

Technical Report - **Product specification**

SmartHome Manager

Disciplina: IES - Introdução à Engenharia de Software

Data: Aveiro, 20 de novembro de 2022

Estudantes: 102477: Artur Afonso Ferra Correia
103453: Bruna de Melo Simões
77036: Daniel Luís Fernandes Carvalho
103925: Diogo Alves e Silva

Abstrato do Projeto: Aplicação web que permite o controlo de uma smart home e todas as suas funcionalidades (tais como regulamento de temperatura, bloqueio e desbloqueio de portas, níveis de monóxido de carbono, ligar e desligar luzes em várias divisões da casa, etc) de modo a aumentar a confortabilidade quando o utilizador quer dentro ou fora de casa. A aplicação web deve ser acessível a todas as pessoas que vivem/gerem a casa, e, dependendo da sua posição relativamente à mesma, podem ter mais ou menos privilégios associados (por exemplo, uma empregada doméstica não poderá ter tantos privilégios como os donos da mesma).

Conteúdo

1 Introdução.....	2
Equipa	2
2 Conceito do produto.....	2
Visão	2
Personas e motivações	3
Cenários Principais.....	3
3 Notas de Arquitetura	4
Requisitos e Restrições Chave.....	4
Vista da Arquitetura	4
Interações de Módulos	5
4 Perspetiva de Informação	7
5 Referências e Recursos	9

1 Introdução

Neste projeto será implementada uma aplicação web para a gestão e monitorização de dispositivos e sensores numa casa inteligente. A associação de diferentes dispositivos à aplicação permitirá o rastreio de várias condições presentes nas diferentes divisões da casa, tais como, indicação da temperatura, indicação da humidade ou a deteção da qualidade do ar, bem como a gestão do gasto de recursos como a energia e a água, e o controlo de dispositivos ligados à smart home (luzes, aquecedores, medidor de monóxido de carbono, entre outros). O administrador da smart home pode conceder privilégios sobre a sua casa a quem quiser, bem como definir os mesmos e a quem os quer atribuir.

Assim, o desenvolvimento deste projeto vai permitir a aplicação de vários conceitos e tecnologias aprendidos ao longo desta disciplina, desde logo a utilização de uma REST API, desenvolvida na framework Spring, para gerir o backend da aplicação e permitir a interação entre a obtenção e processamento dos dados dos sensores, a base de dados persistente e o frontend do sistema. Serão também utilizados e desenvolvidos alguns conceitos de desenvolvimento agile, com a utilização de ferramentas de gestão de backlog, conceitos de desenvolvimento iterativo e contínuo, através da utilização de GitHub, entre outros.

Equipa

- **Team Manager:** Daniel Carvalho
- **Product Owner:** Diogo Alves
- **Architect:** Bruna Simões
- **DevOps Manager:** Artur Correia

2 Conceito do produto

Visão

O produto desenvolvido tem como função primária a criação de uma plataforma que permite registar e gerir todos os sensores/dispositivos do tipo IoT presentes numa casa inteligente, deixando que utilizadores autorizados tenham acesso aos dados atuais e históricos de condições ambientais monitorizadas por estes sensores, bem como tenham a possibilidade de realizar certas ações com base na informação apresentada pela plataforma.

Deste modo, serão registadas para cada divisão da casa os valores de condições tais como a temperatura, a humidade ou a qualidade do ar (por exemplo em termos de níveis de monóxido de carbono, ou de pólen). A plataforma permitirá também a geração de alertas e notificações quando for verificados valores perigosos de qualquer um destes valores. Em termos de ações permitidas, a aplicação poderá, por exemplo, permitir aos utilizadores ligarem/desligarem o ar condicionado de acordo com os valores de temperaturas medidos, controlar o estado de luzes presentes nas diferentes divisões, ou até fechar/abrir remotamente janelas na casa (que poderá ocorrer automaticamente, por exemplo, se forem detetados níveis perigosos de monóxido de carbono).

Simultaneamente, a plataforma permitirá também um controlo dos níveis de energia, água ou gás consumidos, podendo gerar alertas caso um limite imposto pelo utilizador seja atingido.

