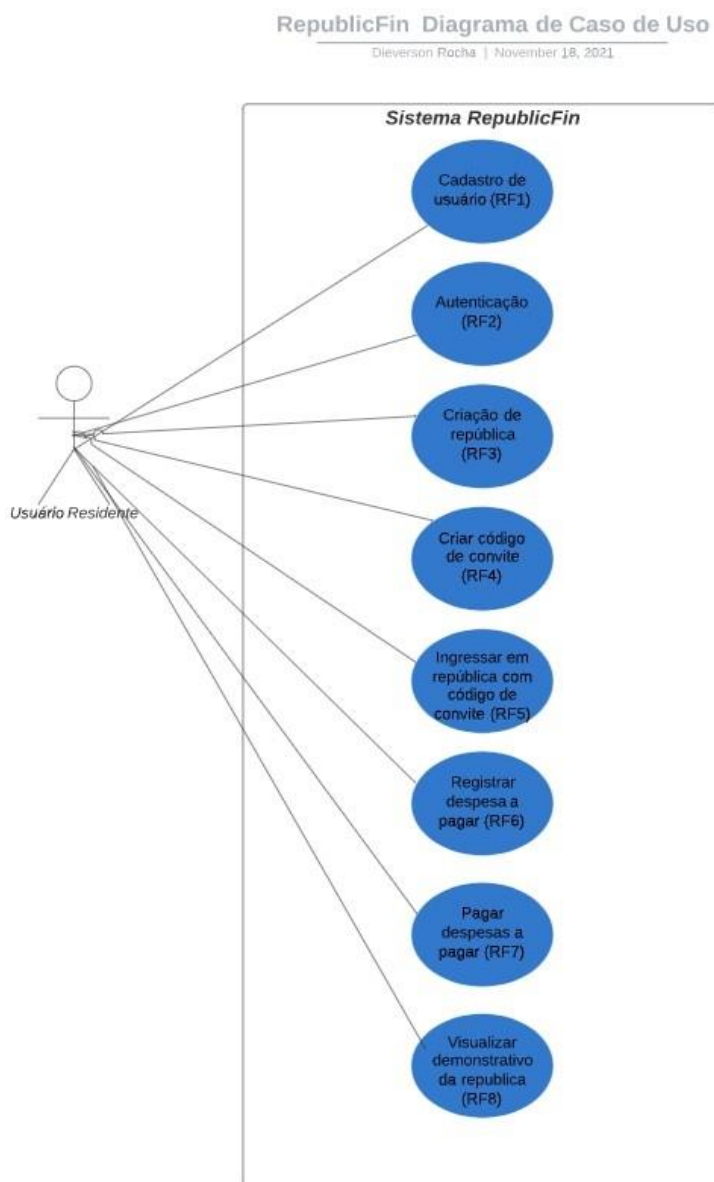
- **Disponibilidade do sistema (RNF1):** O sistema deverá estar disponível para ser acessado a partir de um link através de qualquer computador que tenha conexão com a internet.
- **Hospedagem do sistema (RNF2):** O sistema deverá ser hospedado na nuvem pela plataforma de serviços em nuvem da Microsoft Azure.
- **Armazenamento de dados (RNF3):** O sistema deverá armazenar seus dados em um banco de dados MySQL na nuvem da Microsoft Azure.
- **Armazenamento de arquivos (RNF4):** O sistema deverá armazenar os arquivos do sistema e dos usuários em um container de arquivos na nuvem da Microsoft Azure.

d. DIAGRAMA DE CASO DE USO

O diagrama de caso de uso faz parte da engenharia de requisitos e

permite ter uma visualização mais clara e resumida do sistema. Na Figura 1 está o diagrama de caso de uso do sistema RepublicFin que é o sistema desenvolvido para o presente trabalho.

