

# PROJETO TEMÁTICO EM DESENVOLVIMENTO DE APLICAÇÕES

## SISTEMA DE GESTÃO PARA CASAS INTELIGENTES

### Autores:

Ana Castro (Nº 87908)

Hugo Neves (Nº 88167)

João Santos (Nº 88007)

Rafael Faustino (Nº 80914)

Tiago Silva (Nº 87913)

# PROJETO TEMÁTICO EM DESENVOLVIMENTO DE APLICAÇÕES

## SISTEMA DE GESTÃO PARA CASAS INTELIGENTES

### Autores:

Ana Castro (Nº 87908)

---

Hugo Neves (Nº 88167)

---

João Santos (Nº 88007)

---

Rafael Faustino (Nº 80914)

---

Tiago Silva (Nº 87913)

---

### Orientador:

Fábio Marques

## Resumo

O presente relatório foi elaborado como parte integrante do Projeto Temático em Desenvolvimento de Aplicações, associado às unidades curriculares Engenharia de Software e Sistemas de Bases de Dados, constituindo assim o Módulo Temático em Desenvolvimento de Aplicações, orientado no âmbito da Licenciatura em Tecnologias da Informação da Escola Superior de Tecnologia e Gestão de Águeda. O projeto foi encaminhado pelo docente Fábio Marques durante o primeiro semestre do segundo ano da licenciatura.

Pretendia-se com este projeto analisar os requisitos do sistema e construir o seu modelo concetual, desenvolver um modelo de desenho correspondente, incluindo o esquema relacional da base de dados e o modelo de desenho orientado a objetos da aplicação; implementar bases de dados relacionais num sistema de gestão de bases de dados cliente-servidor, utilizando a linguagem SQL para definir e manipular bases de dados; implementar uma aplicação a instalar em cada cliente com recurso a uma linguagem de programação orientada a objetos (Java), incluindo acesso a bases de dados; e, por fim, adotar uma estratégia de teste e especificar os respetivos casos de teste.

Numa fase inicial descreveu-se o âmbito do problema, baseado no tema escolhido, fez-se o planeamento do projeto, identificou-se e caracterizou-se os requisitos necessários, funcionais, não funcionais e de hardware a ser implementados, bem como os utilizadores do sistema. Posteriormente seguiu-se o desenho do sistema, onde se identificou e descreveu os casos de utilização, identificou e caracterizou todos os *packages*, classes/interfaces, as suas associações, atributos e métodos, e se produziu o modelo de dados persistente. Por fim, na fase de implementação e testes, implementou-se a solução

O presente trabalho culmina com uma reflexão final acerca deste percurso ao longo do desenvolvimento do sistema de gestão para casas inteligentes, bem como algumas sugestões a colmatar lacunas do mesmo.

## ÍNDICE

1. Introdução .....	1
1.1. Visão geral do sistema .....	1
1.2. Cliente.....	1
1.3. Objetivos.....	2
2. Planificação do trabalho .....	3
2.1. Calendarização das tarefas Previsto .....	3
2.2. Calendarização das tarefas executado .....	4
2.3. Requisitos Funcionais .....	4
2.4. Restrições e requisitos não funcionais .....	6
2.4.1. Requisitos de interface e facilidade de uso .....	7
2.4.2. Requisitos de interface com sistemas externos e com ambientes de execução .	7
2.5. Requisitos de hardware .....	8
3. Modelo de Casos de Utilização .....	11
3.1. Visão Geral.....	11
3.2. Atores .....	12
3.3. Descrição dos casos de utilização.....	12
3.3.1. Caso de Utilização #1 .....	12
3.3.2. Caso de Utilização #2 .....	13
3.3.3. Caso de Utilização #3 .....	13
3.3.4. Caso de Utilização #4 .....	14
3.3.5. Caso de Utilização #5 .....	15
3.3.6. Caso de Utilização #6 .....	15
3.3.7. Caso de Utilização #7 .....	16
3.3.8. Caso de Utilização #8 .....	16
3.3.9. Caso de Utilização #9 .....	17
3.3.10. Caso de Utilização #10 .....	18
3.3.11. Caso de Utilização #11 .....	18
3.3.12. Caso de Utilização #12 .....	19
3.3.13. Caso de Utilização #13 .....	19
4. Descrição das Atividades Desenvolvidas .....	21
4.1 Desenvolvimento da Aplicação .....	21

4.2 Diagrama de Classes .....	22
4.3 Modelo de Dados Persistente .....	24
4.4 Interface .....	25
5. Análise de REsultados .....	31
6. Conclusão.....	33
7. Bibliografia .....	35

## ÍNDICE DE TABELAS

Tabela 1 - Requisitos funcionais .....	5
Tabela 2 - Requisitos de interface e usabilidade.....	7
Tabela 3 - Requisitos de interface com sistemas externos e com ambientes de execução .....	8
Tabela 4 - Atores do diagrama de casos de uso .....	12
Tabela 5 - Visualizar os consumos de água, gás e luz dos aparelhos .....	12
Tabela 6 - Mudar estado de dispositivos e equipamentos .....	13
Tabela 7 - Visualizar os alertas .....	14
Tabela 8 - Mudar estado de portas, portões e janelas .....	14
Tabela 9 - Controlar iluminação .....	15
Tabela 10 - Ativar funcionalidades na deteção de utilizadores próximos.....	15
Tabela 11 - Gerir aspirador robot.....	16
Tabela 12 - Gerir e controlar o sistema de irrigação .....	17
Tabela 13 - Executar rotinas .....	17
Tabela 14 - Ativar zonas .....	18
Tabela 15 - Criar zonas .....	18
Tabela 16 - Criar rotinas .....	19
Tabela 17 - Gerir e controlar alarmes .....	19

## ÍNDICE DE FIGURAS

Figura 1 - Calendarização das tarefas previsto.....	3
Figura 2 - Calendarização das tarefas executado .....	4
Figura 3 - Diagrama de casos de uso.....	11
Figura 4 - Diagrama de Classes.....	23
Figura 5 - Modelo de Dados Persistente .....	24
Figura 6 - Janela de Autenticação .....	25
Figura 7 - Janela principal.....	26
Figura 8 - Janela das informações da sala .....	26
Figura 9 - Janela dos dispositivos da sala .....	27
Figura 10 - Janela de aberturas da sala .....	27
Figura 11 - Janela de zonas da sala .....	28
Figura 12 - Janela de editar zonas da sala.....	28
Figura 13 - Janela de gestão dos utilizadores.....	29

# 1. INTRODUÇÃO

## 1.1. VISÃO GERAL DO SISTEMA

Estando a tecnologia cada vez mais presente no nosso quotidiano, é normal que o mercado de automatização esteja em crescimento/desenvolvimento, de forma a simplificar a realização de certas tarefas por parte do ser humano. Como tal, este processo tem sido incorporado em ambientes domésticos, procurando aumentar o nível de comodidade e o tempo para descanso.

O sistema de gestão para casas inteligentes a ser implementado tem como objetivos facilitar o controlo dos equipamentos e otimizar a utilização de recursos, economizando-os, através da automatização da casa, tendo sempre em conta o conforto proporcionado.

O sistema deve centralizar todos os controlos da residência, como por exemplo a eletricidade, a água, o gás, as cortinas e estores, o aquecimento, o sistema de alarme, entre outras funcionalidades. Também deve realizar a automação de algumas tarefas: no caso de um utilizador entrar em casa, as luzes estão programadas para acender automaticamente, caso esteja de noite; ou quando se pretende ver um filme, as luzes da sala são desligadas, o projetor ligado e é estabelecida conexão à *Netflix*. Por último, deve controlar todos os elementos audiovisuais da casa, como sistemas de som, consolas de videojogos e televisores.

## 1.2. CLIENTE

Neste sistema temos duas variantes para o tipo de cliente, podendo este ser qualquer pessoa que resida numa habitação ou uma empresa que pretenda ter uma solução para vender aos seus clientes. Em ambos os casos, é necessário que tenha alguma estabilidade financeira que permita o investimento, e que esteja disposto a pagar por produtos e serviços que o ajude a melhor gerir a casa, mesmo que o interesse na tecnologia em si seja pouco. Destina-se a ser instalado em vivendas e não apartamentos pela questão das burocracias que os segundos implicam.

O cliente, que deverá ser maior de idade, pretende monitorizar, controlar e regular as funções da sua casa, o que será possível com a aquisição do sistema em desenvolvimento. Pode ter como prioridades o conforto e a iluminação, a segurança, a gestão de recursos, o controlo de equipamentos ou ainda o entretenimento, que vão integrar a solução proposta.



