**UNIVERSIDADE VEIGA DE ALMEIDA**

**BACHARELADO EM <NOME DO CURSO DE GRADUAÇÃO>**

<**REMOVER ESTA NOTA EXPLICATIVA:**

AUTOR DESTE TEMPLATE: Professor André Lucio de Oliveira - Universidade Veiga de Almeida. Última atualização em: 30/09/2021

Legenda de cores:

1. Texto escrito em preto = sugestão de escrita. Você pode usar essa sugestão, ou mesmo modificar o texto.
2. Texto escrito em vermelho = texto que deve ser removido do seu trabalho. Em geral, são apenas notas explicativas, ou exemplos de aplicação de alguma técnica. Para deixar ainda mais claro, coloquei o texto “**REMOVER ESTA NOTA EXPLICATIVA**”.
3. Texto em realce amarelo = local onde você preencher com informações especificas sobre o seu trabalho. OBS: não deixar o “<” e o “>”.
4. Texto com realce verde = ponto de atenção (normalmente são observações relanvantes.

>

**<TÍTULO DO TRABALHO>**

**<NOME DO ALUNO>**

**RIO DE JANEIRO**

**<ANO>**

**UNIVERSIDADE VEIGA DE ALMEIDA**

**<NOME DO ALUNO>**

Trabalho de Conclusão de Curso apresentado no curso de <nome

do curso> da Universidade Veiga de Almeida como requisito parcial para obtenção do título de Bacharel em <nome do curso>.

Orientador: André Lucio de Oliveira

**<TÍTULO DO TRABALHO>**

**RIO DE JANEIRO**

**<ANO>**

**<PÁGINA DA BANCA>**

**<Pegar modelo com o nome dos membro da banca - apenas para projetos com defesa para banca>**

*<Dedicatória>*

**AGRADECIMENTOS:**

<Escrever aqui os agradecimentos>

**RESUMO**

<Escrever aqui o Resumo em Português e em parágrafo único>.

Palavras-Chave: <listar palavras-chave>

**SUMÁRIO**

**<Colocar aqui o Sumário>**

**1. INTRODUÇÃO**

**1.1. Nome do Sistema**

<Escreve o nome do Sistema a ser desenvolvido>

**1.2. Cenário Atual**

<Descrever o cenário atual antes do desenvolvimento da aplicação>

**1.3. Objetivos**

<Descrever os objetivos Gerais e específicos>

**1.4. Justificativa**

<Descrever as justificativas pela escolha do tema>

**2. REQUISITOS DO SISTEMA**

**2.1. Descrição Geral do Sistema**

<**REMOVER ESTA NOTA EXPLICATIVA:** Descrever aqui:

1. Uma visão geral do sistema, em parágrafos
2. Todas as principais necessidades e funcionalidades do sistema

OBS: Se necessário, utilize diagrama da UML (ou outros diagramas) para explicar os fluxos, estágios e comportamentos do sistemas, tais como: diagrama de atividades, diagrama de estágios, etc

Você pode usar o template a seguir, que apresenta a Visão do Produto através da técnica elevator pitch [MOORE, Geofrey A. Cross the Chasm: Marketing and selling disruptive products to mainstream customers. Harper Business, 2014]:

Visão do Produto

| **Para** | [cliente final] |
| --- | --- |
| **Cujo** | [problema que precisa ser resolvido] |
| **O** | [nome do produto] |
| **É um** | [categoria do produto] |
| **Que** | [benefício-chave, razão para adquiri-lo], |
| **Diferentemente do** | [alternativa da concorrência], |
| **O nosso produto** | [diferença-chave] |

**OBS**: Referenciar esta tabela no padrão ABNT.

>

**2.2. Partes Interessadas**

Nesta seção são apresentadas as partes interessadas do sistema.

<**REMOVER ESTA NOTA EXPLICATIVA:** Qualquer pessoa afetada ou significativamente interessada no produto de um projeto de software, em algum aspecto específico deste ou em informações geradas na execução do processo de desenvolvimento. As partes interessadas constituem o grupo de indivíduos diretamente envolvidos com as atividades de definição, desenvolvimento e validação do sistema, e podem contribuir de inúmeras maneiras para o projeto, tais como fornecendo recursos, provendo requisitos e/ou feedback no uso do sistema. Também devem ser considerados os indivíduos que influenciam ou sofrem influência do projeto. Clientes, Tomadores de Decisão, Donos do Negócio, Equipes de Desenvolvimento, Equipes de Infra-estrutura, Comunidade de Usuários dentre outros, são exemplos comuns de partes interessadas.

