



**INSTITUTO POLITÉCNICO** Escola Superior de Tecnologia  
**DE BRAGANÇA** e de Gestão

Sistemas Distribuídos

# Aplicação Meteorológica

Trabalho Prático N.2

Pedro Ribeiro a37557

Mário Machado a37549

-Ano Letivo 2018/2019- Professor Rui  
Lopes

# Conteúdo

Introdução .....	2
1.1 Descrição do Sistema .....	3
Descrição da Aplicação .....	3
2.1 Diagrama de Classes.....	3
2.2 Diagrama de Casos de Uso .....	4
2.3 Servidor .....	4
2.4 Cliente .....	6
2.5 Estrutura do sistema .....	6
2.5.1 Servidor .....	6
2.5.2 Cliente .....	7
Conclusão.....	7

## Capítulo 1

### Introdução

O objetivo deste trabalho é implementar uma aplicação web de meteorologia para armazenamento de parâmetros climáticos, constituída por dois módulos principais: o servidor e o cliente.

Este foi desenvolvido em Java, através de um ambiente de desenvolvimento integrado (IDE), recorrendo às tecnologias Enterprise Java Beans (EJB) e Java Persistence Architecture (JPA).

Adicionalmente, era pedida a implementação de forma a ser possível aceder ao servidor por intermédio de plataformas móveis. Este ponto foi implementado com recurso a Web Services.

## 1.1 Descrição do Sistema

A aplicação desenvolvida, de nome WeatherManager, foi desenvolvida recorrendo ao uso de uma arquitetura Cliente-Servidor, onde o cliente acede aos dados e funcionalidades que o servidor disponibiliza.

O sistema possui uma aplicação web, desenvolvida em Bootstrap para possibilitar responsividade, quer em Desktop como em dispositivos móveis. Nesta aplicação, é possível criar, eliminar e atualizar Localidades, assim como criar e eliminar dados de Temperaturas associadas a Localidades já existentes.

# Capítulo 2

## Descrição da Aplicação

A aplicação implementada assenta numa arquitetura Cliente-Servidor composta dois projetos: WeatherManager (Servidor) e WeatherManagerClient (Cliente).

### 2.1 Diagrama de Classes

O nosso sistema baseia-se num simples diagrama de classes, composto apenas por duas entidades: Location que corresponde a cada localização e WeatherData que inclui os vários parâmetros climáticos associados a uma determinada Location.

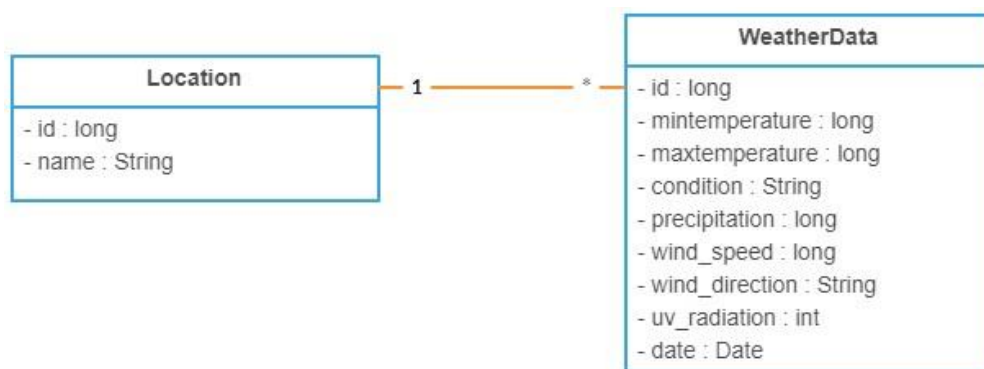
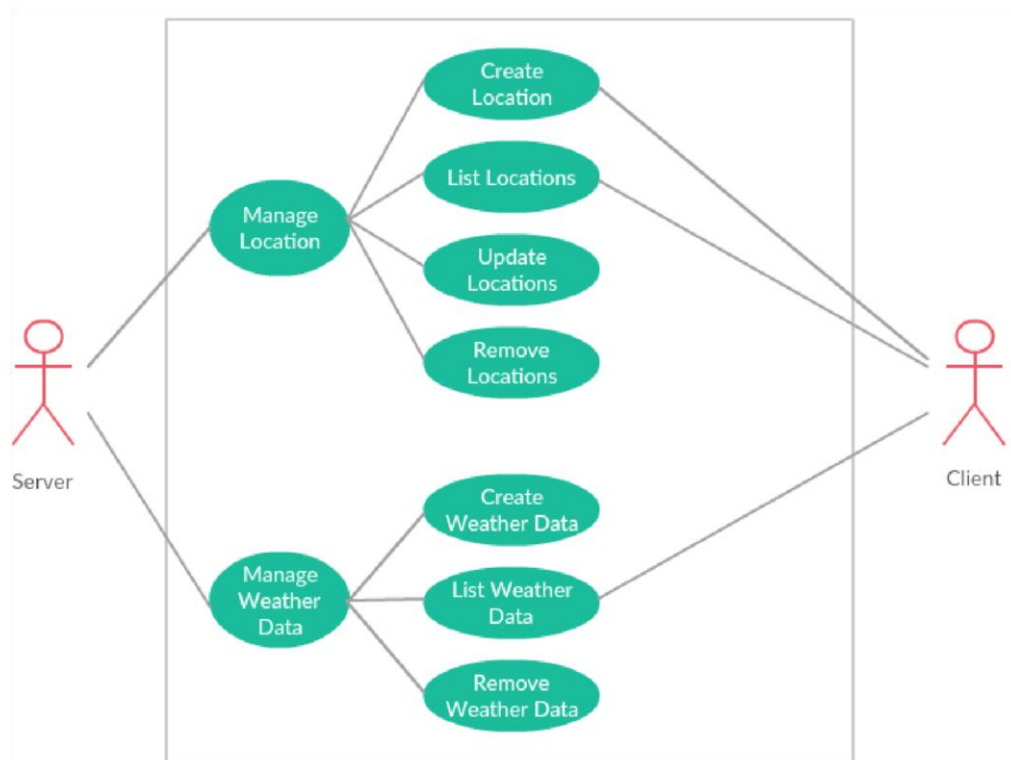


