**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

**Pós-graduação *Lato Sensu* em Engenharia de *Software***

Projeto Integrado

Relatório Técnico

SGR – Sistema de Gestão de Repúblicas

Rafael Simões Coelho de Souza

Belo Horizonte

Abril de 2022.

# Projeto Integrado

**Sumário**

Projeto Integrado 3

1. Cronograma de Trabalho 7

2. Introdução 8

3. Definição Conceitual da Solução 10

3.1 Diagrama de Casos de Uso 11

3.2 Requisitos Funcionais 13

3.3 Requisitos Não-funcionais 14

4. Protótipo Navegável do Sistema 15

5. Diagrama de Classes de Domínio 15

6. Arquitetura da Solução 15

6.1 Padrão Arquitetural 15

6.2 C4 model - Diagrama de Contexto 15

7. Frameworks de Trabalho 18

8. Estrutura Base do Front End 18

9. Modelo Relacional ou Projeto de Banco de Dados NoSQL -

10. Plano de Testes -

11. Relatório de Execução de Testes de Software -

12. Apropriação de Horas no Projeto 23

13. Código da Aplicação 24

14. Avaliação Retrospectiva -

14.1 Objetivos Estimados -

14.2 Objetivos Alcançados -

14.3 Lições aprendidas -

15. Referências -

## Cronograma de Trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 26/11/2021 | 01/12/2021 | 1.Organização inicial das idéias e cenários de aplicação do projeto SGR e seus direcionamentos futuros. | Criação do texto de introdução tratando o cenário macro da aplicação em diversos aspectos. |
| 02/12/2021 | 04/12/2021 | 2. Expansão no detalhamento do texto de introdução partindo dos fatores de aspecto macro para micro, propiciando maior entendimento da aplicação. | Melhor contextualização do problema e separação dos três respectivos cenários que serão implantados futuramente no projeto. |
| 05/12/2021 | 08/12/2021 | 3.Criação do Diagrama de Casos de Uso da aplicação. | Relacionamento entre atores e ações. Enumeração de classes a serem utilizadas posteriormente na confecção do Diagrama de classes de Domínio e determinação de requisitos funcionais |
| 09/12/2021 | 10/12/2021 | 4.Especificação de requisitos funcionais da aplicação. | Enumeração de 24 requisitos funcionais categorizados para a aplicação. |
| 11/12/2021 | 12/12/2021 | 5.Especificação dos requisitos não funcionais da aplicação. | Enumeração de 8 requisitos não funcionais para a aplicação. |
| 13/12/2021 | 16/12/2021 | 6. Criação do Diagrama de Classes de Domínio. | Diagrama com o relacionamento entre as respectivas classes e métodos gerado a partir da Enumeração das classes encontradas no Diagrama de Casos de Uso. Será utilizado posteriormente para a criação das Entidades da aplicação. |
| 17/12/2021 | 22/12/2021 | 7. Criação do Protótipo interativo no Figma abordando os três cenários escolhidos para desenvolvimento dentro do Diagrama de Casos de Uso. | Wireframes, Protótipo no Figma, Vídeo do Protótipo navegável. |
| 23/12/2021 | 23/12/2021 | 8. Criação do repositório da aplicação no GitHub. | Submissão dos arquivos de documentação para a pasta docs do repositório, sendo estes o Relatório Técnico, os diagramas de Casos de Uso e Classes de Domínio, os wireframes e vídeo do Protótipo navegável gerado no Figma. |
| 14/02/2022 | 15/02/2022 | 9. Entrega da 1ª etapa do Projeto Integrado. | Submissão dos itens desenvolvidos na primeira etapa do Projeto Integrado para avaliação do orientador. |
| 22/02/2022 | 01/03/2022 | 10. Correção e ajustes após retorno do orientador na primeira etapa. | Período alocado para eventuais correções apontadas pelo orientador do projeto após a entrega da primeira etapa. |
| 02/03/2022 | 02/03/2022 | 11. Definição do padrão arquitetural da aplicação. | Especificação sobre o padrão escolhido para o projeto e suas tecnologias. |
| 03/03/2022 | 05/03/2022 | 12. Criação do Diagrama C4 Model. | Texto explicativo e Diagrama de contexto da aplicação. |
| 06/03/2022 | 07/03/2022 | 13. Frameworks de trabalho | Especificação dos frameworks utilizados no desenvolvimento da aplicação. |
| 08/03/2022 | 09/03/2022 | 14.Estrutura base do Front End. | Imagens do Layout mestre e menu de opções do sistema. |
| 10/03/2022 | 15/03/2022 | 15. Desenvolvimento de uma funcionalidade. | Funcionalidade implementada e vídeo demonstrativo da mesma. |
| 16/03/2022 | 18/03/2022 | 16. Submissão do Projeto no Heroku. | Configuração do projeto no Heroku e ajustes na ferramenta. |
| 19/03/2022 | 19/03/2022 | 17. Submissão do Relatório Técnico e demais itens desenvolvidos na 2ª etapa do Projeto na pasta docs no GitHub. | Documentação atualizada no repositório do projeto. |
| 14/04/2022 | 15/04/2022 | 18. Entrega da 2ª etapa do Projeto Integrado. | Submissão dos itens da 2ª etapa do Projeto integrado para avaliação do orientador. |
| 22/04/2022 | 25/04/2022 | 19. Correção e ajustes após retorno do orientador na segunda etapa. | Período alocado para eventuais correções apontadas pelo orientador do projeto após a entrega da segunda etapa. |
| 26/04/2022 | 27/04/2022 | 20. Modelo relacional ou Projeto de banco de dados NoSql. | Modelo de banco de dados do projeto e respectiva imagem para este documento. |
| 28/04/2022 | 01/05/2022 | 21. Construção inicial do Projeto com base na documentação levantada nos passos anteriores. | Criação do banco de dados com base no modelo das entidades.  Criação de Entidades, Controladores e Views em seu estado inicial no projeto. |
| 02/05/2022 | 22/05/2022 | 22. Desenvolvimento da aplicação e respectivos casos de uso selecionados. | Período destinado a implantação do sistema de acordo com a documentação estabelecida anteriormente. |
| 23/05/2022 | 25/05/2022 | 23. Desenvolvimento do plano de testes. | Criação de 6 casos de teste para serem aplicados na aplicação e respectivos resultados esperados. |
| 25/05/2022 | 28/05/2022 | 24. Relatório de execução dos testes. | Relatório gerado a partir da execução dos testes propostos anteriormente e seus respectivos resultados coletados. |
| 29/05/2022 | 30/05/2022 | 25. Informações relacionadas a implementação. | Abordagem sobre aquilo que foi implantado no desenvolvimento da aplicação com base na documentação, testes dentre outros. |
| 31/05/2022 | 01/06/2022 | 26. Avaliação retrospectiva. | Analise sobre as etapas e processos percorridos até a conclusão do projeto. |
| 01/06/2022 | 02/06/2022 | 27. Objetivos estimados,Objetivos alcançados e Lições aprendidas. | Delimitação sobre aquilo que foi planejado e executado no desenvolvimento da aplicação. |
| 03/06/2022 | 04/06/2022 | 28. Desenvolvimento do vídeo final de apresentação do projeto integrado. | Vídeo final com apresentação da implementação do Projeto Integrado. |
| 05/06/2022 | 06/06/2022 | 29. Atualização final do GitHub e Heroku com os arquivos e documentação relacionados ao Projeto Integrado. | Organização final de tudo aquilo que foi desenvolvido,testado, documentado e implementado. |
| 14/06/2022 | 15/06/2022 | 30. Entrega da 3ª Etapa do Projeto Integrado. | Submissão final do Projeto integrado para avaliação do orientador. |

