



Relatório de Projeto Análise de Sistemas

(2020/2021)

Grupo Nº	Curso / Nome Projeto				
8	Engenharia Informática / QuickHandler				
Composição do Grupo					
Número / Nome	Esforço (Horas)				
	Pesqui. Web	Reuniões	Elabor. Diag.	Elabor. Relató.	Total
Pedro Oliveira nº 50039057	4 horas	8 horas	12 horas	18 horas	42 horas
Douglas Bewiahn nº 50038630	6 horas	8 horas	9 horas	14 horas	37 horas

Versões do Relatório

Versão	Data	Autor	Descrição
0.5	25/01/2021	Pedro Oliveira	Adicionado as imagens dos diagramas do projeto e uma curta descrição destes e foi feito o sumário executivo.
0.6	27/01/2021	Pedro Oliveira	Corrigido diagrama de blocos, descrição de casos de uso, e
0.7	27/01/2021	Douglas Bewiahn	Desenvolvimento da Introdução
0.8	29/01/2021	Douglas Bewiahn	Manual de Utilização da Aplicação QuickHandler
1.0	31/01/2021	Pedro e Douglas	Bibliografia, Conclusão, Trabalhos futuros, BPMN e Revisão geral

Índice

SUMÁRIO EXECUTIVO	3
ACRÓNIMOS	3
INTRODUÇÃO	4
DIAGRAMA DE CONTEXTO DO SISTEMA QUICKHANDLER	5
DIAGRAMA DE CASOS DE UTILIZAÇÃO DO SISTEMA QUICKHANDLER	6
CASOS DE USO – DESCRIÇÃO GERAL	6
DESCRIPÇÃO DETALHADA DOS CASOS DE UTILIZAÇÃO	7
<i>UC01 – Criar Pedido</i>	7
UC02 – Aceitar Pedido	8
UC03 – Aceitar Pedido	8
MODELO DE DOMÍNIO DO SISTEMA QUICKHANDLER	9
DIAGRAMA MÁQUINA DE ESTADOS QUICKHANDLER	10
Apresentação da Arquitetura do Sistema	11
ESPECIFICAÇÃO DE PROCESSOS NEGÓCIO	12
<i>Processo 1:</i>	13
CONCLUSÕES E TRABALHO FUTURO	13
Referências	14
Biografia dos Autores	14
Anexo A: Manual de Utilização da Aplicação QuickHandler	15
ANEXO B: FRAMEWORKS UTILIZADAS NO DESENVOLVIMENTO DA SOLUÇÃO	16
ANEXO C: LEVANTAMENTO DE REQUISITOS DO SISTEMA QUICKHANDLER	18
<i>Requisitos Funcionais</i>	18
<i>Requisitos Não Funcionais</i>	18
ANEXO D: ESPECIFICAÇÃO DAS INTERFACES API	19
Leaflet – Controlo do Mapa	19
<i>B.2. ESRI-LEAFLET-GEOCODER</i>	19
ANEXO E: FUNCIONAMENTO DA UNIDADE CURRICULAR	20

Sumário Executivo

QuickHandler é uma plataforma que fornece, a pessoas que precisam de um serviço feito, simplicidade e rapidez e segurança no contacto entre o cliente e o profissional. A aplicação torna a ação de procurar por um profissional (carpinteiro, eletricista, pintor, etc.) mais rápida e simples, apresentando o serviço que o cliente precisa a todos os profissionais dessa categoria que se encontram registados no nosso site.

Vamos fornecer aos solicitadores, que procuram serviços de instalação, manutenção e de renovação, a oportunidade de se conectarem com profissionais experientes na área que procuram por uma taxa pré-definida pela QuickHandler. O Cliente submete o seu pedido de serviço, tendo unicamente de colocar a categoria e a data e a descrição, que inclui a informação necessária para o profissional. Ao submeter, vai ser disponibilizado na plataforma para todos os profissionais habilitados que irão aceitar o pedido e ao aceitarem, serão conectados automaticamente com o cliente.

Aos profissionais, fornecemos uma plataforma que os conecta aos interessados no seu serviço, aumentando assim a sua base de possíveis clientes. Deixamos os profissionais serem os seus próprios chefes, aceitando as solicitações disponíveis na plataforma definindo os seus próprios horários, sem se preocupar com burocracias e com mais tempo para se focar na qualidade do seu serviço.

Palavras Chave: Serviços; Profissional; Manutenção; Mão-de-Obra; Localização;

Acrónimos

API	Application Programming Interface
CRUD	Create, Replace, Update or Delete
GUI	Graphical User Interface
GPS	Global Positioning System
Sol	System of Interest
UC	Use Case

1 Introdução

Na arquitetura de software de um sistema, a definição dos componentes de software e suas propriedades externas, como também o relacionamento com outros sistemas e sua documentação são de extrema importância para facilitar a comunicação entre os stakeholders e registrar as decisões iniciais sobre um projeto. A principal função da arquitetura é identificar os principais componentes estruturais de um sistema e desenvolver um modelo que demonstre o relacionamento entre eles.

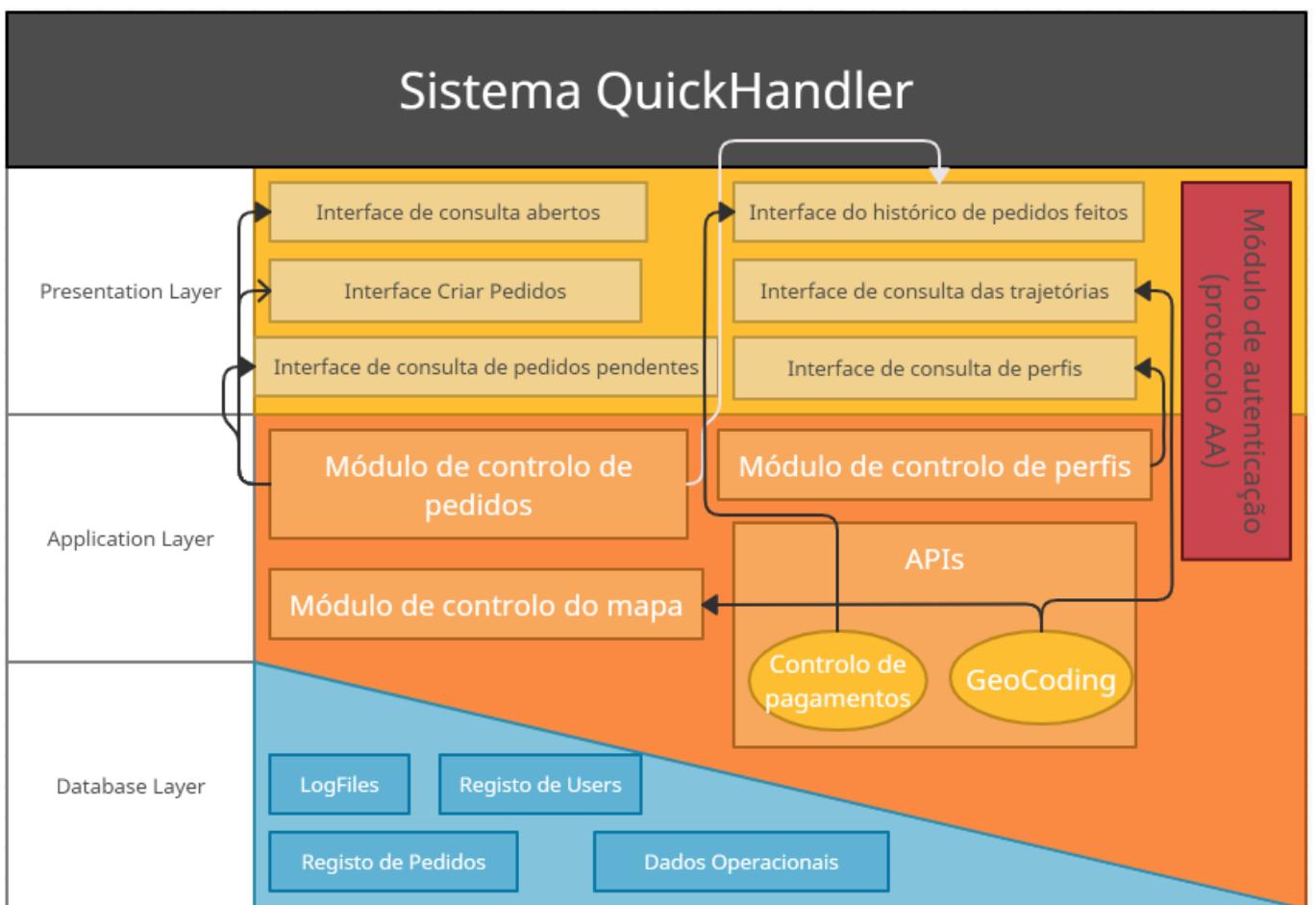
Utilizando a abordagem black-box do diagrama de blocos da arquitetura do sistema foi possível identificar e visualizar em alto-nível os principais componentes do nosso sistema.

