



Universidade
Europeia

LAUREATE INTERNATIONAL UNIVERSITIES



QUICK HANDLER

Pedro Oliveira nº 50039057

Douglas Bewiahn nº 50038630

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

Índice:

Índice:	2
Enquadramento:	3
Proto-Personas:	4
Casos de utilização:	4
Requisitos Funcionais:	6
Sistemas de Informação Geográficos:	8
Gestão de Projetos Informáticos:	8
Ux Journeys:	12
Diagrama de Blocos	13
Diagrama de Máquina de Estados:	13
BPMN:	14
Referências:	15

Enquadramento:

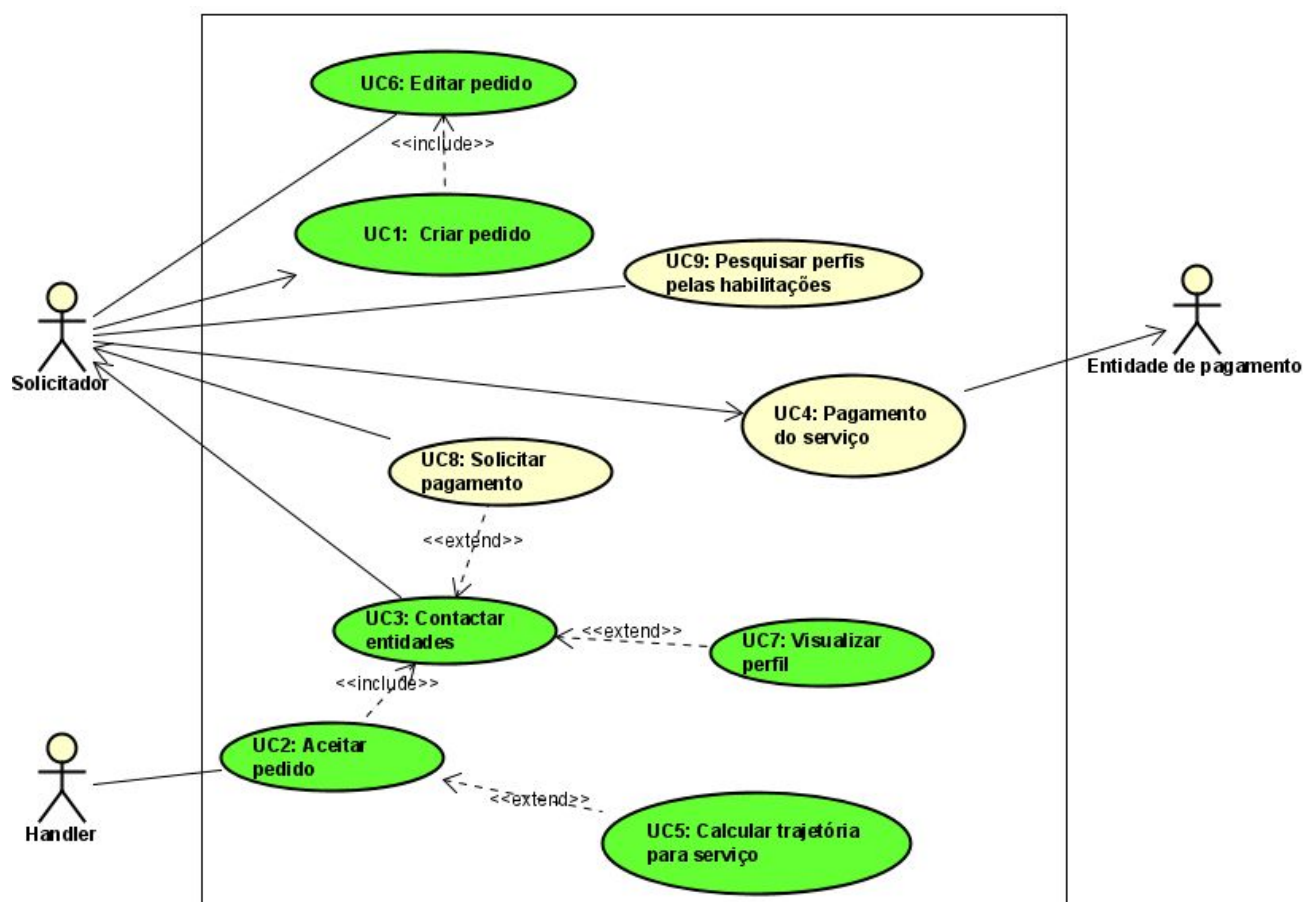
O nosso produto irá fornecer a pessoas que precisem de um serviço feito, uma plataforma que torna o contacto entre o cliente e o profissional mais simples, rápido e seguro.

Alguns produtos semelhantes ao nosso projeto são: faztudo, sosfztudo e starofservice. Entretanto estes não fornecem a rápida ligação entre as partes interessadas, e os pedidos são feitos através de e-mail ou formulários de contato, como no caso da faztudo e sosfztudo, é necessário um colaborador registrar a solicitação, contatar o profissional e enviar ao cliente o orçamento para aprovação. No caso da starofservice, o cliente responde a diversas questões para enviar a solicitação e após aprovada pela equipa é direcionada ao profissional da área que tem de fazer um orçamento e responder ao cliente, que por sua vez analisa e aprova um dos orçamentos.