**Observação: acrescente ou retire linhas, caso seja necessário.**

## Introdução

A Universidade Federal de Ouro Preto possui 59 repúblicas federais que abrigam 800 universitários (Dados referentes a 2022, disponíveis [neste link).](http://www.prace.ufop.br/assistencia-estudantil/moradia-estudantil/modalidade-de-gestao-compartilhada) Estas repúblicas são geridas pelos próprios universitários em um sistema denominado autogestão, sendo algumas destas moradias centenárias.

No município existem inúmeras repúblicas particulares que adotam processo semelhante de admissão de moradores, mas este trabalho focará nas repúblicas federais, tratando as demais de forma empírica em um número que gira em torno de 400 repúblicas.

A organização de uma república ouro pretana atende a um princípio de hierarquia e promoção do universitário por etapas desde seu ingresso na casa como calouro (bixo), passando pela sua escolha como morador e finalmente chegando a sua formatura como ex- aluno.

Em relação ao processo de seleção do calouro, ele poderá ser escolhido ou vetado após um período de batalha. Caso seja escolhido passará a integrar o grupo de moradores em seu mais baixo nível de hierarquia, caso contrário deverá procurar outra república para morar.

Este modelo de gestão proporcionou o desenvolvimento de laços entre gerações, visto que os universitários que se formam seguem tendo vinculo de amizade e acompanhando as decisões dos integrantes que ingressam posteriormente, fortalecendo a república ao longo do tempo.

Para manter uma república preservada é interessante que o custo de permanência estudantil seja baixo, logo a taxa de ocupação deve ser próxima de sua totalidade. De maneira complementar o número de ex alunos formados e calouros (bixos) batalhando são um indicativo de que a república é bem sucedida no ambiente universitário.

Baseado no que foi exposto, o trabalho se pauta no cadastro e edição dos dados de uma república e preservação deste ciclo de hierarquia e registro do mesmo por meio das promoções entre membros ao longo das gerações, perpetuando a organização destas entidades tão importantes e representativas em Ouro Preto.

Por objetivos específicos deste sistema, temos:

1. **Módulo para gestão de uma República.**

Tal módulo tem por finalidade permitir que a história, estrutura, localização e número de vagas limite são elementos fundamentais para a gestão da moradia. As informações aqui obtidas poderão ser repassadas posteriormente (caso necessário) ao frontend da aplicação (caso a república possua um site,cenário não tratado neste projeto). O quantitativo de vagas será utilizado na geração da taxa de ocupação da república gerada via relatório que auxiliará os moradores nesta política de divulgação de vagas de forma que a totalidade do preenchimento esteja sempre próxima do ideal. A saída deste relatório contemplará além da taxa de ocupação, a relação dos membros por curso que residem na república ao longo do tempo.

1. **Módulo para gestão de Membros**

Um membro é uma Entidade que poderá comportar os perfis de Bixo, Morador e Ex Aluno em um período. Um membro possui filiação, telefones de contato, email, status de atividade e falecimento, além do apelido que o tornará único na república. Por meio de um membro podemos navegar para as classes que são referenciadas por ele.

