***I-FAVOR***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

|  |
| --- |
| ***Autores (até 5 – nome completo):***  Murilo de Souza Portela  Nicolas Romero Sicca  Thiago Henrique Xavier Medeiros  ***Turma:***  Engenharia Leibnitz  ***Supervisão:***  Professores(as): Arnaldo Carlos Muller Junior e Dani Juliano Czelusniak |

**Curitiba, 2020**

**Sumário**

[1. Visão Geral do Aplicativo 3](#_Toc37326299)

[2. Requisitos Funcionais e Não Funcionais 4](#_Toc37326300)

[3. Diagrama de casos de uso: 5](#_Toc37326301)

[4. Especificação dos casos de uso: 6](#_Toc37326302)

[5. Arquitetura do Software - Diagrama de Classes 9](#_Toc37326303)

[6. Arquitetura do Software - Diagrama de Entidade-Relacionamento 10](#_Toc37326304)

[7. Interação Humano Computador (IHC) 11](#_Toc37326305)

[8. Prototipagem em Python/Kivy 16](#_Toc37326306)

[9. Referências 17](#_Toc37326307)

[10. Mapas de Avaliação 18](#_Toc37326308)

|  |
| --- |
| Visão Geral do Aplicativo |

O I-Favor trata-se de uma aplicação que tem por finalidade fornecer diversos serviços, facilitando o contato entre prestadores de serviços e seus potenciais clientes. Opta por uma interface simples para facilitar o acesso e manuseio da aplicação e permitir agilidade na solução das necessidades, tem como diferencial a confiabilidade nos profissionais cadastrados através de um processo de verificação.

Trabalha com o lema “Com um simples toque na tela suas necessidades serão prontamente resolvidas”. A linguagem de programação utilizada, python, permite uma programação leve e ágil.

Através de pesquisas feitas no mecanismo de busca Google©, como “aplicações de prestação de serviços” e nas plataformas Google Play© e Apple Store© que oferecem aplicativos, buscando soluções semelhantes as propostas pelo projeto em questão, encontrou-se alguns aplicativos com real similaridade. Entre esses, pode-se enfatizar três em questão: 1) Getninjas, 2) Triider e 3) Chamawill.

Os sistemas de ofertas de serviços em questão foram escolhidos também com relação ao potencial de mercado, vale citar como exemplo o Getninjas sendo considerado pela Forbes, em 2017, como uma das empresas mais promissoras do Brasil [1]. Para a análise com base no projeto realizado foram analisados os seguintes aspectos:

* Simplicidade no acesso e manipulação do aplicativo.
* interface limpa e de linguagem fácil.
* sistema de confiabilidade referente ao prestador de serviço.

1. **Getninjas:** Fundado em 2011 por Eduardo L’Hotellier, disponível para Android e iOS a aplicação trata-se de uma plataforma que oferece serviços diversos como zelador, engenheiro, encanador, entre outros. Existe no aplicativo duas possibilidades de usuário, sendo uma como cliente e outra como prestador de serviços, este, oferta seus serviços definindo sua área de atuação e em seu perfil pode conter recomendações de clientes que já contrataram os serviços [1].

Como cliente é possível escolher os serviços que deseja, entre os vários ofertados podendo definir o “grau” de sua necessidade, ou seja, se necessita por exemplo para 15 dias, 30 dias ou até mesmo se precisa urgentemente do serviço.

Tela de celular com publicação numa rede social

Descrição gerada automaticamenteTela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Figura1. Interface na função cliente Getninja.

Fonte: Reprodução do aplicativo

1. **Triider:** Plataforma de serviços criada por uma equipe de amigos, Aline Murlick, Paulo Guilherme Gil, Thiago Murlick e Juliano Murlick. A Triider oferta serviços diversos para regiões do Brasil trazendo como diferencial um sistema seguro para o cliente, pois conta com um cauteloso processo de verificação dos profissionais cadastrados no sistema, e garantia nos serviços prestados [2].