Figura 1 – Diagrama de caso de uso do sistema RepublicFin

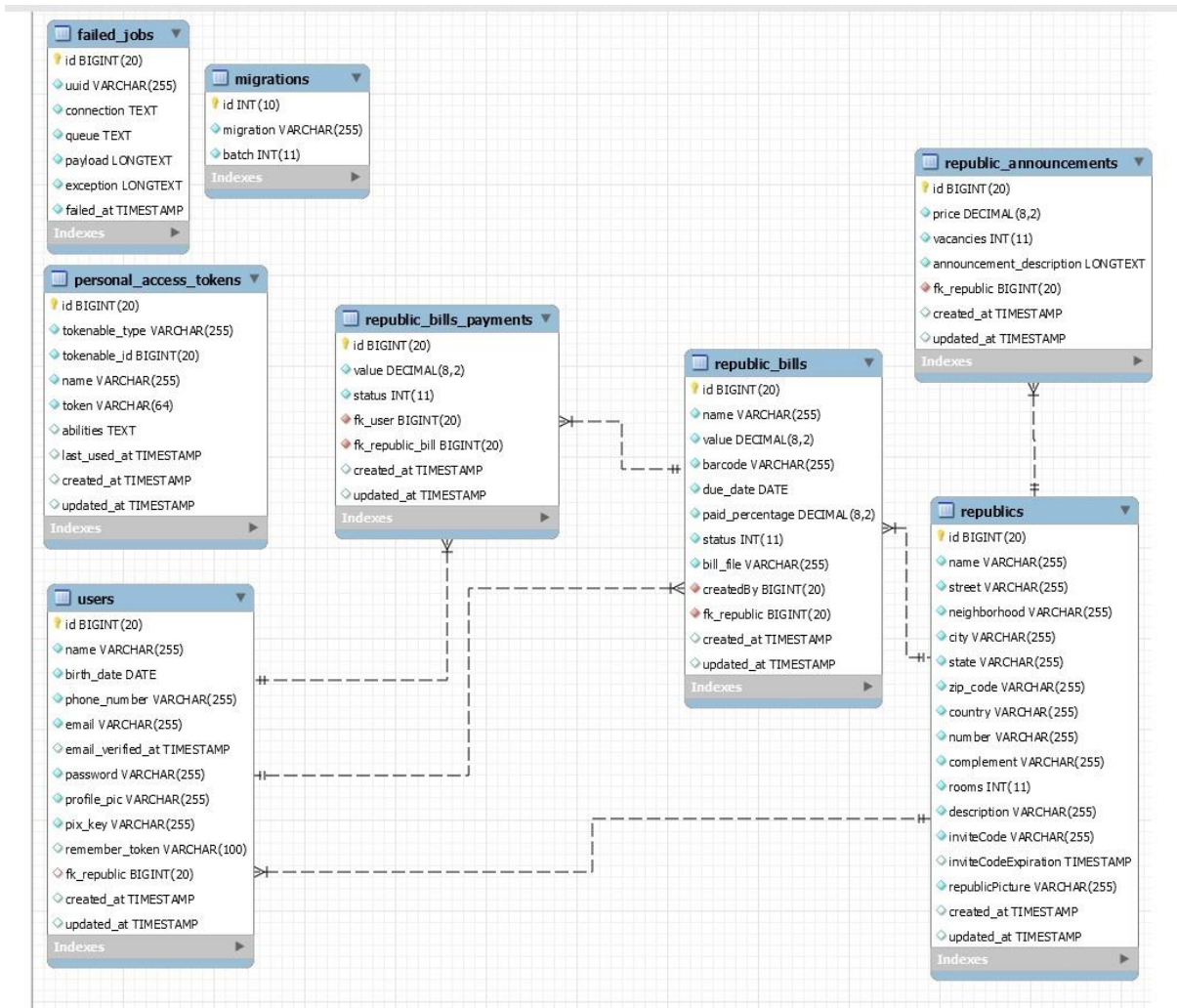


Fonte: Elaborado pelo autor.

e. MODELO DE DADOS

No decorrer do desenvolvimento do sistema RepublicFin, se fez necessário a criação de tabelas, campos e relacionamentos no banco de dados MySQL para armazenar os dados que serão utilizados no sistema. Esses dados foram expressos no modelo de dados relacional na Figura 2 que dá uma visualização geral dos dados existentes no banco de dados.

Figura 2 – Diagrama de dados relacional do banco de dados MySQL



Fonte: Elaborado pelo autor.

As tabelas apresentadas na Figura 2 representam:

1. **Tabela republics:** É a entidade que armazena os dados cadastrais da república que os usuários informam ao criar uma república dentro do sistema.
2. **Tabela users:** A entidade users no banco de dados representa o registro de cada usuário que se cadastrou no sistema e informou seus dados para fazer o cadastro.
 - Os dados de e-mail e password são utilizados para a autenticação do usuário no sistema.
 - O campo fk_republic é um relacionamento com a tabela republic que é preenchido quando o usuário ingressa em alguma república.
 - O campo pix_key é utilizado quando os usuários vão pagar uma despesa registrada por um outro usuário.
3. **Tabela republic_bills:** Representa as despesas que os usuários

registram no sistema. Quando todas as partes das despesas for paga por cada usuário, o status do registro nesta tabela ficara como pago.

- Campo CreatedBy possui relacionamento com a tabela users que indica quem criou o registro e quem vai receber as transferências via pix para pagamento da despesa.

- Campo fk_republic possui relacionamento com a tabela republics que informa de qual república a despesa registrada pertence.

- Campo paid_percentage informa a porcentagem paga de acordo com o pagamento de cada usuário da sua parte da despesa.

4. **Tabela republic_bills_payments:** Representa as partes das despesas registradas para cada usuário, exemplo: Uma republica com três pessoas, um dos integrantes registra uma despesa a pagar, o sistema vai fazer a divisão dessa despesa para os três integrantes e cria um registro na tabela republic_bills_payments para cada um dos três integrantes que vai ficar com status de aguardando pagamento até que o usuário pague a sua parte da despesa.
 - Campo fk_republic_bills possui relacionamento com republic_bills que indica qual despesa aquela parte da despesa faz parte.
 - Campo fk_user possui relacionamento com tabela users para indicar de quem é determinada parte da despesa.
5. **Tabela republic_announcements:** Tabela para implementações futura em que o usuário poderá criar anúncios de vagas na república dentro do próprio sistema.
6. **Tabela personal_access_tokens:** Tabela gerada automaticamente pelo framework Laravel.
7. **Tabela failed_jobs:** Tabela gerada automaticamente pelo framework Laravel.
8. **Tabela migrations:** Tabela gerada automaticamente pelo framework Laravel.

3 RESULTADOS OBTIDOS

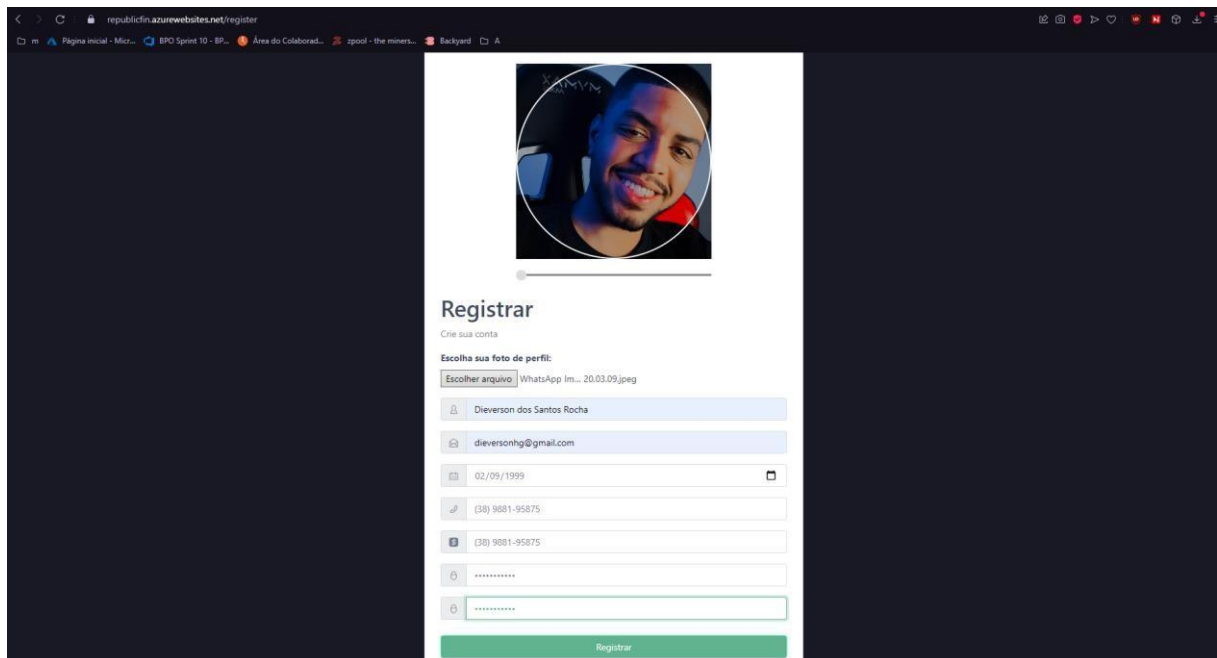
Neste capítulo será demonstrado o resultado obtido depois de todo planejamento, estudo e desenvolvimento do sistema RepublicFin passando por cada etapa do processo dentro do sistema.