QuickHandler é uma plataforma que tem como objetivo automatizar a busca de profissionais que prestam serviços, de forma que uma pessoa que necessite de uma tarefa feita, crie uma solicitação de serviço que fica disponível a todos os profissionais cadastrados. Os profissionais por sua vez analisam as solicitações e aceitam o serviço, então a plataforma conecta as duas partes disponibilizando todas as informações necessárias para que ambos se comuniquem e a tarefa seja executada.

No diagrama de blocos desenvolvido para o sistema QuickHandler, observamos três camadas que são definidas por : camada de apresentação, camada de aplicação e camada de banco de dados.

A camada de apresentação agrupa os elementos de interface, que são disponibilizados para os utilizadores e são usados para interagir com o sistema em questão. Nesta camada os agentes criam, consultam e aceitam pedidos e as interfaces se conectam ao módulo de controle de pedidos da camada de aplicação. A interface de consulta de trajetórias conecta-se com o módulo de controle de mapa, que se encontra na camada de aplicação e através da API de GeoCoding permite a identificação das moradas nos mapas e o cálculo de rotas para as localizações das solicitações de serviço.

Na camada de banco de dados se encontram todas as informações relacionadas aos utilizadores, dados operacionais e pedidos registrados.



2 Diagrama de Contexto do Sistema QuickHandler

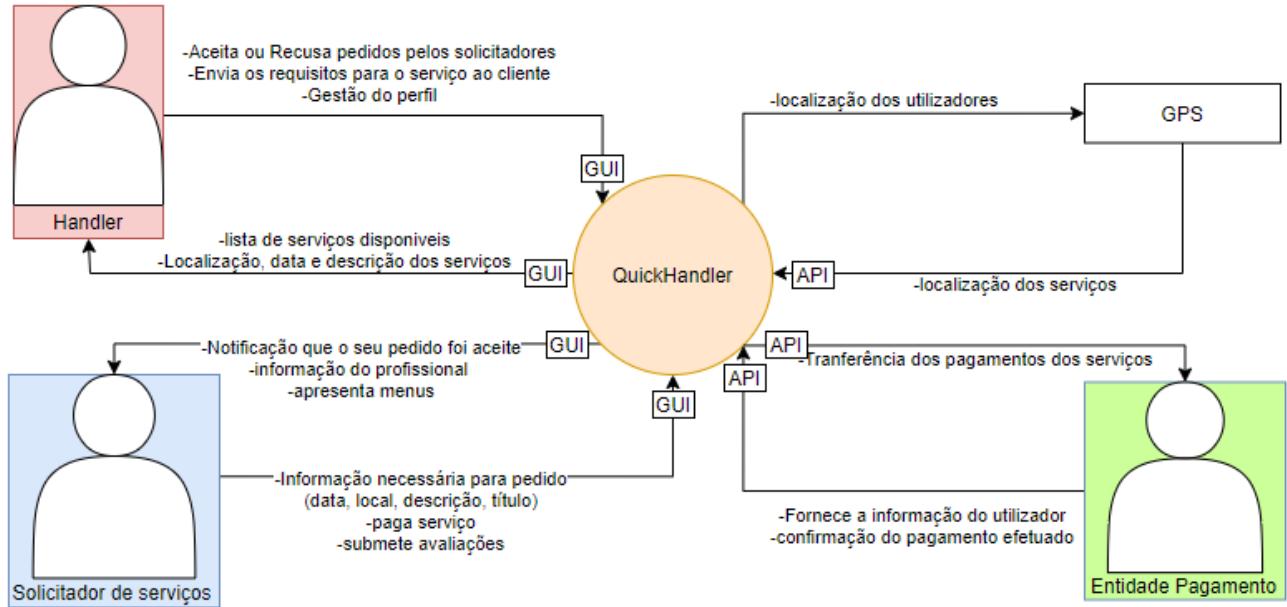
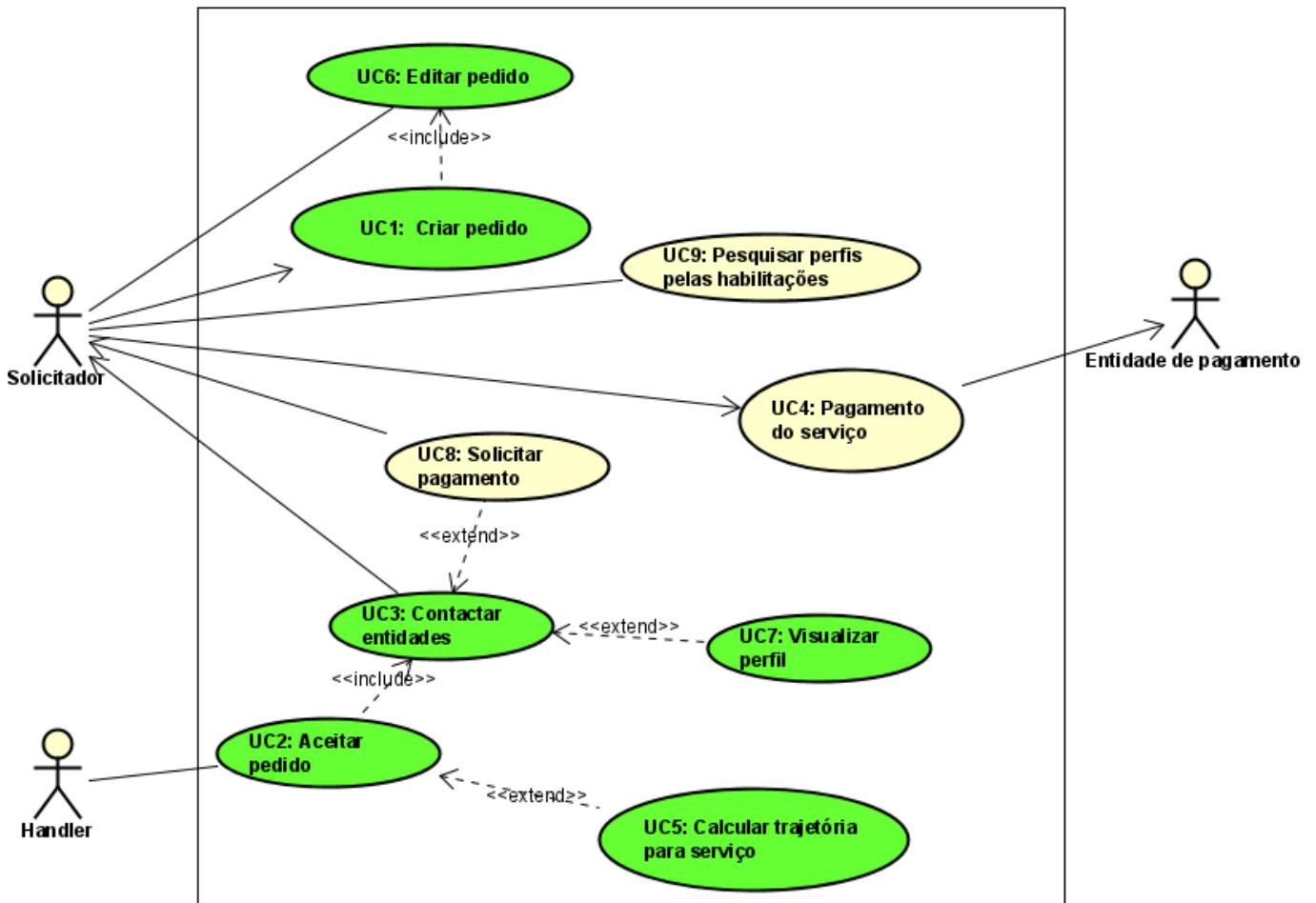


Figura 1. Diagrama de Contexto do sistema QuickHandler

Nome	Função
Solicitador de Serviços (Acordeão)	Adiciona novos pedidos, visualiza o seu histórico de pedidos, visualiza perfis, avalia o Handler e faz o pagamentos.
Handler (Acordeão)	Visualiza e aceita pedidos abertos, define a sua especialidade, recebe pagamentos do serviço e avalia Solicitador de serviços.
Entidade de pagamento (Acordeão)	Entidade responsável pela transferência de dinheiro da conta do solicitador para a conta bancária do Handler.
GPS (Enabling system)	Fornece a localização dos utilizadores, as trajetórias até as moradas do serviço e apresenta no mapa os pedidos.