O nosso projeto é importante porque fornecemos uma plataforma segura e simples que diminui a quantidade de tempo gasto à procura de profissionais qualificados para esse tipo de serviços. A nossa aplicação irá conectar vários profissionais com interessados no seu serviço de uma forma rápida e competitiva.

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.



Proto-Personas:



Carlos
Trabalhador

Sobre:

- 18+ anos
- Casado e com 2 filhos
- Qualquer salário
- Conhecimentos base de tecnologia

Necessidades:


- Reparar a sanita
- Conectar-se com um canalizador rapidamente

Pain Points:

- Mau serviço com Handlers pode motivá-lo a não confiar mais na nossa aplicação.

Motivations:

- Procurar e cuidar dos orçamentos de um canalizador gasta muito tempo



Manuel
Faz-tudo

Sobre:

- 18+ anos
- Casado e com filhos
- Qualquer salário
- Conhecimentos base de tecnologia

Necessidades:

- Complementar a rendimento mensal
- Conectar-se com clientes que precisam de seus serviços

Pain Points:

- Taxas de uso da aplicação serem demasiado elevadas

Motivations:

- Procurar trabalhos pontuais e novos clientes demanda muito tempo e investimento

Casos de utilização:

Nome: Solicitação de serviço

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.

Passo a passo:

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 necessitado.

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).

Pós-condições:

O pedido é guardado na base de dados e apresentado aos Handlers que possuam a categoria necessária do pedido.

Nome: Submissão de serviços

Descrição: O Handler observa as solicitações de serviços que tem a especialidade e a capacidade de cumprir e aceitar os serviços que deseja fazer.

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 habilitações necessárias para cada pedido.

Passo a passo:

Handler observa a lista de serviços à espera de serem aprovados.

Handler seleciona um pedido que deseja aceitar.

Handler é reencaminhado para uma página que apresenta esse pedido com mais detalhe.

Handler aprova o pedido.

O Sistema informa que o Handler só pode cancelar o pedido antes de faltarem 2 horas para o serviço ser feito.

Pós-condições:

O pedido muda de estado para “Aceite”.

Nome: Contacto entre as 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.
Passo a passo: 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).
Pós-condições: O pedido irá mudar o seu estado para "Feito". As avaliações vão ser adicionadas ao respectivo perfil da entidade avaliadas.

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

Documentação REST:

Recurso Pedidos (/api/pedidos)

Listar todos os pedidos pendentes

Devolve a lista de todos os pedidos pendentes com o id do pedido, a data, o estado do pedido, a morada, o username do cliente e a descrição do pedido.

/api/pedidos/pendentes (get)

Sucesso (200):

```
[
  {"id":7,"date":"2021-02-10T08:51:00.000Z","estado":"pendente","morada":"Av da Liberdade, 245","username":"user","descricao":"Reparos diversos no predio."},
  {"id":9,"date":"2021-02-26T23:53:00.000Z","estado":"pendente","morada":"Av. Brasil, 1","username":"user","descricao":"Trocar fichas que nao funcionam"},
  ... ]
```

Erro:

500:

Exemplo:

```
let elemMain = document.getElementById("requestList");
try {
  let pedidos = await $.ajax({
    url: "/api/pedidos/pendentes",
    method: "get",
    dataType: "json"
  });
  let html = "";
```

Obter a informação de um pedido através do seu Id

Vai devolver o id, a data, o estado do pedido mais o username do cliente, o handler_id e o cliente_id

/api/pedidos/:id (get)

Parâmetros:

id (obrigatório): o id do pedido pretendido

Sucesso (200):

```
{ "pedidoID": 9, "date": "2021-02-26T23:53:00.000Z", "estado": "pendente", "morada": "Av. Brasil, 1, Lisboa, Portugal", "descricao": "Trocar fichas que nao funcionam", "username": "user", "handler_id": null, "cliente_id": 1 }
```

Erros:500:

Exemplo:

```
try {  
  let pedido = await $.ajax({  
    url: "/api/pedidos/" + pedidoID,  
    method: "get",  
    dataType: "json"  
  });
```

Obter os pedidos através do id do user e do tipo de user

Vai devolver o id, a data, o estado do pedido, morada mais o username do cliente e a descricao

/api/pedidos/meusPedidos/:userID/:userType (get)

Parâmetros:

id (obrigatório): o id do user logado.

userType (obrigatório): Tipo de user logado

Sucesso (200):

```
[{"id":1,"date":"2021-03-21T15:30:00.000Z","estado":"atribuido","morada":"Rua do Vale de Santo António 180-256, 1170-382, Lisboa","username":"user","descricao":"Cano reventado na casa de banho, modelo X"}, {"id":4,"date":"2021-03-21T15:30:00.000Z","estado":"atribuido","morada":"Av. de Roma, 103, Lisboa, Portugal","username":"user","descricao":"Pintar a sala"}, {"id":7,"date":"2021-02-10T08:51:00.000Z","estado":"atribuido","morada":"Av da Liberdade, 245, Lisboa, Portugal","username":"user","descricao":"Reparos diversos no predio."}]
```

Erros:500

Exemplo:


```
let pedidos = await $.ajax({
    url: "/api/pedidos/meusPedidos/"+userID + "/" + userType,
    method: "get",
    dataType: "json"
});
```

Inserir um novo pedido

Insere um novo pedido à base de dados com a Data, morada, descrição, id de cliente e a categoria do serviço.