O Membro está em cada uma das fases associado a um perfil. Quando criado poderá ser associado ao perfil Bixo, posteriormente as vinculações ocorrerão em Morador e Ex Aluno por meio do método de promoção,respeitando a seguinte ordem; de Bixo para Morador a partir do processo de escolha e de morador para Ex Aluno a partir do processo de promoção para Ex Aluno. Por meio da conferencia de perfil de um Membro podemos determinar a real situação de todos aqueles que passaram pela república.

A associação de membros e suas respectivas classes de apoio ajudarão na composição do número de membros ativos na república no momento, visando auxiliar na geração do relatório da taxa de ocupação.

1. **Promoção de ciclo de passagem ( Calouro-Bixo para Morador).**

“Uma república sem bixos está destinada ao fracasso, vide que os bixos de hoje serão os moradores de amanhã, e os moradores atuais serão os ex alunos que aumentarão a rede de apoio à manutenção da república no futuro”.

Logo, o Bixo, associado ao Membro, deverá ter seus dados do processo seletivo informados. Datas de início e término e se o mesmo foi escolhido ou vetado na batalha de vaga.

Caso seja escolhido, ou promovido a Morador, deverá ser criado um registro novo de Morador, a atualização do perfil atual de Morador associado ao Membro, e inserção deste bixo na mais baixa hierarquia de moradores da república.

O decano é o morador mais velho de uma geração e ocupa o topo da hierarquia, sendo a ele delegado a gestão da república em seu maior nível.

## Definição Conceitual da Solução

## Diagrama de Casos de Uso

|  |
| --- |
| UseCase Diagram0.png |

## Requisitos Funcionais

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descrição Resumida** | **Dificuldade (B/M/A)\*** | **Prioridade**  **(B/M/A)\*** |
| RF01 | O administrador deve ser inserido automaticamente na criação do sistema. | B | A |
| RF02 | O administrador deve cadastrar/editar/atualizar/excluir Cursos que serão vinculados posteriormente aos membros.  Curso com algum Membro vinculado não poderá ser excluído. | M | A |
| RF03 | O administrador deverá listar todos os cursos cadastrados. | M | M |
| RF04 | O administrador deverá listar os membros de acordo com um curso específico. | M | B |
| RF05 | O administrador deve cadastrar/editar/atualizar/excluir Cidades que serão vinculadas posteriormente aos membros.  Cidade com algum Membro vinculado não poderá ser excluída. | M | A |
| RF06 | O administrador deve cadastrar/editar/atualizar/excluir Membros que serão vinculados posteriormente aos perfis de Bixo, Morador e Ex Aluno.  Membros que estejam vinculados em algum destes perfis não poderão ser excluídos.  O apelido de um Membro é um campo único e que não pode se repetir. | M | A |
| RF07 | O administrador deverá listar os membros cadastrados no sistema, apresentando o nível ocupado pelos mesmos no momento (Bixo,Morador, Ex Aluno) e possibilidade de vinculação em um destes perfis de acordo com qual nível o membro esteja associado. | A | A |
| RF08 | O administrador poderá informar o falecimento de um Membro | B | B |
| RF09 | O administrador poderá ativar/desativar um membro específico, sendo que nestes casos:   1. Caso o Membro possua um perfil de Bixo deverá ser inativado e o processo seletivo será marcado como falso se estiver em andamento. 2. Caso o Membro possua um perfil de Morador, a situação de moradia deverá ser marcada como falsa. | A | A |
| RF10 | O administrador deverá cadastrar/editar/atualizar/excluir Bixos que serão associados a um membro específico. | A | A |
| RF11 | O administrador poderá efetuar o veto de um Bixo em processo de batalha, sendo que neste caso, o processo de batalha será finalizado e o membro correspondente desativado.  Uma chamada ao método de geração do relatório da taxa de ocupação da república será feita de forma que o cenário esteja atualizado. | M | A |
| RF12 | O administrador poderá promover o bixo em processo de batalha a morador, sendo que neste caso o processo de batalha será concluído, o bixo será escolhido, sendo gerado um novo perfil de morador associado para o membro em questão, inserido no menor nível de hierarquia.  Uma chamada ao método de geração do relatório da taxa de ocupação da república será feita de forma que o cenário esteja atualizado. | M | A |
| RF013 | O administrador deverá listar todos os moradores cadastrados, residentes e vinculados a membros ativos. | M | M |
| RF14 | O administrador deverá gerenciar a hierarquia de um Morador específico, podendo promovê-lo ou rebaixá-lo. | B | M |
| RF15 | O administrador poderá promover um Morador a Decano. Poderá existir apenas um decano no grupo de moradores ativos e residentes. | M | M |
| RF16 | O administrador poderá promover um perfil de Morador a Ex Aluno. Neste caso será criado um novo registro de Ex Aluno associado ao Membro em questão, com a data de formatura configurada inicialmente como a data atual da promoção, sendo que a mesma poderá ser editada posteriormente.  O perfil de Morador residente na república deverá ser desativado.  Uma chamada ao método de geração do relatório da taxa de ocupação da república será feita de forma que o cenário esteja atualizado. | M | A |
| RF17 | O administrador deverá listar todos os ex alunos cadastrados e ativos ordenados pela data de formatura de forma ascendente. | B | A |
| RF18 | O administrador deverá editar/ atualizar a data de formatura de um Ex Aluno específico. | B | A |
| RF19 | O administrador poderá cadastrar/editar/atualizar/ excluir uma República.  A República deverá possuir ao menos uma vaga cadastrada.  A República não poderá ser excluída caso existam membros cadastrados. | B | A |
| RF20 | O administrador poderá anunciar vagas pela República. | B | M |
| RF21 | O administrador poderá cadastrar/editar/atualizar/ excluir Fotos associadas à República. | M | B |
| RF22 | O administrador poderá cadastrar/editar/atualizar/ excluir Vídeos associados à República. | M | B |
| RF23 | O administrador deverá listar todos os relatórios com a taxa de ocupação ordenada pela data de criação de forma decrescente. | M | A |
| RF24 | O administrador deverá gerar o relatório contendo a lista de Ex Alunos homenageados na Festa do Doze de Outubro no ano corrente. Os Ex Alunos selecionados deverão ter data de formatura proporcionais a 5 anos , ordenados de forma decrescente , os que possuírem maior tempo de formado aparecerão primeiro nesta listagem. | B | A |

