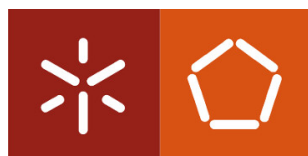


UNIVERSIDADE DO MINHO

ESCOLA DE ENGENHARIA



Projeto de Informática

Mestrado em Engenharia Informática

DTx - iHat

Requisitos e Arquitetura

PG50229 António Fernandes
PG50352 Eduardo Magalhães
PG50483 João Mendes
PG50499 João Torres
PG50518 José Vieira
PG50628 Mariana Rodrigues
PG50745 Rui Alves

Janeiro 2024

Índice

1	Resumo	3
2	Introdução	4
2.1	Contextualização, Motivação e Objetivos	4
2.2	Gestão do Projeto	4
3	Definições do Produto	6
3.1	Utilizadores do Produto	6
3.2	Restrições do Programa	6
3.3	Funcionalidades	6
4	Levantamento e Análise de requisitos	8
4.1	Apresentação da estratégia e método	8
4.2	Descrição geral dos requisitos funcionais	8
4.2.1	Requisitos do Responsável da Obra	8
4.2.2	Requisitos do Sistema	8
4.3	Descrição geral dos requisitos não-funcionais	9
4.3.1	Requisitos de Aparência	9
4.3.2	Requisitos de Usabilidade	9
4.3.3	Requisitos de Desempenho	9
4.3.4	Requisitos Operacionais	10
4.3.5	Requisitos de Manutenção e Suporte	10
4.3.6	Requisitos de Segurança	10
4.3.7	Requisitos Legais	10
5	Casos de Uso	11
5.1	Análise de Casos de Uso	11
5.1.1	Use Case: Adicionar capacetes à obra	11
5.1.2	Use Case: Consultar capacetes	12
5.1.3	Use Case: Consultar Obras	12
5.1.4	Use Case: Login do Administrador/Responsável da obra	12
5.1.5	Use Case: Adicionar um mapa à obra	13
5.1.6	Use Case: Apresentar o mapa da obra	13
5.1.7	Use Case: Apresentar a informação de um capacete	13
5.1.8	Use Case: Alterar zonas de risco a um mapa	14
5.1.9	Use Case: Registrar obra no sistema	14
5.1.10	Use Case: Mudar o estado da obra	14
5.1.11	Use Case: Adicionar/Associar capacete ao sistema	15
5.1.12	Use case: Visualizar a localização dos capacetes	15
5.1.13	Use case: Visualizar dados de um capacete	16
5.1.14	Use case: Atualizar estado do capacete	16
5.1.15	Use case: Consultar Logs de um(a) obra/capacete	16
5.2	Diagrama de Casos de Uso	17
6	Modelo de Dominios	18
7	Diagrama de Classes	19

8	Diagrama de Componentes	20
9	Diagrama de Implementação	21
10	Máquinas de Estados	22
10.1	Máquina de Estados da Obra	22
10.2	Máquina de Estado do Capacete	23
11	Diagrama de Atividades	25
11.1	Tratar mensagens recebidas do Capacete	25
11.2	Adicionar Mapa à Obra	25
12	Mockups	27
12.1	Página Inicial	27
12.2	Página da Obra	28
12.3	Página do Capacete	28
13	Database	30
14	Formato das Mensagens MQTT	33
15	Conclusões	35
15.1	Resultados Obtidos	35
15.2	Trabalho Futuro	35

1 Resumo

O seguinte documento surge como consequência da proposta feita à Universidade do Minho pela DTx, com o objetivo de desenvolver um produto informático no âmbito do Projeto de Informática. A equipa composta pelos autores deste documento ficou encarregue do desenvolvimento do produto informático.

A DTx pretende desenvolver um capacete de obra inteligente que irá monitorizar sinais vitais, detetar quedas e localização geográfica recorrendo a sensores existentes.

A proposta da DTx à UMinho consiste no desenvolvimento de uma WebApp que irá acompanhar esses capacetes da obra. Esta deverá permitir aos utilizadores deste sistema acompanhar os dados recebidos dos capacetes.

Este documento foi desenvolvido no âmbito de uma segunda componente a desenvolver para este projeto. Esta segunda componente do projeto engloba desenvolver um documento onde estão apresentados os requisitos levantados pela equipa e os diagramas UML desenvolvidos para modular a WebApp implementada.

2 Introdução

2.1 Contextualização, Motivação e Objetivos

A construção civil é uma área com um elevado interesse e impacto económico que infelizmente está associado a um elevado número de riscos e a um elevado número de acidentes no trabalho podendo potencialmente levar à mortes dos trabalhadores.

O capacete é um elemento obrigatório e essencial durante a movimentação em locais de construção. Esta obrigação tem como objetivo proteger os trabalhadores e reduzir a probabilidade de acidentes graves.

O proposto com este projeto é utilizar este elemento já existente e complementá-lo com outras tecnologias, de forma a expandir a capacidade de proteção dos capacetes. Assim, com este projeto pretendemos tentar tornar a atividade de construção civil mais eficiente e segura utilizando a tecnologia à nossa disposição para proteger a vida humana e tentar melhorar as condições de trabalho.

O projeto que desenvolvemos consistirá numa plataforma que irá comunicar com capacetes e outros equipamentos necessários. Este sistema irá permitir recolher, analisar e apresentar dados relevantes sobre o estado dos trabalhadores.

Este sistema irá depender de sensores que irão permitir recolher dados relacionados com dados vitais dos trabalhadores, com a localização dos mesmos dentro de uma área de construção, com o ambiente da área e com eventos que poderão ocorrer.

Por sua vez, a comunicação entre a plataforma e o capacete deve ser bidirecional, e o programa desenvolvido deverá ser capaz comunicar com capacete, para que notificá-lo de situações anómalas detetadas.

Por último, este documento foi desenvolvido no âmbito da segunda componente da proposta da DTx. Este documento deverá conter os requisitos levantados e diagramas UML que representem a arquitetura do sistema a implementar.

2.2 Gestão do Projeto

Numa primeira fase, a equipa reuniu de maneira a identificar as tarefas que teriam que ser executadas para um correto desenvolvimento do projeto e definir prazos para a realização das mesmas. Foi criado um diagrama de Gantt com o planeamento concordado pela equipa e pela empresa.



Figure 2.1: Diagrama de Gantt

Em qualquer projeto de desenvolvimento de software é extremamente importante que a equipa de trabalho seja organizada. Para isso, a utilização de um serviço de ferramentas de gestão de projetos são bastante úteis. Desta forma, a equipa de trabalho decidiu utilizar a ferramenta ***Jira Software***, que se trata precisamente de uma ferramenta de gestão de projetos *agile*.

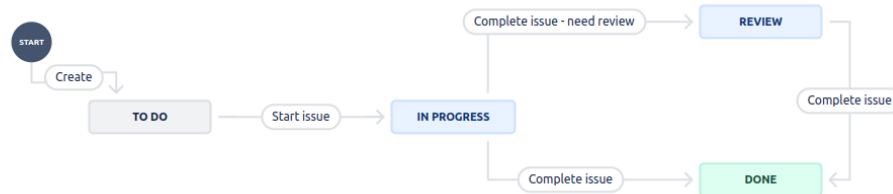


Figure 2.2: Representação do Workflow do Jira

Para além da utilização da ferramenta *Jira Software*, dividiu-se também a equipa em sub-equipas, focando-se cada uma delas no desenvolvimento do *backend* e *frontend* do projeto em questão. A equipa utilizou também a plataforma Github, para o controlo de versões do programa.

Durante o decorrer do projeto, foram realizadas reuniões semanais com os responsáveis pelo projeto do lado da DTx.

3 Definições do Produto

3.1 Utilizadores do Produto

A aplicação desenvolvida destina-se aos responsáveis de obras, isto é, aos trabalhadores encarregues de gerir e monitorizar todas as etapas da construção. A nível do sistema, exerce todas as funções de *admin*. No entanto, preocupa-se principalmente com a vigilância ativa dos trabalhadores da obra que lhe foi atribuída. Para isso utiliza as funcionalidades do sistema para garantir a segurança e integridade do ambiente. Assim, está envolvido na interpretação de dados, identificação de potenciais ameaças e resposta rápida a incidentes de segurança.

3.2 Restrições do Programa

Para garantir o funcionamento do programa será necessário seguir um conjunto de restrições:

- Para assegurar um requisito legal, os trabalhadores deverão autorizar o uso das suas informações e localização.
- Para assegurar o funcionamento do programa, deverá existir no perímetro da obra um mecanismo que permite realizar a leitura de QrCodes. Os QrCodes a ler deverão estar presentes nos capacetes e nos trabalhadores. Se estes não existirem o programa não conseguirá fazer a associação do trabalhador ao capacete.
- Neste programa, a localização dos trabalhadores será obtido em função da sua localização no perímetro da obra, ou seja, a localização dos trabalhadores não será baseada em coordenadas geográficas, mas sim em coordenadas relativas ao perímetro da obra. Assim, é necessário que no perímetro da obra existam beacons, para que os capacetes possam obter a sua localização em referência a estes.