### 1.3. OBJETIVOS

Devido ao facto de os sistemas que se encontram disponíveis no mercado serem de custo elevado ou ainda de não se poderem aplicar na maioria das casas, pretende-se que a solução proposta para o sistema consiga ultrapassar estes dois problemas. Para tal tem de conseguir abranger grande parte das residências para que seja mais fácil a instalação nas mesmas, impondo a sua posição no mercado.

Com este sistema de gestão para casas inteligentes pretende-se estar a par da tecnologia atual, inovando e renovando residências previamente contruídas e/ou por contruir, tentando corresponder ao ideal moderno e funcional; incrementar o conforto ao facilitar grande parte das atividades, para que, por exemplo, não seja necessário o utilizador deslocar-se; aumentar a segurança da casa através da instalação de alarmes e sensores, podendo acompanhar em tempo real todos os movimentos; economizar recursos, ao utilizar o mínimo necessário de eletricidade, gás e água, permitindo pagar o investimento inicial.

## 2. PLANIFICAÇÃO DO TRABALHO

A fase de planeamento deve contemplar várias etapas onde se definem os objetivos gerais e de segunda ordem, o tipo de tarefas, os recursos necessários à concretização dos objetivos e das tarefas, e a pesquisa. É na fase de execução/implementação que se aplicam todas as tarefas e requisitos estabelecidos na fase de planeamento.

### 2.1. CALENDARIZAÇÃO DAS TAREFAS PREVISTO

No âmbito da fase de planeamento, a elaboração de um cronograma é uma forma de organizar e gerir as tarefas durante a execução de um projeto. Consiste na determinação da melhor forma de posicionar as tarefas ao longo do tempo de acordo com a duração das mesmas, das relações de precedência entre elas e dos prazos a cumprir.

Com base nisto, foi elaborado um cronograma correspondente a este projeto no qual estão presentes as atividades previstas bem como o tempo previsto de execução de cada uma delas (Figura 1)

Nome da Tarefa	Duração	Início	Conclusão	Predecessoras
▲ PTDA	68,13 dias	Qui 11/10/18	Ter 15/01/19	
▷ 1.Reunião	65,13 dias	Ter 16/10/18	Ter 15/01/19	
▲ 2.Levantamento e análise de requisitos	9 dias	Qui 11/10/18	Ter 23/10/18	
2.1 Fazer o planeamento do projeto	2 dias	Qui 11/10/18	Sex 12/10/18	
2.2 Pesquisa	2 dias	Seg 15/10/18	Ter 16/10/18	18
2.3 Definir o sistema	2 dias	Qua 17/10/18	Qui 18/10/18	19
2.4 Definir os requisitos	3 dias	Sex 19/10/18	Ter 23/10/18	20
2.5 Modelo de Casos de Utilização	3 dias	Qui 11/10/18	Seg 15/10/18	
▲ 3. Desenho do sistema	11 dias?	Qui 11/10/18	Qui 25/10/18	
3.1 Casos de utilização reais	2 dias	Qua 24/10/18	Qui 25/10/18	21
3.2 Identificar e caracterizar todas as	2 dias	Qua 24/10/18	Qui 25/10/18	21
3.3 Produzir o modelo de dados persistente	2 dias	Qua 24/10/18	Qui 25/10/18	21
3.4 Entrega dos primeiros relatórios	1 dia?	Qui 11/10/18	Qui 11/10/18	
▷ 4.Implementação	24 dias	Seg 05/11/18	Qui 06/12/18	
▷ 5. Finalização do projeto	1 dia	Sex 07/12/18	Sex 07/12/18	

FIGURA 1 - CALENDARIZAÇÃO DAS TAREFAS PREVISTO

## 2.2. CALENDARIZAÇÃO DAS TAREFAS EXECUTADO

Esta fase consistia em aplicar todas as tarefas estabelecidas nas fases iniciais do planeamento. À medida que se iam realizando e implementando as tarefas, observou-se que eram necessários ajustes relativamente às mesmas, como se pode verificar pela Figura 2.

Nome da Tarefa	Duração	Início	Conclusão	Pre
▷ 1.Reunião	65,13 dias	Ter 16/10/18	Ter 15/01/19	
◄ 2.Levantamento e análise de requisitos	9 dias	Qui 11/10/18	Ter 23/10/18	
2.1 Fazer o planeamento do projeto	2 dias	Qui 11/10/18	Sex 12/10/18	
2.2 Pesquisa	2 dias	Seg 15/10/18	Ter 16/10/18	18
2.3 Definir o sistema	2 dias	Qua 17/10/18	Qui 18/10/18	19
2.4 Definir os requisitos	3 dias	Sex 19/10/18	Ter 23/10/18	20
2.5 Modelo de Casos de Utilização	3 dias	Qui 11/10/18	Seg 15/10/18	
◄ 3. Desenho do sistema	14 dias	Qui 11/10/18	Ter 30/10/18	
3.1 Casos de utilização reais	2 dias	Qua 24/10/18	Qui 25/10/18	21
3.2 Elaboração do diagrama de classes	5 dias	Qua 24/10/18	Ter 30/10/18	21
3.3 Produzir o modelo de dados persistente	2 dias	Qua 24/10/18	Qui 25/10/18	21
3.4 Entrega dos primeiros relatórios	1 dia	Qui 11/10/18	Qui 11/10/18	
◄ 4.Implementação	30 dias	Seg 05/11/18	Sex 14/12/18	
4.1 Implementar a solução recorrendo a uma	4 sems	Seg 05/11/18	Sex 30/11/18	26
4.2 Testar/Debug	2 sems	Seg 03/12/18	Sex 14/12/18	29
4.3 Documentar a solução	1 sem	Seg 17/12/18	Sex 21/12/18	30
▷ 5. Finalização do projeto	1 dia	Sex 07/12/18	Sex 07/12/18	

FIGURA 2 - CALENDARIZAÇÃO DAS TAREFAS EXECUTADO

## 2.3.REQUISITOS FUNCIONAIS

Os requisitos funcionais são definidos como as funcionalidades ou atividades que um sistema/*software* deve realizar. Representa o que o software faz, em termos de tarefas e serviços, através de cálculos, detalhes técnicos, manipulação de dados e de processamento e outras funcionalidades específicas que definem o que um sistema, idealmente, será capaz de realizar.

Foram criados alguns conceitos tais como o de “zona” – local criado dentro de uma divisão (um espaço) e que possui os seus próprios comandos ou configurações. As ações realizadas dentro dessa zona não têm qualquer impacto no espaço onde foi criada; “rotina” - tarefa executada

automaticamente, onde os utilizadores só precisam de iniciar ou é iniciado pelo sistema; “quadro elétrico virtual” – pretende simular o quadro elétrico físico presente na casa, onde estão ligados todos os circuitos elétricos da mesma, e onde se pode cortar a alimentação de qualquer tipo de circuito.

Em seguida está apresentada uma tabela (Tabela 1) que contém os requisitos funcionais alusivos ao desenvolvimento do sistema.

**TABELA 1 - REQUISITOS FUNCIONAIS**

<b>Refª</b>	<b>Requisito funcional</b>	<b>Tipo</b>
RF01	O sistema permite visualizar informação relativa ao consumo de cada eletrodoméstico	Alto
RF02	O sistema emite um alerta quando algum tipo de fuga (água ou gás) é detetada e corta o abastecimento do utilitário em questão	Médio
RF03	O sistema permite ligar/desligar o abastecimento de utilitários (água, gás, eletricidade)	Alto
RF04	O sistema possui um quadro elétrico virtual igual ao quadro elétrico físico que existe	Alto
RF05	O sistema permite criar e controlar zonas	Alto
RF06	O sistema permite controlar a iluminação da casa (ligar/desligar, intensidade, cor)	Alto
RF07	O sistema permite acender as luzes automaticamente quando o utilizador entra numa divisão, se a luminosidade for baixa o suficiente, e apagar quando sair	Médio
RF08	O sistema permite ligar/desligar todos os equipamentos	Alto
RF09	O sistema permite controlar os estores da casa (subir/descer)	Alto
RF10	O sistema permite controlar a temperatura da casa	Alto
RF11	O sistema determina a distância a que os utilizadores se encontram da casa para ativação de certas ações	Médio
RF12	O sistema permite abrir/fechar portas e portões	Alto

RF13	O sistema permite destrancar a porta com um código	Médio
RF14	O sistema permite receber <i>feed</i> de vídeo de câmaras	Médio
RF15	O sistema emite um alerta quando tocam à campainha	Alto
RF16	O sistema consegue controlar sistemas audiovisuais	Baixo
RF17	O sistema consegue guardar ficheiros multimédia localmente	Baixo
RF18	O sistema permite gerir o aspirador robot	Alto
RF19	O sistema permite gerir os detetores de fumo (ligar/desligar e consultar estado)	Alto
RF20	O sistema permite gerir as câmaras de vigilância (ligar/desligar, consultar estado e consultar <i>feed</i> )	Alto
RF21	O sistema permite gerir o sistema de irrigação	Alto
RF22	O sistema permite controlar a humidade da casa	Alto
RF23	O sistema permite gerir os ejetores de comida dos animais	Alto
RF24	O sistema permite abrir/fechar a cobertura da piscina	Alto
RF25	O sistema permite a criação e execução de rotinas	Alto
RF26	O sistema permite que o utilizador calendarize ou ative remotamente uma rotina, tendo em conta a sua intenção	Médio
RF27	O sistema permite gerir o sistema de alarme (ligar/desligar, adicionar/remover/alterar códigos)	Alto

## 2.4. RESTRIÇÕES E REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais devem conter elementos específicos, tais como a descrição da tarefa a ser executada pelo *software*, a origem do requisito e o seu utilizador, a relação da passagem de informação entre o *software* e o utilizador e, se existirem, algumas restrições lógicas associadas à tarefa. Dentro dos requisitos não funcionais estão incluídos os requisitos de interface e facilidade de uso, de desempenho, de segurança e integridade de dados, de interface com sistemas externos e ambientes de execução, entre outros.