\*B=Baixa, M=Média, A=Alta.

**Observação: acrescente quantas linhas forem necessárias.**

## Requisitos Não-funcionais

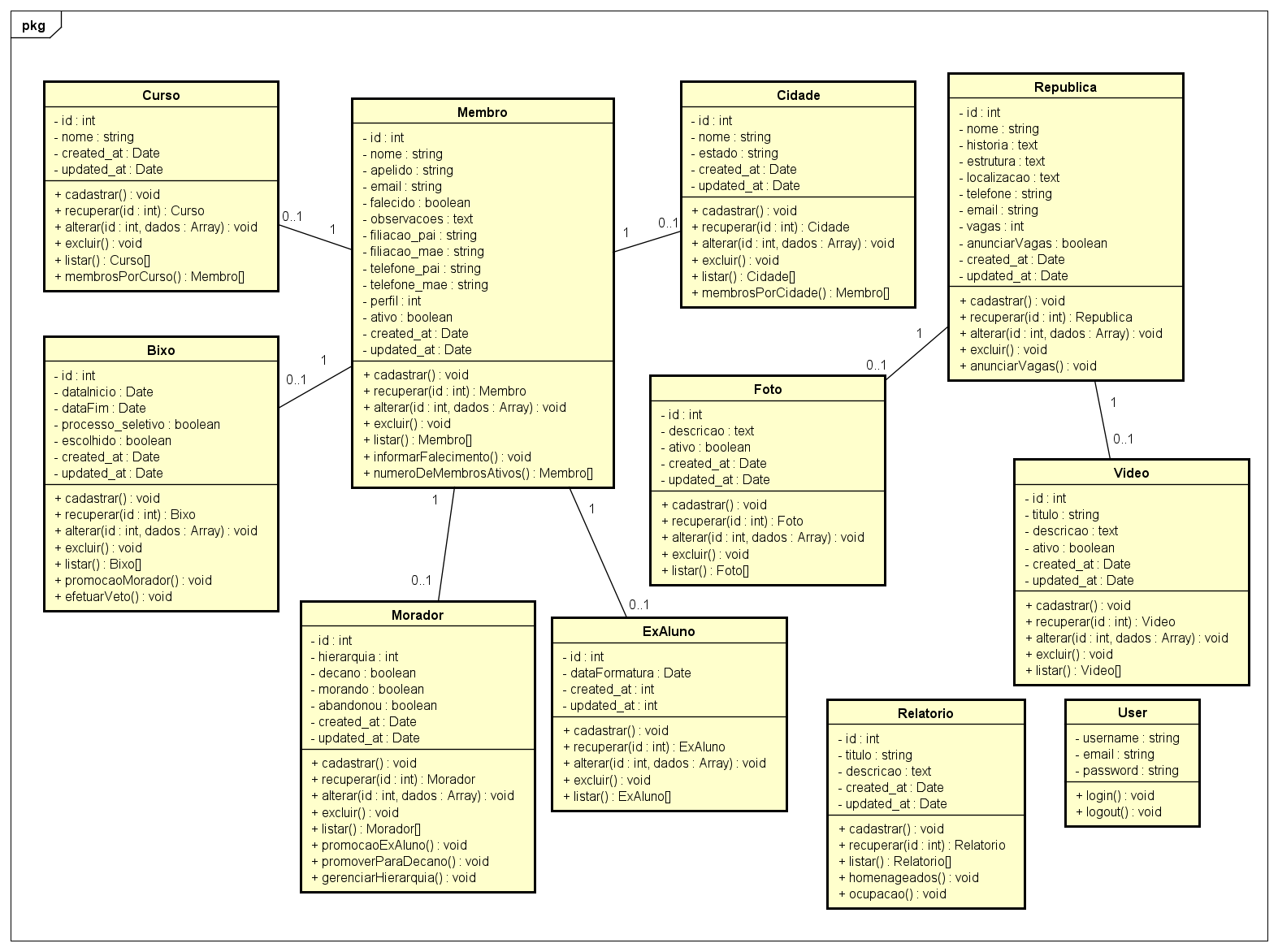
|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Prioridade**  **B/M/A** |
| RNF01 | O sistema deve ter alta disponibilidade, estando ativo 24 horas por dia x 7 dias por semana. | A |
| RNF02 | A Linguagem de programação deve ser o PHP. | A |
| RNF03 | O Framework utilizado deve ser o Laravel em versão superior ou igual a 5.8. | A |
| RNF04 | O banco de dados utilizado na aplicação deve ser o MySQL. | A |
| RNF05 | O layout do sistema deve ser responsivo. | A |
| RNF06 | A biblioteca Bootstrap deve ser utilizada para tratar a responsividade. | A |
| RNF07 | O sistema deverá executar nos navegadores Edge, Google Chrome e Mozilla Firefox | M |
| RNF08 | A paginação de todos os módulos do sistema deverá ser feita a cada 5 registros. | B |