3 Diagrama de Casos de Utilização do Sistema QuickHandler

Este diagrama contém as “functional requirements” que achamos mais importantes e que iríamos implementar no nosso projeto. Um diagrama de casos de usos serve para definir a interação que os atores irão ter com cada funcionalidade, de forma a ajudar nos a definir futuros diagramas.



3.1 Casos de Uso – Descrição Geral

UC01 – Criar Pedido

Descrição	O solicitante coloca uma requisição de um Handler com a descrição do serviço, a data e a morada e a respetiva categoria.
------------------	--

UC02 – Aceitar Pedido

Descrição	O Handler observa as solicitações de serviço e aceita os serviços que deseja fazer.
------------------	---

UC03 – Contactar entidades

Descrição	O Handler e o solicitante são conectados e poderão trocar mais informação entre si
------------------	--

UC04 – Pagamento do Serviço

Descrição	O Solicitador envia os dados do pagamento para a entidade de pagamento para que esta os possa confirmar que foi feito.
------------------	--

UC05 – Calcular trajetória para serviço

Descrição	<i>Handler pode pesquisar pela trajetória até à morada do serviço</i>
------------------	---

UC06 – Editar pedido

Descrição	<i>O solicitador pode alterar a informação do pedido.</i>
------------------	---

UC07 – Visualizar perfil

Descrição	<i>Solicitador e Handler podem visualizar o perfil de um do outro</i>
------------------	---

UC08 – Solicitar pagamento

Descrição	<i>Handler decide o custo do serviço e envia-o para o Solicitador.</i>
------------------	--

UC09 – Pesquisar perfis pelas habilitações

Descrição	<i>O Solicitador pode pesquisar por Handlers específicos pela sua especialidade caso esteja à procura de um serviço específico</i>
------------------	--

3.2 Descrição Detalhada dos Casos de Utilização**UC01 – Criar Pedido**

Descrição	O solicitante coloca uma requisição de um Handler com a descrição do serviço, a data, a morada e a respetiva categoria de serviço. O pedido é guardado na base de dados para ser apresentado aos Handlers.
Pré-Condições	O utilizador tem de estar conectado como solicitante.
Cenário Principal	<ol style="list-style-type: none"> O solicitador clica para adicionar um novo pedido. Solicitador coloca a categoria do serviço. Solicitador preenche a morada. O solicitador preenche a data e hora do serviço. Solicitador adiciona uma descrição do serviço necessário. Sistema verifica se os dados adicionados são bem reconhecidos. (data, morada, categoria) Sistema adiciona pedido à base de dados. Solicitador carrega no botão “Solicitar”. O sistema informa o solicitante que só pode cancelar o serviço até 4 horas antes do horário marcado (sem taxa).
Cenário Alternativo	3.1. Cliente coloca a morada clicando no mapa apresentado
Pós-Condições	O pedido é guardado na base de dados e apresentado aos Handlers que possuam a categoria necessária do pedido.
Cenário de Exceção	<p>6.1. Se a morada do serviço não se encontra na zona coberta pela aplicação.</p> <p>6.1.1. Sistema informa o solicitador que não conseguimos fornecer serviços visto que não temos Handlers suficientes na zona pedida.(insucesso)</p> <p>6.1.2. Sistema informa que o pagamento desse serviço será mais elevado</p> <p>6.2. Se a data do serviço for antes do dia em que se encontra o sistema irá informar o utilizador do problema.</p> <p>8.1. Solicitador cancela antes de faltarem 4 horas ou depois caso não tenha sido aceito por um Handler.</p> <p>8.1.1. O pedido é cancelado</p> <p>8.2. Solicitador cancela depois de faltarem 4 horas.</p> <p>8.2.1. O utilizador tem de pagar uma taxa para o profissional.</p>
Pós-Condições	<p>8.1.2.1. Cliente aceita aumento do pagamento, pedido será guardado com categoria especial e será mostrado aos profissionais mais próximos dessa zona, ou que consigam transportar até à morada.</p> <p>8.1.2.2. Cliente recusa pagar. (insucesso)</p>