Os requisitos não funcionais estão relacionados com os requisitos funcionais e indicam

como o sistema/*software* deve ser feito e como deve funcionar, ou seja, são os critérios que qualificam os requisitos funcionais.

#### 2.4.1. REQUISITOS DE INTERFACE E FACILIDADE DE USO

Os requisitos de interface e facilidade de uso correspondem às expectativas e especificações, desenhados para assegurar que um produto, serviço, processo ou ambiente, seja fácil de utilizar.

Em seguida está apresentada uma tabela (Tabela 2) que contém os requisitos de interface e usabilidade, e respetivos requisitos funcionais, alusivos ao desenvolvimento do sistema.

**TABELA 2 - REQUISITOS DE INTERFACE E USABILIDADE**

Refª	Requisito de interface e usabilidade	Req. funcionais relacionados
RI01	Os consumos individuais de cada equipamento e o equipamento mais consumidor devem ser todos mostrados numa interface dedicada a cada tipo de utilizário	RF01
RI02	As notificações devem ser mostradas num dos cantos e serem sempre visíveis em todas as interfaces	RF02, RF15
RI03	Interface com os controlos básicos de água, eletricidade e gás	RF03, RF04
RI04	Interface de definições onde é permitido criar zonas, adicionar, alterar e remover equipamentos das divisões e zonas	RF05
RI05	Interface para cada divisão com o controlo dos equipamentos da mesma	RF08

#### 2.4.2 REQUISITOS DE INTERFACE COM SISTEMAS EXTERNOS E COM AMBIENTES DE EXECUÇÃO

Os requisitos de interface com sistemas externos e com ambientes de execução podem ser colocados quer no produto, quer no processo, e são derivados do ambiente onde o sistema está a

ser desenvolvido, baseando-se em informação do domínio de aplicação, na necessidade do sistema em interagir com outros sistemas, etc.

Em seguida está apresentada uma tabela (Tabela 3) que contém os requisitos de interface com sistemas externos e com ambientes de execução, e respetivos requisitos funcionais alusivos ao desenvolvimento do sistema.

**TABELA 3 - REQUISITOS DE INTERFACE COM SISTEMAS EXTERNOS E COM AMBIENTES DE EXECUÇÃO**

Refª	Requisito de interface com sistemas externos e com ambientes de execução	Req. funcionais relacionados
RAE01	O sistema deve receber dados dos equipamentos, através de <i>sockets</i>	
RAE02	O sistema deve mandar comandos aos equipamentos, através de <i>sockets</i>	
RAE03	O sistema deverá ser escrito na linguagem Java	
RAE04	O servidor de base de dados deverá ser o <i>PostgreSQL</i>	
RAE05	A interface do sistema deverá ser capaz de correr em Android, iOS e <i>Windows Phone</i>	

## 2.5. REQUISITOS DE HARDWARE

Devido à natureza do nosso projeto só é possível quantificar a maior parte dos componentes na instalação, pois as plantas variam de habitação para habitação, é raro as casas terem o mesmo número de divisões e a mesma estrutura. Visto que será apenas vendido o software, que terá de comunicar com todos os aparelhos da casa, independentemente da sua marca, para que todas as funcionalidades do nosso sistema sejam utilizadas, serão necessários pelo menos um de cada dos seguintes componentes.

- Sensor de caudal;
- Sensor de pressão;
- Válvula Solenóide;
- *Relays*;
- Sensor de gás;

- Sensor de corrente;
- Câmaras de vigilância;
- Detetor de fumo;
- Sensor de humidade;
- Sensor de temperatura;
- Medidor de pH;
- Fechadura elétrica;
- Servidor;
- Controlador.





### 3. MODELO DE CASOS DE UTILIZAÇÃO

#### 3.1. VISÃO GERAL

Um diagrama de casos de utilização de um sistema mostra atores (tipos de utilizadores), casos de utilização e relações entre eles.

O diagrama de casos de uso abaixo (Figura 3) descreve a funcionalidade proposta para o sistema a implementar, facilitando o levantamento dos requisitos funcionais do sistema.



**FIGURA 3 - DIAGRAMA DE CASOS DE USO**

### 3.2. ATORES

O ator especifica um papel executado por um utilizador ou outro sistema que interage com o sistema. O ator deve ser externo ao sistema e deve ter associações exclusivamente para casos de uso, componentes ou classes, podendo herdar o papel de outro. Na tabela abaixo apresentada (Tabela 4 - Atores do diagrama de casos de uso) é possível descrever os atores que interagem com o sistema.

**TABELA 4 - ATORES DO DIAGRAMA DE CASOS DE USO**

Ator	Descrição
Utilizador	Utilizador apenas com permissões para utilizar equipamentos da casa que não perturbem a segurança e normal funcionamento da mesma
Administrador	Utilizador que contém o “poder máximo” na casa, têm mais permissões que os outros utilizadores. Essas permissões podem perturbar o normal funcionamento da casa e a sua segurança daí ser necessário terem de ser ocultas do utilizador comum. Este utilizador é o único que pode editar as configurações do sistema

### 3.3. DESCRIÇÃO DOS CASOS DE UTILIZAÇÃO

#### 3.3.1. CASO DE UTILIZAÇÃO #1

Na tabela seguinte (Tabela 5) é possível observar a descrição do caso de utilização, os atores que envolve, a sua prioridade, finalidade, quais os requisitos funcionais associados e um sumário do caso.

**TABELA 5 - VISUALIZAR OS CONSUMOS DE ÁGUA, GÁS E LUZ DOS APARELHOS**

Nome	Visualizar os consumos de água, gás e luz dos aparelhos
Atores	Utilizador, Administrador
Prioridade (1/3)	1

<b>Finalidade</b>	Receber informações sobre os consumos dos vários aparelhos
<b>Requisitos Funcionais</b>	RF01 - O sistema permite visualizar informação relativa ao consumo de cada eletrodoméstico
<b>Sumário</b>	O utilizador ou o Administrador da casa poderão visualizar a informação sobre os consumos

### 3.3.2. CASO DE UTILIZAÇÃO #2

Na tabela seguinte (Tabela 6) é possível observar a descrição do caso de utilização, os atores que envolve, a sua prioridade, finalidade, quais os requisitos funcionais associados e um sumário do caso.

**TABELA 6 - MUDAR ESTADO DE DISPOSITIVOS E EQUIPAMENTOS**

<b>Nome</b>	<b>Mudar estado de dispositivos e equipamentos</b>
<b>Atores</b>	Utilizador, Administrador
<b>Prioridade (1/3)</b>	1
<b>Finalidade</b>	Mudar o estado de dos dispositivos e equipamentos de ligado para desligado e vice-versa
<b>Requisitos Funcionais</b>	RF03- O sistema permite ligar/desligar o abastecimento de utilitários (água, gás, eletricidade)  RF08 - O sistema permite ligar/desligar todos os equipamentos
<b>Sumário</b>	O utilizador ou o Administrador da casa poderão selecionar qualquer dispositivo e equipamento, e a partir daí terão as opções de ligar ou desligar tal.

### 3.3.3. CASO DE UTILIZAÇÃO #3

Na tabela seguinte (Tabela 5) é possível observar a descrição do caso de utilização, os atores que envolve, a sua prioridade, finalidade, quais os requisitos funcionais associados e um sumário do caso.