**Observação: acrescente quantas linhas forem necessárias.**

## Protótipo Navegável do Sistema

|  |  |
| --- | --- |
| Projeto SGR (GitHub) | [Link do Projeto SGR](https://github.com/rscsouza/sgr) |
| Protótipo interativo (Vídeo) | [Link para vídeo do protótipo interativo em .mp4](https://github.com/rscsouza/sgr/raw/master/docs/prototipo_sgr_v1.mp4) |
| Protótipo interativo (Figma) | [Link para apresentação do protótipo interativo no Figma](https://www.figma.com/proto/mS8Ayxau6kdmbjjz1Yi8NP/Untitled?node-id=43%3A394&scaling=contain&page-id=0%3A1&starting-point-node-id=43%3A394) |
| Protótipo interativo (Wireframes) | [Link para wireframes do protótipo interativo](https://github.com/rscsouza/sgr/raw/master/docs/sgr_wireframes.pdf) |

## Diagrama de Classes de Domínio



## Arquitetura da Solução

## Padrão Arquitetural

O padrão escolhido para o projeto foi o MVC.

O Framework utilizado para a implementação do projeto é o Laravel, sendo a linguagem de programação escolhida para o Backend o PHP.

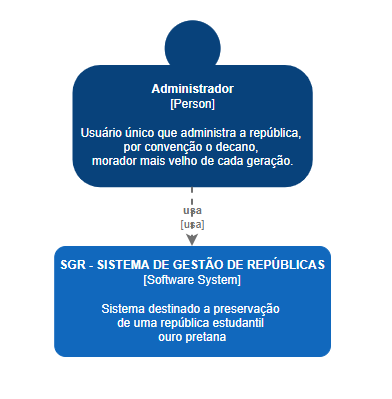
O projeto utiliza a estrutura do Laravel para o frontend denominada “Template Blade” para exibição das Views provenientes dos Controllers.

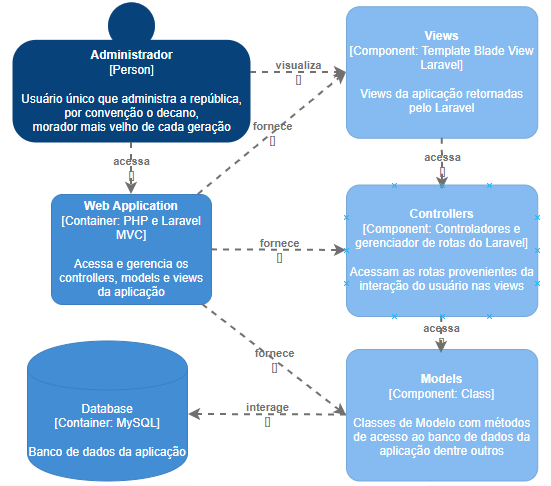
Ainda na parte de Frontend do projeto estão sendo utilizadas a biblioteca JQuery, Javascript nativo para interações do lado do cliente e a biblioteca Bootstrap Twitter em sua versão 5 para tratar a responsividade entre diferentes dispositivos.

## C4 model - Diagrama de Contexto

O primeiro diagrama apresenta a interação entre o usuário único do sistema (administrador), que por convenção é o decano , morador mais velho de cada geração da república.

O segundo diagrama mostra o administrador agindo no ambiente do framework , que retorna um conjunto de views ,controllers e models. Ao interagir com uma destas views, um controller e acionado junto ao gerenciador de rotas e este resgata as informações provenientes deste caminho, que se por ventura envolver acesso ao banco de dados, passa pelas classes de Model, retornando assim o resultado da operação após conclusão do processo para uma view específica no sistema, mantendo a consistência do modelo MVC.





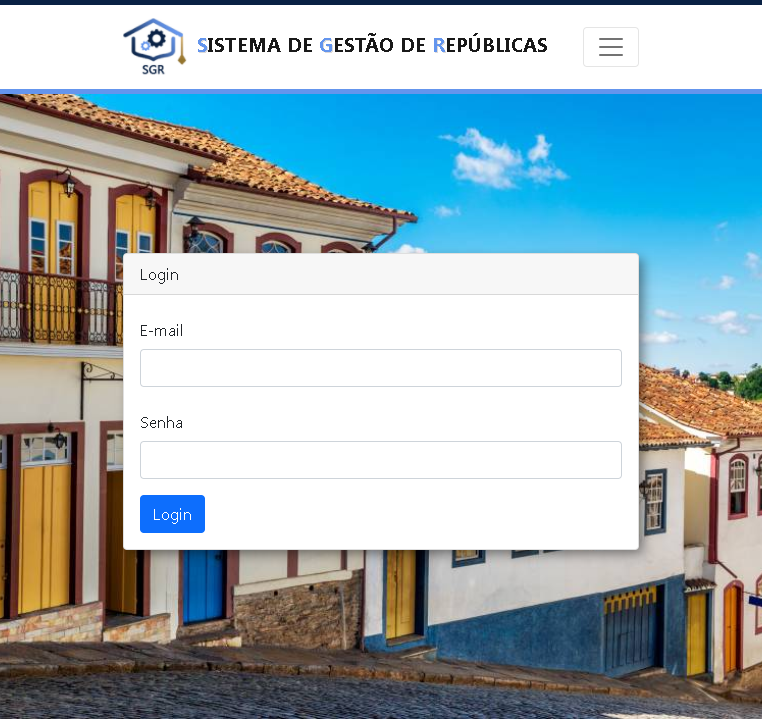
## Frameworks de Trabalho

O framework utilizado no frontend , backend e persistência é o Laravel que é voltado para a linguagem PHP. Na parte de interações com o usuário do lado do cliente temos a utilização da biblioteca JQuery e Javascript nativo.