A aplicação está disponível para Android e iOS, com versões de programas diferentes para cliente e para prestadores de serviços (Triider para profissionais). Como cliente é preciso preencher um formulário, ou acessar pelo face book, para acesso ao aplicativo e para solicitar o serviço, descrever o que será realizado.

Tela de celular com publicação numa rede social

Descrição gerada automaticamenteTela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Figura2. Interface na função cliente Triider.

Fonte: Reprodução do Aplicativo.

1. **Chamawill:** Aplicativo consiste em uma plataforma que ajuda clientes as se conectarem com profissionais que prestam os mais variados tipos de serviços, entre estes serviços de encanador, pedreiro, programador, técnico de informática entre outros. Ao acessar o aplicativo o usuário vai direto para a tela de serviços e seleciona o que deseja, para prosseguir com a solicitação preenche um formulário de cadastro. Como diferencial a opção para prestador de serviço mulher [3].

Uma imagem contendo screenshot, texto

Descrição gerada automaticamenteTela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Figura3. Interface função cliente Chamawill

Fonte: Reprodução do aplicativo.

**Tabela1. Comparação entre os aplicativos similares e a aplicação do projeto**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Aspectos análisados | Getninjas | Triider | Chamawill | I-Favor (projeto) |
| Simplicidade no acesso e manipulação do aplicativo. | NÃO | SIM | SIM | SIM |
| Interface limpa e de linguagem fácil | SIM | NÃO | SIM | SIM |
| sistema de confiabilidade referente ao prestador de serviço | NÃO | SIM | NÃO | SIM |

|  |
| --- |
| Requisitos Funcionais e Não Funcionais |

* Requisitos funcionais:
  + RF01: Permitir que o usuário insira usuário e senha para se autenticar
  + RF02: Listar prestadores de serviço
  + RF03: Informar detalhes do serviço para o usuário
  + RF04: Solicitar um serviço
  + RF05: Listar usuários que querem um serviço para o prestador de serviço
  + RF06: Criar um serviço
* Requisitos não funcionais:
  + RNF01: Rodar em Smartphones com sistema operacional Android

|  |
| --- |
| **Diagrama de casos de uso:** |

|  |
| --- |
|  |

|  |
| --- |
| **Especificação dos casos de uso:** |

Especificação dos casos de uso:

|  |  |
| --- | --- |
| Identificação: UC01 | |
| Nome: Listar serviços | |
| Requisitos envolvidos: RF01, RF02 | |
| Sequência típica de eventos: Usuário acessa o aplicativo | |
|  | |
| *Usuário:* | *Sistema:* |
| * Acessa tela principal do cliente * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | * Busca no banco serviços disponíveis * Envia os serviços para o usuário * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |
| --- | --- |
| Identificação: UC02 | |
| Nome: Detalhar serviço | |
| Requisitos envolvidos: RF01, RF03 | |
| Sequência típica de eventos: Usuário clica no botão de informações do serviço | |
|  | |
| *Usuário:* | *Sistema:* |
| * Acessa tela principal do cliente * Escolhe um serviço * Aperta no botão de informações * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | * Busca informações do serviço escolhido no BD * Redireciona usuário para tela de informações * Apresenta detalhes do serviço * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |
| --- | --- |
| Identificação: UC03 | |
| Nome: Visualizar solicitações de serviço | |
| Requisitos envolvidos: RF01, RF04 | |
| Sequência típica de eventos: Usuário clica no botão de informações do pedido | |
|  | |
| *Usuário:* | *Sistema:* |
| * Acessa tela principal do cliente * Muda para a tela de minhas solicitações * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * Acessa tela principal do prestador de serviços | * Busca informações dos pedidos no BD * Apresenta pedidos para o usuário * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |
| --- | --- |
| Identificação: UC04 | |
| Nome: Cadastrar serviço | |
| Requisitos envolvidos: RF01, RF06 | |
| Sequência típica de eventos: Usuário preenche dados do serviço | |
|  | |
| *Usuário:* | *Sistema:* |
| * Acessa tela principal do prestador de serviços * Muda para tela de meus serviços * Clica no botão de adicionar novo serviço * Preenche dados do serviço * Aperta no botão de salvar | * Envia o usuário para tela de cadastro * Carrega os dados do usuário * Envia dados para o BD * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |
| --- | --- |
| Identificação: UC05 | |
| Nome: Aceitar solicitação de serviço | |
| Requisitos envolvidos: RF01, RF05 | |
| Sequência típica de eventos: Usuário aperta botão de aceitar solicitação | |
|  | |
| *Usuário:* | *Sistema:* |
| * Acessa tela principal do prestador de serviços * Escolhe solicitação * Aperta botão de aceitar solicitação * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | * Carrega solicitações do prestador * Atualiza status do pedido * Atualiza tela do usuário * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |
| --- | --- |
| Identificação: UC06 | |
| Nome: Buscar serviço | |
| Requisitos envolvidos: RF01, RF02 | |
| Sequência típica de eventos: Usuário insere busca e clica para pesquisar | |
|  | |
| *Usuário:* | *Sistema:* |
| * Acessa tela principal do cliente * Insere nome do serviço * Aperta botão de pesquisa * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | * Recebe dados da busca * Pesquisa no BD * Envia lista de serviços para o usuário * Atualiza página * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |
| --- | --- |
| Identificação: UC07 | |
| Nome: Solicitar serviço | |
| Requisitos envolvidos: RF01, RF02, RF04 | |
| Sequência típica de eventos: Usuário acessa informações do serviço e clica para solicitar | |
|  | |
| *Usuário:* | *Sistema:* |
| * Acessa tela principal do cliente * Escolhe um serviço * Aperta para visualizar informações * Aperta botão de solicitar serviço * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | * Carrega serviços para o usuário * Carrega serviço escolhido * Cria solicitação * Envia solicitação para o prestador * Adiciona solicitação na tela de minhas solicitações |

|  |  |
| --- | --- |
| Identificação: UC08 | |
| Nome: Cancelar solicitação de serviço | |
| Requisitos envolvidos: RF01, RF03, RF05 | |
| Sequência típica de eventos: Usuário escolhe serviço e clica para cancelar | |
|  | |
| *Usuário:* | *Sistema:* |
| * Acessa tela de meus serviços (ou tela de serviços caso seja o prestador) * Escolhe serviço * Aperta para visualizar informações * Aperta botão de cancelar serviço | * Carrega pedidos do o usuário * Carrega pedido escolhido * Remove solicitação do BD * Envia usuário para página principal * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |
| --- | --- |
| Identificação: UC09 | |
| Nome: Buscar solicitação | |
| Requisitos envolvidos: RF01, RF03, RF05 | |
| Sequência típica de eventos: Usuário insere nome do serviço e aperta para buscar | |
|  | |
| *Usuário:* | *Sistema:* |
| * Acessa tela de meus serviços (ou tela de serviços caso seja o prestador) * Insere nome do serviço que quer buscar * Aperta botão de pesquisa * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | * Carrega pedidos do o usuário * Busca pedidos com a pesquisa do usuário * Envia para o usuário * Atualiza página * \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |
| --- |
| Arquitetura do Software - Diagrama de Classes |