4.1.1 TELA DE CADASTRO

Esta é a primeira etapa para se utilizar o sistema, na tela de cadastro

o usuário informa os seus dados, uma imagem de perfil e se cadastra no sistema de acordo com o Requisito Funcional (RF1). Essa tela é exibida na Figura 3.

Figura 3 – Tela de cadastro de novos usuários

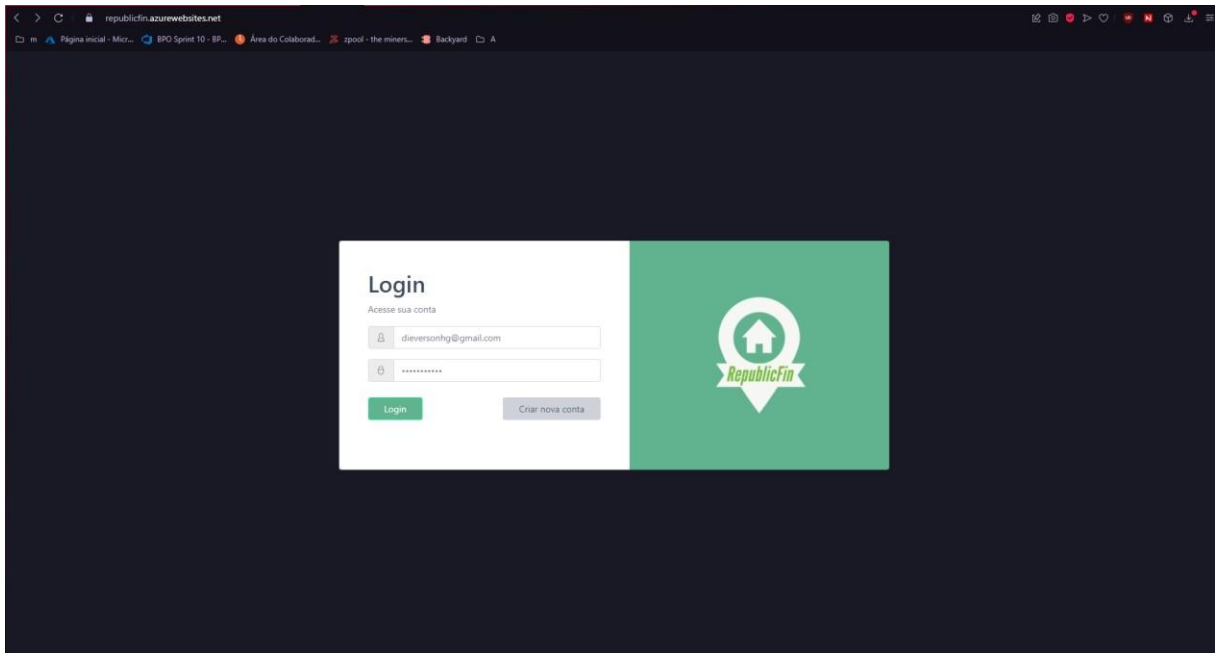


Fonte: Elaborado pelo autor.

4.1.2 TELA DE AUTENTICAÇÃO

Após o usuário se cadastrar no sistema, o sistema poderá ser acessado através da tela de autenticação onde o usuário informa seu e-mail e senha cadastradas no momento de cadastro no sistema de acordo com Requisito Funcional (RF2). Essa tela é exibida na Figura 4.

Figura 4 – Tela de autenticação

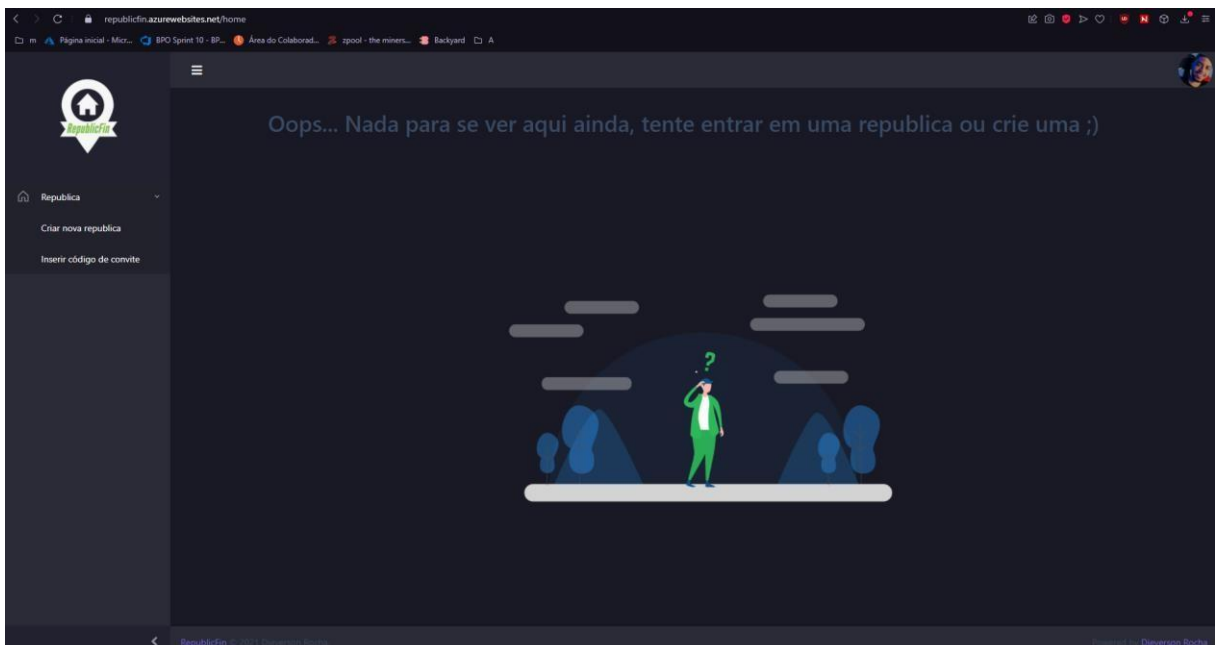


Fonte: Elaborado pelo autor.

4.1.3 TELA HOME

Após o usuário se autenticar no sistema, é exibida a tela home que foi desenvolvida com duas opções para o usuário, sendo elas: Criar uma república ou entrar em alguma república através de um código de convite. Essa tela é exibida na Figura 5.