**TABELA 7 - VISUALIZAR OS ALERTAS**

<b>Nome</b>	<b>Visualizar os alertas</b>
<b>Atores</b>	Utilizador, Administrador
<b>Prioridade (1/3)</b>	1
<b>Finalidade</b>	Acesso aos alertas
<b>Requisitos Funcionais</b>	RF02 - O sistema emite um alerta quando algum tipo de fuga (água ou gás) é detetada e corta o abastecimento do utilitário em questão  RF15- O sistema emite um alerta quando tocam à campainha
<b>Sumário</b>	O utilizador ou o Administrador da casa poderão aceder ao ecrã dos alertas onde terão todas as informações sobre os tais.

#### 3.3.4. CASO DE UTILIZAÇÃO #4

Na tabela seguinte (Tabela 8) é possível observar a descrição do caso de utilização, os atores que envolve, a sua prioridade, finalidade, quais os requisitos funcionais associados e um sumário do caso.

**TABELA 8 - MUDAR ESTADO DE PORTAS, PORTÕES E JANELAS**

<b>Nome</b>	<b>Mudar estado de portas, portões e janelas</b>
<b>Atores</b>	Utilizador, Administrador
<b>Prioridade (1/3)</b>	1
<b>Finalidade</b>	
<b>Requisitos Funcionais</b>	RF09- O sistema permite controlar os estores da casa (subir/descer)  RF12- O sistema permite abrir/fechar portas e portões  RF13- O sistema permite destrancar a porta com um código
<b>Sumário</b>	O utilizador ou o Administrador da casa poderão sempre abrir e fechar uma porta, portão e janela. Em relação à janela, se tiver estouras os utilizadores têm as opções de subir ou descer os estores. Se houver uma porta que necessite de um código para abri-la, os utilizadores poderão só desbloquear a porta se souberem o código correto.

### 3.3.5. CASO DE UTILIZAÇÃO #5

Na tabela seguinte (Tabela 9) é possível observar a descrição do caso de utilização, os atores que envolve, a sua prioridade, finalidade, quais os requisitos funcionais associados e um sumário do caso.

**TABELA 9 - CONTROLAR ILUMINAÇÃO**

<b>Nome</b>	<b>Controlar iluminação</b>
<b>Atores</b>	Utilizador, Administrador
<b>Prioridade (1/3)</b>	
<b>Finalidade</b>	
<b>Requisitos Funcionais</b>	RF04- O sistema possui um quadro elétrico virtual igual ao quadro elétrico físico que existe  RF06- O sistema permite controlar a iluminação da casa (ligar/desligar, intensidade, cor)
<b>Sumário</b>	O utilizador ou o Administrador da casa terá a sua disposição uma interface onde poderá ligar/desligar a iluminação, também terá a opção de regular a intensidade da luz e por fim poderá mudar a cor da iluminação.  Ainda existirá outra opção que funcionará como quadro elétrico digital.

### 3.3.6. CASO DE UTILIZAÇÃO #6

Na tabela seguinte (Tabela 10) é possível observar a descrição do caso de utilização, os atores que envolve, a sua prioridade, finalidade, quais os requisitos funcionais associados e um sumário do caso.

**TABELA 10 - ATIVAR FUNCIONALIDADES NA DETEÇÃO DE UTILIZADORES PRÓXIMOS**

<b>Nome</b>	<b>Ativar funcionalidades na deteção de utilizadores próximos</b>
<b>Atores</b>	Utilizador, Administrador
<b>Prioridade (1/3)</b>	
<b>Finalidade</b>	Executa certas tarefas automáticas na presença de um utilizador

<b>Requisitos Funcionais</b>	RF07- O sistema permite acender as luzes automaticamente quando o utilizador entra numa divisão, se a luminosidade for baixa o suficiente, e apagar quando sair
	RF11- O sistema determina a distância a que os utilizadores se encontram da casa para ativação de certas ações
<b>Sumário</b>	O utilizador ou o Administrador só necessitarão de ter consigo algo que os identifique. Quando o sistema reconhecer irá executar as tarefas programadas.

### 3.3.7. CASO DE UTILIZAÇÃO #7

Na tabela seguinte (Tabela 11) é possível observar a descrição do caso de utilização, os atores que envolve, a sua prioridade, finalidade, quais os requisitos funcionais associados e um sumário do caso.

**TABELA 11 - GERIR ASPIRADOR ROBOT**

<b>Nome</b>	<b>Gerir aspirador robot</b>
<b>Atores</b>	Utilizador, Administrador
<b>Prioridade (1/3)</b>	
<b>Finalidade</b>	Criar e editar tarefas para o robot
<b>Requisitos Funcionais</b>	RF18- O sistema permite gerir o aspirador robot
<b>Sumário</b>	O utilizador ou o Administrador poderá configurar para “trabalhar” quando quiserem, podendo se tornar uma rotina

### 3.3.8. CASO DE UTILIZAÇÃO #8

Na tabela seguinte (Tabela 12) é possível observar a descrição do caso de utilização, os atores que envolve, a sua prioridade, finalidade, quais os requisitos funcionais associados e um sumário do caso.

**TABELA 12 - GERIR E CONTROLAR O SISTEMA DE IRRIGAÇÃO**

<b>Nome</b>	<b>Gerir e Controlar o sistema de irrigação</b>
<b>Atores</b>	Utilizador, Administrador
<b>Prioridade (1/3)</b>	
<b>Finalidade</b>	Criar e editar tarefas para os sistemas de irrigação
<b>Requisitos Funcionais</b>	RF21- O sistema permite gerir o sistema de irrigação
<b>Sumário</b>	O utilizador ou o Administrador poderá configurar o sistema de irrigação de vários sítios para “trabalhar” quando quiserem, podendo se tornar uma rotina

### 3.3.9. CASO DE UTILIZAÇÃO #9

Na tabela seguinte (Tabela 13) é possível observar a descrição do caso de utilização, os atores que envolve, a sua prioridade, finalidade, quais os requisitos funcionais associados e um sumário do caso.

**TABELA 13 - EXECUTAR ROTINAS**

<b>Nome</b>	<b>Executar Rotinas</b>
<b>Atores</b>	Utilizador, Administrador
<b>Prioridade (1/3)</b>	
<b>Finalidade</b>	Criar e editar tarefas para serem feitos automaticamente
<b>Requisitos Funcionais</b>	RF26- O sistema permite que o utilizador calendarize ou ative remotamente uma rotina, tendo em conta a sua intenção
<b>Sumário</b>	O utilizador ou o Administrador poderá configurar qualquer equipamento ou dispositivo do sistema para fazer algo que não necessita da “mão” do utilizador ou administrador.



### 3.3.10 CASO DE UTILIZAÇÃO #10

Na tabela seguinte (Tabela 14) é possível observar a descrição do caso de utilização, os atores que envolve, a sua prioridade, finalidade, quais os requisitos funcionais associados e um sumário do caso.

**TABELA 14 - ATIVAR ZONAS**

<b>Nome</b>	<b>Ativar zonas</b>
<b>Atores</b>	Utilizador, Administrador
<b>Prioridade (1/3)</b>	1
<b>Finalidade</b>	Colocar a zona funcional
<b>Requisitos Funcionais</b>	RF05- O sistema permite criar e controlar zonas
<b>Sumário</b>	Depois da zona ser criada pelo o Administrador, O utilizador ou o Administrador poderão ativar a zona sempre que quiserem.

### 3.3.11. CASO DE UTILIZAÇÃO #11

Na tabela seguinte (Tabela 15) é possível observar a descrição do caso de utilização, os atores que envolve, a sua prioridade, finalidade, quais os requisitos funcionais associados e um sumário do caso.

**TABELA 15 - CRIAR ZONAS**

<b>Nome</b>	<b>Criar zonas</b>
<b>Atores</b>	Administrador
<b>Prioridade (1/3)</b>	1
<b>Finalidade</b>	Criar uma zona
<b>Requisitos Funcionais</b>	RF05- O sistema permite criar e controlar zonas
<b>Sumário</b>	O Administrador insere no sistema uma nova zona.  Essa zona poderá ser ativada mais tarde quer seja pelo administrador quer

seja pelo utilizador

### 3.3.12. CASO DE UTILIZAÇÃO #12

Na tabela seguinte (Tabela 16) é possível observar a descrição do caso de utilização, os atores que envolve, a sua prioridade, finalidade, quais os requisitos funcionais associados e um sumário do caso.

**TABELA 16 - CRIAR ROTINAS**

<b>Nome</b>	<b>Criar rotinas</b>
<b>Atores</b>	Administrador
<b>Prioridade (1/3)</b>	1
<b>Finalidade</b>	Criar uma rotina
<b>Requisitos Funcionais</b>	RF25- O sistema permite a criação e execução de rotinas
<b>Sumário</b>	O Administrador insere no sistema uma nova rotina.  Essa rotina poderá ser usada mais tarde quer seja pelo administrador quer seja pelo utilizador

### 3.3.13. CASO DE UTILIZAÇÃO #13

Na tabela seguinte (Tabela 17) é possível observar a descrição do caso de utilização, os atores que envolve, a sua prioridade, finalidade, quais os requisitos funcionais associados e um sumário do caso.