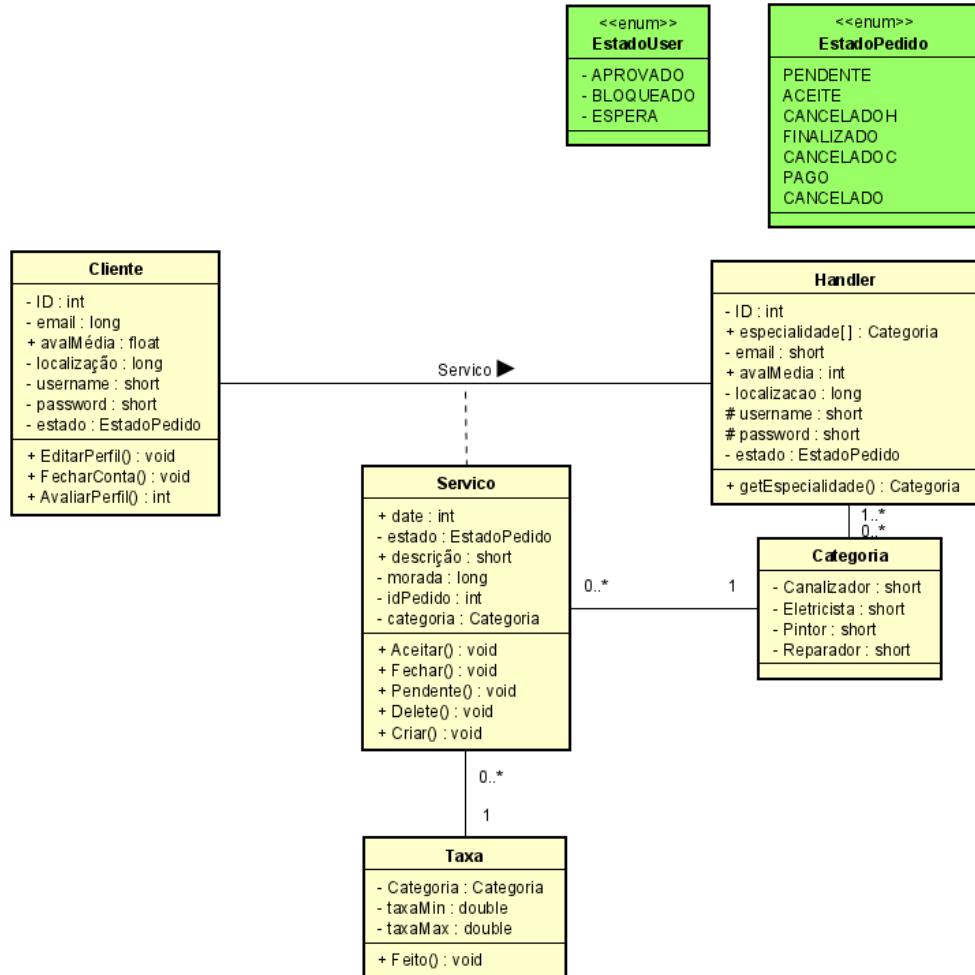
UC02 – Aceitar Pedido

Descrição	O Handler observa as solicitações de serviços que não foram aceites e que tem a especialidade e a capacidade de cumprir e aceitar os serviços que deseja fazer e muda o estado do pedido para aceite ao aceitar esse pedido.
Pré-Condições	Estar conectado como Handler. As capacidades do Handler devem condizer com a categoria colocada, de forma a garantir que o Handler tem as habilidades necessárias para cada pedido.
Cenário Principal	<ol style="list-style-type: none"> 1. Handler observa a lista de serviços à espera de serem aprovados. 2. Handler seleciona um pedido que deseja aceitar. 3. Handler é reencaminhado para uma página que apresenta esse pedido com mais detalhe. 4. Handler aprova o pedido. 5. O Sistema informa que o Handler só pode cancelar o pedido antes de faltarem 2 horas para o serviço ser feito.
Cenário Alternativo	3. Handler pode visualizar as rotas até o local do serviço
Pós-Condições	O pedido muda de estado para “Aceite”.
Cenário de Exceção	<ol style="list-style-type: none"> 5.1. Handler cancela serviço antes de faltarem 2 horas para o serviço ser feito. 5.2. Handler cancela depois de faltarem 2 horas para o serviço ser feito.
Pós-Condições	<ol style="list-style-type: none"> 5.1.1. Pedido volta a ser colocado na tabela de pedidos pendentes. 5.2.1. Handler terá de pagar uma taxa por ter cancelado o serviço demasiado tarde, e fica responsável em garantir que esse serviço é feito.

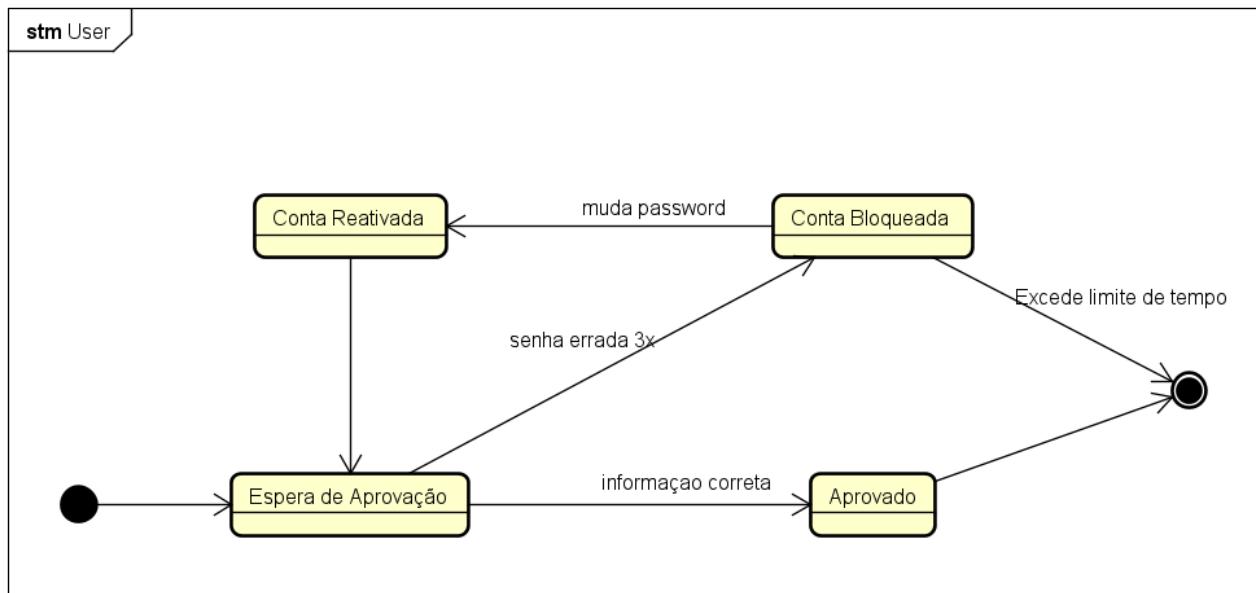
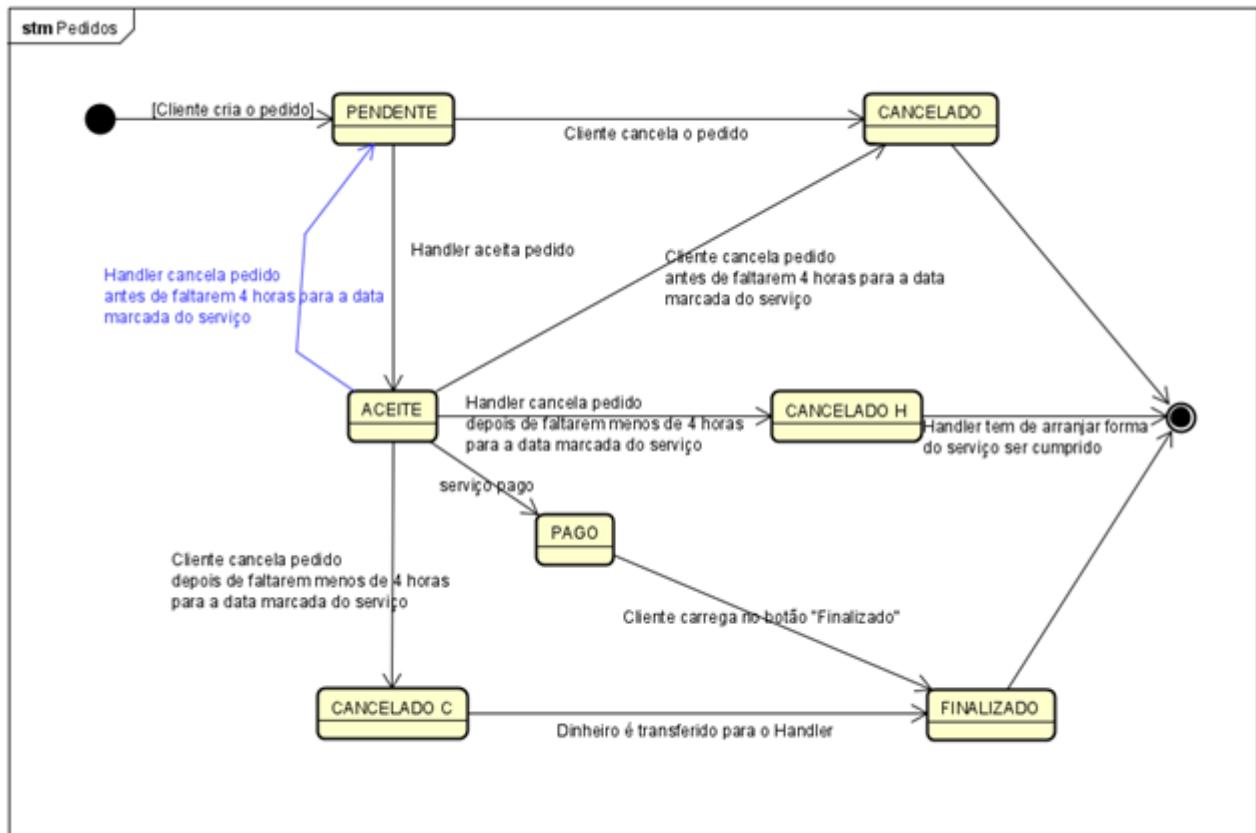
UC03 – Contactar Entidades

Descrição	O Handler e o solicitante recebem a informação um do outro e poderão trocar mais informações entre si.
Pré-Condições	Estar logado como solicitante/Handler. O seu pedido ter sido aceite por um Handler/ter aceite um pedido.
Cenário Principal	<ol style="list-style-type: none"> 1. Handler recebe a informação do Cliente. 2. Cliente recebe a informação do Handler que aceitou o seu pedido. 3. Cliente faz a transferência necessária para o pagamento do serviço. 4. Handler coloca uma fotografia do serviço prestado, como confirmação que o pedido foi feito. 5. Handler carrega no botão "Concluído" de forma a informar que o serviço foi concluído 6. Transferência do pagamento para o Handler. 7. O solicitador e o Handler avaliam a sua experiência (review da outra entidade).
Cenário Alternativo	5. Handler recebe uma notificação mal acaba o dia do serviço para carregar no botão "Concluir", caso ainda não tenha carregado.
Pós-Condições	O pedido irá mudar o seu estado para “Feito”. As avaliações vão ser adicionadas ao respectivo perfil da entidade avaliadas.
Cenário de Exceção	<ol style="list-style-type: none"> 3.1. Cliente não paga serviço. <ol style="list-style-type: none"> 3.1.1 Sistema informa que se o cliente não fizer a transferência o serviço irá ser cancelado. 5.1. Handler não carrega no botão 7.1. O solicitante não se encontra satisfeito com o serviço.
Pós-Condições	<ol style="list-style-type: none"> 3.1.2. serviço é cancelado caso ainda não tenha sido pago até quatro horas antes da data e hora marcada pelo cliente. 5.1.1. Handler não recebe o seu pagamento 5.1.2. Perguntado ao Cliente se o serviço foi feito. 7.1.1. O sistema procura outro profissional que compra o serviço que foi mal realizado e o Handler que não cumpriu o serviço fica responsável pelos custos.