Figura 5 – Tela home

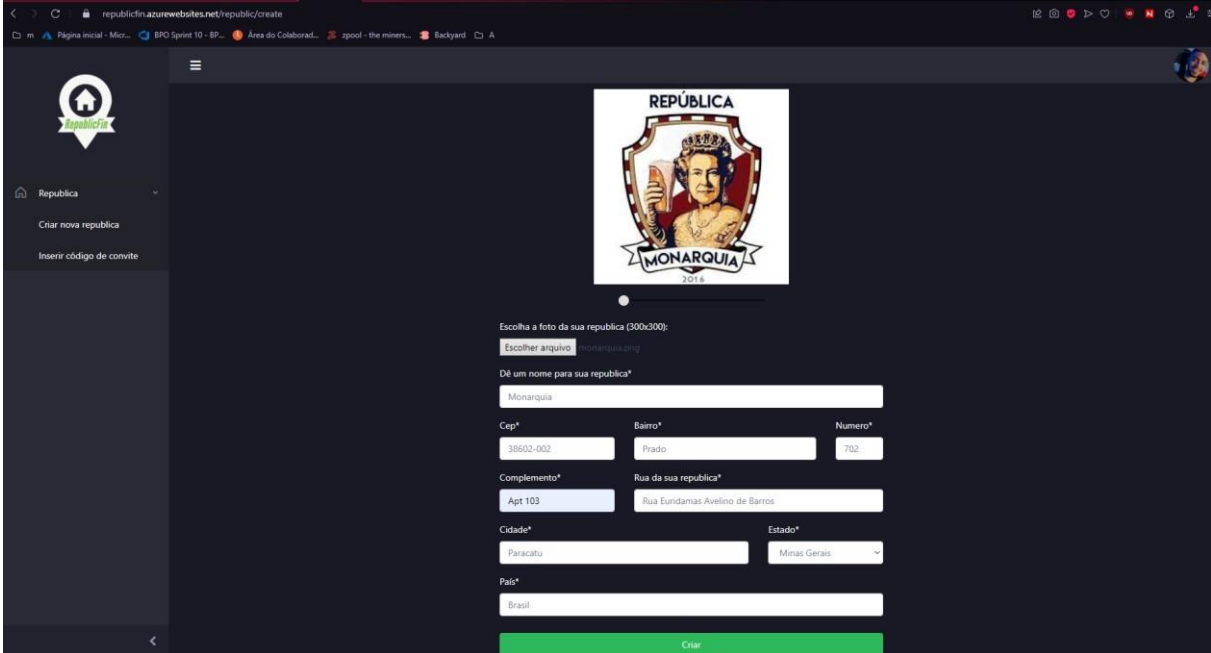


Fonte: Elaborado pelo autor.

4.1.4 TELA NOVA REPÚBLICA

De acordo com a (RF3), foi desenvolvido uma tela onde os usuários possam criar repúblicas com suas informações. Essa tela é exibida na Figura 6.

Figura 6 – Tela nova república



The screenshot shows a web browser window with the URL 'republicas.azurewebsites.net/repUBLIC/create'. The page has a dark theme. On the left, there is a sidebar with a home icon and a 'República' menu. The main content area features a form for creating a new republic. At the top of the form is a preview of a republic logo with a crown and the text 'REPÚBLICA' and 'MONARQUIA 2016'. Below this, there is a section for uploading a photo (300x300 pixels) with a file selection button. The form then asks for the name of the republic, which is filled with 'Monarquia'. Below that are fields for 'Cep*' (38602-002), 'Bairro*' (Prado), and 'Numero*' (702). There are also fields for 'Complemento*' (Apt 103) and 'Rua da sua republica*' (Rua Euzébio de Barros). The 'Cidade*' is Paracatu and the 'Estado*' is Minas Gerais. The 'País*' is Brasil. A green 'Criar' button is at the bottom of the form.

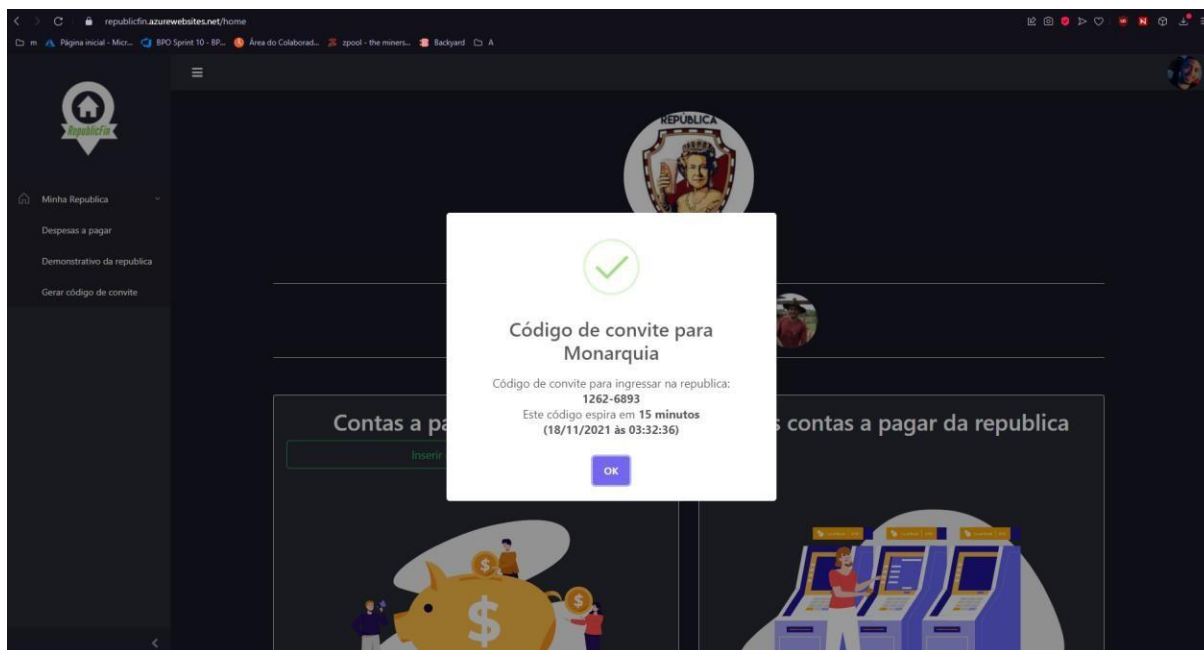
Fonte: Elaborado pelo autor.

É importante citar que no desenvolvimento desta tela, pode-se observar que poderia ser feita uma melhoria no preenchimento desses dados, então a partir do momento do preenchimento do CEP, é feita uma consulta a uma API que faz o retorno dos dados do endereço e já é preenchido automaticamente na tela, facilitando assim o preenchimento das informações da república.