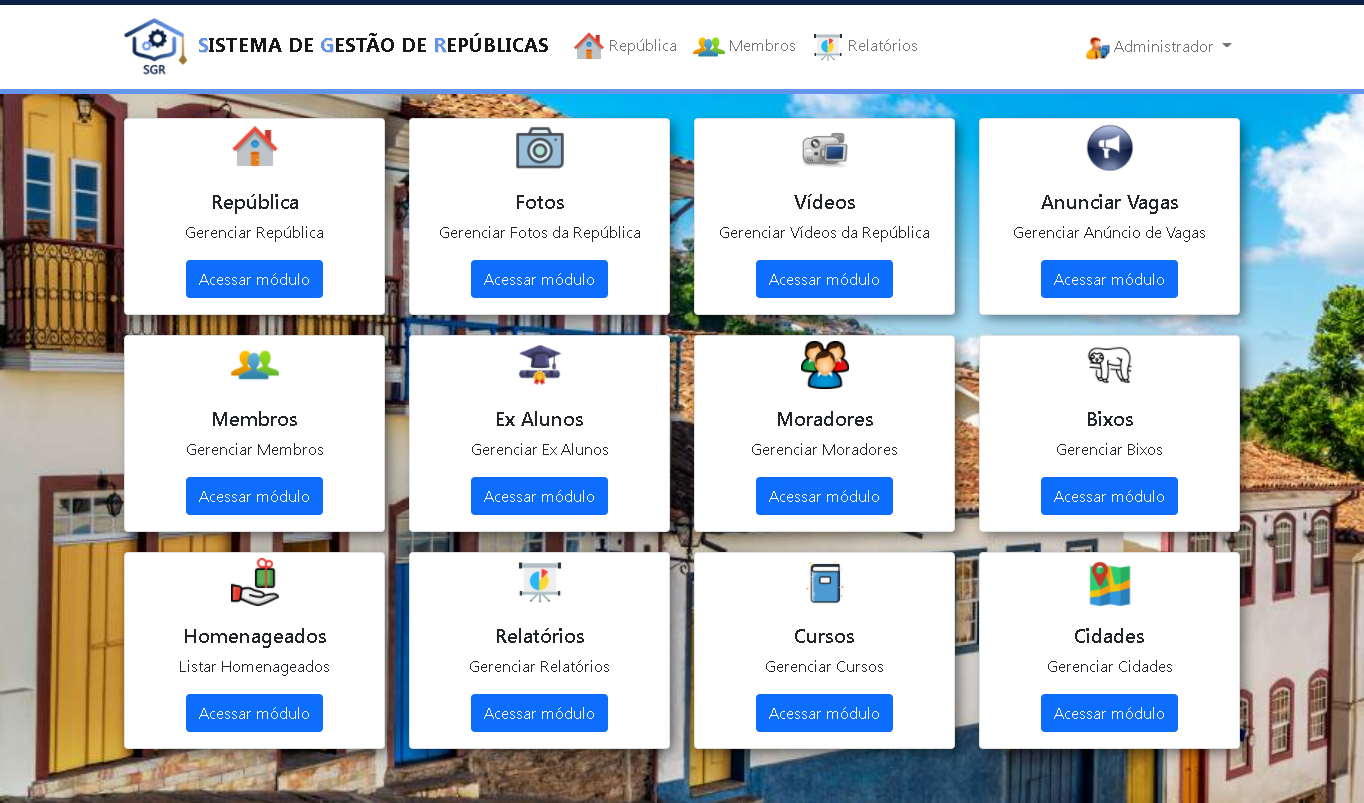
## Estrutura Base do Front End



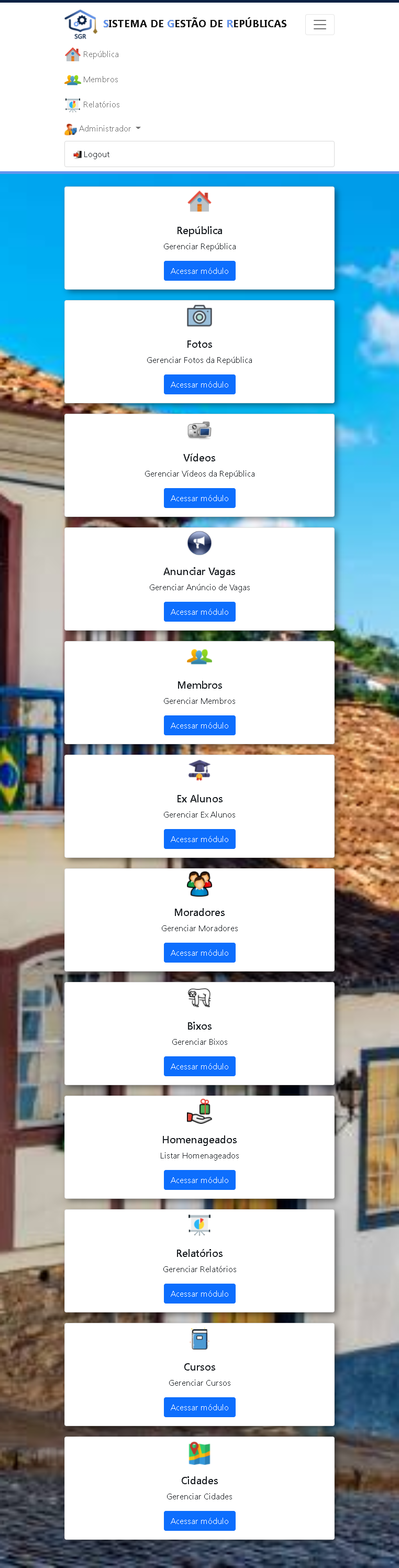
Tela de login (desktop)