/api/pedidos/ (post)

Dados:

```
{ "date": "2021-01-13", "time": "10:30:00", "morada": "IADE", "description": "Montagem de armário de cozinha, modelo X", "cliente_id": "2", "handlerType": "Pedreiro"/}
```

Sucesso (200):

```
{
}
```

Erros:

500:

Exemplo:

```
let pedido = {
    date: document.getElementById("date").value+"
"+document.getElementById("time").value,
    morada: document.getElementById("morada").value,
    descricao: document.getElementById("description").value,
    cliente_id: userID,
    categoria: document.getElementById("handlerType").value
};

try {
    let result = await $.ajax({
        url: "/api/pedidos/",
        method: "POST",
        dataType: "json",
        data: JSON.stringify(pedido),
        contentType: "application/json"
    });
```

Aceitar pedido

Muda o estado de um pedido para “atribuido”

/api/pedidos/status/ (put)

Dados:

```
{ "handler_id": "2", "pedidoID": "11", "estado": "atribuído" }
```

Sucesso (200):

```
{  
}
```

Erros: 500:

Exemplo:

```
let aPedido = {  
  handler_id: userID,  
  pedidoID: pedidoID,  
  estado: "atribuido"  
};  
  
let result = await $.ajax({  
  url: "/api/pedidos/status/",  
  method: "PUT",  
  dataType: "json",  
  data: JSON.stringify(aPedido),  
  contentType: "application/json"  
});
```

Recurso Users (/api/user/)

Obter Id de um Utilizador dado o username e o tipo

Recebe o nome do Utilizador e o tipo de utilizador e devolve o Id do utilizador em questão para simular o login

/api/user/getId/:user/:userType (get)

Dados:

```
{ "user": "handler", "userType": "Handler" }
```

 todos os dados são obrigatórios

Sucesso (200):

```
[{"handlerID":1}]
```

Erros: 500:

Exemplo:

```
let userID = await $.ajax({  
  url: "/api/user/getId/" + user + "/" + userType,  
  method: "get",  
  dataType: "json"  
});
```

Obter Informação de um User através do seu id e do seu tipo

Recebe o email, nome e número de telemóvel do utilizador.

/api/user/getInfo/:userID/:userType (get)

Dados:

```
{ "userID": "2", "userType": "Handler" }
```

Parâmetros:

userId: o id do utilizador;

userType: o tipo de utilizador (Handler/Cliente)

Sucesso (200):

```
[{"email": "user2@example.com", "nome": "user exemple2", "telemovel": 999999998}]
```

Erros:

500:

Exemplo:

```
let userInfo = await $.ajax({  
  url: "/api/user/getInfo/" + userID + "/" + userType,  
  method: "get",  
  dataType: "json"  
});
```

Sistemas de Informação Geográficos:

Neste Projeto temos três funcionalidades que utilizam sistemas de informação geográficos sendo estas:

- **GeoCoding**, em que utilizamos a biblioteca [esri-leaflet-geocoder](https://esri.github.io/esri-leaflet/) (<https://esri.github.io/esri-leaflet/>) que contém um conjunto de serviços ArcGIS para o leaflet (<https://leafletjs.com/>) em que buscamos à base de dados a morada do serviço e transformamos esse dado em coordenadas para o podermos colocar no nosso mapa de forma ajudar os nossos clientes a visualizarem os dados dessa página;
- **Reverse GeoCoding** em que transformamos as coordenadas de um certo ponto no mapa na respetiva morada, utilizando a mesma biblioteca que a funcionalidade anterior;
- **Cálculo de rotas entre dois pontos**, para cumprir esta funcionalidade, utilizamos a biblioteca [GraphHopper](https://www.graphhopper.com/) (<https://www.graphhopper.com/>). Após o utilizador selecionar um pedido, pedimos ao utilizador que clique numa posição no mapa, de forma a definir o início da sua rota e guardamos as coordenadas dessa posição, apresentando-a de seguida com a rota ao serviço selecionado.

Encontramos problemas com a biblioteca GraphHopper por apresentar algumas rotas em que nos questionamos algumas escolhas das rotas que esta nos apresenta.

Gestão de Projetos Informáticos:

Ao longo deste semestre desenvolvemos vários documentos que nos ajudaram a planificar melhor as decisões a tomar para este projeto. Ao produzirmos o Project Charter, mais especificamente a análise SWOT, encontramos os nossos pontos fortes, pontos fracos, oportunidades e ameaças para o nosso projeto.