4 Modelo de Domínio do Sistema QuickHandler



5 Diagrama Máquina de Estados QuickHandler



6 Apresentação da Arquitetura do Sistema

Para identificar o fluxo de informação entre os componentes do nosso sistema utilizamos a abordagem Black-box do diagrama de blocos. Identificamos que as funções do Solicitador de Serviços é criar pedidos para estes depois serem visualizados e aceites pelos Handlers que criam os seus perfis e alteram-nos para conseguirem arranjar mais pedidos de serviço. Os Solicitadores comunicam com a entidade pagamento, fornecendo os dados do pagamento feita para esta os puder conferir.

Funcionalidades	Entidades					
	Solicitador	Handler	Serviço	Categoria	Taxa	Entidade de Pagamento
Criar pedidos	CRUD		C	U	RU	
Troca de Contactos	R	R	RUD			
Pagamento	R	RU	CRUD	RU	CRUD	RU
localização serviço	CRUD	CRUD	CRUD			
Calcular Trajetória	R	R	RU			RU
Criar User	CRUD	CRUD	RU	CRU		
Avaliar Perfis	CRU	CRU	CRU			
Gerir Perfil	CRUD	CRUD	RU	CRUD		

Definimos que os módulos necessários para este sistema ter o comportamento desejado são o módulo de perfil e o módulo de pedidos. Para facilitar a leitura da informação fornecida pelos outros módulos decidimos também implementar o módulo de controle do mapa de forma a conseguir implementar o cálculo da trajetória também para diminuir o trabalho dos nossos clientes.

O Solicitador e o Handler comunicam com todos os módulos apresentados (módulo de controle de pedidos; módulo de controle de perfis e módulo de controle do mapa). O Solicitador consegue criar, alterar, visualizar e apagar pedidos, o Handler consegue somente visualizar e modificar pedidos. O solicitador cria, visualiza, altera e apaga perfis e o Handler também. Por fim, ambos o Solicitador e o Handler conseguem ler, criar, alterar e apagar as localizações no mapa.