|  |
| --- |
|  |

|  |
| --- |
| Arquitetura do Software - Diagrama de Entidade-Relacionamento |

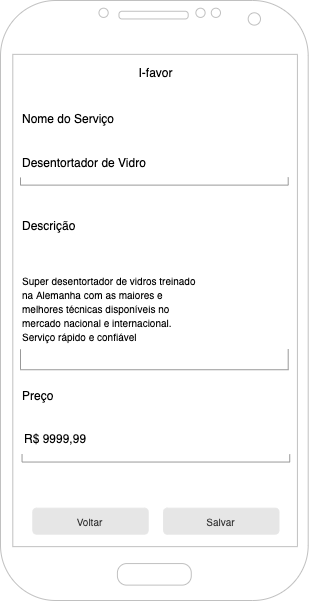
|  |
| --- |
|  |

|  |
| --- |
| Interação Humano Computador (IHC) |

Apresente aqui esboços das telas do aplicativo, conforme instruções constantes na segunda parte da pesquisa efetuada em TDE.

|  |  |
| --- | --- |
| Tela 1- Tela de Login | Tela 2 - Tela de Listagem dos serviços disponíveis para o Cliente |
| Tela 3 - Tela de informações de um serviço para o Cliente | Tela 4 - Tela de listagem dos serviços solicitados pelo Cliente |

|  |  |
| --- | --- |
| Tela 5 - Tela de informações do serviço solicitado pelo Cliente    Tela 7 - Tela de informações de um cliente e serviço para o Prestador de Serviços | Tela 6 - Tela de serviços solicitados ao Prestador de Serviço      Tela 6 - Tela de serviços que o Prestador de Serviços oferece |



Tela 7 - Tela de edição de um serviço oferecido pelo Prestado de Serviços

* Descreva aqui quais **componentes visuais** serão utilizadas no aplicativo e com **qual finalidade**.

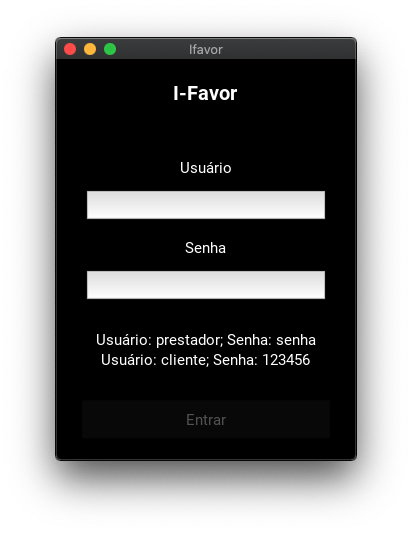
|  |
| --- |
| **Label:** serão utlizadas para desmonstra informações para o usuário (cliente e prestador de serviços). Com ela serão mostrados o nome da aplicação, as informações de um serviço (para o cliente e para o prestador de serviços), as informações de um usuário e serviço solicitado (para o prestador de serviços) |
| **Campo de texto:** campos para o usuário inserir informações. Tela 1 - Utilizado para informar as credenciasis do usuário. Telas 2, 4 e 6 – campo de filtro de um serviço pelo nome do mesmo. Tela 9 – campos de informações de um serviço a ser editado/criado por um Prestador de Serviços. |
| **Navegação por abas:** Telas 2 e 4 – navegação por abas da área do Cliente. Telas 6 e 8 – navegação por abas da área do Prestador de serviço |
| **Lista:** Tela 2 – lista de todos os serviços disponíveis na plataforma que o Cliente pode solicitar. Tela 4 – lista detodos os serviços solicitados pelo Cliente. Tela 6 – lista de todos os serviços solicitados ao Prestador de Serviços. Tela 8 – lista de todos os serviços que um Prestador de Serviços cadastrou |
| **Botão:** Tela 1 – botão para acessar a aplicação (Tela 2 caso seja um Cliente ou Tela 6 caso seja um Prestador de Serviços) . Tela 2 – botão de desconectar da conta (Sair), botão de busca e botão para acessar tela de informações de um serviço (Tela 3). Tela 3 -botão para voltar para a página principal do Cliente (Tela 2) e botão para solicitar um serviço. Tela 4 - botão de desconectar da conta (Sair), botão de busca e botão para acessar tela de informações de um serviço solicitado (Tela 5). Tela 5 -botão para voltar para a página principal do Cliente (Tela 2) e botão para cancelar um serviço solicitado. Tela 6 - botão de desconectar da conta (Sair, volta para a Tela 1), botão de busca, botão para acessar tela de informações de um serviço solicitado ao Prestador de Serviços (Tela 7) e botões de aceitar e rejeitar solicitação de serviço. Tela 7 - botão para voltar para a página principal do Prestador de Serviços (Tela 6) e botão para cancelar um serviço solicitado. Tela 8 - botão de desconectar da conta (Sair, volta para a Tela 1), botão de adicionar novo serviço (Tela 9) e botão editar um serviço cadastrado (Tela 9). Tela 9 - botão para voltar para a página principal do Prestador de Serviços (Tela 6) e botão para salvar o serviço criado/editado. |