**TABELA 17 - GERIR E CONTROLAR ALARMES**

<b>Nome</b>	<b>Gerir e controlar alarmes</b>
<b>Atores</b>	Administrador
<b>Prioridade (1/3)</b>	1

<b>Finalidade</b>	Garantir a segurança da casa
<b>Requisitos Funcionais</b>	RF25- O sistema permite gerir o sistema de alarme (ligar/desligar, adicionar/remover/alterar códigos)
<b>Sumário</b>	O Administrador será o único no sistema a poder adicionar/remover/alterar códigos de alarme

## 4. DESCRIÇÃO DAS ATIVIDADES DESENVOLVIDAS

### 4.1 DESENVOLVIMENTO DA APLICAÇÃO

Para o desenvolvimento da aplicação foi adotada uma arquitetura de camadas, em que foram adotadas 3 camadas a de interface, a lógica e a de dados.

A camada de interface fornece ao utilizador uma interface gráfica com vista a fornecer uma experiência de usabilidade melhor, no entanto não foram implementados muitos conceitos de Interação Humano-Computador.

A camada lógica processa toda a informação dos sensores e fornece a mesma à camada de interface. Esta camada também permite consultar e alterar os estados dos equipamentos. Esta camada é responsável por verificar periodicamente se existem rotinas a ser executadas.

A camada de dados guarda permanentemente dados relativos ao sistema, ou seja, guarda todos os equipamentos presentes na casa, todas as zonas existentes e todas as rotinas presentes.

Estas camadas necessitam de comunicar entre si para ser possível de partilhar dados entre as mesmas. Para a comunicação com a base de dados foi usada a biblioteca do PostgreSQL para o Java e que agiliza a conexão com a camada de dados. A restante comunicação é realizada por sockets.

Sockets são mecanismos de comunicação, usado normalmente para executar um modelo cliente/servidor, que permite a troca de mensagens entre os processos de uma máquina/aplicação servidor e de uma máquina/aplicação cliente.

TCP significa “Protocolo de Controle de transmissão” sendo um dos principais protocolos de envio e recebimento de dados. TCP é o protocolo mais usado isto porque fornece garantia na entrega de todos os pacotes entre um PC emissor e um PC recetor.

Posto isto o grupo decidiu usar este protocolo de comunicação para estabelecer contacto entre a interface e o servidor, servidor e equipamentos, estando incluídos os sensores. Foi criada a classe SendCommand que permite enviar comandos via sockets em que o comando tem o formato: objeto a ser aplicado, método a executar e parâmetros para o mesmo método.

## 4.2 DIAGRAMA DE CLASSES

Um diagrama de classes serve para modelar o vocabulário de um sistema, do ponto de vista do utilizador/problema ou do implementador/solução.

Um diagrama de classes é constituído por itens que podem representar as classes que serão realmente programadas, assim como os principais objetos ou a interação entre classes e objetos do sistema, sendo eles classes, atributos, operações e associações.

Uma classe contém a especificação do objeto, as suas características, atributos e métodos.

Um atributo define as características da classe, como a sua visibilidade, nome, tipo de dados, multiplicidade, valor inicial e propriedade. Quanto à visibilidade, esta pode ser pública (representada pelo símbolo "+") ou privada (representada pelo símbolo "-"). O nome corresponde à identificação do atributo, o tipo de dados indica a especificação dos dados do atributo, o valor inicial e as suas propriedades dependem da linguagem usada na programação (Java) e a multiplicidade indica a possibilidade de fazer relações com outras classes.

Uma operação trata da função requerida a um objeto abstrato e contém características como nome, visibilidade e parâmetros.

Uma associação trata da capacidade das classes se relacionarem. Também pode conter nome, multiplicidade e tipo de navegação, que indica de onde partem as informações da classe e para onde irão.

Na figura abaixo (Figura 4) está representado o diagrama de classes correspondente ao sistema em desenvolvimento.

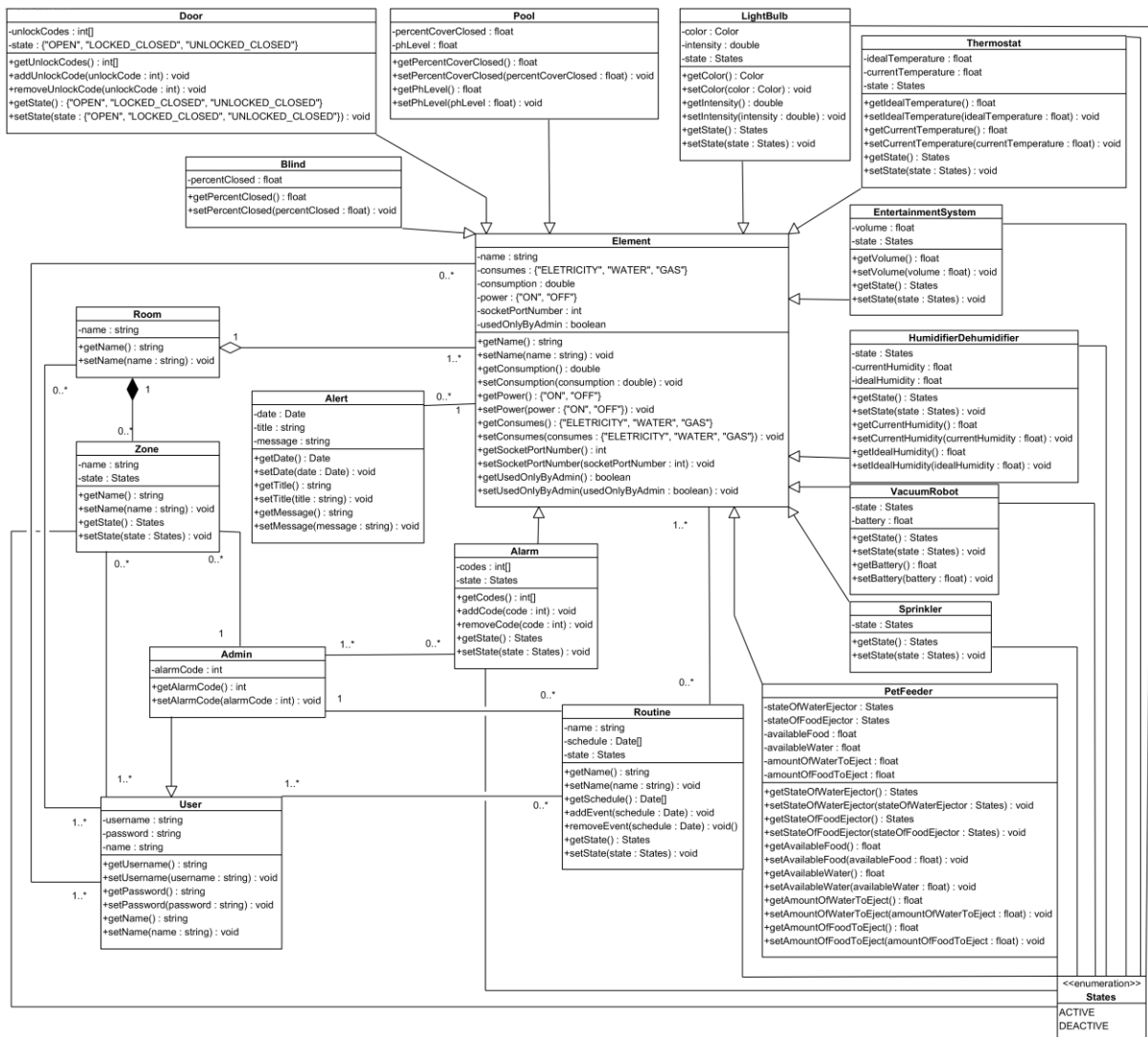


FIGURA 4 - DIAGRAMA DE CLASSES

### 4.3 MODELO DE DADOS PERSISTENTE

Modelos de dados persistentes são utilizados para projetar a estrutura dos data stores persistentes utilizados pelo sistema, fornecendo um conjunto de elementos de modelagem que podem ser utilizados para desenvolver o design detalhado de tabelas no banco de dados e modelar o layout de armazenamento físico do banco de dados.

Na figura abaixo (Figura 5) está representado o modelo de dados persistente correspondente ao sistema em desenvolvimento.