Personas e motivações

Nome: Jaime Soares

O Jaime, de 37 anos de idade, trabalha como gestor comercial numa multinacional tecnológica. Ele é casado e tem 3 filhos, vivendo com a sua família numa casa que comprou recentemente. Após esta compra, o Jaime decidiu instalar vários sensores e dispositivos inteligentes, que monitorizam diferentes condições da sua casa, e realizam certas ações em resposta às medições efetuadas.

Devido ao seu trabalho, o Jaime viaja bastante, passando muito tempo fora de casa. Assim, ele gostaria de ter uma plataforma onde poderia verificar remotamente o estado da sua casa, por exemplo a qualidade do ar, visto que um dos seus filhos sofre de alergias, bem como os níveis de eletricidade ou gás consumidos, de maneira a garantir que os valores se mantêm dentro do orçamento definido para o mês.

Nome: Eduarda Santos

A Eduarda, de 45 anos, trabalha como empregada doméstica numa moradia inteligente. Durante o seu horário de trabalho, a Eduarda encontra-se sozinha, uma vez que os habitantes da moradia estão também eles no seu local de trabalho, ou nas suas escolas. Deste modo, a Eduarda precisa de acesso à plataforma que gere as funcionalidades inteligentes da moradia.

Cenários Principais

- Após a compra da nova casa, o Jaime verificou que algum do equipamento existente não está nas melhores condições – por exemplo, o sistema central de aquecimento não funciona corretamente, podendo ter algumas fugas. Assim, o Jaime instalou sensores de monóxido de carbono pela casa, querendo utilizar a nossa plataforma para registar os valores deste gás medidos ao longo do dia, sendo notificado caso algum destes valores seja perigoso. Além disso, quer definir que, se a concentração numa dada divisão for demasiado elevada, as janelas desta divisão deverão ser abertas.
- O Jaime pretende manter um olho sobre o consumo de energia, de água e de gás na sua casa, enquanto está fora em viagem. Assim, ele utiliza a nossa plataforma para verificar os dados de consumo destes recursos.
- A Eduarda encontra-se sozinha a limpar a sala da sua moradia. Ela repara que está demasiado quente, e pretende aceder à aplicação para controlar baixar a temperatura através do ar condicionado.
- O Jaime tem um filho que sofre de várias alergias e doenças respiratórias. Assim, o Jaime quer instalar e monitorizar sensores de qualidade no ar no quarto deste filho, definido a notificação de alertas caso estes valores fujam do normal.
- A Eduarda quer definir os horários de ativação dos regadores da moradia onde trabalha, de acordo com os dados de medição da temperatura externa, registada nos jardins desta.

3 Notas de Arquitetura

Requisitos e Restrições Chave

- Os utilizadores necessitam de uma ligação à internet, devendo aceder ao website do sistema.
- Para se registarem na plataforma, os utilizadores necessitam de introduzir parâmetros tais como o seu email, nome e password. Caso já tenha sido registado, tem de efetuar o seu login.
- O utilizador deve introduzir quais as divisões que tem, bem como registar todos os sensores para cada divisão da sua casa, sendo que a qualquer altura pode editar estas informações.
- O sistema deve receber e enviar valores de forma automática, sendo que os mesmos são gerados através de sensores virtuais que simulam as condições de temperatura, qualidade do ar e humidade para cada divisão da casa.
- Após deteção de valores excessivamente elevados/baixos, serão executados determinados protocolos automaticamente de modo a regular esses valores para a normalidade, bem como será enviado um alerta para notificar o utilizador, na aplicação.
- O backend funciona como um middleware, que estabelece uma ligação entre os sensores/dispositivos conectados e o frontend.
- O sistema deve armazenar os valores medidos, de modo a providenciar ao utilizador um histórico de registos acessível ao mesmo.
- O sistema deve permitir ao utilizador definir parâmetros personalizados para reagir mediante os dados medidos no sensor (por exemplo, ligar/desligar AC de acordo com temperaturas definidas, definir horários/temperaturas para os regadores do exterior se ativarem, entre outros)
- Não deve ser permitido acesso de terceiros a dados confidenciais, pelo que a informação acessível pela REST API deve ser protegida, de forma que apenas utilizadores autenticados consigam contactar os REST endpoints desta.
- O frontend deve ser reativo e responder a alterações da informação recebida do sistema.

