**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

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

Projeto Integrado

Relatório Técnico

BarberShop

Leandro Oliveira Pereira

Belo Horizonte

Fevereiro, 2022

# Projeto Integrado

**Sumário**

Projeto Integrado 3

1. Cronograma de Trabalho 4

2. Introdução 5

3. Definição Conceitual da Solução 6

3.1 Diagrama de Casos de Uso 6

3.2 Requisitos Funcionais 6

3.3 Requisitos Não-funcionais 7

4. Protótipo Navegável do Sistema 7

5. Diagrama de Classes de Domínio 8

6. Arquitetura da Solução 8

6.1 Padrão Arquitetural 8

6.2 C4 model - Diagrama de Contexto 9

7. Frameworks de Trabalho 10

8. Estrutura Base do Front End 10

9. Modelo Relacional ou Projeto de Banco de Dados NoSQL 10

10. Plano de Testes 11

11. Relatório de Execução de Testes de Software 11

12. Apropriação de Horas no Projeto 11

13. Código da Aplicação 12

14. Avaliação Retrospectiva 12

14.1 Objetivos Estimados 13

14.2 Objetivos Alcançados 13

14.3 Lições aprendidas 13

15. Referências 13

## Cronograma de Trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 10/10/2021 | 20/10/2021 | 1. Definição dos requisitos funcionais e não-funcionais | Requisitos funcionais e não-funcionais |
| 21/10/2021 | 30/10/2021 | 1. Realização dos casos de uso | Diagrama de casos de uso |
| 01/11/2021 | 20/11/2021 | 1. Estudo de layouts e criação de design das telas | Protótipo navegável |
| 21/11/2021 | 10/12/2021 | 1. Criação do diagrama de classes | Diagrama de classes |
| 15/01/2022 | 20/02/2022 | 1. Escrita da Etapa 1 do trabalho | Etapa 1 do projeto |
| 20/02/ 2022 | 23/02 /2022 | 1. Definição do padrão arquitetural | Padrão arquitetural |
| 07/03/2022 | 09/03/2022 | 1. Definição do framework de trabalho | Frameworks utilizados no desenvolvimento |
| 10/03/2022 | 13/03/2022 | 1. Codificação da estrutura base do front end da aplicação | Estrutura base do front-end |
| 14/03/2022 | 03/05/2022 | 1. Desenvolvimento da primeira funcionalidade do sistema (front end, back end e persistência de dados) (crud de usuários e empresa) | Primeira funcionalidade do MVP funcional |
| 03/05/2022 | 03/05/2022 | 1. Arquitetura da solução c4 model | Arquitetura projeto c4 model |
| 03/05/2022 | 14/05/2022 | 1. Escrita da Etapa 2 do projeto | Etapa 2 do projeto |
| 16/05/2022 | 22/05/2022 | 1. Criação do modelo relacional do banco de dados | Modelo Relacional do banco de dados |
| 23/06/2022 | 28/06/2022 | 1. Desenvolvimento de funcionalidades do sistema para completar o MVP (front end, back end e persistência de dados) | MVP do sistema codificado |
| 29/07/2022 | 05/07/2022 | 1. Plano e Relatório de testes de software | Plano e relatório de testes do sistema |
| 06/08/2022 | 12/08/2022 | 1. Avaliação retrospectiva | Avaliação retrospectiva |

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

## Introdução

A sociedade passa por mudanças, entre essas, a maneira como o homem enxerga o momento de cuidar de sua aparência. A área da estética e cuidados pessoais sempre tiveram como prioridade o público feminino, porém vem acontecendo uma mudança no comportamento masculino. Surgiu um novo perfil, com novos hábitos, que deixa de preocupar-se apenas com sua higiene pessoal e passa a preocupar-se com sua aparência. (LIMA, 2016).

De acordo com Alves (2015), com a valorização da estética pelo público masculino, as barbearias tradicionais estão trazendo novas tendências para o mercado. O sucesso de um negócio depende da capacidade de prestar um serviço de maneira que atenda as necessidades do público-alvo e satisfaça os clientes. Segundo dados da Associação Brasileira da Indústria de Higiene Pessoal, Perfumaria e Cosméticos (ABIHPEC), 83% dos homens entrevistados afirmaram que “o tempo em que só mulheres cuidavam de sua aparência acabou” e que 54% frequentam regularmente barbearias.