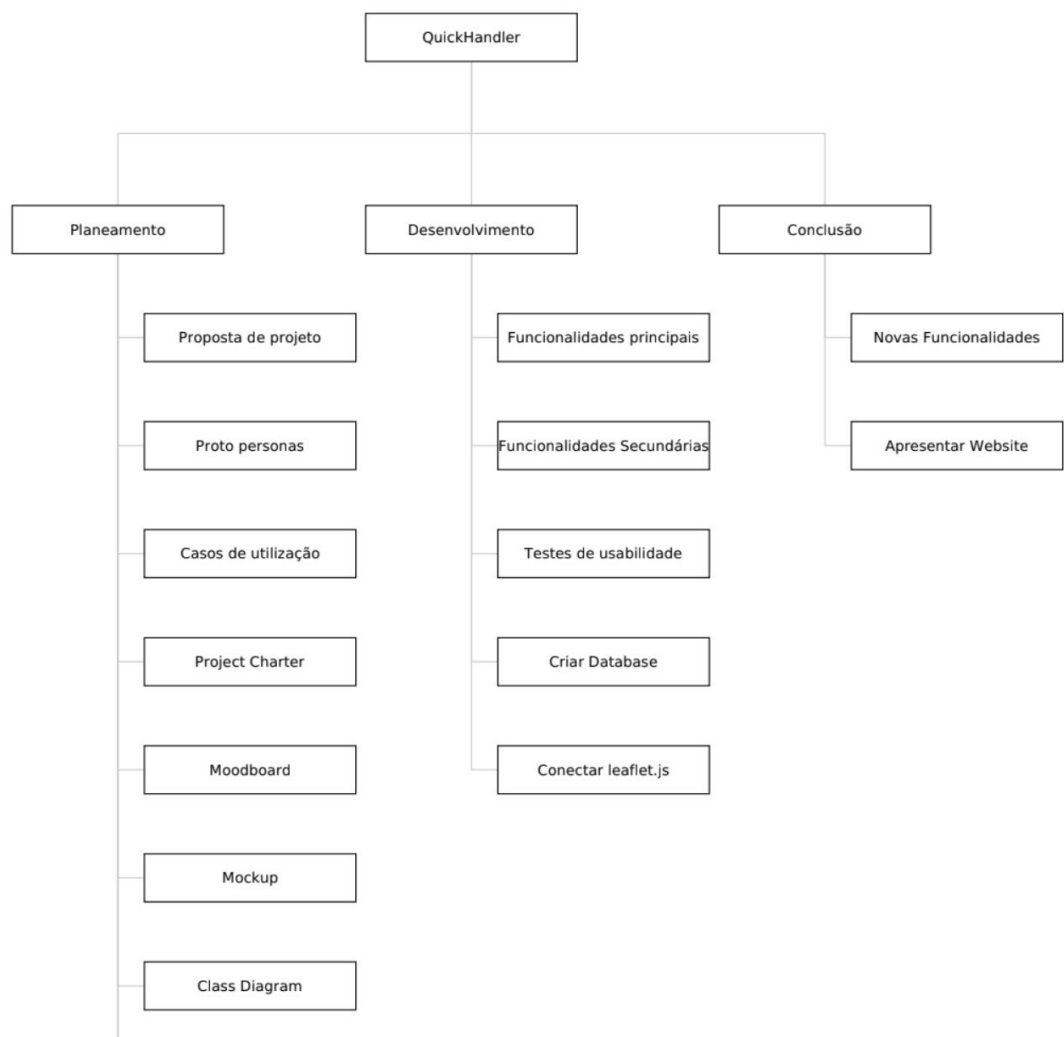
Pontos Fortes	Pontos Fracos	Oportunidades	Ameaças
Processo simples e rápido para solicitação de serviços	Fiabilidade da nossa aplicação depende dos nossos clientes	Podemos facilmente aumentar o tipo de serviços que fornecemos.	Handlers que aceitam pedidos que não vão cumprir

O project Charter ajudou-nos também a definir melhor o nosso Scope do projeto e Visão do projeto e ajudou a descobrir que partes da nossa aplicação nos deveríamos focar principalmente.

Fizemos também o software project plan que nos ajudou mais a prevenir as futuras implementações ao nosso projeto em cada cadeira e como nos iríamos agendar utilizando o diagrama de Gantt para definir datas limites para cada implementação.

Utilizamos o software do site clickup (<https://app.clickup.com/>) para produzirmos o nosso software project plan, principalmente utilizando-o para produzirmos o diagrama de Gantt.