* Descreva aqui quais **componentes sonoros[[1]](#footnote-1)** serão utilizadas no aplicativo e com **qual finalidade**.

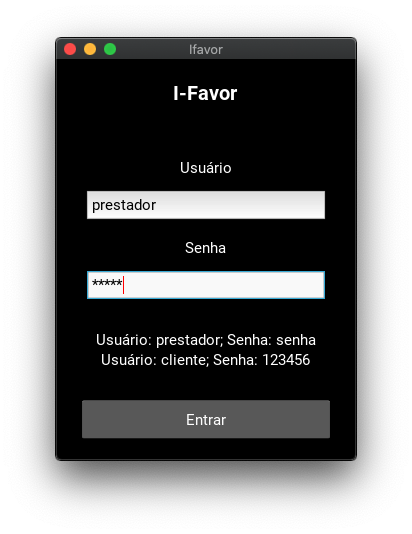
|  |
| --- |
| Não serão utilizados componentes sonoros |

|  |
| --- |
| Prototipagem em Python/Kivy |

Apresentar, além das telas construídas em Kivy, quais métodos das classes são ativados pelos botões e eventuais outros componentes de tela.



Tela 1 sem nenhuma informação de usuário informada.



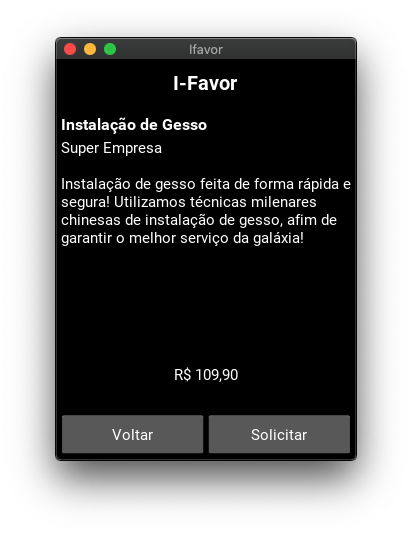
Tela 1 após usuário inserir dados de login.

* O botão “Entrar” aciona o método entrar() da classe Login



Tela 2 mostrando os serviços disponíveis para o Cliente.

* Ao clicar em “Minhas Solicitações” o usuário é alterado para a Tela 4, isso ocorre sem chamar nenhum método já que está sendo utilizado o widget TabbedPanel.
* O botão com a lupa faz uma busca de serviços que contenham o texto do TextField ao seu lado. Ele ativa o método popular\_listagem() da classe PrincipalCliente.
* O botão com o “i” abre a Tela 3 com as informações do serviço selecionado. Ativa o método exibir\_detalhes() da classe Linha.
* O botão “Sair” desconecta da conta e volta para a Tela 1. Executa o método sair() da classe PrincipalCliente.
* Ao entrar na Tela 2 o método popular\_listagem() da classe PrincipalCliente é executado.