Figura 1 - Diagrama de Classes

## 2.2 Diagrama de Casos de Uso

A aplicação possui determinadas especificidades para cada entidade representada no diagrama de classes.

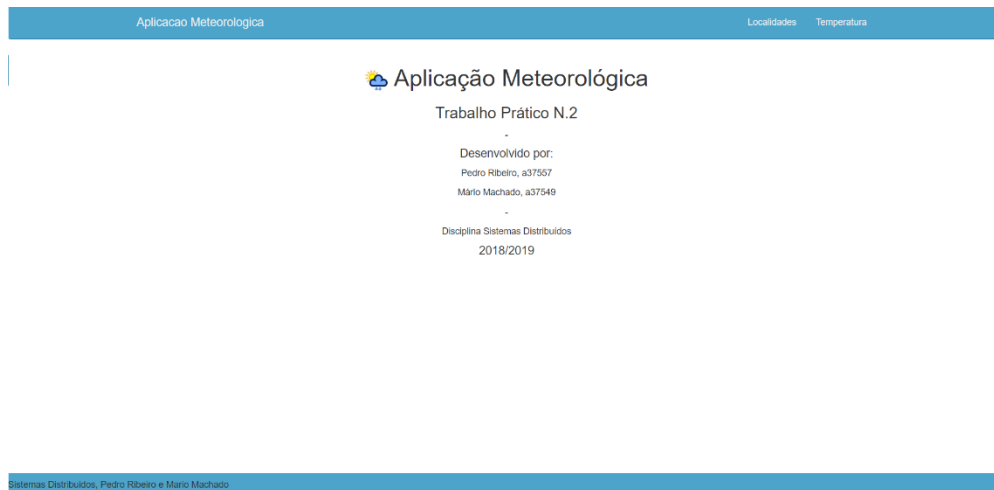
No servidor assenta a administração dos dados, estando implementadas as operações Create, Read, Update and Delete (CRUD). No cliente apenas estão implementadas algumas das operações do servidor para demonstração do funcionamento.