Algumas questões podem ser úteis na identificação das partes interessadas:

* Quem é o cliente (financiador do projeto)?
* Quem define os requisitos do projeto de software?
* Quem influencia o desenvolvimento do projeto de software?
* Quem irá avaliar e homologar o sistema quando este estiver pronto?
* Existe uma área de negócio responsável pela parte contratutal?
* Os testes do sistema serão realizados por uma equipe externa ao time do projeto?

**OBS**: Replique a tabela a seguir para cada parte interessada identificada no projeto

>

| **Nome** | **<Nome da parte interessada.>** |
| --- | --- |
| **Papel no desenvolvimento** | <Descrever brevemente a importância e o papel desempenhado pela parte interessada no processo de software.> |
| **Insumos ao sistema** | <Listar os produtos a serem entregues para servir de insumo ao desenvolvimento do projeto de software. Exemplo: reequistos do sistema, manuais operacionais, plano de testes, modelo de contrato> |
| **Representante** | <Identificação da(s) pessoa(s) e respectivo(s) órgão(s) que representa(m) a parte interessada> |

**2.3. Atores**

Nesta seção são apresentadoss os atores do sistema.

<**REMOVER ESTA NOTA EXPLICATIVA:** Um ator, por definição, já é uma parte interessada. Sendo assim, como já serão listados nesta seção, não é necessário incluí-los na seção anterior.

Um ator pode ser *uma entidade humana ou não-humana que interage diretamente com o sistema. Atores não-humanos podem ser hardware ou outros sistemas. Um ator participa em um cenário enviando ou recebendo mensagens do sistema.*

*Algumas questões podem ser úteis na identificação de atores de um sistema:*

* *Quem fornecerá, usará ou recuperará informações do sistema?*
* *Quem irá operar o sistema?*
* *De onde o sistema obtém as informações necessárias à sua operação?*
* *Quais sistemas externos interagem com o sistema?*

**OBS**: Replique a tabela a seguir para cada parte interessada identificada no projeto

>

| **Nome** | **<Nome do Ator>** |
| --- | --- |
| **Papel** | <Descrever brevemente a importância e o papel desempenhado pelo ator no sistema.> |
| **Insumos ao sistema** | <Listar os produtos e/ou informações a serem entregues pelo ator como insumo ao funcionamento do sistema, caso existam  Exemplo:   * Fornecer os dados do paciente * Analisar o retorno das consultas gerenciais * Agendar consultas no sistema   > |
| Representante | <Identificação da(s) pessoa(s) e respectivo(s) órgão(s) que representa(m) o ator (ou grupo de atores).> |

**2.4. Especificação dos Requisitos Funcionais**

<**REMOVER ESTA NOTA EXPLICATIVA:** o início desta seção deve conter uma explicação mais geral sobre o sistema, sem entrar em detalhes sobre cenários de testes e regras de negócio.

Vocês podem colocar uma descrição textual sobre a aplicação e, para ilustrar, caso vejam necessidade, incluir diagramas da UML, como:

* Diagrama de Atividades
* Diagrama de Estágios
* Diagrama de Casos de Uso

**OBS1**: Vocês não podem propor outros diagramas ou figuras, que não sejam da UML, mas ajudam no entendimento da aplicação.

**OBS2**: Os protótipos das telas das funcionalidades não devem ser colocados aqui (bem como as telas finais da aplicação), uma vez que devem ser colocadas junto com as Histórias de Usuário / Casos de Uso associados às mesmas.

>

**2.4.1. <Histórias de Usuário / Especificação de Casos de Uso>**

Nesta seção serão apresentadas as <histórias de usuário [você pode citar o livro do Mike Cohn - veja essa sugestão na seção referências, no final deste documento] / especificações dos casos de uso> do sistema.