4.1.5 GERAR CÓDIGO DE CONVITE

De acordo com o Requisito Funcional (RF4), foi desenvolvido uma funcionalidade onde o usuário integrante de uma república, possa gerar um código de convite para uma pessoa que ainda não esteja na sua república. Na Figura 7 demonstra como é gerado este código.

Figura 7 – Gerar código de convite



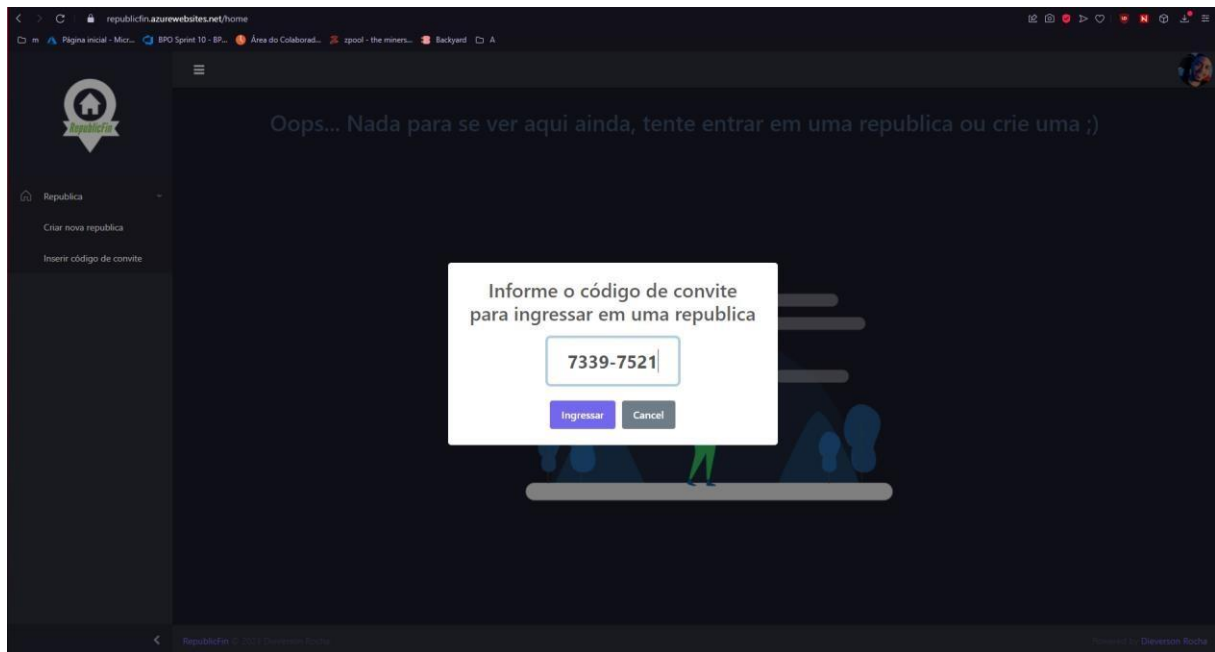
Fonte: Elaborado pelo autor.

Por questão de segurança, para que o mesmo código não fique disponível para ingressar a qualquer momento, o código de convite que é gerado tem um tempo de expiração de quinze minutos.

4.1.6 INSERIR CÓDIGO DE CONVITE

De acordo com o Requisito Funcional (RF5), alternativamente, caso o usuário não queira criar uma república e queira ingressar em uma república já existente, foi desenvolvido essa tela onde o usuário pode inserir o código gerado por um integrante de outra república e ingressar na mesma. A inserção deste código de convite é demonstrada na Figura 8.

Figura 8 – Inserção de código de convite

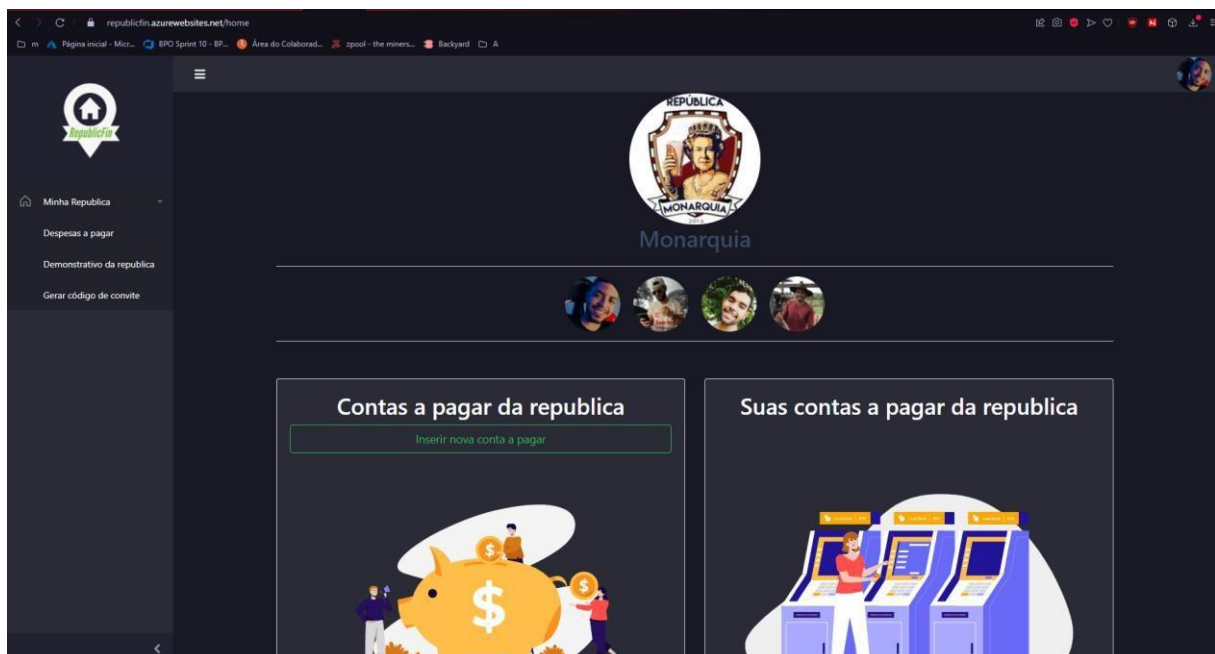


Fonte: Elaborado pelo autor.