*Figura 2 - Diagrama de Casos de Uso*

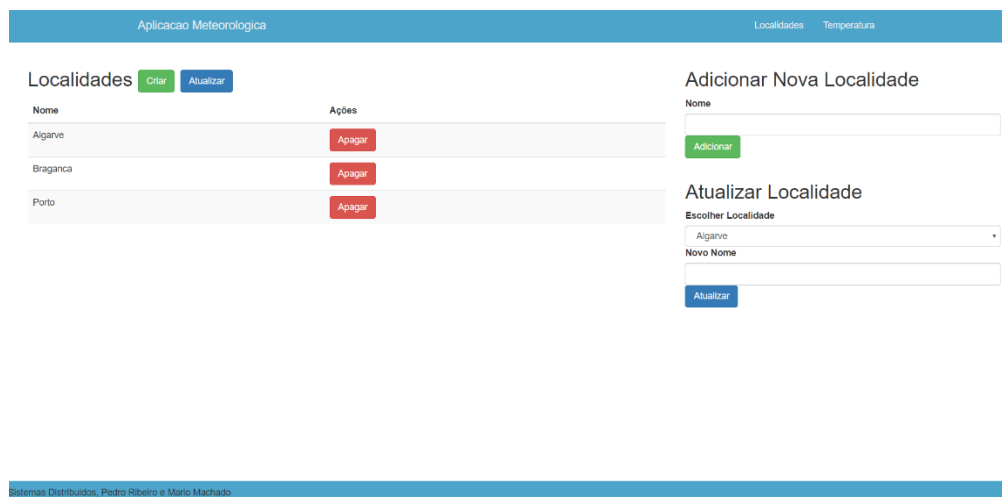
## 2.3 Servidor

O servidor consiste num projeto de web dinâmico, desenvolvido em Bootstrap para permitir responsividade em diferentes dispositivos.



*Figura 3 - Página inicial*

Na página representada na seguinte imagem (Figura 4), é possível adicionar, atualizar e remover Localidades assim como ver uma listagem das mesmas.



*Figura 4 – Página das Localidades*

Na página da seguinte imagem (Figura 5), é possível adicionar e remover Dados Meteorológicos assim como ver uma listagem dos mesmos.

Aplicação Meteorológica

Localidades Temperatura

### Temperatura

[Criar](#)

Escolher Localidade Ver Tudo [OK](#)

Localidade	Data	Min	Max	Condições	Precipitação	Vento	UV	Ações
Algarve	Sat Mar 04 17:11:11 GMT 2017	12	21	Céu Limpo	12	12 N	5	<a href="#">Apagar</a>
Bragança	Sun Mar 04 17:11:11 GMT 2018	12	21	Céu Limpo	12	12 N	5	<a href="#">Apagar</a>
Porto	Mon Mar 04 17:11:11 GMT 2019	12	21	Céu Limpo	12	12 N	5	<a href="#">Apagar</a>

### Adicionar Novo Registro

Localidade

Temperatura Mínima (°C)

Temperatura Máxima (°C)

Condição

Precipitação

Vento

Radiação UV

Data

[Adicionar](#)

Sistemas Distribuídos, Pedro Ribeiro e Mario Machado

Figura 5 - Página dos Dados da Temperatura

## 2.4 Cliente

No cliente foi desenvolvido uma aplicação Window Builder: WeatherDataManager. A aplicação deveria aceder ao servidor através de “*Context ctx = new InitialContext();*”. Mas por algum motivo desconhecido não funciona.

## 2.5 Estrutura do sistema

A solução encontrada engloba os módulos de servidor e de cliente, cada um composto por várias camadas.

### 2.5.1 Servidor

- **pt.ipb.sd.wm.entity**

Todas as entidades do sistema (Figura 1), onde se definem os atributos e métodos básicos para a funcionalidade do mesmo.

- **pt.ipb.sd.wm.ejb**

Camada de persistência de dados, composta pelos *beans* e as suas interfaces remotas que permitem realizar as operações básicas de Create Retrieve Update Delete (CRUD) sobre as entidades do sistema.

- **pt.ipb.sd.wm.backingbean**

Todos os Backing Beans necessários para a gestão das páginas Web.

- **pt.ipb.sd.wm.ws**

Onde são definidos todos os web services do sistema, as funcionalidades disponibilizadas para a aplicação de cliente.

- **WebApp**

Conjunto de páginas que permitem o acesso à aplicação através de um navegador Web.

### **2.5.2 Cliente**

- **pt.ipb.sd.WindowBuilder**

Aplicação Window Builder encarregue de apresentar os dados na aplicação de cliente Java.

## **Capítulo 3**

## **Conclusão**

Com a conclusão deste trabalho, achamos que ficamos aquém dos requisitos propostos. Foi implementado um serviço que permite fazer a manutenção de parâmetros meteorológicos de forma simples. A implementação do sistema revelou-se algo problemática em partes, com a ocorrência de certos erros que nos atrasaram um pouco no trabalho. Um desses erros ficou sem resolução, o que leva a que, cada vez que se inicie o Cliente seja executado o Servidor.

# Bibliografia

- [1] Retrieving JPA Entity Objects:  
<http://www.objectdb.com/java/jpa/persistence/retrieve>
- [2] Bootstrap : <http://getbootstrap.com/components>
- [3] Bootstrap Tutorial : <https://www.w3schools.com/bootstrap/>
- [4] WindowBuilder: <https://eclipse.org/windowbuilder>
- [5] RESTful Web Services: [https://www.tutorialspoint.com/restful/restful\\_jax\\_rs.htm](https://www.tutorialspoint.com/restful/restful_jax_rs.htm)