<**REMOVER ESTA NOTA EXPLICATIVA:** Após escolher a forma de espeficicar os requisitos (história de usuário ou casos de uso), vocês devem escolher o um dos nomes acima para a seção 2.4.1

**OBS1**: A tabela seguir apresenta um exemplo de especificação de história de usuário, sendo apenas uma proposta. Vocês podem propor outros modelo, tanto para histórias de usuário, como de especificação de casos de uso.

**OBS2**: Não misture no seu trabalho História de Usuário com Caso de Uso. Escolha um única abordagem.

**OBS3**: A tabela exemplo a seguir deverá ser apresentada para cada história de usuário.

>

| **HU-1** | **Incluir Aluno** |
| --- | --- |
| **COMO** Funcionário da Secretaria  **POSSO** Incluir um novo aluno  **PARA** que a vida acadêmica do aluno possa ser gerenciada pelo sistema de controle acadêmico. | |
| **Cenários de Aceitação** | |
| **Cenário 1 <Sucesso>:** Inclusão do Aluno com sucesso  **DADO** que estou na tela de inclusão de novo aluno **E** preechi todos os dados obrigatórios (conforme definido na “RN1- Dados do Aluno”)  **QUANDO** clico no botão [Salvar**]**  **ENTÃO** exibir a mensagem “Aluno cadastrado com sucesso” | |
| **Cenário 2 <Navegabilidade>:** CPF informado pertence a um aluno egresso  **DADO** que estou na tela de inclusão de novo aluno  **QUANDO** informo o CPF de um aluno que já foi aluno da universidade  **ENTÃO** importar os dados do aluno e exibir na tela | |
| **Cenário 3 <Erro>:** Dados obrigatório não informados  **DADO** que estou na tela de inclusão de novo aluno **E** não informei todos os dados obrigatórios do Aluno (conforme definido na “RN1- Dados do Aluno”)  **QUANDO** clico no botão [Salvar**]**  **ENTÃO** exibir a mensagem “Informe todos os dados obrigatório” **E** sinalizar com borda “vermelha” todos os campos obrigatórios não informados | |
| **Cenário 4 <Erro>:** CPF inválido  **DADO** que estou na tela de inclusão de novo aluno  **QUANDO** informo um CPF inválido (conforme regra definina na “RN2 - Cálculo do Dígito Verificador do CPF”)  **ENTÃO** exibir a mensagem “CPF inválido” (abaixo do campo) | |
| **...** | |
| **Cenário n <Erro>:** ... | |
| **Prototipação** | |
| <**REMOVER ESTA NOTA EXPLICATIVA:** colocar aqui imagens de protótipos de tela que refletem esta história de usuário  Exemplos de Ferramentas para gerar protótipos: Figma, Balsamic, Draw.io, Pencil, etc>. | |
| **Telas da Aplicação** | |
| <**REMOVER ESTA NOTA EXPLICATIVA: OBS:** Apenas em TCC 2. | |

**2.4.2. Regras de Negócio**

<**REMOVER ESTE TRECHO:** inserir aqui as regras de negócio citadas pelas históias de usuário ou casos de uso. Por exemplo, as RN1 e RN2, citadas no exemplo acima poderim ser incluídas nesta seção:

**OBS**: Os exemplo abaixo são apenas para ilustar questões que podem ser inseridas nesta seção. Você não precisa necessáriamente usar essas regras de negócio

**RN1 - Dados do Aluno**

| **Campo** | **Obrigatório** | **Alterável?** | **Domínio** | **Máscara** | **Obs.** |
| --- | --- | --- | --- | --- | --- |
| Nome | Sim | Sim | - | - | - |
| Sexo | Sim | Sim | Masculino, Feminino, Outros | - | - |
| CPF | Sim | Não | - | 999.999.999-99 | - |
| ... |  |  |  |  |  |

**RN2 - Cálculo do Dígito Verificador do CPF**

O cálculo para validar um CPF é especificado pelo [Ministério da Fazenda](http://www.receita.fazenda.gov.br/aplicacoes/atcta/cpf/consultapublica.asp), que disponibiliza no próprio site as [funções](http://www.receita.fazenda.gov.br/aplicacoes/atcta/cpf/funcoes.js) (em javascript) para validação de CPF. Vamos entender como funciona.

O CPF é formado por 11 dígitos numéricos que seguem a máscara "###.###.###-##", a verificação do CPF acontece utilizando os 9 primeiros dígitos e, com um cálculo simples, verificando se o resultado corresponde aos dois últimos dígitos (depois do sinal "-").