Vista da Arquitetura

A arquitetura será composta por 4 camadas principais: a geração de dados e o seu envio para uma *message queue*, o *backend* que envolve o processamento lógico dos dados, a lógica de negócio, a REST API e o modelo de dados, o *frontend* e a camada de persistência.

A geração de dados será feita com recurso a sensores virtuais, escritos em linguagem Python, que simulam sensores do mundo real. A informação gerada será enviada através de um sistema de fila de mensagens para o *backend*, utilizando como *message broker* o RabbitMQ. Neste, deverão ser criados tópicos para a receção de dados de cada sensor desenvolvido.

Por sua vez, *backend* será construído e implementado na linguagem Java, utilizando a Spring *framework*. O *backend* vai possuir uma camada de processamento dos dados recebidos através do *message broker*, após o qual os dados serão enviados para a camada de persistência, para os mesmos serem armazenados. Esta camada de persistência será desenvolvida através do modelo relacional, sendo implementada com recurso ao MySQL.

A interação com a base de dados é feita através do Spring Data (*object-database mapping framework*). No entanto, se aquando do processamento dos dados recebidos se detetar valores

anormais ou perigosos para as condições medidas, deverá ser enviado uma notificação de alerta diretamente para a camada do *frontend*, através de websockets. Assim, a camada de processamento destes dados poderá envolver também scripts Python.

O *backend* engloba também a criação de uma REST API para a interação com o *frontend* (client-side), em Spring Boot. Deverão ser desenvolvidos mecanismos de autenticação e autorização utilizados para proteger o acesso aos métodos desenvolvidos para a API, de forma a que apenas utilizadores registados possam ter acesso à plataforma, e de maneira a poder distinguir entre os privilégios dos diferentes utilizadores.

Por fim, o *frontend* será implementado utilizando a biblioteca ReactJS, que permite desenhar a aplicação de acordo com o modelo SPA, cuja estrutura deverá ser alterada conforme a atualização de dados (isto é, a plataforma deve ser responsiva).

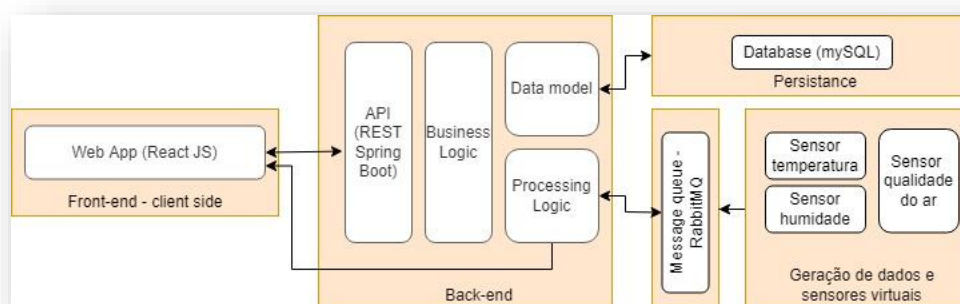


Figura 1 – Diagrama de arquitetura para a aplicação SmartHome Manager.

Interações de Módulos

Os diferentes módulos comunicam constantemente entre si, de modo a suportar ao utilizador toda a informação em tempo real e atualizada.

O primeiro diagrama retrata os acontecimentos sucedidos após um dado utilizador consultar o histórico de temperaturas registadas num dado intervalo de tempo.

A aplicação deve sempre ter dados atualizados da temperatura, medidos constantemente pelo sensor, que envia tais dados para o MessageBroker. De seguida, serão processados os tais dados e enviados para a base de dados que os armazena, com a sua referência temporal.

Assim sendo, quando o utilizador consultar o histórico na aplicação web, esta última contacta a API, que irá pedir a informação ao data access, que obtém os dados da base de dados, registados anteriormente na mesma. Quando o data access recebe os dados, envia os mesmos para serem processados, sendo recebidos pela API, que se encarrega de enviar os dados para a responsive app, que os demonstra ao utilizador.