Segundo Campos (2014), é uma tendência mundial trazer uma experiência masculina para as barbearias. As necessidades do público masculino englobam não apenas ter um corte de cabelo rápido e com qualidade, mas também permitir que o homem possa se sentir à vontade, para relaxar, conversar com amigos, tomar uma cerveja enquanto assiste um jogo de futebol e sair com uma aparência mais atrativa. Dessa forma, é possível escapar de um momento em que anteriormente não era agradável ficar esperando para tornar-se um momento prazeroso de encontro com amigos e muita conversa.

Não é fácil encontrar uma barbearia que proporcione ao cliente a satisfação esperada, pensando nisso este estudo concentra-se em resolver 2 problemas principais: facilitar a busca de barbearias que promovam uma experiência positiva no cliente final e fazer o gerenciamento de agenda tanto do cliente como do prestador de serviços da barbearia. Desta forma, será possível pesquisar por barbearias baseada em avaliações de outros clientes, por proximidade com seu endereço e por maior nota de avaliação. Além disso, cliente e barbeiro poderão ter suas agendas de marcações sempre atualizadas fazendo o acompanhamento dos serviços agendados.

Com a facilitação de encontrar uma barbearia, ambos os lados saem beneficiados, o cliente encontra a barbearia para realizar o serviço que procura e o administrador tem um aumento de receita decorrente do aumento de clientes atendidos.

O objetivo deste trabalho é apresentar a descrição do projeto de uma aplicação para facilitar o encontro de barbearias pelo cliente, bem como expor no site uma lista de barbearias e serviços cadastrados, podendo fazer o gerenciamento de agenda do cliente e do administrador da barbearia.

Os objetivos específicos são:

* Desenvolvimento de um site para auxiliar no gerenciamento de uma barbearia com agendamentos, relatórios e históricos de agendamentos;
* Criação da arquitetura backend para guardar e fornecer dados ao front-end;
* Desenvolvimento das funcionalidades: crud de usuário, adicionar empresa, crud funcionários e serviços, geração de relatórios, histórico de atendimentos e avaliação de atendimento;

## Definição Conceitual da Solução

## Diagrama de Casos de Uso

Com o intuito de descrever as funcionalidades propostas para o sistema a ser projetado e ajudar no levantamento de requisitos foi realizado um diagrama de casos de uso que pode ser verificado abaixo:

Diagrama de casos de uso também disponível no repositório github:

https://github.com/leandrobhte1/ETAPA1-PUC

|  |
| --- |
|  |

## Requisitos Funcionais

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descrição Resumida** | **Dificuldade (B/M/A)\*** | **Prioridade**  **(B/M/A)\*** |
| RF01 | O usuário deve auto cadastrar-se no sistema. | B | A |
| RF02 | O cliente pode buscar uma barbearia pelo nome da empresa | B | A |
| RF03 | O cliente pode buscar uma barbearia por proximidade informando seu endereço | A | B |
| RF04 | O cliente filtrar uma busca de barbearia pelas melhores notas de avaliação | B | B |
| RF05 | O cliente pode buscar por um serviço oferecido pela barbearia | B | A |
| RF06 | O cliente pode filtrar a lista de serviços pelo preço | B | B |
| RF07 | O cliente pode filtrar a lista de serviços pelos melhores avaliados | B | B |
| RF08 | O cliente pode agendar um serviço em uma barbearia | A | A |
| RF09 | Ao realizar um agendamento o cliente pode escolher o profissional que irá atendê-lo. | M | M |
| RF10 | O cliente poderá ver sua agenda com marcações. | M | M |
| RF11 | O cliente poderá ver seu histórico de agendamentos | M | M |
| RF12 | O cliente poderá avaliar os serviços da barbearia | M | B |
| RF13 | O administrador pode incluir uma ou mais unidades da barbearia | M | B |
| RF14 | O administrador pode incluir funcionários em suas unidades | M | M |
| RF15 | Um funcionário pode atender em mais de uma unidade em dias diferentes | A | B |
| RF16 | O administrador pode incluir/alterar/excluir os serviços prestados em cada unidade | B | B |
| RF17 | Um profissional pode realizar mais de um serviço | B | M |
| RF18 | O administrador pode ver a agenda de marcações de cada unidade | M | M |
| RF19 | O administrador pode incluir um novo horário na agenda como ocupado | B | M |
| RF20 | O administrador deve conseguir gerar relatórios administrativos | A | M |
| RF21 | Os relatórios deverão ter opções de filtragem de tempo de últimos 7 dias, 15 dias, mês e anual. | M | M |
| RF22 | O administrador poderá visualizar e responder as avaliações de seus clientes | M | B |
| RF23 | O administrador pode incluir horários de funcionamento de cada unidade | B | B |