### Validação do primeiro dígito

Primeiramente multiplica-se os 9 primeiros dígitos pela sequência decrescente de números de 10 à 2 e soma os resultados. Assim:

*5 \** ***10*** *+ 2 \** ***9*** *+ 9 \** ***8*** *+ 9 \** ***7*** *+ 8 \** ***6*** *+ 2 \** ***5*** *+ 2 \** ***4*** *+ 4 \** ***3*** *+ 7 \** ***2***

O resultado do nosso exemplo é:

***295***

O próximo passo da verificação também é simples, basta multiplicarmos esse resultado por 10 e dividirmos por 11.

*295 \** ***10*** */* ***11***

O resultado que nos interessa na verdade é o RESTO da divisão. Se ele for igual ao **primeiro dígito verificador** (primeiro dígito depois do '-'), a primeira parte da validação está correta.

**Observação Importante:** Se o resto da divisão for igual a 10, nós o consideramos como 0.

Vamos conferir o primeiro dígito verificador do nosso exemplo:

*O resultado da divisão acima é '268' e o RESTO é* ***2***

Isso significa que o nosso CPF exemplo passou na validação do primeiro dígito.

### Validação do segundo dígito

A validação do segundo dígito é semelhante à primeira, porém vamos considerar os 9 primeiros dígitos, mais o primeiro dígito verificador, e vamos multiplicar esses 10 números pela sequencia decrescente de 11 a 2. Vejamos:

*5 \** ***11*** *+ 2 \** ***10*** *+ 9 \** ***9*** *+ 9 \** ***8*** *+ 8 \** ***7*** *+ 2 \** ***6*** *+ 2 \** ***5*** *+ 4 \** ***4*** *+ 7 \** ***3*** *+ 2 \** ***2***

O resultado é:

***347***

Seguindo o mesmo processo da primeira verificação, multiplicamos por 10 e dividimos por 11.

*347 \** ***10*** */* ***11***

Verificando o RESTO, como fizemos anteriormente, temos:

*O resultado da divisão é '315' e o RESTO é* ***5***

Verificamos, se o resto corresponde ao segundo dígito verificador.

Com essa verificação, constatamos que o CPF 529.982.247-25 é válido.

>

**2.5 Especificação dos Requisitos Não Funcionais**

Nesta seção serão apresentadas os requisitos não funcionais do sistema.

<**REMOVER ESTA NOTA EXPLICATIVA:** Vocês podem escrever na forma de subseções, ou mesmo colocar os requisitos não funcionais em um tabela.

Nesta seção, vocês devem incluir questões como:

* Usabilidade: Ex: Help Online, Notas explicativas (hint), configurações de grid de resultados, etc
* Confiabilidade: Ex: Tempo médio entre falhas, Período do Dia que a aplicação estará disponível para uso, etc
* Desempenho: Ex: Tempo médio de retorno de uma consulta, numero máximo de usuários logados simultâneamente, etc
* Ambiente: Ex: Browsers e versão suportadas pela aplicação , Sistema Operacional necessário para executar a aplicação, etc
* **OBS**: Vocês pode pensar em outros requisitos não funcionais, exemplo: questões éticas, de legislação, etc

>

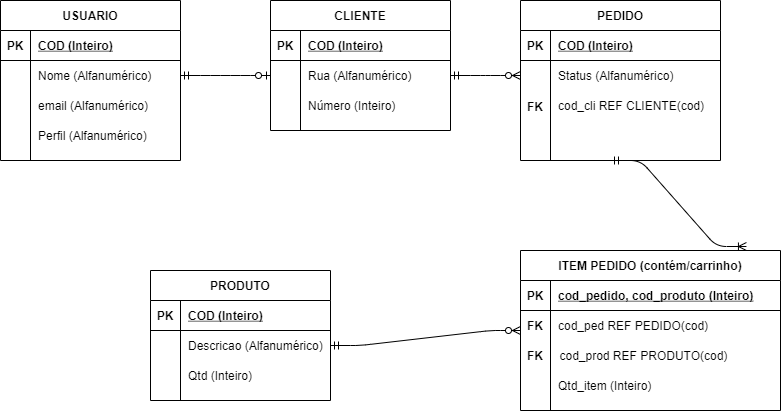
**3. MODELO DE DADOS**

Nesta seção será apresentado o modelo de dados da aplicação <nome do sistema>. A Tabela <99> apresenta o dicionário de dados referente a este modelo.