3.3 Funcionalidades

A partir da análise do ficheiro com a proposta do projeto, recolhemos as seguintes funcionalidades. Estas poderão ser completadas com outras funcionalidades identificadas após o levantamento de requisitos.

- Permitir introduzir um mapa da obra em formato IFC 3D
- Permitir introduzir os capacetes a utilizar
- Permitir alterar os capacetes que estão a ser utilizado (adicionar/remover)
- Identificar zonas de perigo no mapa
- Permitir começar e terminar a obra
- Apresentar o mapa da obra
- Apresentar a localização dos trabalhadores no mapa da obra
- Apresentar uma visão geral do estado de todos os trabalhadores, que seja fácil de visualizar

- Apresentar os dados de cada capacete
- Apresentar notificações na aplicação
- Enviar notificações para o capacete (Alertas)
- Permitir associação entre capacete e trabalhador a partir da leitura de QrCodes

4 Levantamento e Análise de requisitos

4.1 Apresentação da estratégia e método

O primeiro contacto que tivemos com o projeto foi a partir da leitura da descrição da proposta feita à UMinho. A partir deste documento, foram levantados alguns requisitos para o programa a desenvolver.

Posteriormente, foi realizada uma reunião com dois representantes da DTx, onde foram apresentados alguns outros requisitos e restrições do projeto.

Por último, foram realizadas reuniões onde foram apresentados os requisitos levantados. Estes foram validados pelos representantes.

4.2 Descrição geral dos requisitos funcionais

4.2.1 Requisitos do Responsável da Obra

- O responsável da obra deve poder consultar as obras existentes no sistema.
- O responsável da obra deve poder consultar os capacetes do sistema.
- O responsável da obra deve autenticar-se para poder aceder às funcionalidades. A autenticação deve ser feita com email e password.
- O responsável da obra deve introduzir o mapa da obra a utilizar quando se dá início a uma nova obra.
- O responsável da obra deve identificar as zonas de risco no mapa da obra no início da obra.
- O responsável da obra deve poder alterar as zonas de risco durante a realização da obra.
- O responsável da obra deve ser capaz de adicionar e remover capacetes de uma obra.
- O responsável da obra deve ser capaz de visualizar os trabalhadores da obra, o capacete que utilizam e localização dos mesmos no mapa.
- O responsável da obra deve poder visualizar os dados recebidos do capacete.
- O responsável deve ser capaz de alterar o estado da obra
- O responsável deve ser capaz de alterar o estado do capacete

4.2.2 Requisitos do Sistema

- O sistema deve ser capaz de apresentar o mapa e as zonas de risco da obra.
- O sistema deve ser capaz de associar um capacete a um trabalhador.
- O sistema deve ser capaz de enviar mensagens/ notificações ao capacete.
- O sistema deve ser capaz de identificar situações em que deverão ser enviadas notificações (por exemplo, quedas detetadas, sinais vitais irregulares, proximidade a uma zona de risco)

- O sistema deve ser capaz de receber dados do capacete.
- O sistema deve apresentar os dados recebidos pelo capacete, e sinalizar situações irregulares.
- O sistema deve ser capaz de representar a posição do capacete no mapa.
- O sistema deve ser capaz de apresentar notificações ao responsável da obra. (ex: trabalhador em zona de perigo, queda, etc..)

4.3 Descrição geral dos requisitos não-funcionais

- Programa desenvolvido deve ser escalável. Descrição: O programa deve ser capaz de escalar e responder a todos os pedidos dos utilizadores da aplicação e receber e "tratar" todas as mensagens dos capacetes
- Usar o formato IFC 3D
- Backend da ferramenta deve ser desenvolvido em .NET v7
- API apresentar uma interface swagger
- FrontEnd da ferramenta deve ser desenvolvido em vue.js
- Dados a considerar da api: *quedas/impacto (solução comercial com acelerómetro), temperatura corporal e ritmo cardíaco anormal (printed sensor), gases (solução comercial), proximidade (printed sensor), posição do utilizador (solução comercial) de queda sem movimento do capacete e do utilizador também.*

4.3.1 Requisitos de Aparência

- A interface deve utilizar o esquema de cores da empresa DST

4.3.2 Requisitos de Usabilidade

- A interface deve ser intuitiva e fácil de utilizar.

4.3.3 Requisitos de Desempenho

- O capacete tem de identificar autonomamente situações anómalas ou críticas urgentes, sem necessidade de comunicação com a plataforma e notificar o trabalhador. Deve indicar também quando não deteta nenhuma situação anómala.
- A deteção de uma anomalia deve ativar um alarme tanto na plataforma como no capacete, por exemplo, através de uma luz de urgência.
- Se a plataforma detetar uma anomalia deve comunicar com o capacete de forma a este alertar o trabalhador.

4.3.4 Requisitos Operacionais

- O capacete deve dispor de formas redundantes de comunicação para situações de falha do canal normal e para situações de anomalia urgentes.
- Os mapas de localização têm de estar no formato IFC 2D.
- A comunicação do capacete é feita via LoRa.
- Os sensores no capacete contemplam: quedas/impacto (solução comercial com acelerómetro), temperatura corporal (printed sensor) e ritmo cardíaco anormal, gases (solução comercial), proximidade (printed sensor: para deteção ao topo), posição do utilizador (solução comercial) de queda sem movimento do capacete e pressupõem-se que do utilizador também
- O DTx deve partilhar a arquitetura da plataforma com os copromotores CENTI e INL
- A definição dos intervalos de segurança para os parâmetros medidos pelo capacete feita na plataforma e transmitida para os capacetes.

4.3.5 Requisitos de Manutenção e Suporte

- Tem de existir uma forma de carregar o capacete em modo urgente quando tal não é possível via painel solar.

4.3.6 Requisitos de Segurança

- Autenticação e autorização

4.3.7 Requisitos Legais

- O sistema não pode apresentar informações pessoais dos trabalhadores

5 Casos de Uso

Após introduzirmos os requisitos que o programa deverá assegurar e apresentarmos o utilizador do sistema, desenvolvemos um diagrama de casos de uso. Este diagrama permite apresentar de forma explícita as funcionalidades que o sistema vai disponibilizar ao "*Responsável da Obra*".

5.1 Análise de Casos de Uso

Nesta secção apresentamos uma análise detalhada de cada um dos casos de uso deste programa. A definição dos Use Cases deve facilitar a implementação de cada funcionalidade do sistema.

O formato seguido para cada caso de uso será:

Descrição: ...

Pré-condição: ...

Pós-condição: ...

Fluxo Normal:

1. ...

Fluxo de exceção [Condição] (passo x)

1. ...

Fluxo Alternativo [Condição] (passo x)

1. ...

Este formato permite que seja perceptível o fluxo sequencial da interação do ator com o sistema.

5.1.1 Use Case: Adicionar capacetes à obra

Descrição: O *use case* Adicionar capacetes à obra consiste em associar capacetes livres a uma obra

Pré-condição: Capacete encontra-se no estado Livre e com nenhuma obra associada

Pós-condição: Capacete adicionado à obra

Fluxo Normal:

1. Sistema fornece a lista de todos os capacetes possíveis de adicionar
2. O Responsável da obra seleciona um ou mais capacetes e confirma a seleção
3. O Sistema regista a adição dos capacetes à obra e informa que a adição foi concluída com sucesso.

Fluxo de exceção [Não existem capacetes livres disponíveis] (passo 2)

1. O Responsável da obra é incapaz de selecionar um capacete
2. Sistema não regista nenhum capacete novo à obra.

5.1.2 Use Case: Consultar capacetes

Descrição: O *use case* Consultar capacete consiste em listar todos os capacetes presentes no sistema

Pré-condição: Responsável da obra encontra-se na pagina inicial

Pós-condição: Sistema apresenta lista dos capacetes do sistema

Fluxo Normal:

1. Responsável da obra pede a lista de todos os capacetes do sistema
2. Sistema apresenta lista de todos os capacetes do sistema

Fluxo de exceção [Não há capacetes no sistema] (passo 2)

1. Sistema informa que não existe nenhum capacete no sistema

5.1.3 Use Case: Consultar Obras

Descrição: O *use case* Consultar obras consiste em listar todos as obras presentes no sistema

Pré-condição: Responsável de obra encontra-se na pagina inicial **Pós-condição:** Sistema apresenta lista as obras do sistema

Fluxo Normal:

1. Responsável da obra pede a lista de todos as obras do sistema
2. Sistema apresenta lista de todas as obras do sistema

Fluxo de exceção [Não há obras no sistema] (passo 2)

1. Sistema informa que não existem obras

5.1.4 Use Case: Login do Administrador/Responsável da obra

Descrição: O *use case* Login do Administrador consiste na autenticação com credenciais de acesso para entrar no sistema, sendo que o login é feito pelo email e *password*.