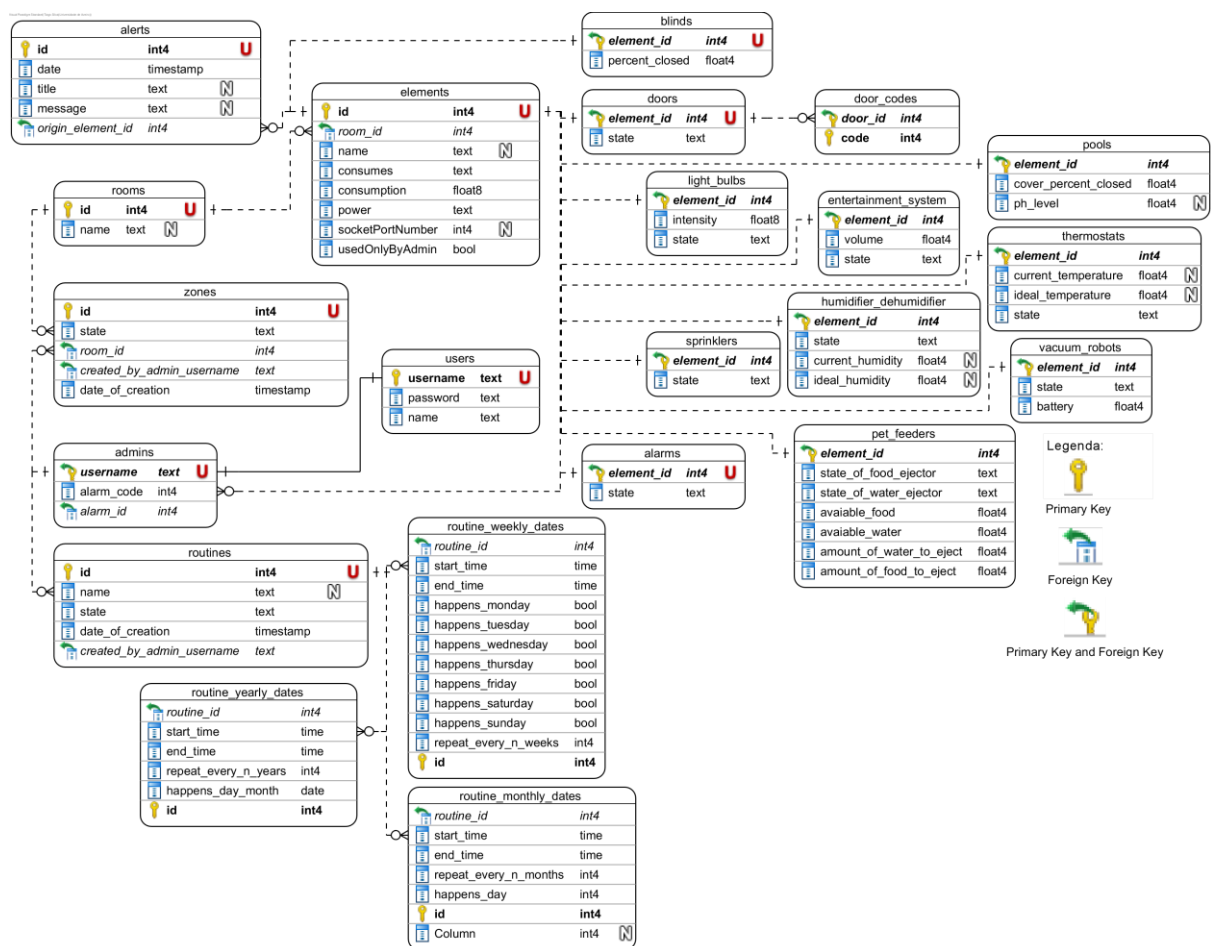


FIGURA 5 - MODELO DE DADOS PERSISTENTE

## 4.4 INTERFACE

Uma interface de utilizador é o espaço onde a interação entre humanos e máquinas ocorre. Tem como objetivo a operação e controle efetivos da máquina no lado do utilizador e o feedback da máquina, que auxilia o operador na tomada de decisões operacionais. Inclui componentes de hardware (físico) e software (lógico) e fornecem um meio de entrada, permitindo ao utilizador manipular o sistema, e de saída, permitindo ao sistema produzir os efeitos (respostas) das ações do utilizador.

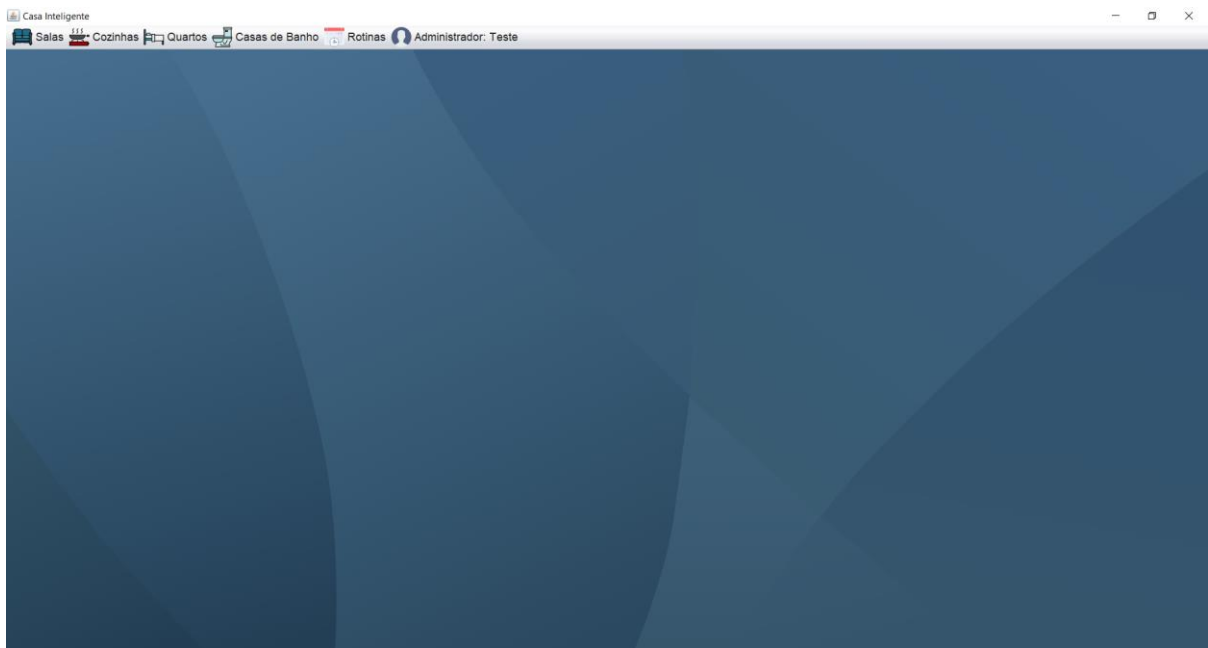
Na figura abaixo (Figura 6) está representada a janela de autenticação, onde o utilizador insere o nome de utilizador e a palavra-passe correspondente. Completando a autenticação, passa para o ecrã seguinte.



FIGURA 6 - JANELA DE AUTENTICAÇÃO

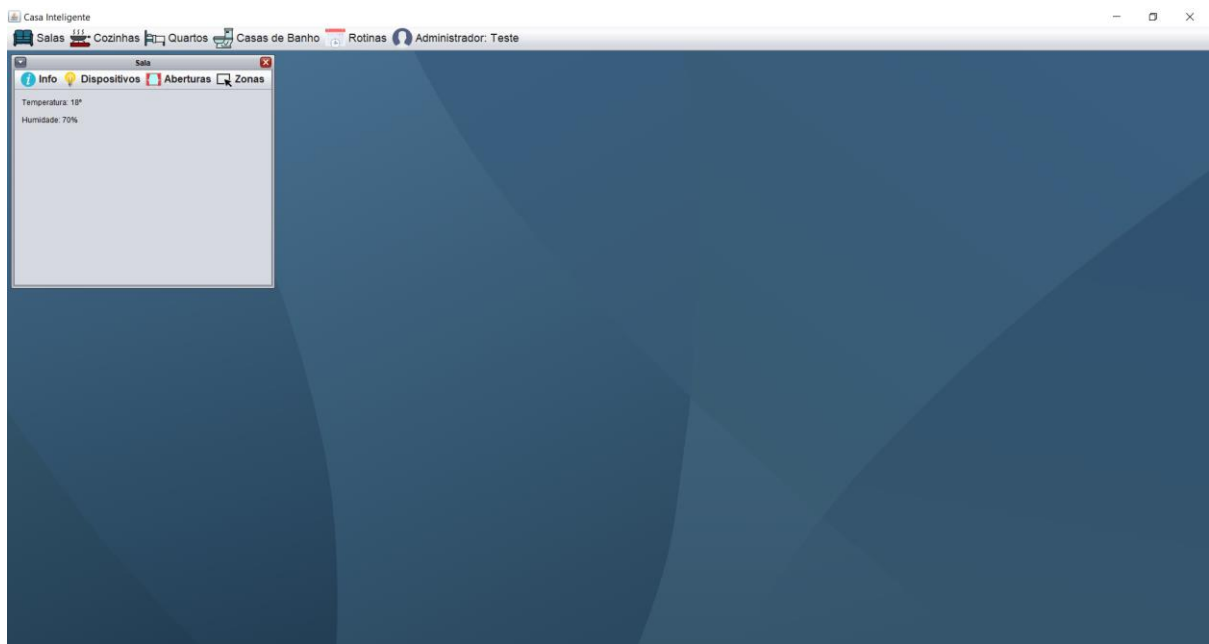
Na figura abaixo (Figura 7) está representada a janela principal, que surge após a autenticação, onde, neste caso, o administrador pode escolher controlar salas, cozinhas, quartos, casas de banho, rotinas e gerir utilizadores.