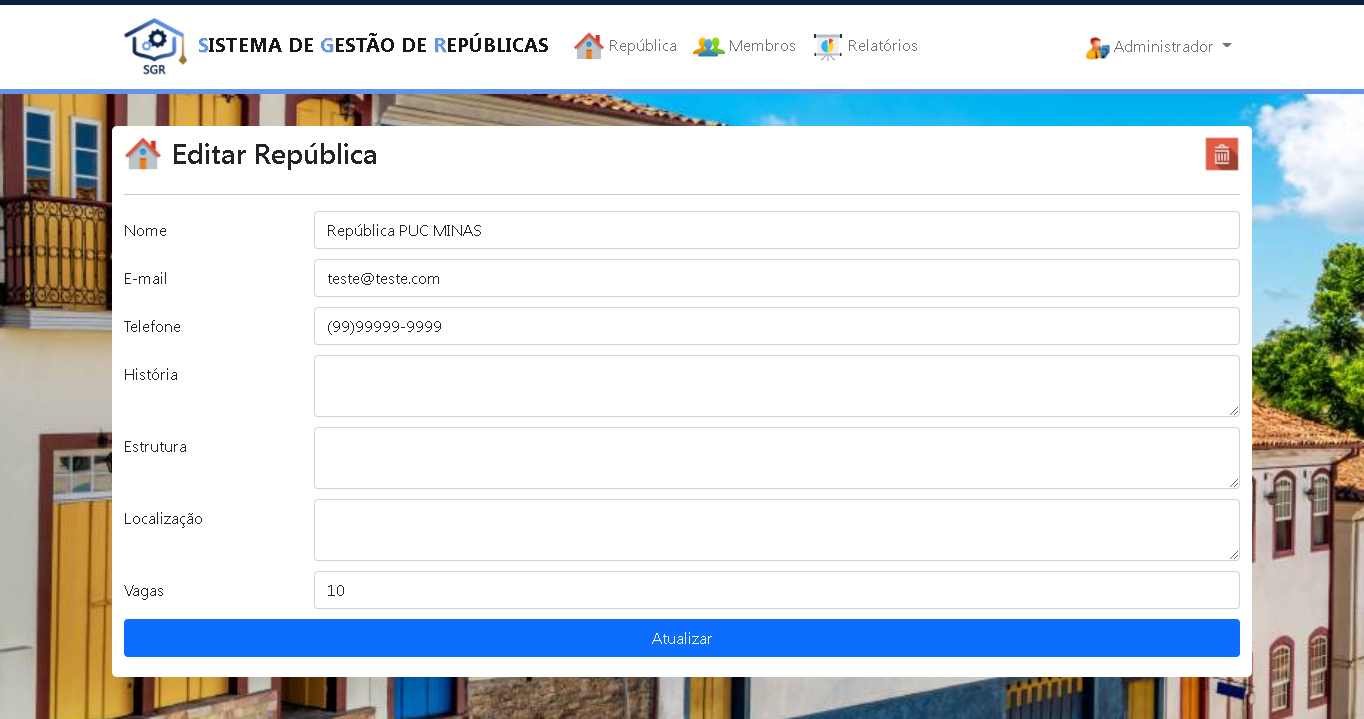
Tela de login (mobile)



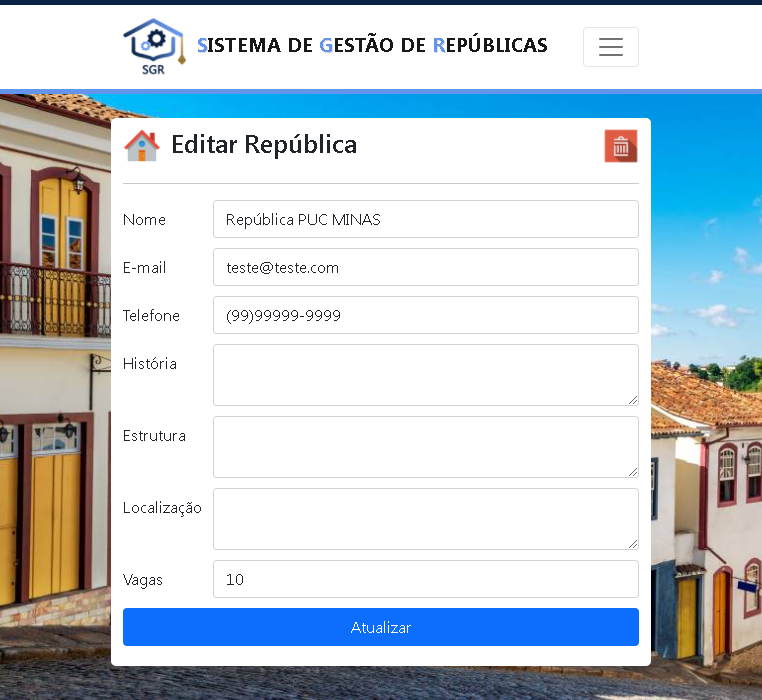
Tela de layout base (desktop) e menu do sistema ao topo