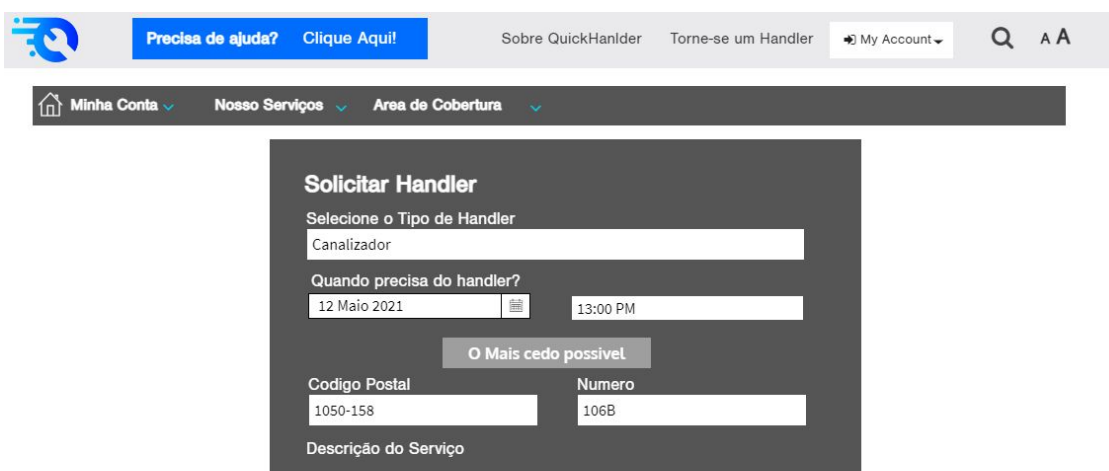
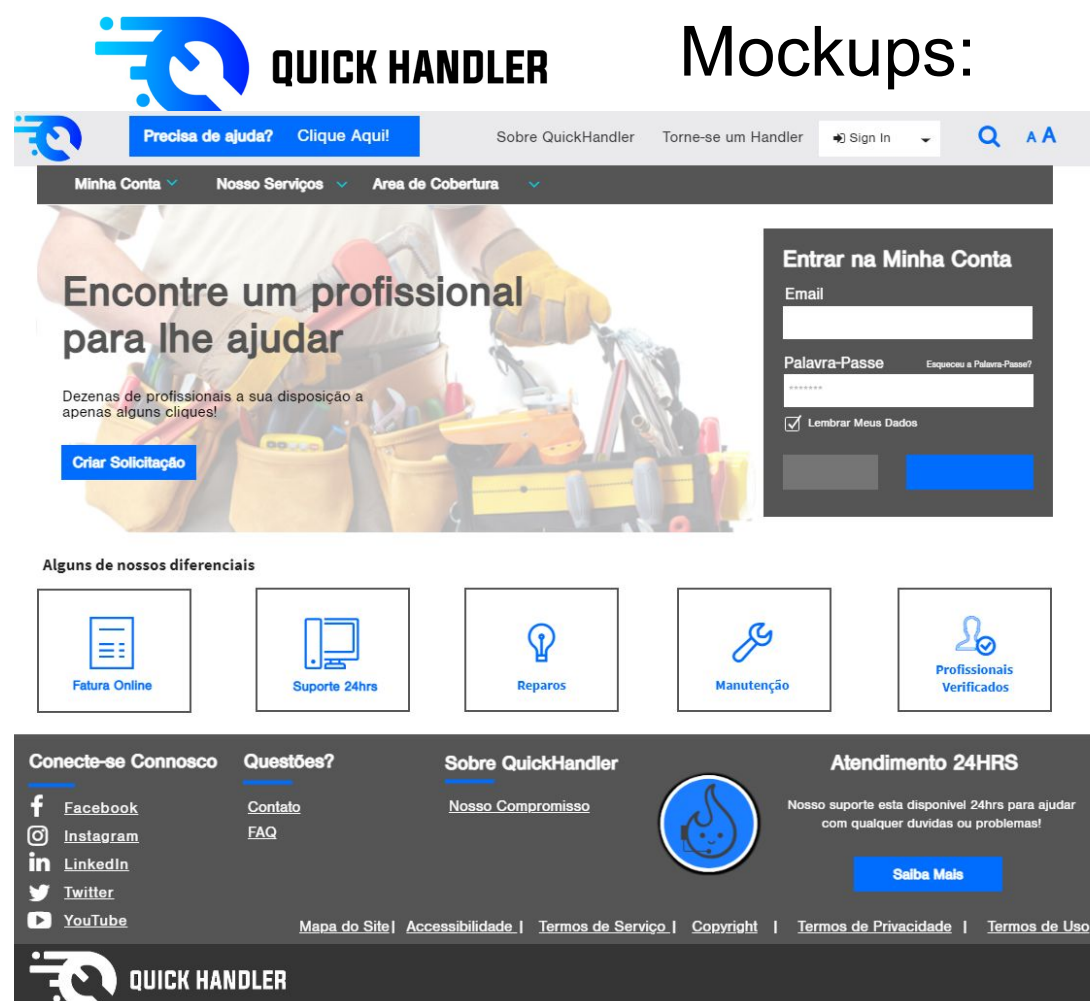
Utilizamos o WBS para definir as principais fases do desenvolvimento do nosso projeto e que documentos/ ações se incluíam em cada fase.



Branding:

Ao trabalharmos na nossa Brand (QuickHandler) decidimos focar na apresentação de um bom logótipo em que representava todas as palavras chave que caracterizam o nosso produto: Mão-de-obra; Movimento. O nosso logótipo representa também a letra inicial do nosso projeto "Q". O movimento provém do rastreio deixado atrás da bola.

Realizamos também testes de usabilidade ao criarmos os nossos primeiros mockups para perceber o que os nossos futuros clientes esperam ver na nossa aplicação.





Precisa de ajuda? [Clique Aqui!](#)

[My Account](#)



[Minha Conta](#) [Em Aberto](#) [Concluídas](#) [Canceladas](#)

Solicitações Disponíveis

Pesquisar Tipo de Handler



Tipo de Handler

Canalizador

Data do Serviço

12 Maio 2021

13:00 PM

Codigo Postal

1050-158



Descrição do Serviço

Um vazamento nos canos que ficam em baixo da pia da cozinha.