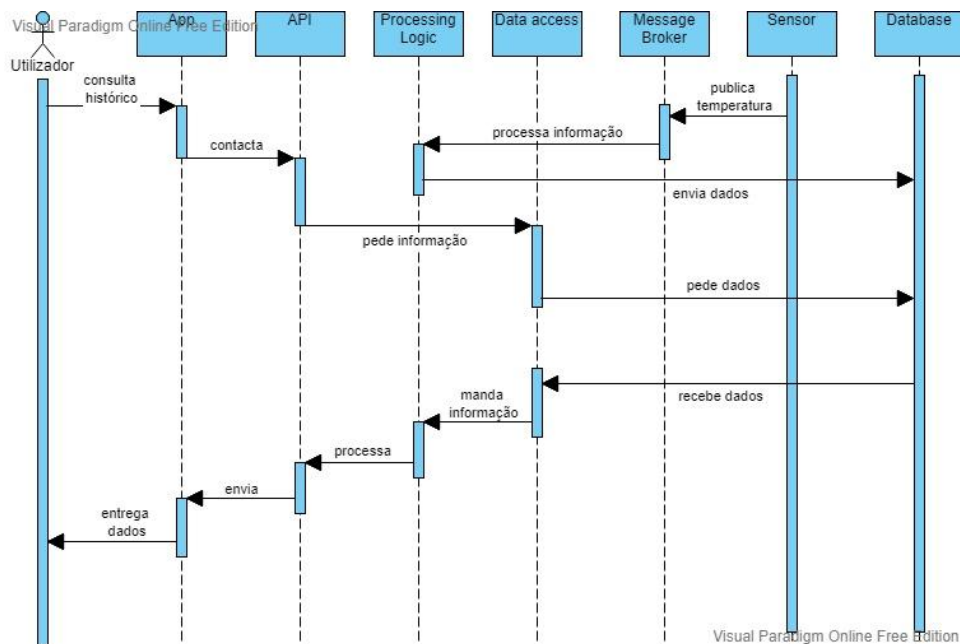


Figura 2 – Diagrama de interação que representa uma ação de um utilizador para acesso a uma funcionalidade do sistema, neste caso, o pedido de consulta do histórico de temperaturas para uma dada divisão.

Já neste segundo diagrama é retratado o evento em que um determinado sensor deteta um nível anormal de um gás tóxico.

Neste caso, o sensor envia, como de costume, os dados que recebe, para o message broker, que processa a informação no processing logic e, enviando os dados para a base de dados para serem gravados (e potencialmente mais tarde, acedidos), verifica a anormalidade dos seus valores, enviando por web socket para a aplicação, que alerta o utilizador da informação recebida.

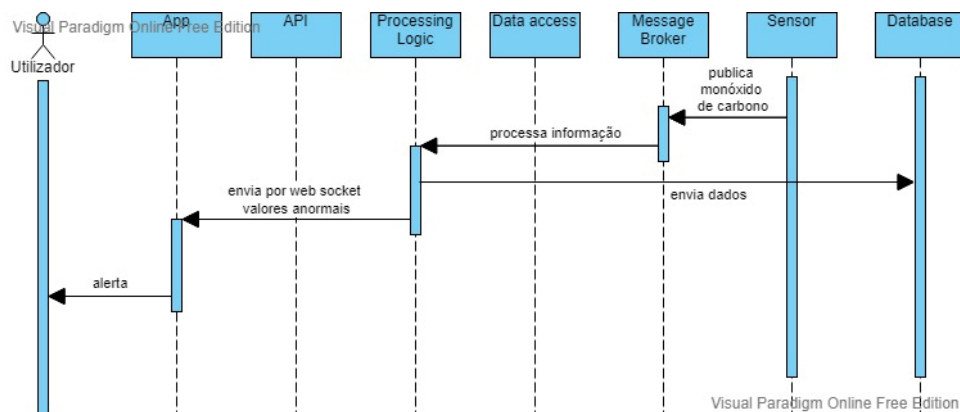


Figura 3 – Diagrama de interação que representa o envio de um alerta ao utilizador, após processamento de um valor anómalo de monóxido de carbono num sensor registado na plataforma.

4 Perspetiva de Informação

Para representar as classes definidas e as suas relações foi desenhado um diagrama do domínio.