Pré-condição: Ator não estar autenticado

Pós-condição: Ator estar autenticado

Fluxo Normal:

1. Sistema pede as credenciais de acesso;
2. Responsável da obra fornece o email e a password;
3. Sistema valida o email e a password;
4. Sistema que informa que o administrador foi autenticado com sucesso.

Fluxo de exceção [email inexistente] (passo 3)

1. Sistema informa que o email não existe.

Fluxo Alternativo [password incorreta] (passo 3)

1. Sistema informa que a password não está associada ao email fornecido.
2. Voltar ao passo 1.

5.1.5 Use Case: Adicionar um mapa à obra

Descrição: O Responsável da obra adiciona um ficheiro IFC 3D que contém o mapa da obra

Pré-condição: A obra não tem nenhum mapa registado

Pós-condição: O mapa da obra é atualizado

Fluxo Normal:

1. Responsável da obra indica o ficheiro com o mapa que pretende adicionar à obra.
2. Sistema recebe o ficheiro IFC 3D.
3. Sistema obtém as plantas 2D do ficheiro de cada piso.
4. Sistema verifica se pode adicionar o mapa à obra.
5. Sistema guarda os ficheiros associados ao mapa.
6. Sistema informa que o mapa foi adicionado com sucesso, mostrando o mapa.

Fluxo de exceção [Não é possível adicionar mapa à obra] (passo 4)

1. Sistema notifica o Responsável da obra que não é possível adicionar um mapa à obra

5.1.6 Use Case: Apresentar o mapa da obra

Descrição: Responsável da obra pretende visualizar o mapa para uma obra específica

Pré-condição: Existe pelo menos uma obra e essa obra tem um mapa associado

Pós-condição: Mapa apresentado pelo sistema

Fluxo Normal:

1. Administrador escolhe obra para a qual pretende visualizar o mapa
2. Sistema apresenta o mapa da obra pretendida

5.1.7 Use Case: Apresentar a informação de um capacete

Descrição: O Responsável da obra pretende visualizar a informação de um determinado capacete

Pré-condição: Existe pelo menos um capacete no sistema

Pós-condição: Informação do trabalhador apresentado pelo sistema

Fluxo Normal:

1. Responsável da obra escolhe o capacete que pretende visualizar
2. Sistema apresenta a informação do capacete

Fluxo de exceção [Não existem informações do capacete] (passo 2)

1. Sistema informa que não existe informação do capacete disponível

5.1.8 Use Case: Alterar zonas de risco a um mapa

Descrição: Permite que se selecione uma zona de um mapa que irá corresponder a uma zona de risco

Pré-condição: Existe pelo menos uma obra no sistema

Pós-condição: Mapa atualizado com as novas zonas de risco

Fluxo Normal:

1. Responsável da obra escolhe uma obra
2. Sistema apresenta o mapa da obra pretendida
3. Responsável da obra adiciona/altera/remove uma ou mais zonas de risco.
4. Responsável da obra termina a edição efetuada
5. Sistema atualiza as zonas de risco do mapa e informa o Responsável da obra.

Fluxo de exceção [Não existe mapa associado à obra] (passo 2)

1. Sistema informa o Responsável da obra que não existe nenhum mapa associado à obra selecionada

5.1.9 Use Case: Registrar obra no sistema

Descrição: Responsável da obra pretende registar uma obra no sistema.

Pré-condição:

Pós-condição: Obra foi registada no sistema com sucesso.

Fluxo Normal:

1. Responsável da obra indica ao sistema que pretende registar uma obra.
2. Sistema apresenta formulário de registo com o nome da obra
3. Responsável da obra preenche o nome e submete o formulário.
4. Sistema verifica se o formulário se encontra bem preenchido.
5. Sistema comunica ao responsável da obra que a obra foi registada com sucesso.

Fluxo de exceção 1 [Nome da obra já existe] (passo 4)

1. Sistema informa que já existe uma obra com o nome fornecido.
2. Volta ao passo 2.

5.1.10 Use Case: Mudar o estado da obra

Descrição: Responsável da obra pretende alterar o estado de uma obra.

Pré-condição: Existe pelo menos uma obra no sistema

Pós-condição: Estado da obra foi alterado com sucesso.

Fluxo Normal:

1. Responsável da obra seleciona uma obra
2. Sistema apresenta a informação da obra selecionada

3. Responsável da obra indica que pretende mudar o estado da obra
4. Sistema lista as opções de estados disponíveis
5. Responsável da obra seleciona um estado
6. Sistema solicita o Responsável da obra para a confirmação da ação.
7. Responsável da obra confirma a ação.
8. Sistema comunica que o estado da obra foi alterado com sucesso.

Fluxo de exceção 1 [Responsável da obra não confirma a ação.] (passo 7)

1. Volta ao passo 2.

Fluxo de exceção 2 [A obra encontra-se Cancelada ou Finalizada] (passo 4)

1. Sistema indica que não é possível alterar o estado da obra.

5.1.11 Use Case: Adicionar/Associar capacete ao sistema

Descrição: Responsável da obra pretende adicionar um capacete ao sistema

Pré-condição:

Pós-condição: Novo capacete associado ao sistema

Fluxo Normal:

1. Responsável da obra indica ao sistema que pretende registar um capacete.
2. Sistema apresenta formulário de registo com o id do capacete
3. Responsável da obra preenche o id e submete o formulário.
4. Sistema verifica se o formulário se encontra bem preenchido.
5. Sistema comunica ao responsável da obra que o capacete foi registada com sucesso.

Fluxo de exceção [Já existe um capacete com o id fornecido] (passo 4)

1. Sistema indica que já existe um capacete com o id fornecido
2. Volta ao passo 2.

5.1.12 Use case: Visualizar a localização dos capacetes

Descrição: Responsável da obra pretende visualizar a localização dos capacetes

Pré-condição: Capacetes tem de estar associados à obra

Pós-condição: Visualizar os diversos capacetes associados à obra no mapa

Fluxo Normal:

1. Responsável da obra abre a página onde a obra está representada
2. Sistema renderiza capacetes no mapa

Fluxo de exceção [Capacete não existe] (passo 2)

1. Capacetes não tem localização associada

5.1.13 Use case: Visualizar dados de um capacete

Descrição: Responsável da obra pretende visualizar os dados de um capacete

Pré-condição: Capacete tem de estar com o estado "Em uso"

Pós-condição: Visualizar os dados do capacete selecionado

Fluxo Normal:

1. Responsável da obra a partir da pagina obra seleciona capacete que pretende analisar
2. Sistema renderiza a página com todos os dados associados ao capacete selecionado

Fluxo de exceção [Condição] (passo 2)

1. Capacete não tem dados associados

5.1.14 Use case: Atualizar estado do capacete

Descrição: Alterar o estado de um capacete

Pré-condição: Existe pelo menos um capacete no sistema

Pós-condição: O estado do capacete foi alterado com sucesso

Fluxo Normal:

1. Responsável da obra indica que pretende mudar o estado do capacete na página da respetiva obra
2. Sistema lista as opções de estados disponíveis
3. Responsável da obra seleciona um estado
4. Sistema solicita o Responsável da obra para a confirmação da ação.
5. Responsável da obra confirma a ação.
6. Sistema comunica que o estado da obra foi alterado com sucesso.

Fluxo de exceção 1 [Responsável da obra não confirma a ação.] (passo 7)

1. Volta ao passo 2.

Fluxo de exceção 2 [O Capacete encontra-se Em Uso] (passo 4)

1. Sistema indica que não é possível alterar o estado da obra.

5.1.15 Use case: Consultar Logs de um(a) obra/capacete

Descrição: Consultar os logs de um(a) obra/capacete

Pré-condição: Existe pelo menos um(a) obra/capacete no sistema

Pós-condição: O sistema lista os logs da/do obra/capacete

Fluxo Normal:

1. Responsável da obra seleciona um(a) obra/capacete
2. Sistema apresenta os logs de um(a) obra/capacete

Fluxo de exceção [Não existe logs da/do obra/capacete selecionada] (passo 2)