<**REMOVER ESTA NOTA EXPLICATIVA:** Incluir nesta seção o modelo lógico de dados da aplicação. Você pode usar a notaçao de MER, proposta por Peter Chan (conforme pode mostrada na livro do Navathe [Navathe. S e Elmasri, E. – Sistemas de Banco de Dados - 6ª ed – Addison-Wesley (2010)], ou a notação pé de galinha, proposta por James Martin. A seguir pode ser conferido um exemplo dessa segunda notação. Você também pode usar um diagrama de classes, mas peço bastante cuidado com os padrões definidos pela UML.

Após o desenho do modelo de dados, sugiro mostrar uma tabela com o dicionário de dados, explicando as entidades e os seus atributos.

>

****

| **Dicionário de Dados** | | |
| --- | --- | --- |
| **Entidade** | <Nome da Entidade-1> | |
| **Atributo** | **Tipo** | **Descrição** |
| <preencher> | <preencher> | <preencher> |
| **Entidade** | <Nome da Entidade-2> | |
| **Atributo** | **Tipo** | **Descrição** |
| <preencher> | <preencher> | <preencher> |
| **Entidade** | <Nome da Entidade-3> | |
| **Atributo** | **Tipo** | **Descrição** |
| <preencher> | <preencher> | <preencher> |

**4. GESTÃO DE PROJETO**

<**REMOVER ESTA NOTA EXPLICATIVA:** Nesta seção serão incluídas as informações sobre a forma como a gestão do projeto foi conduzida durante o desenvolvimento do sistema, tais como:

* Gestão em ferramentas: Jira, Trello, Planilhas.
  + OBS: Vocês podem colar telas dessas ferramentas, se julgarem necessário.
* Como os itens de backlog / tarefas foram organizadas e distribuídas entre os recursos.

>

**5. SOLUÇÃO ARQUITETURAL E TECNOLÓGICA**

**<EM CONSTRUÇÃO DO TEMPLATE. AGUARGEM …>**

**6. PROJETO E EXECUÇÃO DOS TESTES**

**<EM CONSTRUÇÃO DO TEMPLATE. AGUARGEM …>**

**7. CONCLUSÕES E TRABALHOS FUTUROS**

**<EM CONSTRUÇÃO DO TEMPLATE. AGUARGEM …>**

**REFERÊNCIAS**

<**REMOVER ESTA NOTA EXPLICATIVA:** Lista referências, seguindo o padrão da ABNT.

**OBS**: Todas as referências listadas nesta seção devem ser citada em algum ponto do texto.

**Algumas Referências Clássicas**:

**Engenharia de Software**

* Ian, Sommerville. Engenharia de Software – 10ª - Parson Addison-Wesley (2019)
* Roger S. Pressman, Engenharia de Software – McGraw-Hill. 8ª ed. (2016)

**UML**

* Fowler M. UML Essencial - Um breve guia para a linguagem-pdrão de modelagem de objetos. 3ª edição. Porto Alegre: Bookman 2005

**Banco de Dados**

* Date, C. J – Introdução a Sistemas de Banco de Dados – GEN LTC; 1ª ed (2004)
* Navathe. S e Elmasri, E. – Sistemas de Banco de Dados - 6ª ed – Addison-Wesley (2010)

**Machine Learning / Mineração de Dados**

* Han, J.; Kamber, M.; Pei, J. Data Mining: Concepts and Techniques. 3. ed. Boston: Morgan Kaufmann, 2011. (The Morgan Kaufmann Series in Data Management Systems). ISBN 978-0-12-381479-1.
* Ian H. Witten; Eibe Frank; Mark A. HallData Mining - Practical Machine Learning - Tools and Techniques - 3 ed.ISBN 978-0-12-374856-0

**Agile**

* M. Cohn, User Stories Applied: For Agile Software Development: Addison-Wesley Professional, 2004
* MOORE, Geofrey A. Cross the Chasm: Marketing and selling disruptive products to mainstream customers. Harper Business, 2014

>