\* 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 ser construído utilizado o framework React no frontend | A |
| RNF02 | Um relatório não deve demorar mais que 45 segundos para ser gerado | M |
| RNF03 | O carregamento da agenda não deve demorar mais que 30 segundos para ser carregada | A |
| RNF04 | Em caso de falhas, o sistema deverá voltar a funcionar em no máximo 24 horas | A |
| RNF05 | O sistema deverá ser desenvolvido com Java no backend usando framework spring boot | B |
| RNF06 | O banco utilizado deverá ser Postgres MySQL | M |
| RNF07 | O sistema deverá rodar no browser Google Chrome a partir da versão 90.0 | M |

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

## Protótipo Navegável do Sistema

O protótipo a ser apresentado tem por objetivo mostrar a tela inicial da aplicação, assim como a sequência de telas para 4 casos de usos principais, são eles:

1. Agendamento de horário para ser atendido (cliente)
2. Visualizar histórico de agendamentos e avaliar serviço (cliente)
3. Acesso a relatórios das unidades (Admin)
4. Visualizar agenda de marcações (Cliente e Admin)

Protótipo disponível para uso no link:

https://www.figma.com/proto/6WDA6g9jTCvAhos7OWxcQE/BarberShop-Prototype?node-id=1792%3A670&scaling=scale-down&page-id=10%3A2&starting-point-node-id=1792%3A670&show-proto-sidebar=1

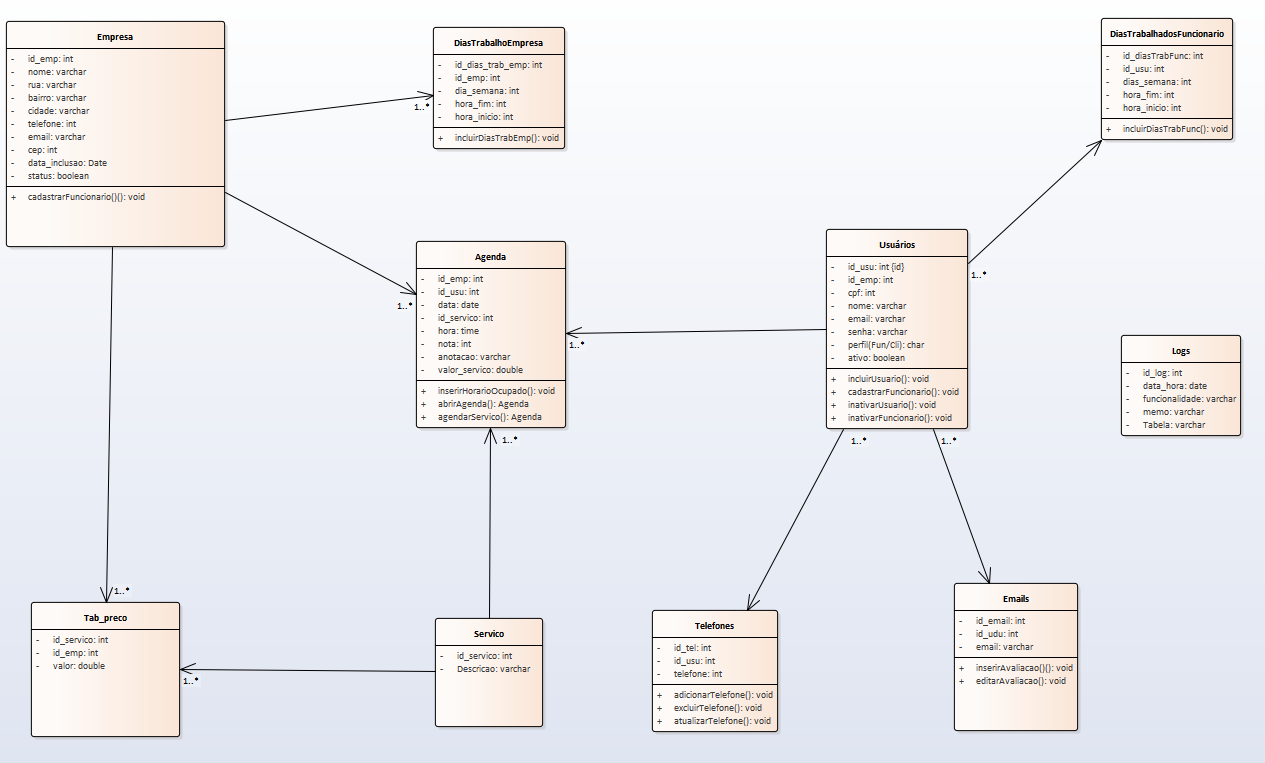
Vídeo gravado mostrando uso do protótipo navegável disponível no repositório github:

https://github.com/leandrobhte1/ETAPA1-PUC

## Diagrama de Classes de Domínio

Para representar a estrutura e relações entre as classes do sistema foi feito o diagrama de classes, que auxiliará a programação do sistema, a definição dos campos de cada classe, métodos e relacionamentos.