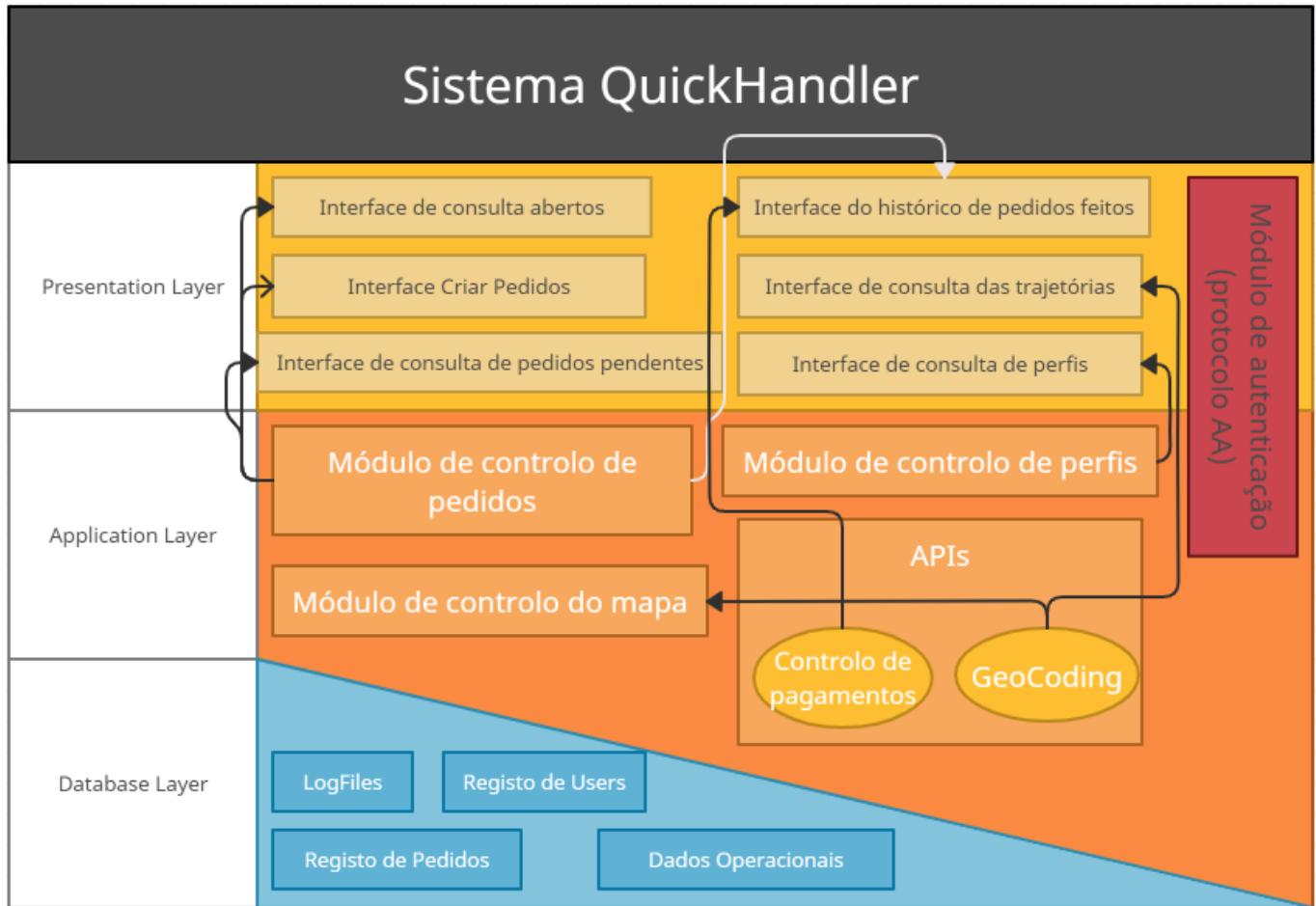


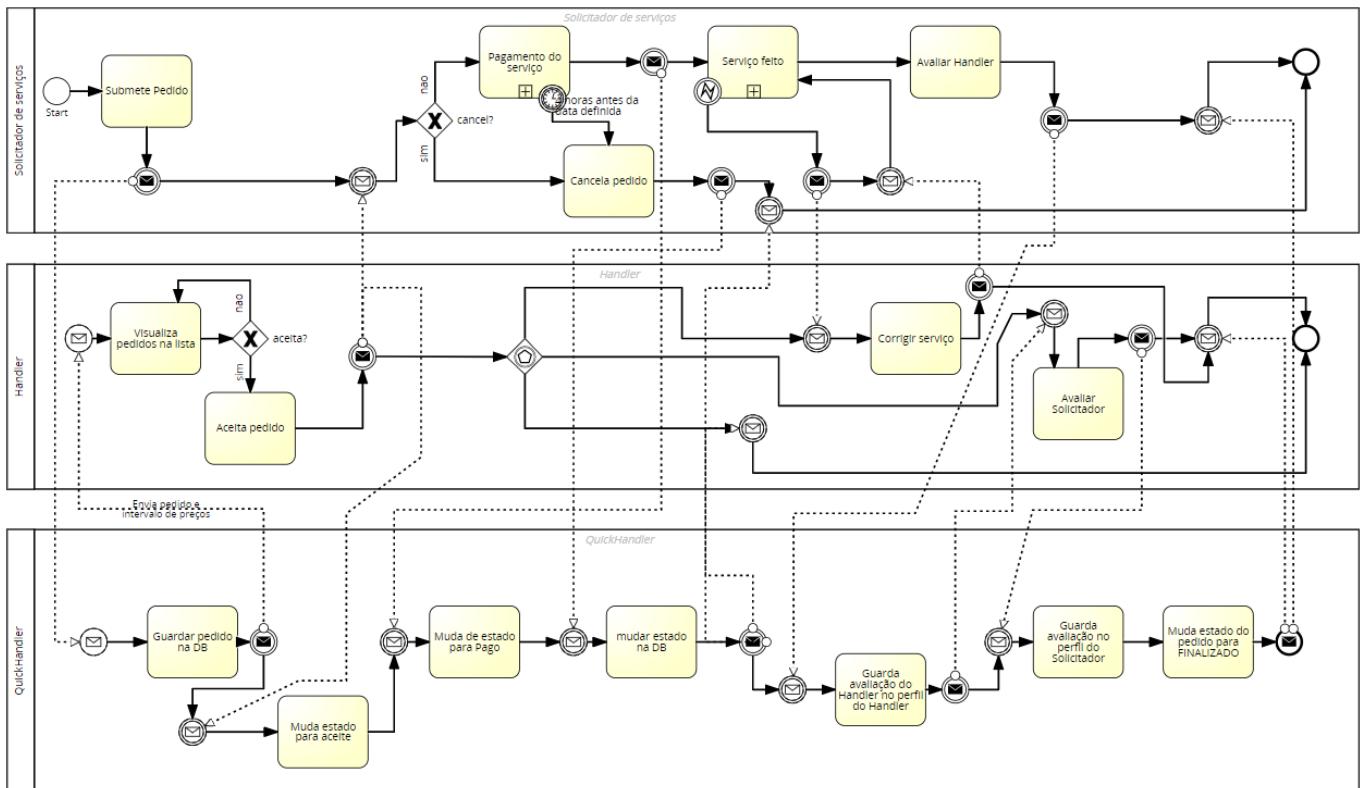
Figura 3. Diagrama de blocos da proposta QuickHandler.

7 Especificação de Processos Negócio

Ao preencher um pedido, o Solicitador submete-o para a Base de Dados para este ser visto pelo Handlers da nossa aplicação. Um Handler ao ver esse pedido pode aceitar ou ignorar este pedido para ver outros. Ao aceitar, define o custo do serviço e envia-o para o solicitador que pode cancelar o serviço ou fazer o pagamento até quatro horas antes da data definida para o serviço. Ao fazer o pagamento, o Solicitador só tem de esperar que o Handler faça o serviço. Caso haja algum problema, com o cumprimento do serviço, o Handler fica responsável por cumprir o serviço de qualquer forma. Ao terminar, o solicitador avalia o Handler que depois irá avaliar o solicitador e essas avaliações vão ser submetidas para a base de dados.

Processo 1:

O solicitador necessita de ter preenchido a informação para criar um novo pedido. Não implementamos a definição do custo do pedido ao aceitar o pedido nos casos



8 Conclusões e Trabalho Futuro

O nosso sistema é uma plataforma para serviços de manutenção que pode facilmente ser alterado para condizer com outros serviços variados. Assumimos que o pagamento de cada serviço não iria necessitar de um orçamento para facilitar o nosso pensamento ao desenvolver a aplicação, mas permitimos que o preço seja mudado conforme as necessidades de cada Handler.

Alguns constrangimentos que chegamos foram arranjar uma forma fiável de definir que um serviço foi concluído, de forma, que decidimos colocar que iria ser submetida uma fotografia pelo Handler e carregado um botão por parte do Solicitador para haver provas dos dois lados que o serviço foi feito.

#	Nome do módulo	Fundamentação
1	Módulo dos pedidos	É o módulo mais importante pois contém a ideia base do nosso sistema que contém a criação e modificação de pedidos para que os serviços possam ser feitos
2	Módulo dos perfis	Segundo módulo mais importante pois define os cargos dos utilizadores no nosso sistema tornando mais simples a interação entre estes.
3	Módulo de controlo do mapa	Módulo para o controlo do mapa para servir de apoio visual à informação dos outros dois módulos

Trabalho Futuro:

- Desenvolvimento de uma aplicação mobile para maior engajamento do público alvo.
- Desenvolvimento de um módulo de vendas de materiais que são utilizados para fazer os serviços disponibilizados ou mesmo uma ligação comissionada com fornecedores de materiais para a venda casada de serviços e produtos.
- Desenvolvimento de uma plataforma de ajuda remota, onde os prestadores de serviço podem auxiliar de caráter emergencial aos utilizadores.

Referências

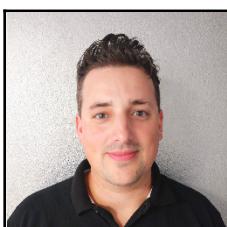
- [1] "Business Process Model and Notation", *En.wikipedia.org*, 2021. [Online]. Disponível em: https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation. [Acedido: 27- Jan- 2021].
- [2] "Black Box Modelo: Diagrama de Blocos", sodiummedia, 2021. [Online]. Disponível em: <https://pt.sodiummedia.com/4295835-black-box-model-block-diagram>. [Acedido: 28- Jan- 2021].
- [3] "O que é um diagrama de máquina de estados?", *Lucidchart*, 2021. [Online]. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-maquina-de-estados-uml>. [Acedido: 22- Dez- 2020].
- [4] *Quatinetwork.files.wordpress.com*, 2021. [Online]. Disponível em: https://quatinetwork.files.wordpress.com/2012/04/modelo_dominio.pdf. [Acedido: 26- Jan- 2021].
- [5] *Homepages.dcc.ufmg.br*, 2021. [Online]. Disponível em: https://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/uml-casos-de-uso_v02-1.pdf. [Acedido: 29- Nov- 2020].
- [6] *Capsi.apsi.pt*, 2021. [Online]. Disponível em: <http://capsi.apsi.pt/index.php/capsi/article/viewFile/102/97>. [Acedido: 31- Jan- 2021].
- [7] A. BPMN, A. BPMN and C. Processo, "O que é BPMN? A notação mais usada para modelar processos", *HEFLO PT*, 2021. [Online]. Disponível em: <https://www.heflo.com/pt-br/bpm/notacao-bpmn/>. [Acedido: 30- Jan- 2021].
- [8] "Eletricista, Chaveiro, Canalizador | 24H | em Lisboa | S.O.S Faz Tudo", *SOS Faz Tudo*, 2021. [Online]. Available: <https://sosfaztudo.pt/>. [Acedido: 05- Nov- 2020].
- [9] "Reparações ao Domicílio Lisboa | Faz tudo", *Faztudo.com.pt*, 2021. [Online]. Disponível em: <https://www.faztudo.com.pt/>. [Acedido: 05- Nov- 2020].
- [10] "Encontre profissionais locais para todos os seus projetos", *StarOfService*, 2021. [Online]. Disponível em: <https://www.starofservice.pt/>. [Acedido: 05- Nov- 2020].

Biografia dos Autores



(50039057) Pedro Caldas de Oliveira

Estudante Universitário do curso de Engenharia Informática na Universidade Europeia. Possui conhecimento das linguagens Java, JavaScript, HTML, CSS, C, Python, MySQL e Pascal. As suas Soft Skills são Criatividade e Hardworking.



(50038630) Douglas William Bewiahn

Profissional experiente em negócios, formado em Business no Brasil, mudou-se para Portugal para estudar Engenharia Informática. Tem conhecimentos em Java, C, MySQL, HTML e JavaScript. Proatividade, organização e raciocínio lógico são características fortes.