Tela de layout base (mobile) e menu do sistema ao topo



Tela da funcionalidade desenvolvida para segunda etapa (desktop)



Tela da funcionalidade desenvolvida para segunda etapa (mobile)

## Modelo Relacional ou Projeto de Banco de Dados NoSQL

<Cole aqui uma imagem legível do modelo relacional ou do projeto de banco de dados NoSQL de todo o sistema.>

## Plano de Testes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de uso** | **Objetivo do caso de teste** | **Entradas** | **Resultados esperados** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Relatório de Execução de Testes de Software

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de teste** | **Saída esperada** | **Resultados encontrados** | **Aprovado?** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Apropriação de Horas no Projeto

|  |  |  |
| --- | --- | --- |
| **Histórico de apropriação de horas** | | |
| **Data do registro** | **Atividade** | **Quantidade de horas** |
| 26/11/2021 -01/12/2021 | Organização inicial das idéias e cenários de aplicação do projeto SGR e seus direcionamentos futuros. | 30 |
| 02/12/2021 - 04/12/2021 | Expansão no detalhamento do texto de introdução partindo dos fatores de aspecto macro para micro, propiciando maior entendimento da aplicação. | 16 |
| 05/12/2021 -08/12/2021 | Criação do Diagrama de Casos de Uso da aplicação. | 21 |
| 09/12/2021 - 10/12/2021 | Especificação de requisitos funcionais da aplicação. | 8 |
| 11/12/2021 - 12/12/2021 | Especificação dos requisitos não funcionais da aplicação. | 8 |
| 13/12/2021 - 16/12/2021 | Criação do Diagrama de Classes de Domínio. | 24 |
| 17/12/2021 - 22/12/2021 | Criação do Protótipo interativo no Figma abordando os três cenários escolhidos para desenvolvimento dentro do Diagrama de Casos de Uso | 40 |
| 23/12/2021 - 23/12/2021 | Criação do repositório da aplicação no GitHub | 2 |
| 07/02/2022 | Entrega dos itens relativos a primeira etapa do projeto integrado | 1 |
| 02/03/2022 | Definição do padrão arquitetural da aplicação. | 2 |
| 03/05/2022 -05/03/2022 | Criação do Diagrama C4 Model. | 3 |
| 06/03/2022 - 07/03/2022 | Frameworks de trabalho | 1 |
| 08/03/2022 - 09/03/2022 | Estrutura base do Front End. | 5 |
| 10/03/2022 -15/03/2022 | Desenvolvimento de uma funcionalidade. | 24 |
| 15/03/2022 -16/03/2022 | Correção e ajustes após retorno do orientador na primeira etapa. | 12 |
| 16/03/2022 - 18/03/2022 | Submissão do Projeto no Heroku. | 2 |
| 30/03/2022 - 31/03/2022 | Submissão do Relatório Técnico e demais itens desenvolvidos na 2ª etapa do Projeto na pasta docs no GitHub. | 3 |
| 31/03/2022 | Entrega da 2ª etapa do Projeto Integrado. | 1 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Código da Aplicação

Repositório no GitHub: [Link do Projeto SGR](https://github.com/rscsouza/sgr)

Endereço da aplicação no Heroku: [Link do Projeto SGR no Heroku](http://app-sgr.herokuapp.com/login)

Usuário: [admin@sgr.com](mailto:admin@sgr.com)

Senha: senha

Vídeo com o protótipo desenvolvido na primeira etapa: [Link do vídeo](https://github.com/rscsouza/sgr/raw/master/docs/prototipo_sgr_v1.mp4)

Vídeo com a primeira funcionalidade desenvolvida para a segunda etapa (desktop): [Link do vídeo](https://github.com/rscsouza/sgr/raw/master/docs/crud_republica_desktop.mp4)

Vídeo com a primeira funcionalidade desenvolvida para a segunda etapa (móbile): [Link do vídeo](https://github.com/rscsouza/sgr/raw/master/docs/crud_republica_mobile.mp4)

## Avaliação Retrospectiva

< Nesta seção, você deve apresentar uma avaliação do processo de desenvolvimento do trabalho.>

## Objetivos Estimados

<Descreva, de forma sucinta, quais eram os objetivos estimados para a execução do projeto.>

## Objetivos Alcançados

< Descreva, de forma sucinta, quais foram os objetivos que realmente foram alcançados no projeto.>

## Lições aprendidas

<Descreva, de forma sucinta, quais foram as lições aprendidas na execução do projeto. A coluna “Classificação” deve ser preenchida com “Positiva” ou “Negativa”.>

|  |  |  |
| --- | --- | --- |
|  | **Retrospectiva (Lições Aprendidas)** | |
|  | **Descrição da Lição** | **Classificação** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |

## Referências

<Esse trabalho não requer revisão bibliográfica e, por isso, a inclusão das referências não é obrigatória, embora seja recomendada. Caso você deseje incluir referências empregadas em seu trabalho, relacione-as de acordo com as normas ABNT, disponíveis em [www.pucminas.br](http://www.pucminas.br), no *link*: <http://portal.pucminas.br/imagedb/documento/DOC_DSC_NOME_ARQUI20160217102425-n.pdf>.

Exemplo:

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.>