1. O sistema indica que não existem logs para esta/este obra/capacete.

5.2 Diagrama de Casos de Uso

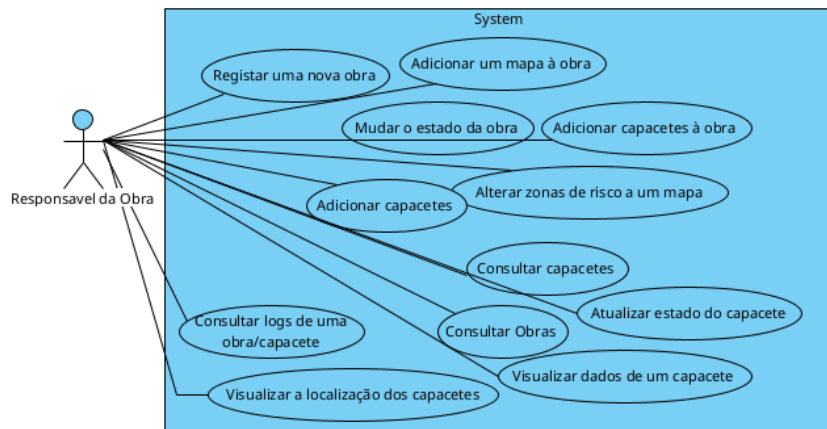


Figure 5.1: Diagrama de Casos de Uso

6 Modelo de Domínios

O Diagrama de Modelo de Domínio UML é uma representação visual que oferece uma visão abrangente das entidades e relacionamentos essenciais dentro do nosso projeto. Este diagrama é particularmente útil na fase inicial de análise e design de sistemas, permitindo uma compreensão clara das entidades fundamentais e das interações entre elas. Com base nos requisitos concluímos que necessitávamos de definir uma entidade **Obra**, esta obra tem vários estados associados (Finalizada, Pendente, Em curso, Cancelada), as respectivas **Plantas** e as suas **Zonas de Risco**. A Obra contém também **Capacetes** e é gerida por um **Responsável de Obra**. Os capacetes possuem um **Identificador** único e recolhem vários dados provenientes de sensores (Ritmo Cardíaco, Gases, Posição, Quedas/Impacto, Temperatura Corporal, Proximidade), existe também um estado relativo a cada capacete (Livre, Não Operacional, Em Uso), dependendo do seu estado o capacete terá um **Trabalhador** associado.

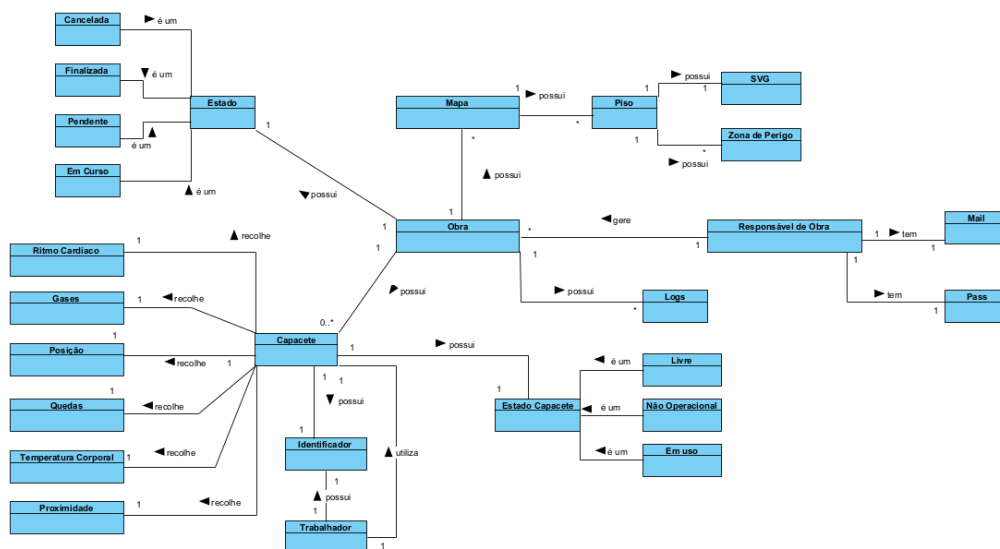


Figure 6.1: Modelo de Domínio

7 Diagrama de Classes

Os diagramas de classe representam a estrutura estática de um sistema, mostrando as classes que o compõem, seus atributos, métodos e os relacionamentos entre as diversas classes.

Na figura que se segue, figura 7.1, apresentamos o diagrama de classes elaborado para o desenvolvimento da aplicação.

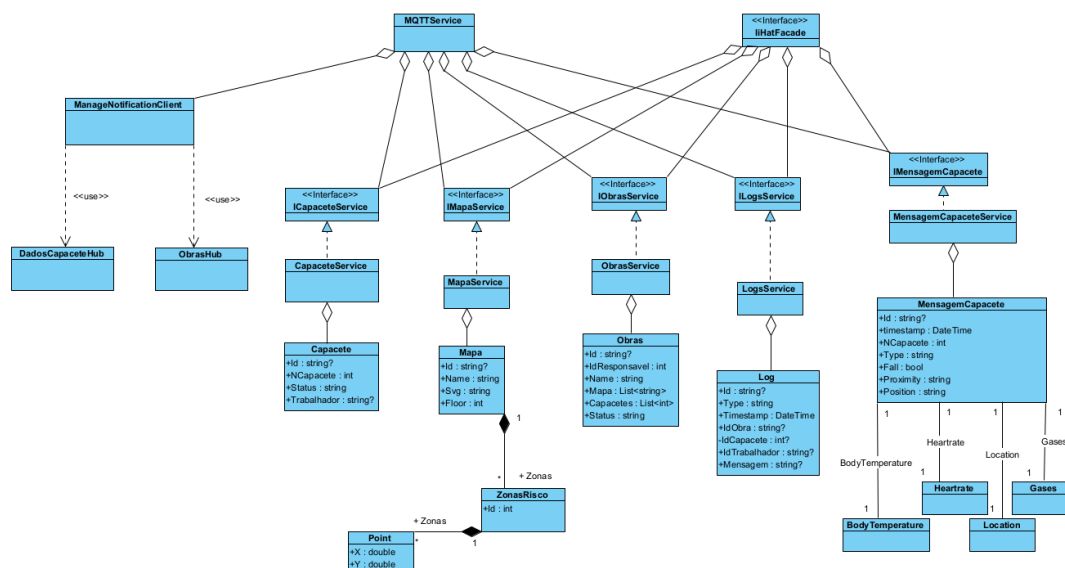


Figure 7.1: Diagrama de Classes

O programa final deverá possuir duas classes que serão responsáveis por receber pedidos e enviar respostas a pedidos vindos de diferentes protocolos. Estas classes serão o *IiHatFacade* e o *MQTTService*. A interface *IiHatFacade* irá possuir as funções necessárias para dar resposta a pedidos HTTP vindos do frontend da *WebApp*. A classe *MQTTService* deverá receber e tratar as mensagens MQTT do Capacete.

Na implementação final, pretendemos que as classes *Capacete*, *Mapa*, *Obras*, *Log* e *MensagemCapacete* contenham dados que serão relevantes para o bom funcionamento do programa. Estas classes deverão também representar objetos a guardar na base de dados.

As interfaces "Service" deverão permitir o consultar e guardar objetos que se encontram na base de dados. Cada "Service" deverá estar responsável por um dado objeto. Estes "Service" serão também utilizados pelo *IiHatFacade* e pelo *MQTTService*.

Por último, as classes *DadosCapaceteHub*, *ObrasHub* e *ManageNotificationClient* serão responsáveis por fornecer dados em tempo real ao frontend. Os Hubs deverão registar os clientes que pretendam receber dados em tempo real. O *ManageNotificationClient* irá conter as funções que permitirão notificar os clientes dos Hubs. Graças a estas classes, é possível implementar um padrão *Observer* que permite que o frontend receba dados em tempo real.

8 Diagrama de Componentes

Este diagrama serve para representar a estrutura e as relações entre os componentes do sistema. Assim sendo, conseguimos observar como os diferentes componentes e as suas interfaces estão organizados, bem como são estabelecidas as conexões entre os diferentes componentes.

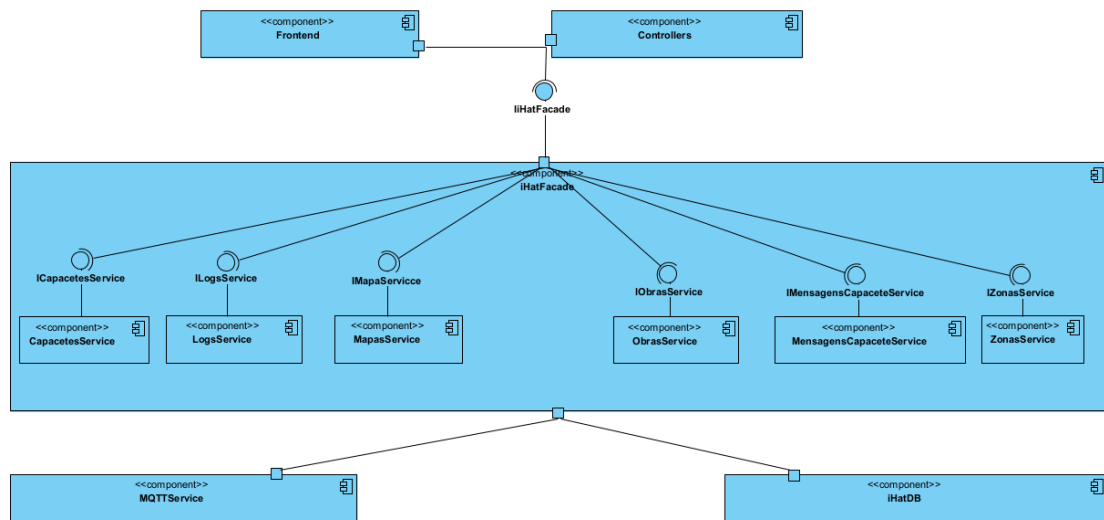


Figure 8.1: Diagrama de Componentes

9 Diagrama de Implementação

Os diagramas de implementação modelam a arquitetura física de um sistema e mostram os relacionamentos entre os diferentes componentes de software e hardware no sistema e a distribuição física do processamento.