4.1.7 TELA DE DESPESAS A PAGAR

Na tela de despesas a pagar foi desenvolvido duas sessões onde o usuário pode visualizar as despesas a pagar da república e suas despesas a pagar da república (Caso tenha despesas a pagar no momento). A tela é exibida na Figura 9.

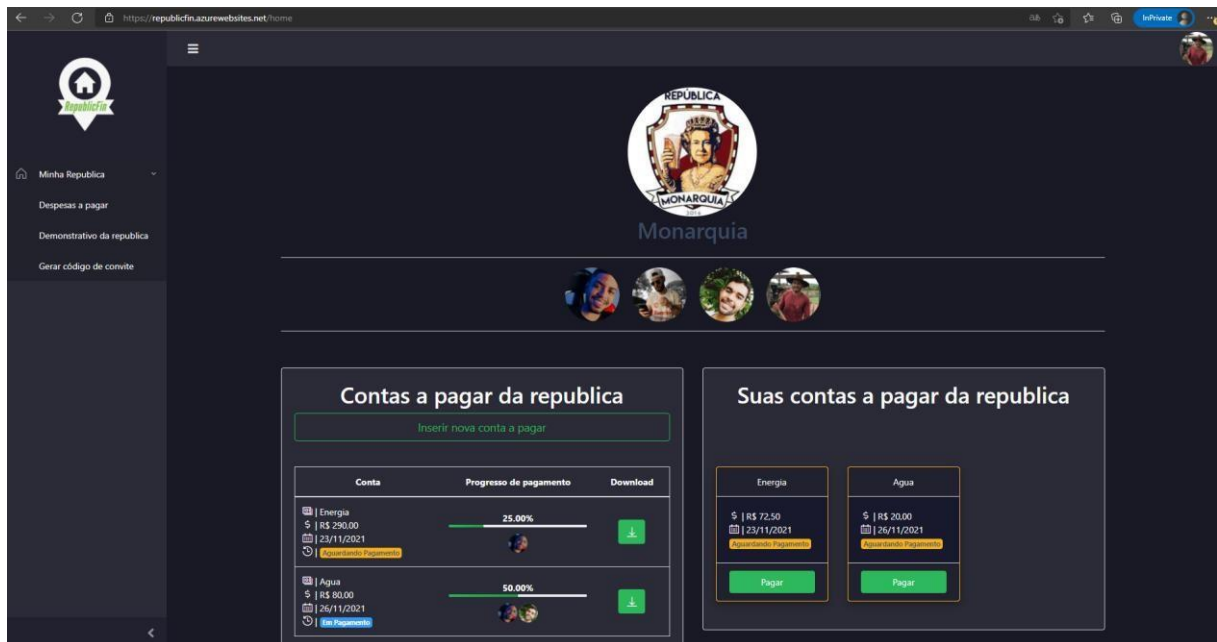
Figura 9 – Tela de despesas a pagar



Fonte: Elaborado pelo autor.

Com a inserção de algumas despesas a pagar a tela passa a ter uma exibição diferente com as contas a pagar no lado esquerdo, com o status atual, o progresso do pagamento e quem já pagou até o momento. E ao lado direito as despesas do usuário logado que ainda não foram pagas e que está aguardando o pagamento. A visualização desta tela pode ser feita na Figura 10.

Figura 10 – Tela com despesas a pagar

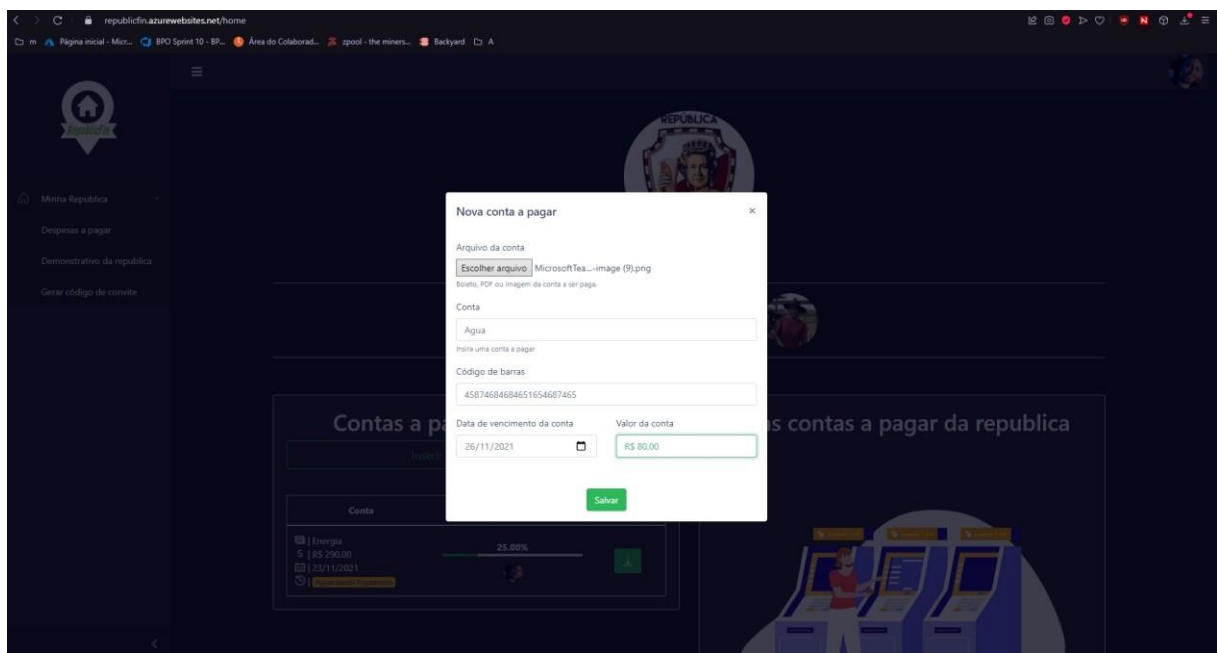


Fonte: Elaborado pelo autor.

4.1.8 TELA DE NOVA DESPESA A PAGAR

De acordo com o Requisito Funcional (RF6). Foi desenvolvido essa tela onde o usuário pode registrar uma nova despesa para que ela possa ser dividida em partes iguais para cada integrante da república. A tela é exibida na Figura 11.