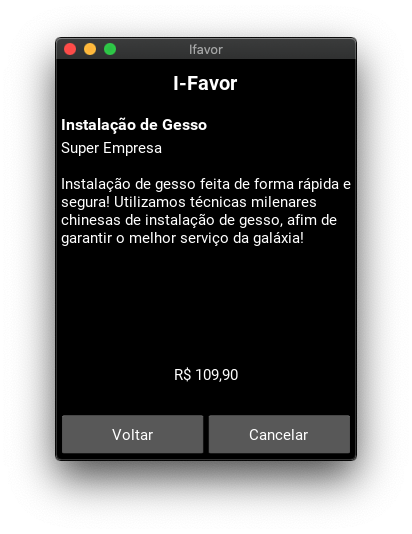
Tela 3 mostrando as informações do serviço selecionado na Tela 2 (clicando no botão “i”).

* O botão “Voltar” volta para a Tela 2 sem fazer nenhuma ação com o serviço. Executa o método voltar() da classe VisualizarCliente.
* O botão “Solicitar” solicitar o serviço. Executa o método solicitar() da classe VisualizarCliente.
* Ao entrar na Tela 3 o método preencher\_detalhes() da classe VisualizarCliente é executado, nesse caso a variável id\_servico será igual ao id do serviço selecionado na tela anterior.



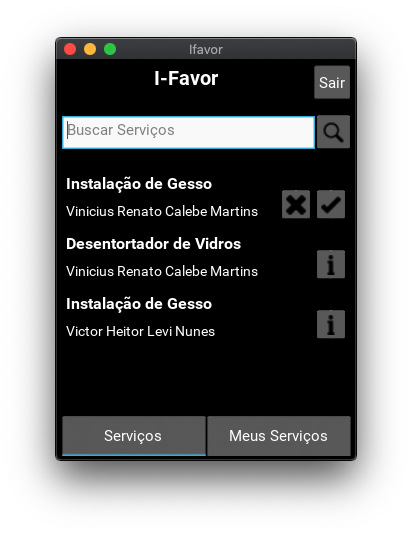
Tela 4 mostrando os serviços solicitados pelo Cliente.

* Ao clicar em “Serviços” o usuário é alterado para a Tela 2, isso ocorre sem chamar nenhum método já que está sendo utilizado o widget TabbedPanel.
* O botão com a lupa faz uma busca de serviços que contenham o texto do TextField ao seu lado. Ele ativa o método popular\_listagem() da classe PrincipalCliente.
* O botão com o “i” abre a Tela 5 com as informações do pedido selecionado. Ativa o método exibir\_detalhes() da classe Linha.
* O botão “Sair” desconecta da conta e volta para a Tela 1. Executa o método sair() da classe PrincipalCliente.
* Ao entrar na Tela 4 o método popular\_listagem() da classe PrincipalCliente é executado.



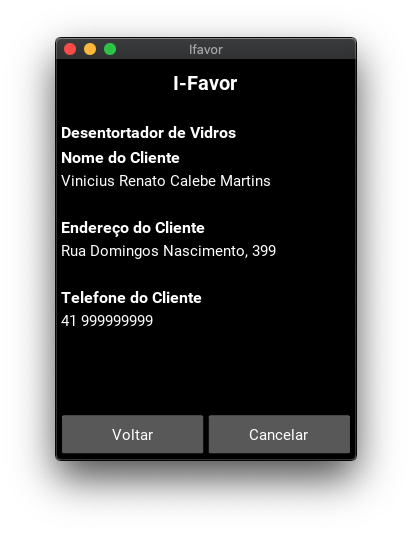
Tela 5 mostrando as informações do serviço selecionado na Tela 4 (clicando no botão “i”).

* O botão “Voltar” volta para a Tela 2 sem fazer nenhuma ação com o pedido. Executa o método voltar() da classe VisualizarCliente.
* O botão “Cancelar” cancela o pedido. Executa o método solicitar() da classe VisualizarCliente.
* Ao entrar na Tela 5 o método preencher\_detalhes() da classe VisualizarCliente é executado, nesse caso a variável id\_pedido será igual ao id do pedido selecionado na tela anterior.