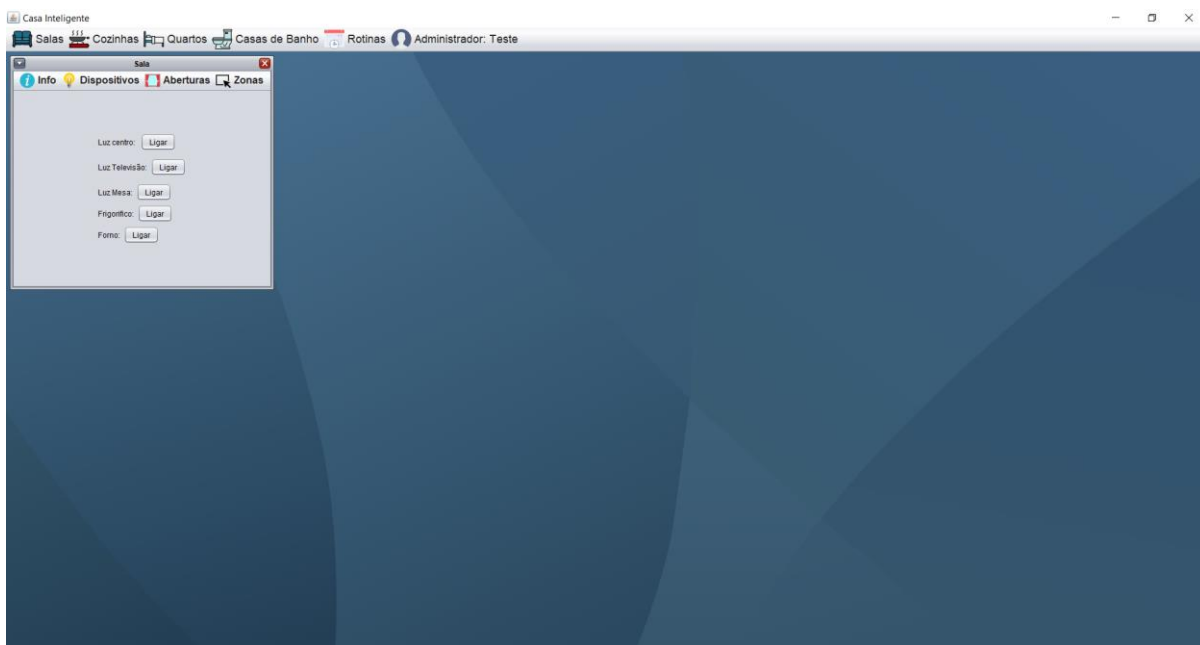


**FIGURA 7 - JANELA PRINCIPAL**

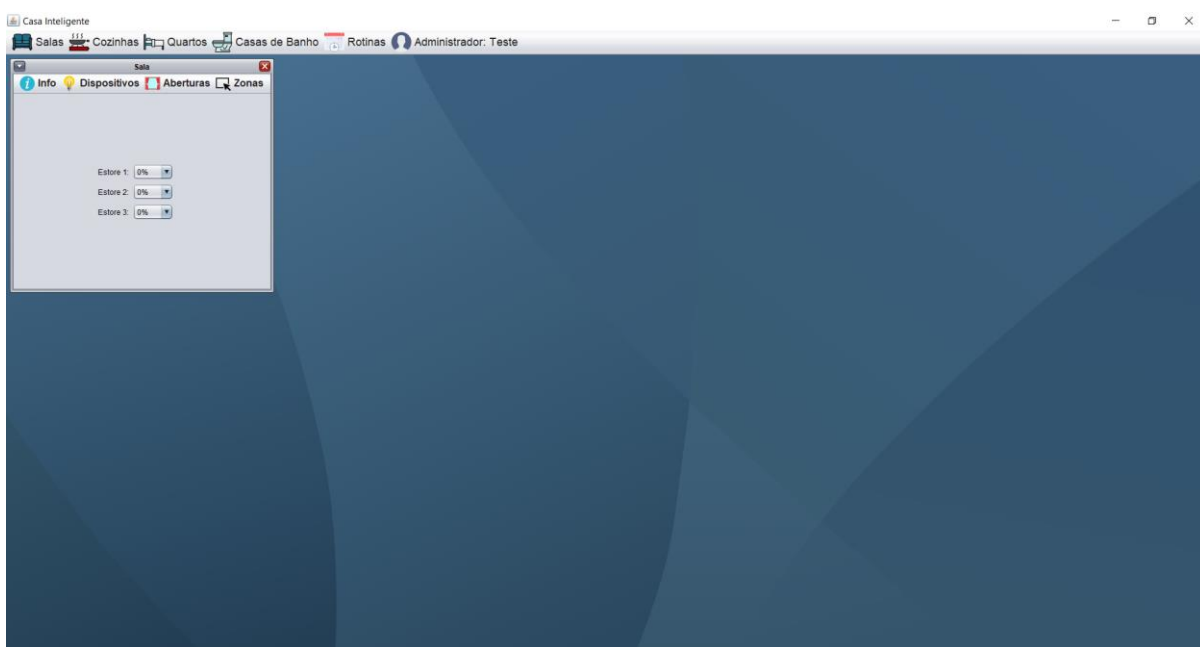
Após selecionar a opção pretendida, o utilizador, no caso de escolher uma divisão, pode obter informações acerca da sua temperatura e humidade, ligar e desligar os dispositivos que a mesma contém, controlar a abertura de cada estore, ativar/desativar zonas e editá-las ao adicionar/remover elementos, estando estas opções representadas, respetivamente, nas seguintes figuras (Figura 8, Figura 9, Figura 10, Figura 11 e Figura 12).



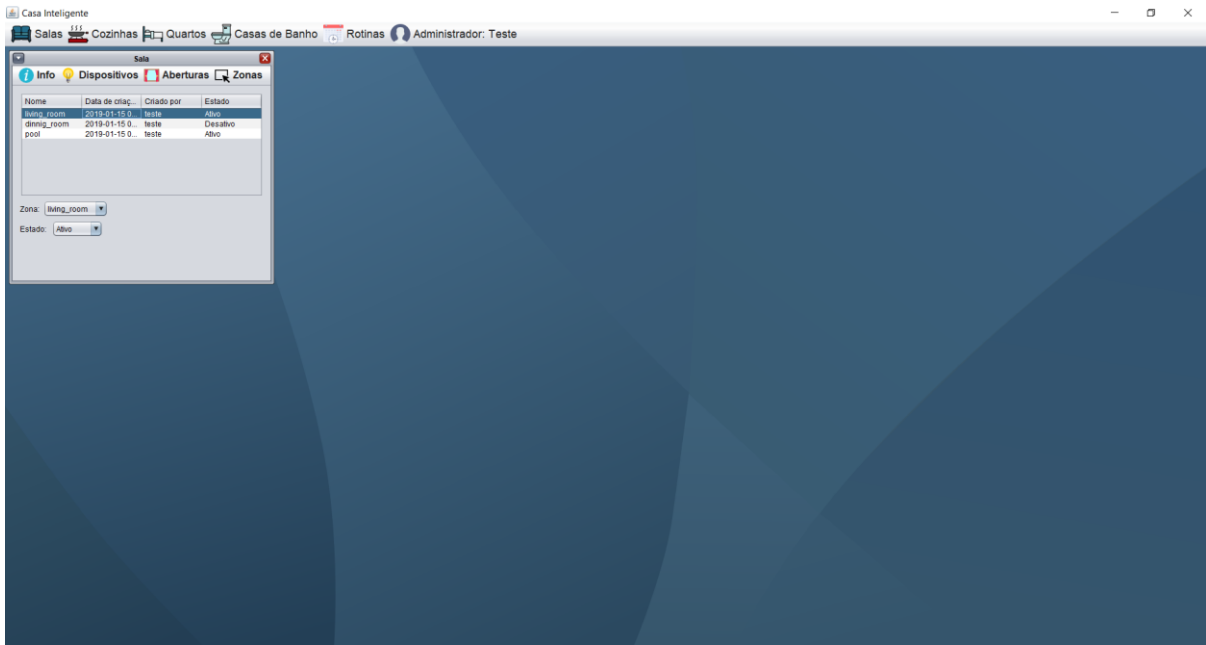
**FIGURA 8 - JANELA DAS INFORMAÇÕES DA SALA**