Na Figura 9.1 , é apresentado o diagrama de implementação da aplicação desenvolvida.

Neste sistema, uma máquina do cliente irá aceder através de um *Web Browser* a um servidor de frontend. Os pedidos deste Web Browser serão realizados sobre HTTP. O frontend disponibilizará as páginas que o utilizador pretender consultar. Por sua vez, o servidor de frontend comunica com o servidor de backend também por HTTP através de uma API REST. Para alguns casos particulares, a comunicação será realizada através de HTTP Websockets, para que dados em tempo real possam ser apresentados aos utilizadores.

O servidor de backend comunica também com a base de dados, em MongoDB, através de drivers e com um serviço definido em python, que serve para tratar os ficheiros IFC recebidos. A máquina 'Capacete' da figura representa os Capacetes que existirão no sistema. Estes capacetes serão serviços externos a esta implementação. Estes deverão ser capacetes de obra com sensorização que permitem recolher alguns dados do trabalhador. Espera-se que capacetes interajam com o servidor de backend através do protocolo MQTT.

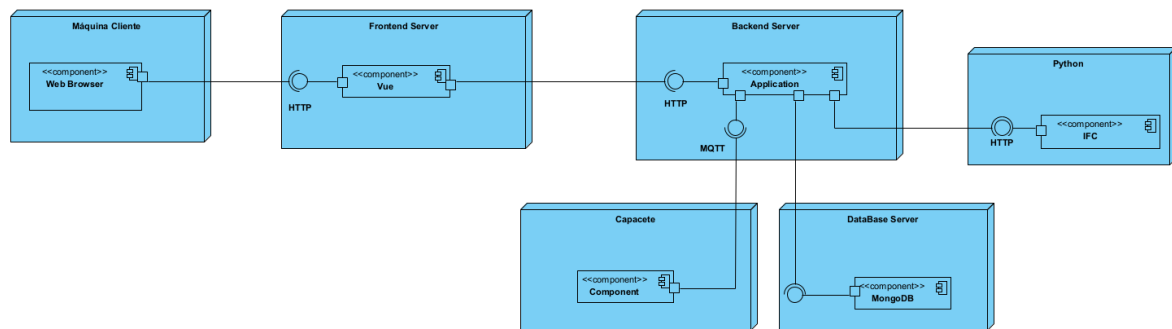


Figure 9.1: Diagrama de Implementação

10 Máquinas de Estados

As máquinas de estado proporcionam uma representação visual e lógica das transições de estados de um objeto. Estes diagramas facilitam a compreensão e análise do comportamento dinâmico de sistemas complexos.

Para este projeto, identificamos que a Obra e o Capacete seriam classes para as quais seria relevantes apresentar diagramas de Máquinas de Estados.

10.1 Máquina de Estados da Obra

Para este projeto, foram considerados diferentes estados para a obra que serão ilustrados na seguinte Máquina de Estados.

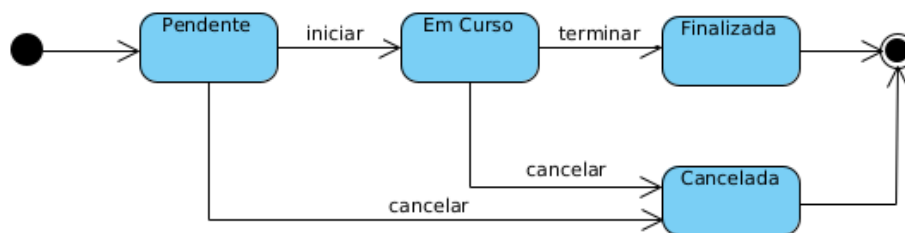


Figure 10.1: Diagrama de Máquina de Estado da Obra

De seguida serão enumeradas as operações que podem ser efetuadas quando uma obra se encontra em cada um dos diferentes estados possíveis.

Pendente

- Alterar o nome da obra
- Alterar o estado da obra para qualquer um dos estados, exceto Finalizada
- Alterar o mapa da obra
- Definir zonas de risco
- Adicionar capacetes
- Não é possível receber mensagem dos capacetes

Em curso

- Alterar o nome da obra
- Alterar o estado da obra para qualquer Finalizada ou Cancelada
- Alterar o mapa da obra
- Definir zonas de risco
- Adicionar capacetes
- Receber mensagem dos capacetes

Finalizada/Cancelada

- Não é possível alterar o nome da obra
- Não é possível alterar o mapa da obra
- Não é possível alterar o estado da obra
- Não é possível definir zonas de risco
- Todos os capacetes associados são "removidos" da lista de capacetes
- Não é possível receber mensagem dos capacetes

10.2 Máquina de Estado do Capacete

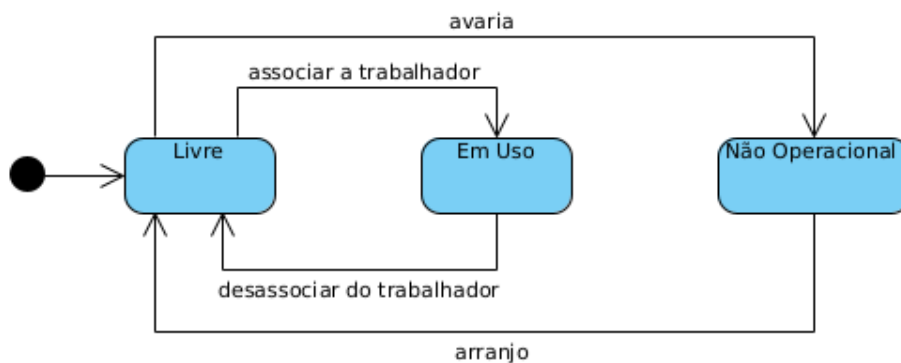


Figure 10.2: Diagrama de Máquina de Estado da Capacete

De seguida serão enumeradas as operações que podem ser efetuadas quando um capacete se encontra em cada um dos diferentes estados possíveis.

Livre

1. Primeiro Estado
2. Capacete pode ser associado a trabalhadores
3. Estado pode ser atualizado para não operacional, em uso (associado à obra)
4. Mensagens recebidas do capacete são ignoradas
5. Pode consultar o "histórico" (neste caso não seria Real-Time)

Em Uso

1. Capacete não pode ser alterado para livre de forma manual
2. Capacete não pode ser associado a trabalhadores, nem a obras
3. Capacete pode ser desassociado do (trabalhador e da obra atual)
4. Mensagens deste capacete serão tratadas
5. Pode consultar o "histórico" (neste caso vai ser Real-Time)

Não operacional

1. Capacete não pode ser associado a trabalhadores nem a obras (não pode passar para em uso)
2. Mensagens do capacete não serão tratadas
3. Estado pode ser atualizado para livre
4. Pode consultar o "histórico" (neste caso não seria Real-Time)

11 Diagrama de Atividades

Nesta secção apresentamos alguns diagramas de atividades que permitem representar de forma clara o fluxo de ações e decisões feitas para uma operação.

Neste documento, optamos por abordar apenas as atividades que consideramos que apresentam o fluxo mais complexo, visto que será nestas situações que estes diagramas de atividades serão mais úteis.

11.1 Tratar mensagens recebidas do Capacete

No projeto a desenvolver, é necessário que exista um mecanismo que seja capaz de receber mensagens do capacete. Nestas mensagens do capacete deverão vir informações recolhidas pelos sensores existentes num capacete "físico". O processo que deverá ser realizado para receber e tratar as mensagens recebidas são representados no seguinte diagrama de atividades.