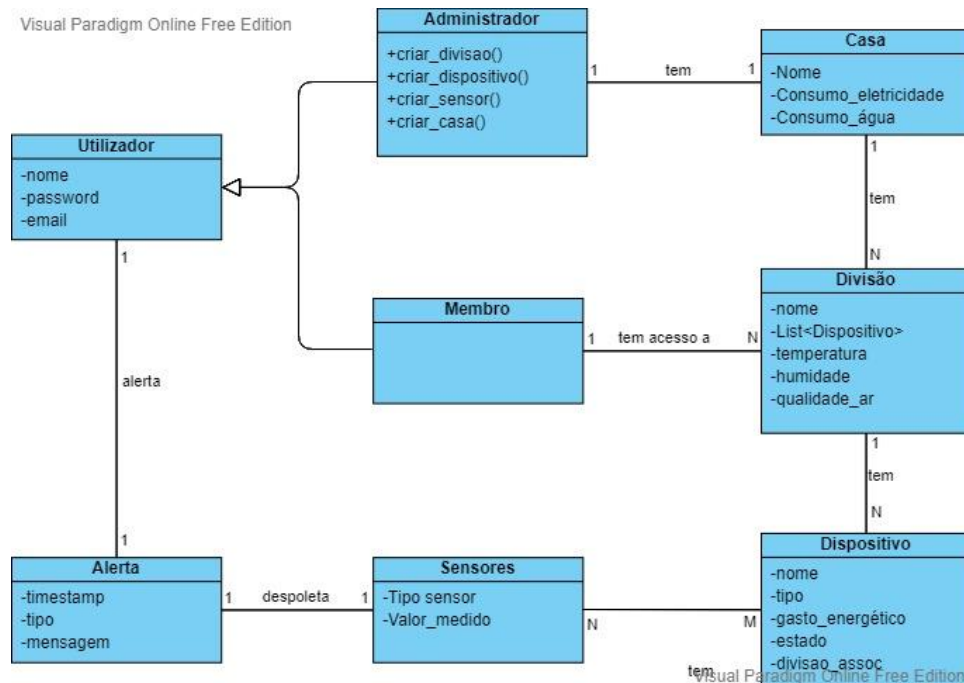


Figura 4 – Diagrama de classes UML que representa as entidades do domínio da aplicação desenvolvida.

Após a análise de requisitos foram definidas as seguintes entidades:

Utilizador

O utilizador é quem interage com a aplicação e efetua a maior parte das ações, como por exemplo associar dispositivos, programar temporizadores, etc. Para além disso este também se subdivide em dois tipos, que serão explicados seguidamente.

Administrador

O administrador é o utilizador com mais permissões. É o responsável por criar uma casa e gerir as suas divisões. Pode adicionar, remover ou editar divisões, e os dispositivos que lhe estão associados. Além de ter acesso a todas as restantes funcionalidades permitidas a qualquer membro.

Membro

Um membro é um utilizador com acesso restrito à aplicação, ou seja, apenas poderá efetuar as ações permitidas pelo administrador. Por exemplo, um utilizador apenas poderá ligar ou desligar dispositivos se tiver o acesso necessário a eles.

Casa

A casa é o elemento principal da aplicação, incluirá várias divisões associadas a ela pelo administrador e também terá estatísticas gerais dessas divisões.

Divisão

Uma divisão é uma entidade que faz parte da casa, possuirá uma lista de dispositivos associados pelo administrador e outros dispositivos especiais, os sensores, que forneceram dados úteis sobre a divisão, temperatura, humidade, etc.

Dispositivo

Um dispositivo pode ser várias coisas, um eletrodoméstico ou uma janela, que será associado a uma divisão pelo administrador. Esse dispositivo terá um gasto energético e um estado (ligado, desligado), estado esse que poderá ser alterado por um utilizador.

Sensor

Dispositivo responsável por detetar e enviar periodicamente os valores de humidade/temperatura/qualidade do ar. No caso, iremos simular os sensores através de geradores, que criam periodicamente valores pseudoaleatórios realistas.

Alerta

O alerta é uma entidade especial no sentido que será gerado na parte de processamento caso exista uma avaliação anómala de um determinado dado, temperatura elevada, níveis de CO elevados, etc. Esse alerta poderá depender dependendo da sua magnitude, por isso tendo um tipo e serve para informar ao utilizador de alguma informação pertinente através do envio de uma mensagem.

5 Referências e Recursos

ReactJS

<https://reactjs.org/docs/getting-started.html>

RabbitMQ

<https://www.rabbitmq.com/documentation.html>

Spring Boot

<https://spring.io/projects/spring-boot>

MySQL

<https://dev.mysql.com/doc/>