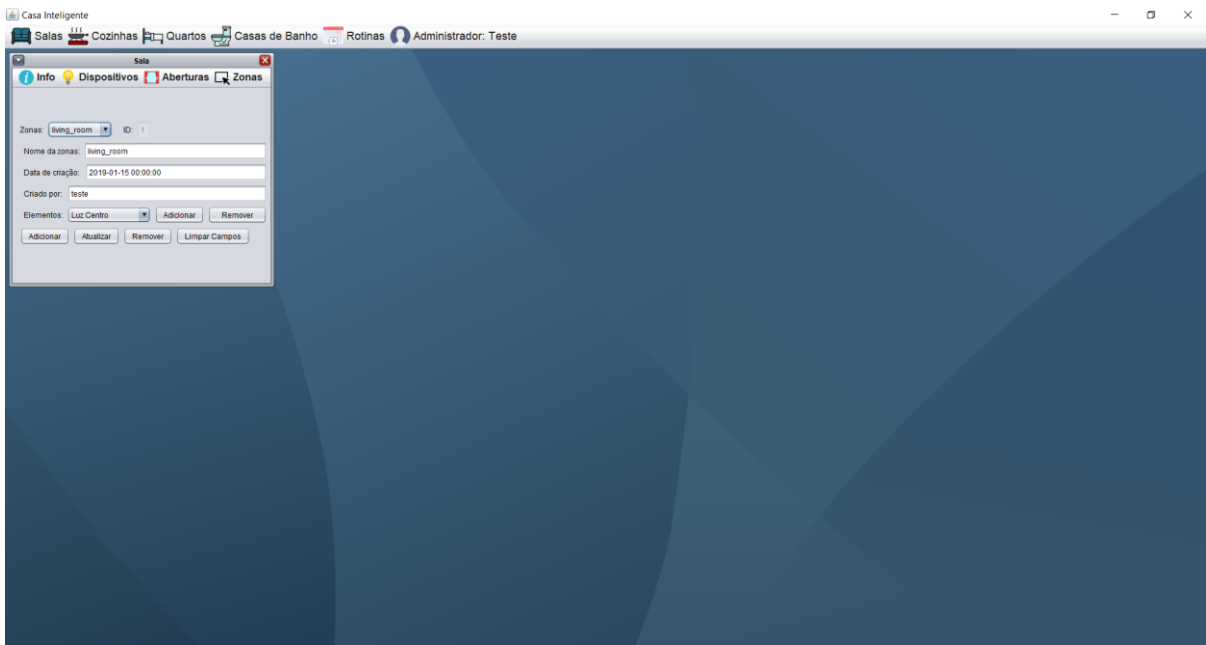
**FIGURA 9 - JANELA DOS DISPOSITIVOS DA SALA**



**FIGURA 10 - JANELA DE ABERTURAS DA SALA**



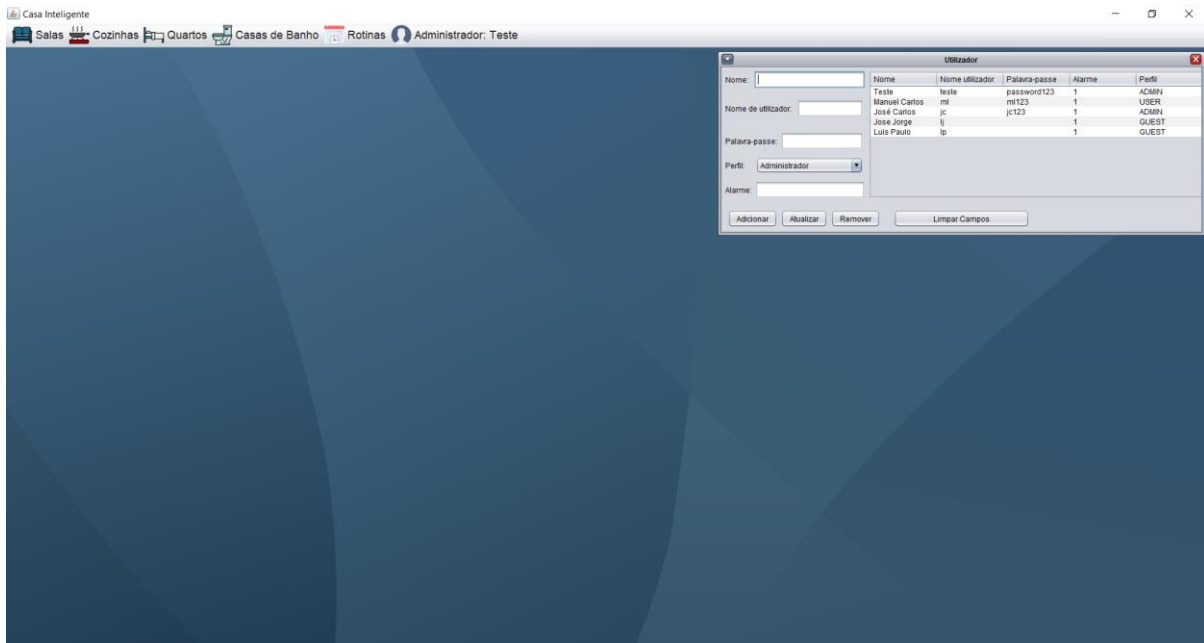
**FIGURA 11 - JANELA DE ZONAS DA SALA**



**FIGURA 12 - JANELA DE EDITAR ZONAS DA SALA**

Na Figura 13 está representada a gestão dos utilizadores do sistema, podendo-se adicionar/remover utilizadores e atualizar os seus dados. É possível ainda alterar o perfil do utilizador em questão, concedendo ou retirando-lhe permissões (“ADMIN”, “USER”). A criação de

um novo utilizador envolve um nome, um nome de utilizador, uma palavra-passe, definição do perfil e alarme.



**FIGURA 13 - JANELA DE GESTÃO DOS UTILIZADORES**



## 5. ANÁLISE DE RESULTADOS

Requisitos Funcionais	Cumprido	Não cumprido
RF01	X	
RF02	X	
RF03	X	
RF04	X	
RF05	X	
RF06	X	
RF07	X	
RF08	X	
RF09	X	
RF10	X	
RF11		X
RF12	X	
RF13		X
RF14		X
RF15		X
RF16	X	
RF17		X
RF18	X	
RF19	X	
RF20	X	
RF21	X	
RF22	X	
RF23	X	

<b>RF24</b>	X	
<b>RF25</b>		X
<b>RF26</b>		X
<b>RF27</b>	X	

## 6. CONCLUSÃO

Com a realização deste projeto, foram postos em prática os conhecimentos adquiridos das unidades curriculares associadas ao Projeto Temático, Sistemas de base de dados e Engenharia de Software, lecionadas pelos docentes Luís Jorge Gonçalves e Joaquim Ferreira.

Durante a realização do projeto houve algumas mudanças nas ideias originais do grupo e foram feitas alterações durante o processo de desenvolvimento para melhorar a qualidade do que foi proposto desenvolver.

Concluindo, o grupo superou a maior parte das dificuldades com que se deparou, como por exemplo o estabelecimento da conexão à base de dados e o processo de comunicação por sockets, e conseguiu desenvolver e implementar grande parte das tarefas que foram propostas. Porém, é notório que ainda é necessário aperfeiçoar e melhorar o sistema apresentado, como é o caso da interface, que pode ser melhorada de forma a ser ainda mais utilizada em dispositivos móveis e a ter maior compatibilidade com outros eletrodomésticos.





## 7. BIBLIOGRAFIA

- [1] R. Ramakrishnan, Database Management Systems, McGraw Hill , 1998.
- [2] Software Engineering, Ian Sommerville, Addison-Wesley, 9th edition, 2010.
- [3] Database Modeling and Design. Autores: Toby Teorey, Sam Lightstone, Tom Nadeau and H: Jagadish. ISBN: 978-012-382-020-4. Morgan Kaufmann, 5th Edition.
- [4] Bernd Bruegge and Allen H. Dutoit, 2011, Object-Oriented Software Engineering: Using UML, Patterns and Java. Prentice Hall.