Anexo A: Manual de Utilização da Aplicação QuickHandler

Criar Serviço :

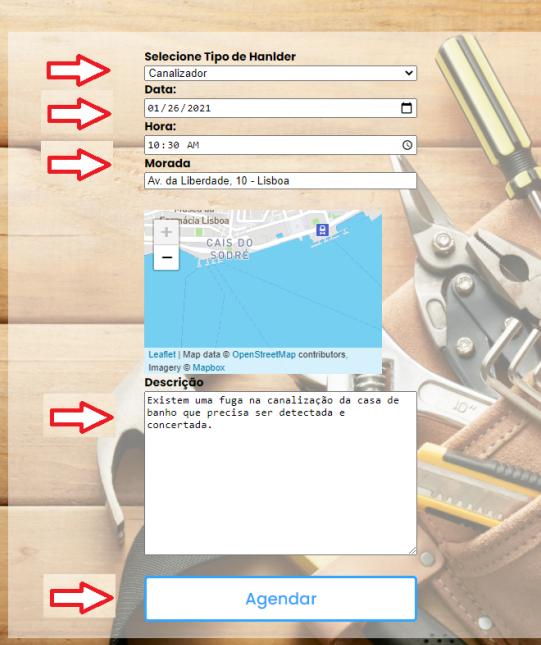
O utilizador clica no botão “login” e autentica-se como solicitante, clica no botão “Solicitar Handler” para adicionar um novo pedido.

O solicitante seleciona a categoria do serviço e preenche a data, hora, morada e uma descrição do serviço necessário e clica em “Agendar”.



Encontre um profissional para lhe ajudar
Dezenas de profissionais à sua disposição a apenas alguns cliques!

Solicitar Handler



Criar pedido de Handler

Selecionar Tipo de Handler
Canalizador

Data:
01 / 26 / 2021

Hora:
10 : 30 AM

Morada:
Av. da Liberdade, 10 - Lisboa

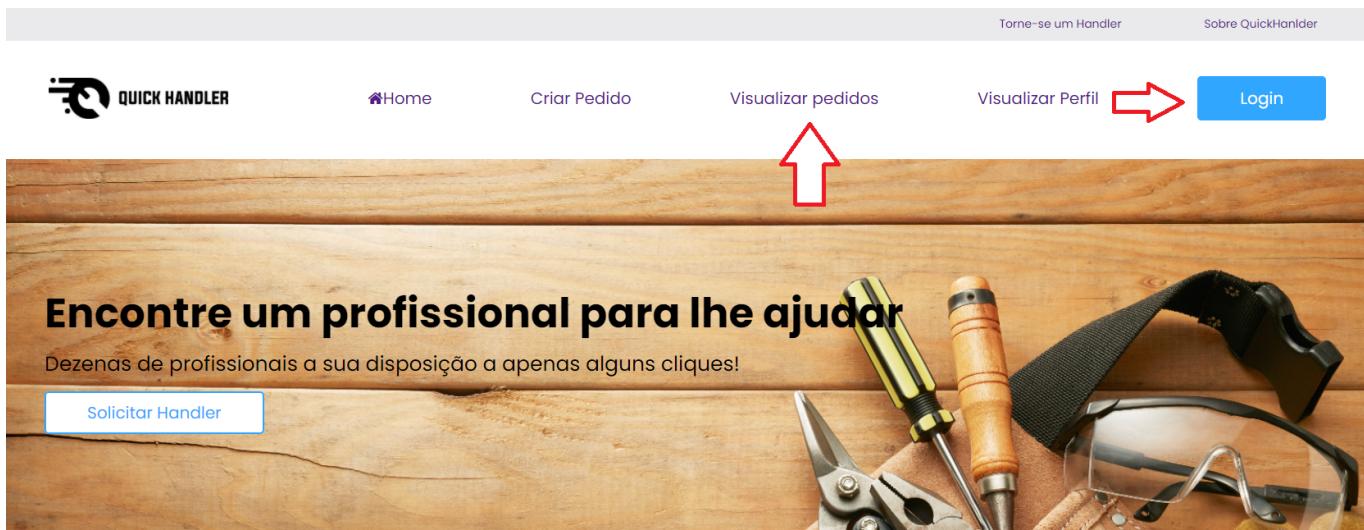
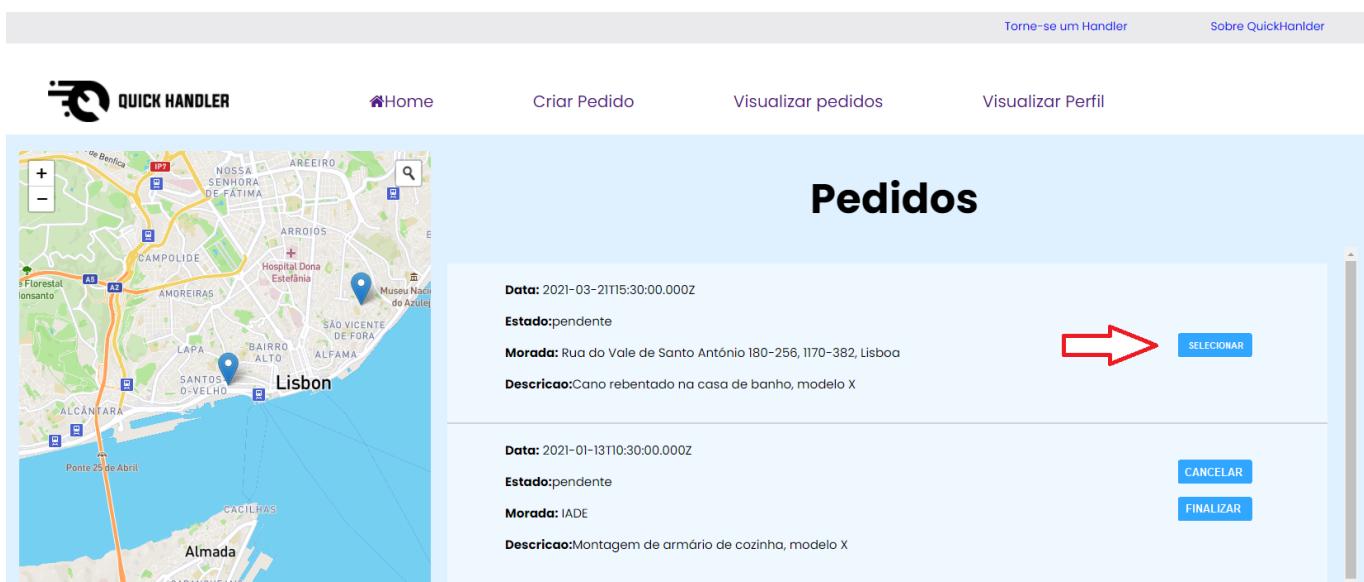
Descrição:
Existem uma fuga na canalização da casa de banho que precisa ser detectada e concertada.

Agendar

Aceitar Serviços :

O utilizador clica no botão “login” e autentica-se como handler, clica no menu “Visualizar pedidos” para ser encaminhado para a página de pedidos disponíveis.

O utilizador observa a lista de serviços disponíveis e clicar em “selecionar” no pedido que deseja aceitar. Na página seguinte tem todos os detalhes do pedido, clica em aceitar e o sistema informa que o pedido só pode ser cancelado em até duas horas antes do horário agendado.

The screenshot shows the "Pedidos" (Orders) page. On the left, there is a map of Lisbon with several locations marked. The main area displays two service requests:

- Data:** 2021-03-21T15:30:00.000Z
Estado: pendente
Morada: Rua do Vale de Santo António 180-256, 1170-382, Lisboa
Descrição: Cano rebentado na casa de banho, modelo X
- Data:** 2021-01-13T10:30:00.000Z
Estado: pendente
Morada: JADE
Descrição: Montagem de armário de cozinha, modelo X

For each request, there are three buttons: "SELECIONAR" (with a red arrow pointing to it), "CANCELAR", and "FINALIZAR".



The screenshot shows a map of Lisbon with various neighborhoods labeled. A repair request is displayed on the right side of the screen:

- Data:** 2021-03-21T15:30:00.000Z
- Cliente:** user
- Estado:** pendente
- Morada:** Rua do Vale de Santo António 180-256, 1170-382, Lisboa
- Descrição:** Cano rebentado na casa de banho, modelo X

A blue button labeled "ACEITAR" is visible, with a red arrow pointing to it from the left.