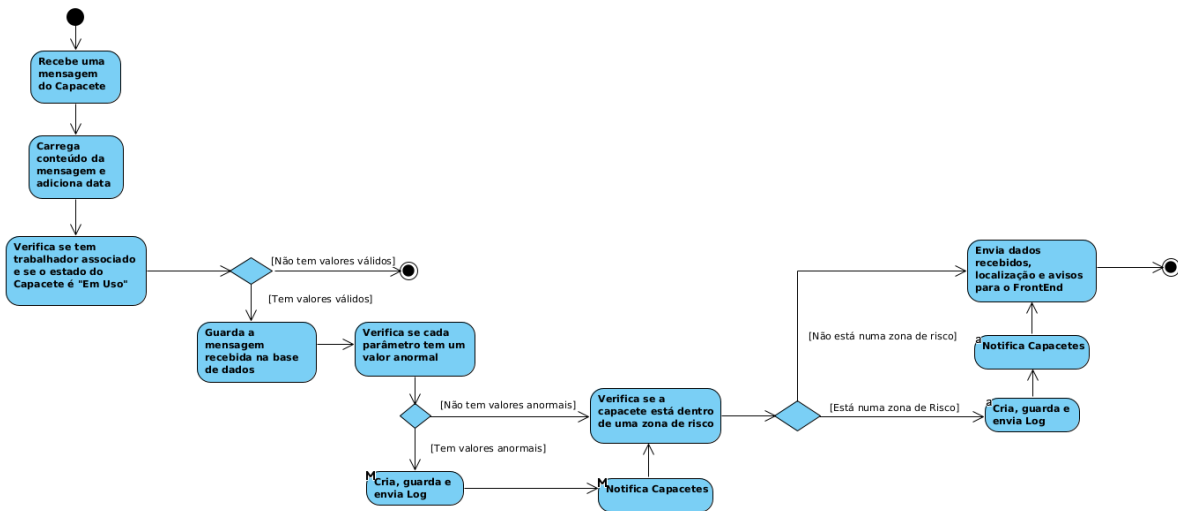


Figure 11.1: Diagrama de atividades relativo ao tratamento de mensagens recebidas do Capacete

11.2 Adicionar Mapa à Obra

Para um bom funcionamento do sistema é preciso garantir que o mapa da obra é adicionado corretamente. De forma a ser mais perceptível o processo de adicionar um mapa definimos o seguinte diagrama de atividades que demonstra a forma como o mesmo é adicionado e como o servidor de Backend comunica com o serviço *model2SVG*, serviço este definido em *Python* que permite o tratamento dos ficheiros IFC recebidos.

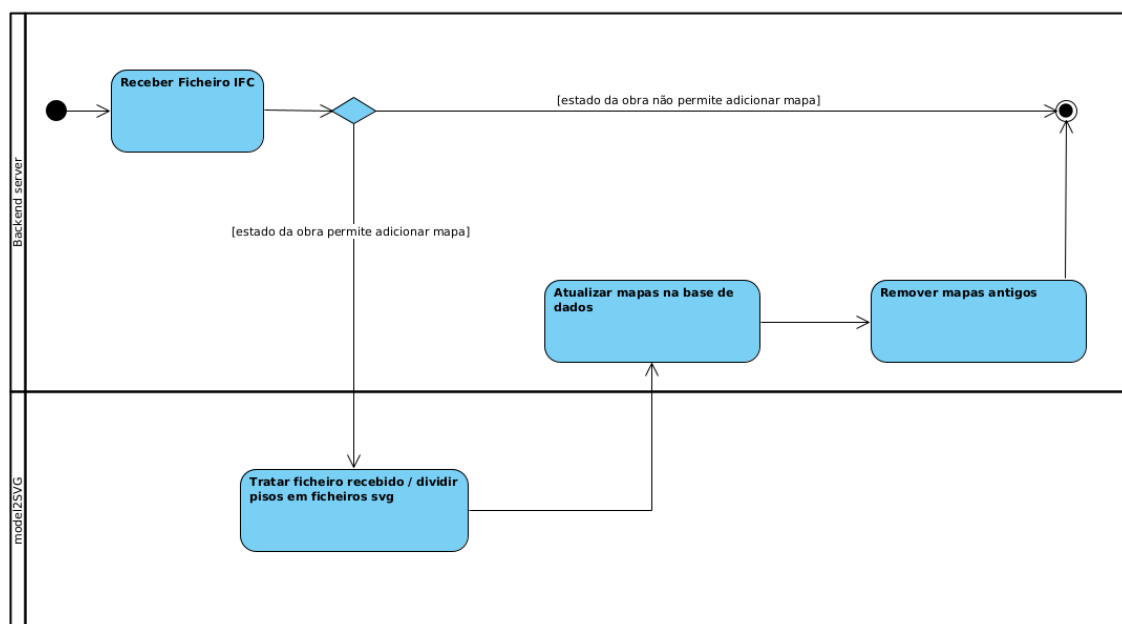


Figure 11.2: Diagrama de atividades relativo a adicionar um mapa a uma obra

12 Mockups

Nesta secção apresentamos visualizações gráficas que destacam a estrutura e o layout das páginas web propostas.

Estes mockups oferecem uma representação visual clara da disposição dos elementos, facilitando a compreensão da arquitetura da informação e da experiência do usuário. Cada mockup foi criado para ilustrar a organização dos componentes, orientando as decisões de design relacionadas à navegação, hierarquia de conteúdo e disposição estratégica.

Todas as páginas contêm uma app bar no topo da página que irá conter opções que serão acessíveis em qualquer página da aplicação.

12.1 Página Inicial

A mockup representada na Figura 12.1 ilustra o desenho inicial para a página inicial da interface da aplicação.

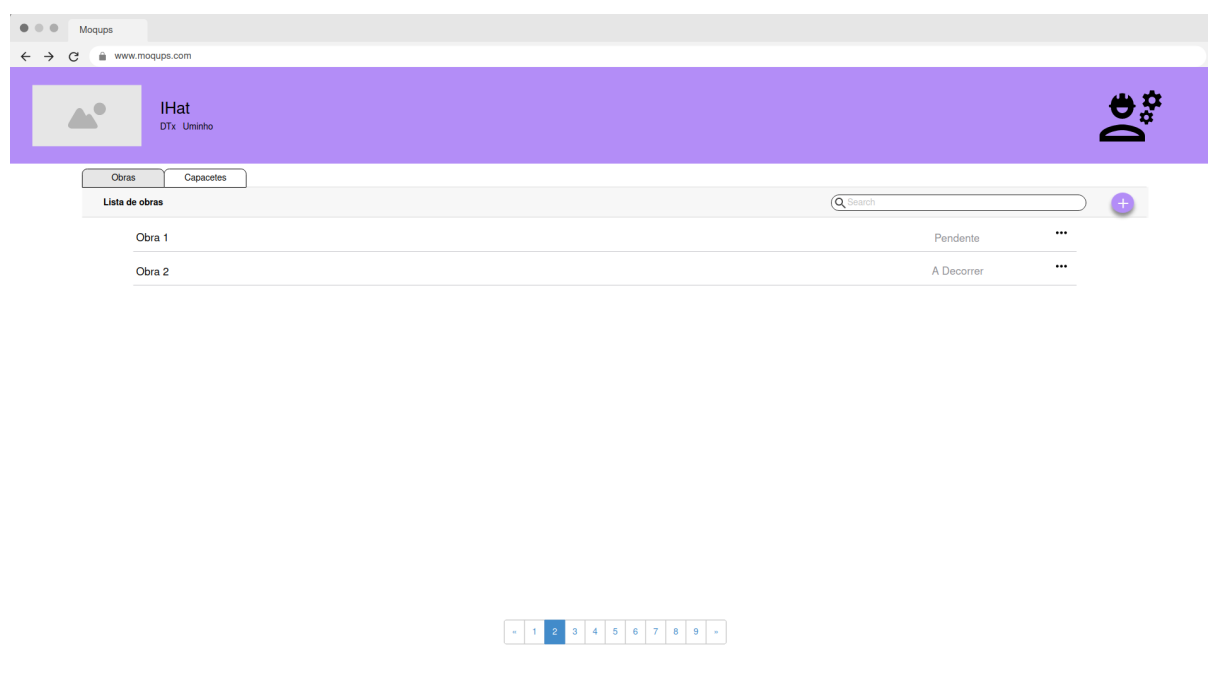


Figure 12.1: Mockup da Página Inicial

Iremos ter uma tabela para listar tanto os capacetes como as obras do sistema, esta escolha será feita através dos botões imediatamente acima da tabela que funcionaram como tabs.

A tabela terá ainda a opção de pesquisa e de adição de novas obras ou capacetes consoante a tab seleccionada, assim como paginação.

Cada elemento da tabela terá as informações relativamente ao elemento, o nome e o estado.

12.2 Página da Obra

A mockup representada na Figura 12.2 ilustra o desenho inicial para a página da obra da interface da aplicação.