Figura 11 – Tela de nova despesa a pagar

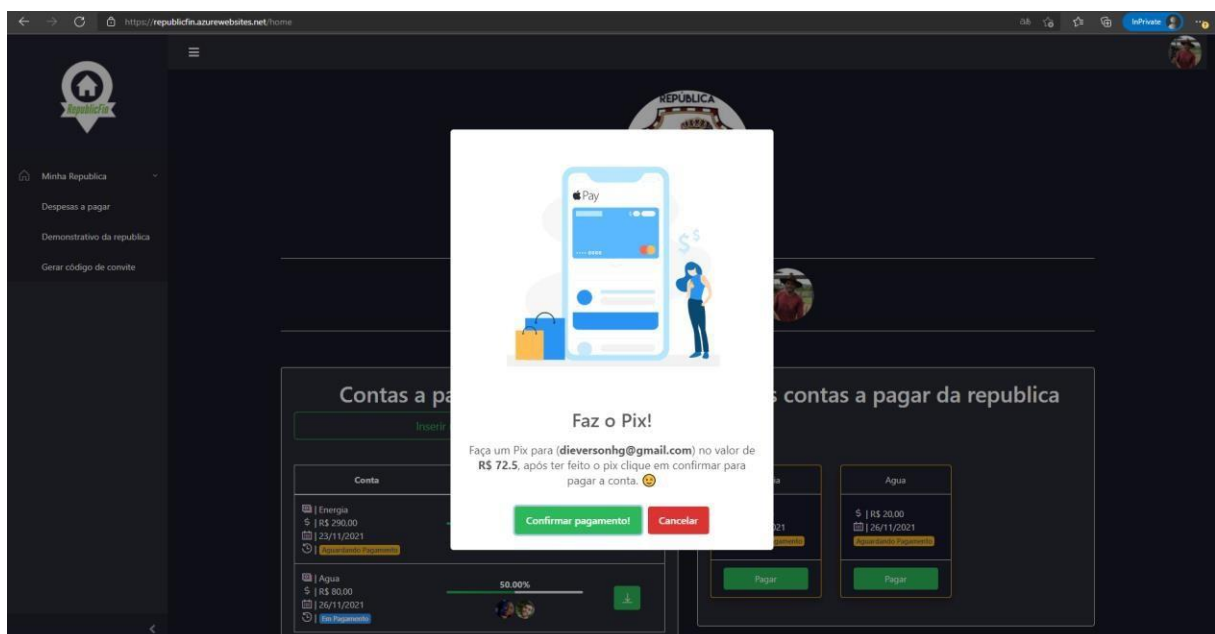


Fonte: Elaborado pelo autor.

4.1.9 PAGAMENTO DE DESPESAS

De acordo com o Requisito Funcional (RF7). Foi desenvolvido a funcionalidade onde o usuário pode pagar a sua parte na despesa da república. O pagamento é feito através da chave pix do usuário que registrou a despesa para que ele pague a despesa assim que todos tiverem pagado a sua parte e terem reunido o dinheiro com a pessoa responsável pela despesa. A tela é exibida na Figura 12.

Figura 12 – Tela de pagamento de despesas

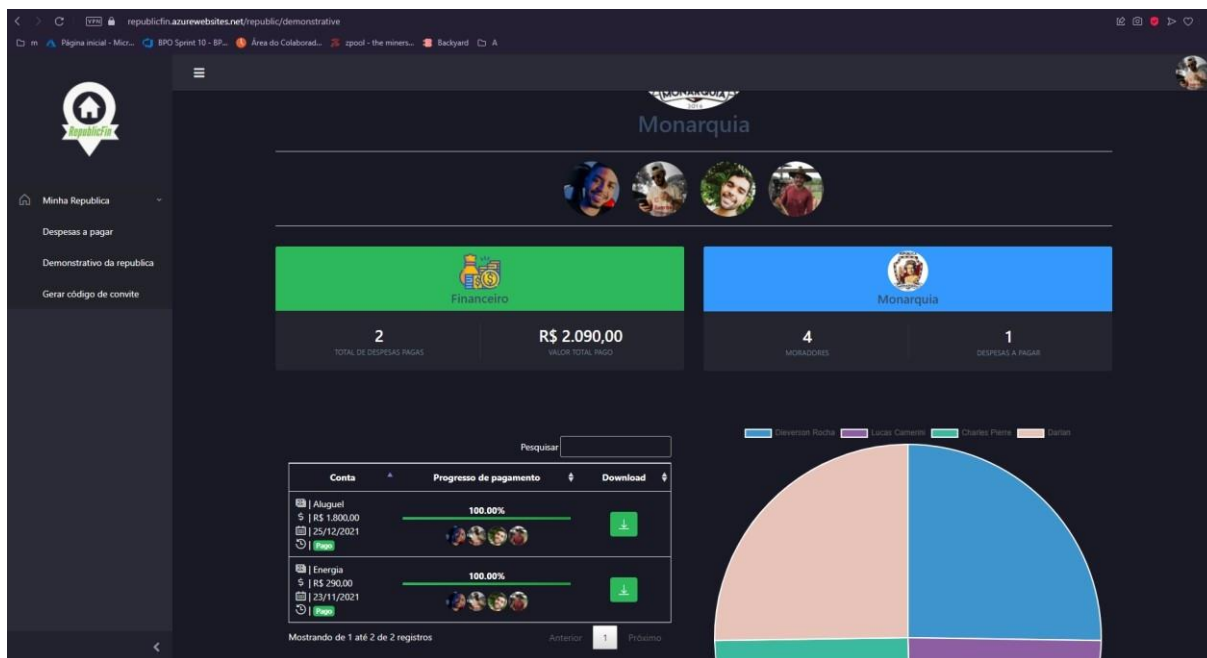


Fonte: Elaborado pelo autor.

4.2.0 TELA DEMONSTRATIVO DA REPÚBLICA

De acordo com o Requisito Funcional (RF8). Foi desenvolvido uma tela onde os usuários residentes possam ter uma visão geral da república através de um demonstrativo com diversas informações como despesas pagas, valor total pago, quantidade de moradores, quantidade de despesas a pagar, gráfico com quantidade que cada integrante já gastou com despesas e contas pagas. A tela é exibida na Figura 13.

Figura 13 – Tela de demonstrativo da república



Fonte: Elaborado pelo autor.

4 CONSIDERAÇÕES FINAIS

Tendo em vista o grande problema que é a falta de organização de despesas financeira em repúblicas universitárias, e a falta de ferramenta específica para isso, viabilizou o desenvolvimento do sistema RepublicFin.

O desenvolvimento e a hospedagem do sistema em uma plataforma em nuvem, torna o sistema RepublicFin uma ferramenta prática e fácil de utilização para que os integrantes de repúblicas possam organizar a vida financeira de suas repúblicas, evitando assim eventuais conflitos decorrentes de despesas financeira.