Tela 6 mostrando os serviços solicitados ao Prestador de Serviços.

* Ao clicar em “Meus Serviços” o usuário é alterado para a Tela 8, isso ocorre sem chamar nenhum método já que está sendo utilizado o widget TabbedPanel.
* O botão com a lupa faz uma busca de serviços que contenham o texto do TextField ao seu lado. Ele ativa o método popular\_listagem() da classe PrincipalPrestador.
* O botão com o “i” abre a Tela 7 com as informações do cliente do pedido selecionado. Ativa o método exibir\_detalhes\_cliente() da classe LinhaPrestador.
* O botão com o ✖ (xis) rejeita o pedido selecionado. Ativa o método rejeitar\_pedido() da classe LinhaPrestador.
* O botão com o **✓** (check) aceita o pedido selecionado. Ativa o método aceitar\_pedido() da classe LinhaPrestador.
* O botão “Sair” desconecta da conta e volta para a Tela 1. Executa o método sair() da classe PrincipalPrestador.
* Ao entrar na Tela 6 o método popular\_listagem() da classe PrincipalPrestador é executado.



Tela 7 mostrando as informações do cliente do pedido selecionado na Tela 4 (clicando no botão “i”).

* O botão “Voltar” volta para a Tela 6 sem fazer nenhuma ação com o pedido. Executa o método voltar() da classe PrestadorInfoCliente.
* O botão “Cancelar” cancela o pedido. Executa o método solicitar() da classe PrestadorInfoCliente.
* Ao entrar na Tela 7 o método preencher\_detalhes() da classe PrestadorInfoCliente é executado, nesse caso a variável id\_pedido será igual ao id do pedido selecionado na tela anterior.



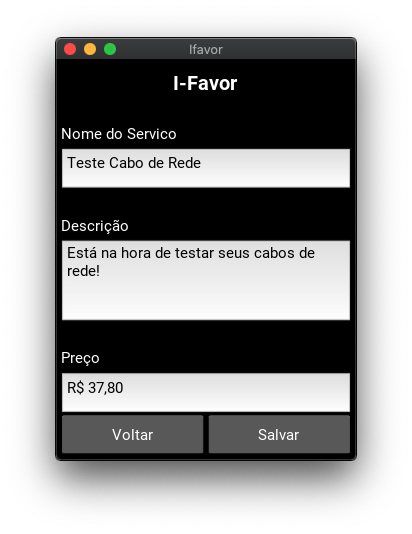
Tela 8 mostrando os serviços cadastrados pelo Prestador de Serviços.

* Ao clicar em “Serviços” o usuário é alterado para a Tela 6, isso ocorre sem chamar nenhum método já que está sendo utilizado o widget TabbedPanel.
* O botão “Adicionar Novo” abre a Tela 9 para que o usuário cadastre um novo serviço. Ele ativa o método novo\_servico() da classe PrincipalPrestador.
* O botão com o lápis abre a Tela 9 com as informações do serviço selecionado para que o usuário possa editá-lo. Ativa o método editar\_servico() da classe LinhaPrestador.
* O botão “Sair” desconecta da conta e volta para a Tela 1. Executa o método sair() da classe PrincipalPrestador.
* Ao entrar na Tela 8 o método popular\_listagem() da classe PrincipalPrestador é executado.



Tela 9 mostrando o serviço a ser editado.

* O botão “Voltar” volta para a Tela 6 sem fazer nenhuma ação com o serviço. Executa o método voltar() da classe VisualizarPrestador.
* O botão “Editar” libera os campos para edição e altera seu próprio texto para “Salvar”. Executa o método editar() da classe VisualizarPrestador.
* Ao entrar na Tela 9 o método preencher\_detalhes() da classe VisualizarPrestador é executado, nesse caso a variável id\_pedido será igual ao id do pedido selecionado na tela anterior.