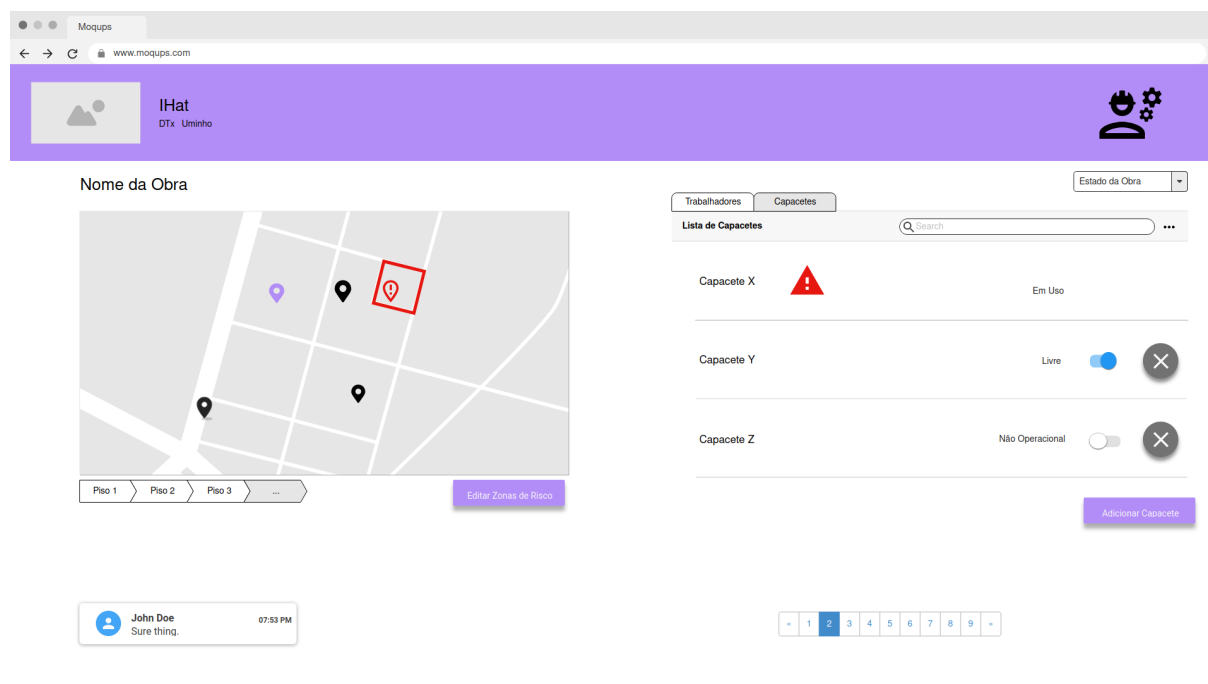


Figure 12.2: Mockup da Página da Obra

O desenho para a página da obra tem três secções:

- **Mapa**, imagem da planta de um piso para a visualização da posição dos capacetes e zonas de risco, com opção de selecionar outros pisos e de editar as zonas de risco do piso selecionado.
- **Lista dos Capacetes**, uma tabela semelhante à da página inicial com os capacetes pertencentes à obra, com opção de pesquisa, remoção, adição e alteração do estado do capacete
- **Logs**, irá conter os alertas gerados pelos capacetes dessa obra

12.3 Pagina do Capacete

A mockup representada na Figura 12.3 ilustra o desenho inicial para a página que irá apresentar os dados do capacete.

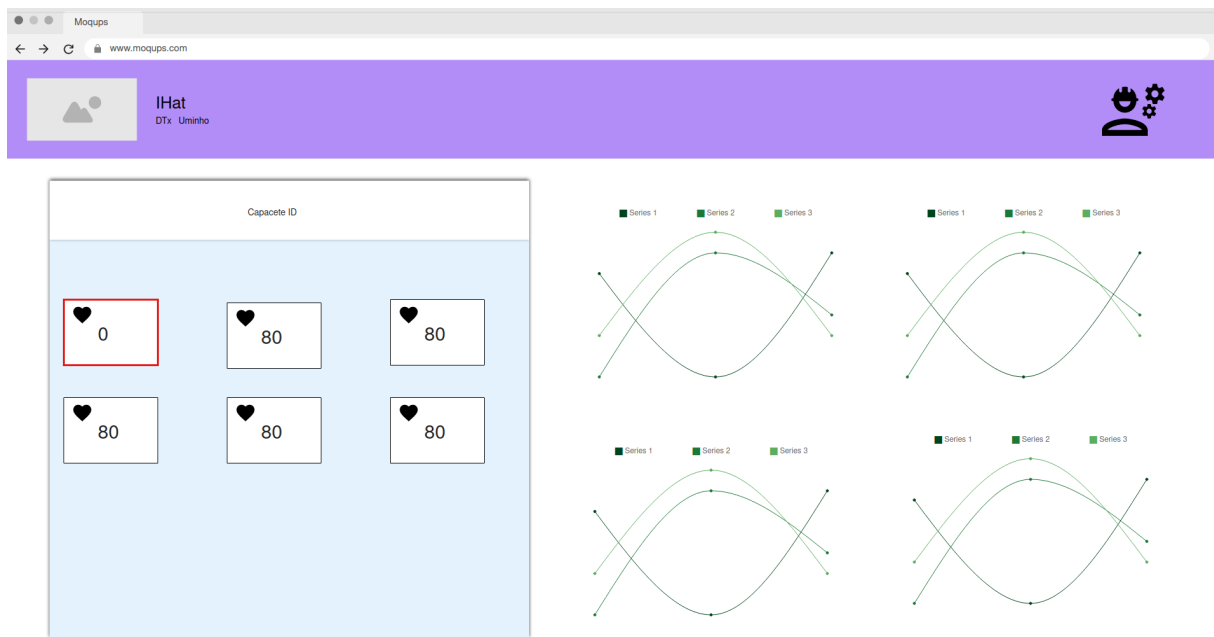


Figure 12.3: Mockup da Página de Capacete

O desenho para a página individual de cada capacete terá duas secções:

- **Informações em Tempo Real**, irá conter as últimas informações relativas ao sensores do capacete.
- **Gráficos**, para a visualização da variação das informações num período de tempo.

13 Database

O projeto IHat utiliza como sistema de gestão de base de dados, o MongoDB. Optamos por esta base de dados documental devido à familiaridade do grupo com a ferramenta e a sua adaptabilidade aos requisitos do projeto.

As coleções que foram criadas de forma a armazenar os dados necessários para o correto funcionamento do programa foram:

- **Obra**

Esta coleção irá armazenar as informações relativas a obras em curso, nomeadamente, *IdResponsavel* (identificador do responsável da obra), *Nome* (nome da obra), *Mapa* (informações do mapa), *Capacetes* (números dos capacetes associados) e *Estado* (estado da obra)

Id	IdResponsavel	Name	Mapa	Capacetes	Status
"001"	1	"Primeira Obra"	["001", "002"]	[]	"Pendente"
"002"	1	"Segunda Obra"	[]	[1,2]	"Em Curso"

Table 13.1: Coleção de Obras

- **Capacete**

Contém os detalhes sobre os capacetes. Isto é, *Numero* (identificador do capacete), *Estado* (estado de utilização), *Trabalhador* (id do trabalhador associado) e *Obra* (id da obra associada)

Id	NCapacete	Status	Trabalhador
"001"	1	"Em Uso"	"a01"
"002"	2	"Livre"	null

Table 13.2: Coleção de Capacetes

- **Logs**

Reservada para registo de eventos, está pronta para armazenar informações sobre os eventos do sistema. Nomeadamente, *Type* (tipo de gravidade do log), *Timestamp* (marca de tempo do log), *IdObra*, *IdCapacete*, *IdTrabalhador* que estão os ids associados ao log, *Mensagem* (mensagem do log) e *Vista* que indica se o log foi visualizado.

Id	Timestamp	IdObra	IdCapacete	IdTrabalhador	Mensagem
"001"	04/12/2023 12:00:00	"002"	1	"a01"	"Warning: Fall detected!"
"002"	04/12/2023 12:00:00	"002"	1	"a01"	"Warning: Unusual heartrate!"

Table 13.3: Coleção de Logs

- **Mensagens Capacete**

Coleção responsável por armazenar mensagens e dados enviados pelos capacetes. Apresenta *NCapacete* (número do capacete), *Tipo* (tipo de mensagem), *Queda* (in-

formação sobre queda), *TemperaturaCorporal* (temperatura corporal), *FrequenciaCardiaca* (frequência cardíaca), *Proximidade* (proximidade a zonas de risco), *Localizacao* (posição do capacete) e *Gases* (concentração dos diferentes gases).

BodyTemperature	Heartrate	Proximity	Location
36	120	10	(1,4,1)
0	0	20	(2,4,0)

Gases
{Metano: 0, Monoxido de Carbono: 2}
{Metano: 0, Monoxido de Carbono: 0}

Table 13.4: Coleção de mensagens dos capacetes

• Mapas

Projetada para armazenar informações relacionadas a mapas. Apresenta os seguintes campos, *Name* (nome do mapa), *Svg* (string para renderizar o mapa), *Floor* (pisos a que corresponde o mapa) e *Zonas* (que contém as zonas de risco, isto é, as coordenadas do mapa)

Id	Name	Svg	Zonas
"001"	"Level 1"	<xml.../>	[...]
"002"	"Level 2"	<xml.../>	[]

Table 13.5: Coleção de Mapas

Estas coleções permitem armazenar as diferentes informações relativas aos dados do nosso programa. No código é feita a conexão ao localhost onde implementamos a base de dados e onde iremos procurar e armazenar as diferentes informações. Foi criada uma script que cria as coleções necessárias e insere dados iniciais, estabelecendo a base para o desenvolvimento e testes do projeto.