<https://quickhandler.herokuapp.com/>

Anexo B: Frameworks utilizadas no desenvolvimento da solução

Diagrama de Contexto: Draw.io, é uma aplicação online de construção de diagramas para fazer flow charts, diagramas de processos, UMLs, etc. url: <https://app.diagrams.net/> ;

Diagrama de Blocos: creately.com, é um software visual gratuito para desenhar e colaborar em ideias, conceitos e processos; <https://creately.com/>

Diagrama BPMN: Signavio Process Manager, é uma ferramenta de Business Process Modelling online vendida pela empresa Signavio; <https://www.signavio.com/>

Desenho dos Mockups: Mockflow.com, é um software wireframe que permite aos designers colaborarem em tempo real em protótipos de interface para sites e software.; <https://mockflow.com/>

Diagramas UML: Astah UML, é um editor simples para UMLs feito para aumentar a acessibilidade de fazer estes diagramas; <https://astah.net/products/astah-community/>

Matriz CRUD: Google sheets programa para produzir spreadsheets que faz parte da ferramenta Google Docs Editors gratuita produzida pela Google; <https://docs.google.com/spreadsheets/>

APIs: Leaflet.js, biblioteca de JavaScript utilizada para construir web mapping apps; <https://leafletjs.com/>

esri-leaflet, conjunto de ferramentas que implementa funcionalidades ArcGIS; <https://esri.github.io/esri-leaflet/>,

express.js, framework de aplicação Back end para Node.js utilizada para construir aplicações na internet e APIs; <https://expressjs.com/>;

Presentation Layer: HTML, CSS, Node.js

Coding: Microsoft Visual Studio Code,

Dados: MySQL, remotemysql.com

Anexo C: Levantamento de Requisitos do Sistema QuickHandler

Requisitos Funcionais

#	Nome do Requisito	Descrição	Pri.
FR01	Criar pedidos	O solicitador conseguir criar os pedidos com o devido conteúdo	High
FR02	Visualizar pedidos pendentes	O Handler conseguir visualizar os pedidos	High
FR03	Aceitar pedidos pendentes	O Handler conseguir aceitar os pedidos	High
FR04	Conectar-se como Handler ou solicitador	Poder definir que utilizador é Handler ou solicitador	High
FR05	Localizar o serviço	O utilizador pode definir onde é o serviço	High
FR06	Editar serviço	Poder editar o pedido depois de submetido	Medium
FR07	Finalizar o serviço	O utilizador poder finalizar o pedido após ter sido concluído	High
FR08	Avaliar perfis	Poder avaliar os perfis dos utilizadores apos serviços	Medium
FR09	Gestão do perfil	Poder gerenciar a informação do seu perfil	Low
FR10	Apresentar pedidos através da proximidade	Handler veem os pedidos em relação à proximidade	Medium
FR11	Localizar o utilizador	O sistema saber a localização do utilizador (Handler/cliente)	High
FR12	Calcular rotas para o serviço	O sistema calcular a rota para a morada do serviço	Low
FR13	Visualizar perfis	O utilizador pode visualizar os perfis de outros utilizadores	High
FR14	Colocar as habilitações do Handler	O sistema saber as habilitações de cada Handler	Medium
FR16	Dividir os pedidos por habilitações	Apresentar respetivos pedidos a pessoas capazes de o cumprir	Medium

Requisitos Não Funcionais

#	Nome do Requisito	Descrição	Pri.
NFR01	Calcular rotas para o serviço	Handlers podem calcular a rota para o serviço escolhido	Medium
NFR02	Novas Categorias de Serviço	Solicitador tem acesso a mais tipos de serviços disponíveis	Low
NFR03	Chat entre usuário e Handler	Handler e solicitador poderem comunicar entre si	Low

Anexo D: Especificação das Interfaces API

Leaflet – Controlo do Mapa

			Descrição: Apresentar o mapa de forma a ajudar o Solicitador e o Handler a terem uma melhor ideia da localização do local do serviço.	Versão: 1.7.1
1	Apresentar a morada dos serviços		Descrição: apresentar a localização do serviço de forma ao Handler saber onde tem de se deslocar.	
Parâmetros de Input				
	Atributo	Tipo	Obr.	Descrição do Atributo
	Latitude	double	S	Latitude da localização do serviço
	Longitude	double	S	Longitude da localização do serviço
	Nome do Solicitador	string	N	Nome do usuário que criou esse serviço para mostrar no popup.
Parâmetros de Output				
	Atributo	Tipo	Obr.	Descrição do Atributo
	Map	L.map	S	Mapa para apresentar a localização dos pedidos
2	Adicionar markers através de cliques no mapa		Descrição: permitir a interação com o mapa permitindo a seleção de localizações específicas através de cliques de mapa.	
Parâmetros de Input				
	Atributo	Tipo	Obr.	Descrição do Atributo
Parâmetros de Output				
	Atributo	Tipo	Obr.	Descrição do Atributo

B.2. Esri-leaflet-Geocoder

			Descrição: utiliza a funcionalidade de GeoCoding do ArcGis para o mapa leaflet. Devolve as coordenadas das moradas colocadas.	Versão: 2.3.4
1	Procurar coordenadas de uma morada		Descrição: procura as coordenadas da morada colocada pelo utilizador	
Parâmetros de Input				
	Atributo	Tipo	Obr.	Descrição do Atributo
	morada	String	S	A morada que o cliente colocou para o serviço em questão
Parâmetros de Output				
	Atributo	Tipo	Obr.	Descrição do Atributo
	latlong	latlong	S	Coordenadas da localização do serviço
2	Buscar morada através de coordenadas		Descrição: procura a morada de um ponto do mapa colocado pelo utilizador de forma a obter a morada para colocar esse pedido.	
Parâmetros de Input				
	Atributo	Tipo	Obr.	Descrição do Atributo
	latlong	latlong	S	Coordenadas da localização que o utilizador colocou.
Parâmetros de Output				
	Atributo	Tipo	Obr.	Descrição do Atributo
	morada	String	S	A morada desta localização.

Anexo E: Funcionamento da Unidade Curricular

Para um engenheiro informático uma das mais importantes habilidades seria o domínio da análise de sistemas, tanto para o próprio planejamento de aplicações onde seria necessário utilizar efetivamente todos os conhecimentos adquiridos na unidade curricular, bem como programando o sistema que foi desenhado por analistas de sistemas, pois devemos entender o que foi proposto e construir da melhor forma possível.

Aprendemos nessa unidade curricular todos os passos para o planejamento de um sistema de forma que possa ser desenvolvido de forma mais eficiente e completa, evitando ao máximo possível falhas e para isso utilizamos várias ferramentas para entender o que precisa ser desenvolvido, como por exemplo o diagrama de blocos, que possibilita identificar em alto-nível os principais componentes de nosso sistema ou também o diagrama de contexto que identifica os atores que interagem com o nosso Sol e os tipos de interfaces que devem existir para que o ator possa interagir com o sistema e os “enabling systems” que dão suporte a aplicação de forma a torná-la mais completa.

Os casos de utilização são uma forma de descrever exaustivamente os principais cenários de interação entre os atores e o sistema, o diagrama de casos de utilização é uma representação visual das interações destes atores.

Modelo de domínio representa uma estrutura de diagramas de classes e suas relações e é uma modelagem muito importante para o desenvolvimento de sistemas, pois tem como objetivo definir todas as classes que o sistema necessita possuir e as interações entre elas.

As máquinas de estado são uma representação da execução de um processo no sistema e os estados que o um objeto pode se encontrar, passando em um estado inicial para um estado final em várias transições.

Pensamos que futuras alterações nesta cadeira poderiam ser: Mais feedback em relação aos projetos fora dos pitches porque sentimos que o feedback dos pitches era muito útil e tentamos ao máximo corrigir o que nos foi indicado mas sentimos falta de apoio fora dessas avaliações.

Apreciamos muito o facto das aulas teóricas terem uma grande componente prática porque nos ajudou a compreender melhor os diagramas que estavam a ser apresentados.