Tela 9 mostrando o serviço liberado para ser editado.

* O botão “Voltar” volta para a Tela 6 sem fazer nenhuma ação com o serviço. Executa o método voltar() da classe VisualizarPrestador.
* O botão “Salvar” enviar as alterações para o banco e volta para a Tela 6. Executa o método editar() da classe VisualizarPrestador.
* Ao entrar na Tela 9 o método preencher\_detalhes() da classe VisualizarPrestador é executado, nesse caso a variável id\_pedido será igual ao id do pedido selecionado na tela anterior.



Tela 9 mostrando o serviço liberado para ser editado.

* O botão “Voltar” volta para a Tela 6 sem fazer nenhuma ação com o serviço. Executa o método voltar() da classe VisualizarPrestador.
* O botão “Salvar” enviar o novo serviço para o banco e volta para a Tela 6. Executa o método editar() da classe VisualizarPrestador.
* Ao entrar na Tela 9 o método preencher\_detalhes() da classe VisualizarPrestador é executado, nesse caso a variável id\_pedido será igual ao id do pedido selecionado na tela anterior.

|  |
| --- |
| Referências |

[1] L’Hotellier, E. (2020). *Plataforma de contratação de serviços*. Acesso em Junho de 2020, disponível em **Getninjas** : https://www.getninjas.com.br/quem-somos

[2] Triider. (2019). *Plataforma de serviços*. Acesso em Junho de 2020, disponível em **Triider**: https://www.triider.com.br/

[3] Techtudo. (Junho de 2020). *Aplicativo de diarista e serviços gerais: veja lista com melhores apps*. Fonte: **Techtudo**: https://www.techtudo.com.br/listas/2019/07/aplicativo-de-diarista-e-servicos-gerais-veja-lista-com-melhores-apps.ghtml

|  |
| --- |
| Mapas de Avaliação |

Para maiores detalhes quanto às fichas de avaliação, vide o Plano de Ensino da disciplina disponível no AVA Blackboard®.

**Composição da Avaliação da Especificação do Projeto**

Itens do Projeto com os Respectivos Ids de Avaliação

|  |  |  |
| --- | --- | --- |
| ID | Itens Contemplados | Nota Obtida |
| **ID 1** | 2 – Requisitos Funcionais e Não Funcionais, 3 – Diagrama de casos de uso e 4 – Especificação dos casos de uso |  |
| **ID 2** | 5 – Arquitetura do Software – Diagrama de Classes |  |
| **ID 3** | 6 – Arquitetura do Software – Diagrama de Entidade-Relacionamento |  |
| **ID 7** | 1 – Visão Geral do Aplicativo, 7 – Interação Humano Computador (IHC) e 9 – Referências |  |

**Composição da Avaliação da Defesa do Projeto**

Itens do Projeto com os Respectivos Ids de Avaliação

|  |  |  |
| --- | --- | --- |
| ID | Itens Contemplados | Nota Obtida |
| **ID 1** | 2 – Requisitos Funcionais e Não Funcionais, 3 – Diagrama de casos de uso e 4 – Especificação dos casos de uso |  |
| **ID 2** | 5 – Arquitetura do Software – Diagrama de Classes |  |
| **ID 3** | 6 – Arquitetura do Software – Diagrama de Entidade-Relacionamento |  |
| **ID 4** | 8 *– Prototipagem em Python/Kivy e análise do código-fonte construído juntamente ao banco de dados e sua persistência. IDs lançados pelos professores da Implementação* | |
| **ID 5** |
| **ID 6** |
| **ID 7** | 1 - Visão Geral do Aplicativo, 7 - Interação Humano Computador (IHC) e 9 - Referências |  |

1. Se forem utilizados tais tipos de componentes. [↑](#footnote-ref-1)