[Aceitar](#)

Tipo de Handler

Eletricista

Data do Serviço

12 Maio 2021

13:00 PM

Codigo Postal

1050-158



Descrição do Serviço

Refazer o cabeamento de tomadas que estão com mal contato

[Aceitar](#)

Tipo de Handler

Carpinteiro

Data do Serviço

23 Abril 2021

13:00 PM

Codigo Postal

1050-158



Descrição do Serviço

Instalação de duas portas de madeira nos quartos.

[Aceitar](#)

Ux Journeys:

Solicitador de Serviços

Manuel Guerreiro



Age
20 years

Highest Level of Education
High school degree

Industry
Student

Organization Size
Self-employed



Handler

Bruno Marques



Age
35 anos

Highest Level of Education
High School Graduation

Industry
Manufacturing

Organization Size
Self-employed

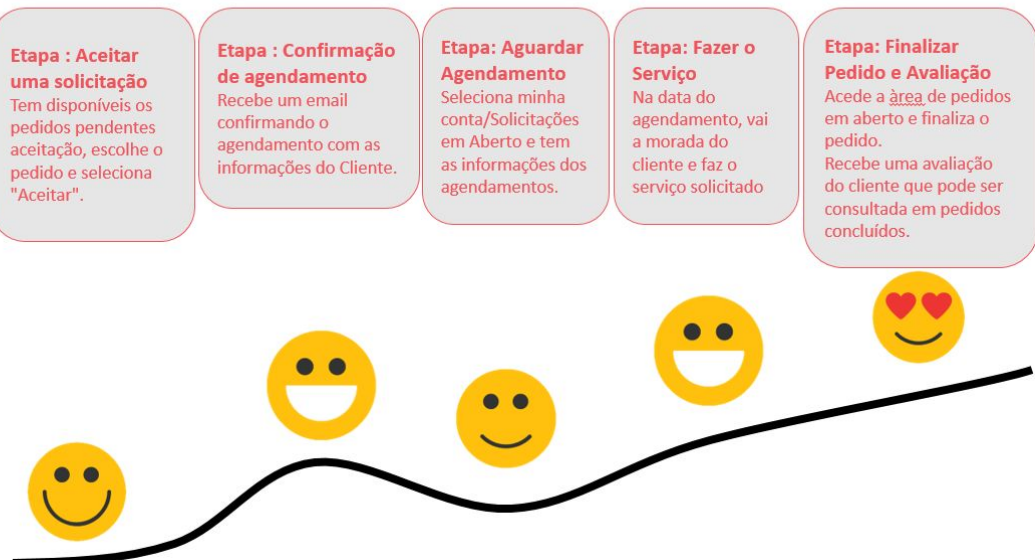


Diagrama de Blocos

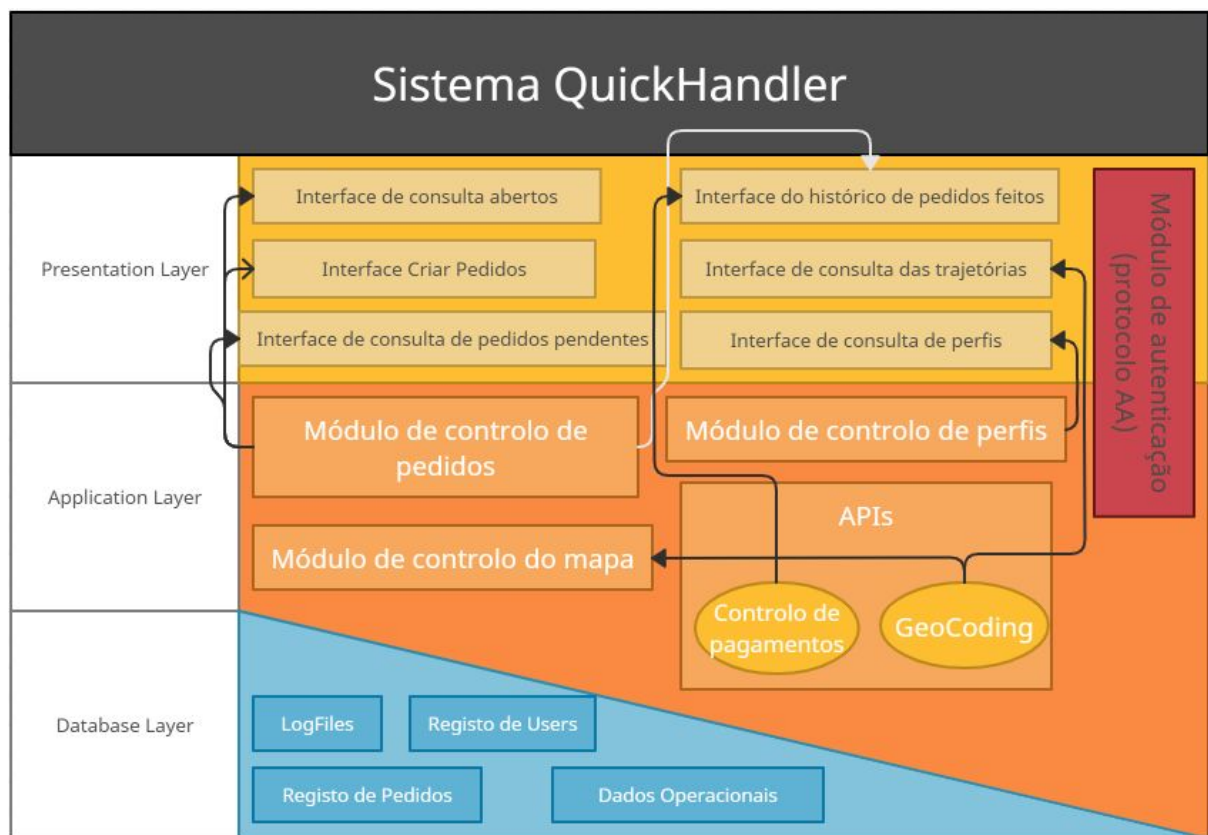
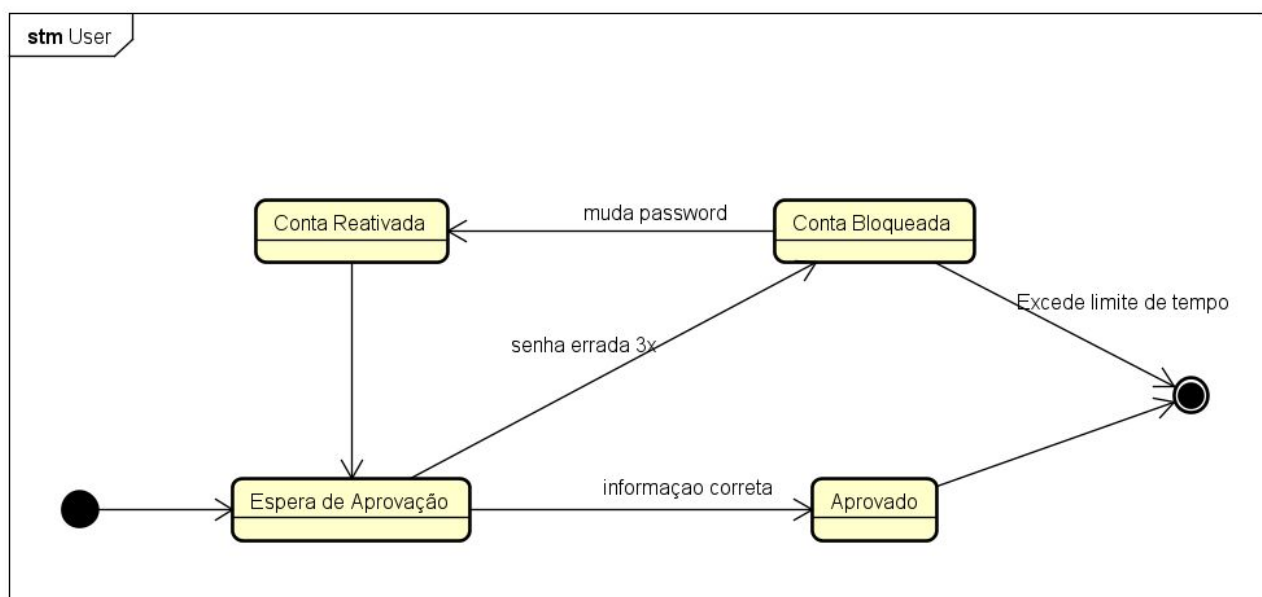
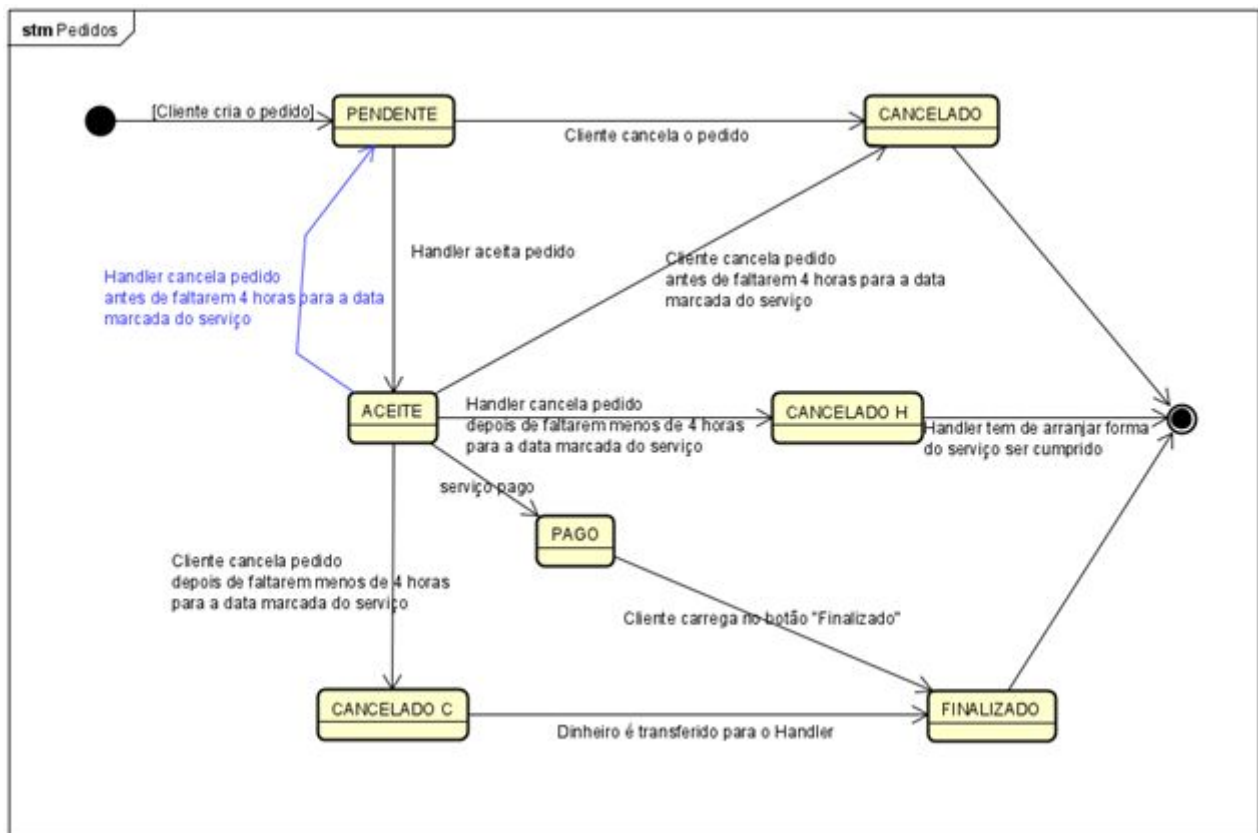
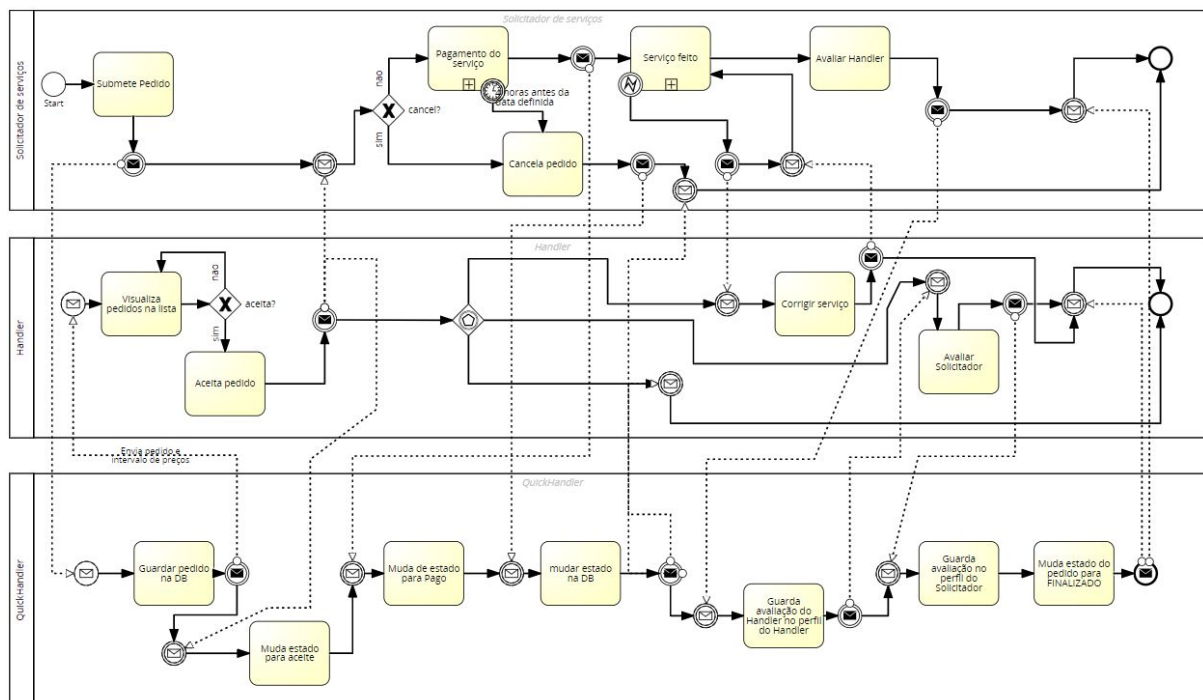


Diagrama de Máquina de Estados:





BPMN:



Referências:

SOS Faz Tudo. 2020. *Eletricista, Chaveiro, Canalizador | 24H | Em Lisboa | S.O.S Faz Tudo*. [online] Available at: <<https://sosfaztudo.pt/>> [Accessed 1 November 2020].

Faztudo.com.pt. 2020. *Reparações Ao Domicílio Lisboa | Faz Tudo*. [online] Available at: <<https://www.faztudo.com.pt/>> [Accessed 1 November 2020].

StarOfService. 2020. *Encontre Profissionais Locais Para Todos Os Seus Projetos*. [online] Available at: <<https://www.starofservice.pt/>> [Accessed 1 November 2020].

Alshawi, M., & Ingirige, B. (2003). Web-enabled project management: an emerging paradigm in construction. *Automation in Construction*, 12, 349-364. [Accessed 20 November 2020].

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].

Make My Persona. (2021). Retrieved 4 February 2021, from <https://www.hubspot.com/make-my-persona>