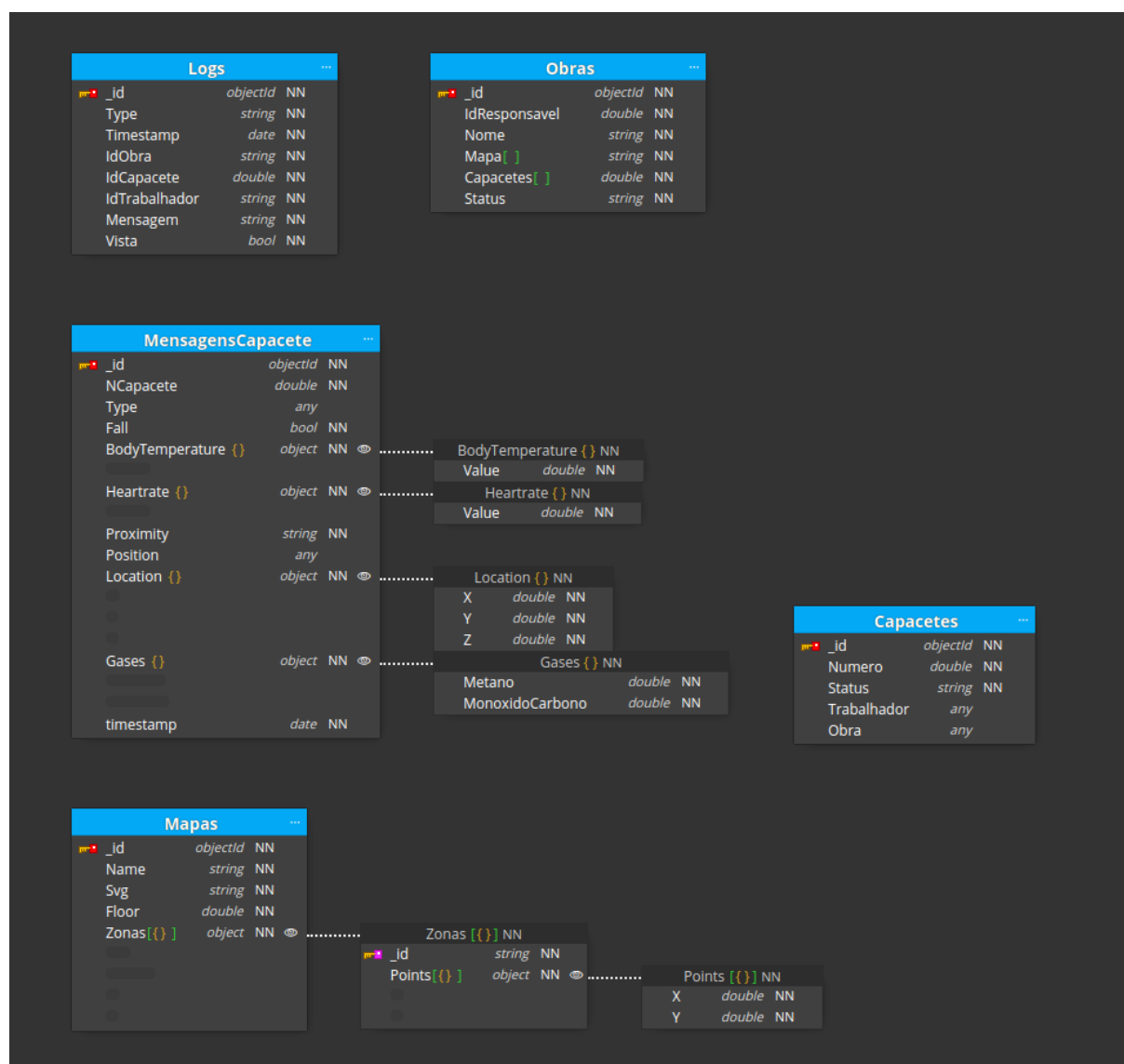


Figure 13.1: Modelo Lógico

14 Formato das Mensagens MQTT

O protocolo MQTT segue um modelo cliente-servidor. Os dispositivos que pretendem trocar mensagens através do protocolo MQTT devem conectar-se a um servidor centralizado, o "broker". Os clientes devem publicar mensagens para um tópico, e subscrever a um tópico para receber mensagens enviadas para este. O "broker" ficará responsável por enviar as mensagens de cada tópico para os clientes inscritos no mesmo.

O serviço será também capaz de receber e validar mensagens MQTT que informem de uma associação (ou desassociação) de um capacete com um trabalhador.

Espera-se que sejam trocadas mensagens em três diferentes tópicos, que são apresentados de seguida:

- **my/topic** O backend subscreve a este tópico para poder receber mensagens enviadas pelos capacetes, que contêm os dados que estes recolhem. O formato da mensagem recebida deve ser:

```
{
  { "HelmetNB", numero_capacete },
  { "TypeMessage", "ValueUpdate" },
  { "Fall", false },
  { "BodyTemperature", 38 },
  { "Heartrate", 100 },
  { "Proximity", "10" },
  { "Position", "10" },
  { "Location", { { "X", 0 }, { "Y", 0 }, { "Z", 0 } } },
  { "Gases", { { "Metano", 0 }, { "MonoxidoCarbono", 0 } } }
};
```

- **my/topic/{numero capacete}** O backend deverá enviar uma mensagem para este tópico quando detetar valores anormais numa mensagem do capacete {numero capacete}. A mensagem enviada para este tópico tem como objetivo alertar o capacete. O formato das mensagens enviadas para este tópico é:

```
{
  { "Notify", true }
};
```

- **ihat/obras** O backend subscreve a este tópico para receber as mensagens de associação e desassociação de um capacete a um trabalhador. O tipo da mensagem de associação deve ser "Pairing" e o da mensagem de desassociação deve ser "Disconnect". O formato das mensagens recebidas deve ser:

```
{
  { "type", "Pairing" },
  { "numero", numero_capacete },
  { "idTrabalhador", trabalhador_id },
  { "obra", obra_id }
```

```
};  
{  
    { "type", "Disconnect" },  
    { "numero", numero_capacete },  
    { "idTrabalhador", trabalhador_id },  
    { "obra", obra_id }  
};
```

15 Conclusões

15.1 Resultados Obtidos

Como resultados obtidos conseguimos desenvolver uma plataforma web que cumpre integralmente com todas as funcionalidades planeadas.

Com efeito, atendemos todos os requisitos esperados. Nos funcionais, destacam-se a receção de notificações no sistema nos casos em que o capacete recebe anomalias e a delimitação de zonas de perigo num mapa de obra. Já nos requisitos não funcionais é evidente a fácil usabilidade do programa devido à implementação de uma interface simples e dinâmica.

Além disso, foi ainda possível complementar o trabalho pretendido com um simulador. Este extra serve para simular um cenário real, como a movimentação de trabalhadores dentro da obra. Deste modo é possível ter uma boa perceção de como o sistema funciona com a variação constante das posições dos trabalhadores e com a variação dos diferentes parâmetros que serão recolhidos por um capacete. Assim, este simulador permite observar os avisos que a plataforma recebe quando determinadas ações consideradas anormais ocorrem, como a deslocação de um trabalhador numa zonas de perigo, por exemplo.

15.2 Trabalho Futuro

Apesar deste projeto ser um grande passo para o aumento da segurança na construção civil, ainda existem possibilidades de o tornar mais eficiente.

O código atual já se encontra bem modularizado em componentes, o que é essencial para garantir que o sistema tenha fácil manutenção e escalabilidade. No entanto, para trabalho futuro pretendemos realizar uma integração completa do projeto em *containers Docker* para permitir uma maior portabilidade, escalabilidade e uma gestão mais simples do desenvolvimento do projeto.

Além disso, pretendemos também introduzir um sistema de autenticação e melhorar a gestão de permissões na aplicação web. Na solução atual não estamos a gerir os utilizadores, visto que estes se encontram numa *active directory* que é da empresa e, portanto, não temos acesso a essa informação. Dito isto, na página atual, o utilizador terá acesso a todas as permissões.

Dito isto, a implementação de um sistema de *login* é crucial para que os utilizadores se autenticuem de forma segura, o que garante que todos sejam facilmente identificáveis. Por fim, pretendemos adicionar o cargo de administrador ao sistema para algumas entidades possam ter controlo sobre funcionalidades relacionadas à configuração e administração do sistema.