Diagrama de classes disponível também no repositório do GitHub para melhor visualização: https://github.com/leandrobhte1/ETAPA1-PUC



## Arquitetura da Solução

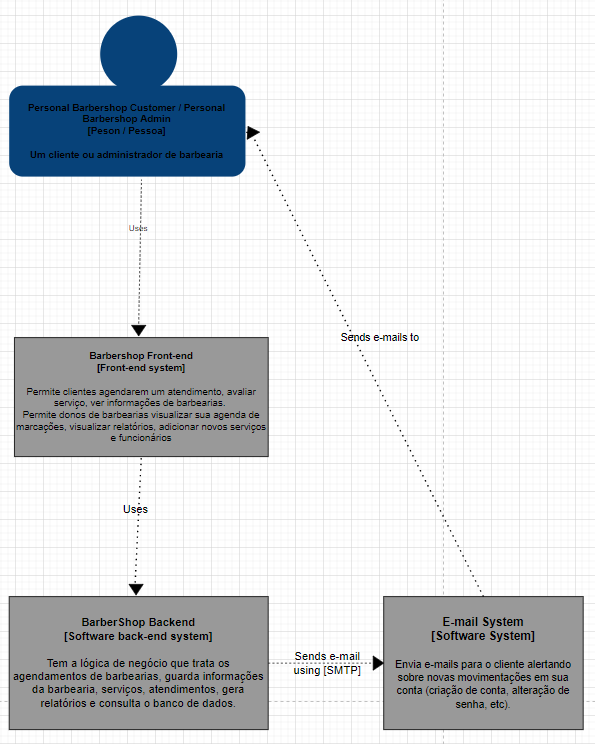
## Padrão Arquitetural

O back-end foi construído utilizando o padrão arquitetural MVC utilizando a linguagem JAVA e framework Spring (Spring boot e Spring Security) e o banco de dados escolhido foi o PostgreSQL.

A escolha do padrão MVC para o backend foi devido a sua simplicidade e larga utilização no mercado, sendo um dos padrões mais comuns e reconhecidos na programação pois facilita a troca de informações entre a interface do usuário ao sistema backend e banco de dados. Uma de suas responsabilidades é a otimização de velocidade entre as requisições feitas pelo comando dos usuários.

O front-end foi construído utilizando o framework React pois é um framework SPA focado no desempenho e simplicidade, além de ser um framework amplamente conhecido pela comunidade, o que facilita o desenvolvimento com auxílio de fóruns de desenvolvimento.

## C4 model - Diagrama de Contexto



**Figura 1 - Visão Geral da Solução. Diagram Context**

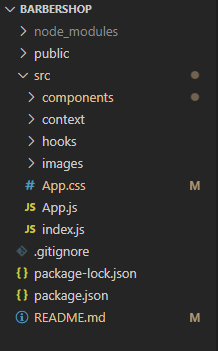
A Figura 1 mostra o diagrama de contexto da solução proposta, com todos os seus principais módulos: O front-end, site que o usuário do sistema Barbershop irá utilizar, que por sua vez conversa através de requisições http com o sistema de backend da Barbershop, módulo responsável por ter a lógica de negócio, tratar os agendamentos, informações de usuários e de barbearias. O sistema de back-end pode enviar e-mails aos usuários do sistema para notificar a criação de sua conta e alteração de senha.

## Frameworks de Trabalho

No front-end foi utilizado o framework React na versão 18.1, no backend foi utilizado o framework Spring boot e o banco de dados utilizado foi o PostgreSQL 13.

## Estrutura Base do Front End

Para o desenvolvimento do fornt-end a estrutura de pastas utilizada foi uma pasta para os componentes a serem re-utilizados na aplicação, uma pasta pros contextos e outra pros hooks conforme a imagem a seguir:

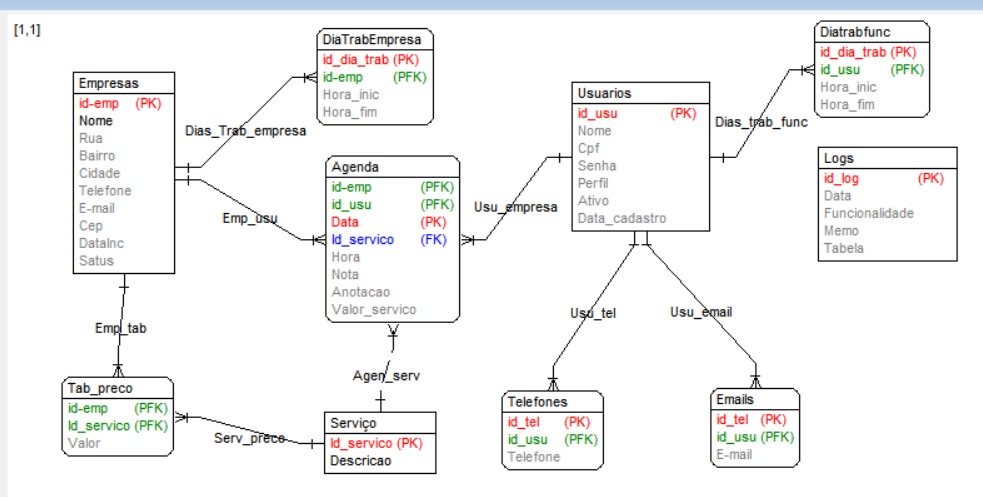


Os menus do sistema dependem do perfil do usuário (que pode ser user, manager, admin ou super admin). Para os perfis com maior acesso o menu aparecerá da seguinte maneira:



## Modelo Relacional ou Projeto de Banco de Dados NoSQL

Segue o Modelo Relacional utilizado para o desenvolvimento do projeto:



## 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** |
| 18/10/2021 | Definição dos requisitos funcionais | 12h |
| 20/10/2021 | Definição dos requisitos não-funcionais | 6h |
| 30/10/2021 | Construção dos casos de uso | 5h |
| 20/11/2021 | Criação de protótipo navegável | 40h |
| 10/12/2021 | Criação do diagrama de classes | 20h |
| 20/02/2022 | Escrita da introdução do trabalho | 12h |
| 01/04/2022 | Padrão arquitetural escolhido | 2h |
| 01/04/2022 | Definição dos frameworks de trabalho | 2h |
| 01/05/2022 | Estrutura base do front-end | 20h |
| 01/06/2022 | Desenvolvimento da primeira funcionalidade | 80h |
| 06/06/2022 | Arquitetura da solução C4 model | 2h |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Código da Aplicação

Código repositório front-end: <https://github.com/leandrobhte1/barbershopFrontPUC>

Código repositório back-end: <https://github.com/leandrobhte1/barbershopBack>

Repositório que contém o vídeo (no canvas diz que o vídeo deve ter 3 minutos e no template em word diz 5 minutos.. na dúvida fiz em 3 minutos para não ser penalizado) falando sobre as funcionalidades entregues na Etapa 2 (vídeos em formato .mp4 e .wmv): <https://github.com/leandrobhte1/ETAPA2-PUC>

Endereço site da aplicação: https://barbershop-site.herokuapp.com/

Credenciais de acesso:

- Perfil Admin:

Username: barbershoppuc@gmail.com

Password: 12345678

- Perfil Manager: **(OBS: Se atente ao . no e-mail)**

Username: barbershop.puc@gmail.com

Password: 12345678

- Perfil User: **(OBS: Se atente ao . no e-mail)**

Username: barber.shop.puc@gmail.com

Password: 12345678

< Informe aqui o ***link*** para seu **repositório público de código**.

Informe também o **endereço do *site* de sua aplicação**. É importante observar que, no ambiente fornecido para a avaliação, a base de dados deverá apresentar exemplos de teste previamente cadastrados que permitam visualizar o correto funcionamento do sistema. Indique também as **credenciais de acesso**, para **todos os perfis de usuários** da aplicação, que devem ser **empregadas pelos avaliadores**.

Indique o ***link*** para acesso ao **vídeo de apresentação de seu projeto**. Espera-se a produção de um vídeo sintético de, **no máximo**, **5 minutos**, no formato **MP4**, apresentando o projeto e a solução desenvolvida. Sugere-se que o aluno apresente um ciclo completo do que pode ser realizado pelos usuários principais da solução.>

## 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

ASSOCIAÇÃO BRASILEIRA DA INDÚSTRIA DA HIGIENE PESSOAL, PERFUMARIA E COSMÉTICOS (ABIHPEC). Mercado masculino de HPPC segue em crescimento. 04 ago 2016. Disponível em: https://abihpec.org.br/release/mercado-masculino-de-hppc-segue-em-crescimento/ segueem-crescimento. Acesso em: 12/02/2022.

< 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.>