É importante citar que no desenvolvimento do sistema proposto, pode-se notar que o sistema pode oferecer mais algumas funcionalidades que não está no escopo do projeto atual, mas que futuramente ao implementar traria benefícios a mais para os usuários do sistema, como por exemplo: a divulgação de vagas em repúblicas para encontrar outros moradores dentro do próprio sistema e a integração com uma API de pagamentos para fazer os pagamentos das despesas dentro do sistema. Essas são implementações futuras que em um segundo momento na continuação do projeto será feito.

Este Trabalho de Conclusão de Curso foi muito proveitoso, tanto para o aprofundamento em prática na utilização de tecnologias para desenvolver soluções, quanto para a comunidade de estudantes que moram em repúblicas universitárias. Sendo assim, finalizados o desenvolvimento e a publicação do sistema na hospedagem em nuvem, os objetivos propostos para este TCC foram atingidos.

REFERÊNCIAS

IBGE. **Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal 2**

018. Rio de Janeiro, 2020. 12 p. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/livros/liv101705_informativo.pdf. Acesso em: 21 abr. 2021.

LACALLE, DANILO. **DataFala! mostra percentual de estudantes universitários que moram sozinhos**. Fala Universidades. 2018. Disponível em: <https://falauniversidades.com.br/data-fala-mostra-percentual-de-estudantes-universitarios-que-moram-sozinhos/>. Acesso em: 21 abr. 2021.

DELABRIDA, Zenith Nara Costa; SANTOS, Carla Mayra de Jesus; BARLETTA, Janaína Bi anca. **Habilidades sociais, estresse, desempenho acadêmico em universitários de moradias coletivas**. Rev. bras. ter. cogn., Rio de Janeiro, v. 14, n. 1, p. 21-

30, jun. 2018. Disponível em http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1808-56872018000100004&lng=pt&nrm=iso. acessos em 17 abr. 2021.

Laranjeira, C. A. (2009). **O contexto organizacional e a experiência de estresse: uma perspectiva integrativa**. Revista de salud pública, 11(1),123-133.

Furtado, E. S., Falcone, E. M. O., & Clark, C. (2003). **Avaliação do estresse e das habilidades sociais na experiência acadêmica de estudantes de medicina de uma universidade do Rio de Janeiro**. Interação em Psicologia, 7(2),43-51.

NASPOLINI, Rodrigo Benedet. **As primeiras faculdades de Direito: São Paulo e Recife**. Florianópolis: 10 mai. 2008. Disponível em: . Acesso em: 10 nov. 2010.

SARDI, Jaime A. **Virtù et appetitus: aprendendo a conviver com o prazer**. In: MACHAD O, Otávio Luiz (Org.). **As repúblicas de Ouro Preto e Mariana: Trajetórias elmportância**. Recife: Universidade Federal de Pernambuco, 2007

VASCONCELOS, Ana Maria Nogales. **Juventude e ensino superior no Brasil**. Disponível e m: <http://repositorio.ipea.gov.br/bitstream/11058/9062/1/Juventude%20e%20ensino.pdf>. Acesso 26/08/2019. p.127-137. 2016.

SANTANA SILVA, Luzia Wilma et al. **A arte do viver no contexto das repúblicas universitárias**. Revista de Enfermagem UFPE on line, [S.l.], v. 7, n. 2, p. 518-526, dez. 2012. ISSN 1981-

8963. Disponível em: <https://periodicos.ufpe.br/revistas/revistaenfermagem/article/view/10263/10889>. Acesso em: 02 maio 2021.

MASSARI, Jorge. **Padrão MVC | Arquitetura Model-View-Controller**. Portal GSTI. 2018. Disponível em: <https://www.portalgsti.com.br/2017/08/padrao->

mvc-arquitetura-model-view-controller.html#:~:text=O%20que%20é%20Model-View,e%20o%20controlador%20(controller).&text=O%20padrão%20MVC%20foi%20desenvolvido,como%20arquitetura%20para%20aplicativos%20desktop.. Acesso em: 16 mai.2021.

PHP Group. **What can PHP do?**. PHP. Disponível em: <https://www.php.net/manual/en/intro-whatcando.php>. Acesso em: 16 mai. 2021.

Creative Tim. **What is Laravel? Explain it like I'm five..** Dev.to. 2019. Disponível em: https://dev.to/creativetim_official/what-is-laravel-explain-it-like-i-m-five-19eb. Acesso em: 23 mai. 2021.

Yandex. **What is HTML (And why it matters a lot)**. Medium. 2019. Disponível em: <https://medium.com/practicum-by-yandex/what-is-html-and-why-it-matters-a-lot-458f838062e1#:~:text=HTML%20is%20a%20markup%20language,a%20href%3D%22...%22>. Acesso em: 23 mai. 2021.

MARQUES, Rafael. **O que é HTML? Entenda de forma descomplicada**. Home host. 2019. Disponível em: <https://www.homehost.com.br/blog/tutoriais/o-que-e-html/>. Acesso em: 23 mai. 2021.

BOYETT, Richard. **What is MySQL: MySQL Explained For Beginners**. Hostinger. 2021. Disponível em: <https://www.hostinger.com/tutorials/what-is-mysql>. Acesso em: 26 mai. 2021.

MySQL Tutorial. **What Is MySQL**. MySQLTutorial. Disponível em: <https://www.mysqltutorial.org/what-is-mysql/>. Acesso em: 27 mai. 2021.

KS, Ashiq . **Working With MySQL**. Medium. 2019. Disponível em: <https://medium.com/@ashiqgiga07/working-with-mysql-dae8f149aa57>. Acesso em: 27 mai. 2021.

KOFLER, Michael. **What Is MySQL?**. Springer. 2001. Disponível em: https://doi.org/10.1007/978-1-4302-0853-2_1. Acesso em: 27 mai. 2021.

TURINI, Rodrigo. **PHP e Laravel: Crie aplicações web como um verdadeiro artesão**. Editora Casa do Código, v. 3, f. 110, 2015. 219 p.

CRUZ, Bruna Heloisa; KROETZ, Marilei ; FÁVERI, Dinorá Baldo. **GESTÃO FINANCEIRA PESSOAL: UMA APLICAÇÃO PRÁTICA**. Disponível em: <https://www.aedb.br/seget/arquivos/artigos12/19116831.pdf>. Acesso em: 27 mai. 2021.

CHEROBIM A. P. M. S; ESPEJO M.M.S.B. **Finanças pessoais: conhecer para enriquecer!**São Paulo: Atlas, 2010. 147 p.

BEZERRA, E. Princípios de Análise e Projeto de Sistema com UML. 2. ed. Rio de Janeiro: ELSEVIER, 2007. v. 6.