

Documentação Arquitetural Projeto ARQSI

**PEDRO GOMES – 1130383@ISEP.IPP.PT**

**RUBEN TEIXEIRA – 1140780@ISEP.IPP.PT**

**PEDRO ABREU – 1131219@ISEP.IPP.PT**

**ÍNDICE**

1. FURPS+.. ………………………………….. 2,3
2. Vista Lógica.. ………………………………….. 4
3. Vista Implantação. ………………………………….. 5
4. Vista Implementação.. ………………………………….. 6
5. Vista de Cenarios.. ………………………………….. 7

**FURPS+**

Funcionalidades

* ... desenvolver um widget para ambiente Web que permita a consulta seletiva desses dados.
* A consulta deve ser feita por facetas definidas pela informação dos dados existentes na base de dados.
* Quaisquer funcionalidades adicionais devem ser antecipadamente acordadas com o cliente/docente, pela apresentação de mokups ou protótipos da UI
* Uma aplicação denominada “Lugares” que permita o CRUD de Pontos de Interesse (POI)
* Uma API denominada “Cancela” que permita CRUD dos dados dos sensores e dos POI;
* Uma aplicação denominada “Visita” que permita a consulta de dados de sensores lidos

o mais próximo possível dum determinado POI a determinada data+hora (+/- uma diferença definida pelo utilizador).

Usabilidade

* Deve ser tomado em consideração que o utilizador espera que tarefas similares sejam feitas de maneira similar. Para semânticas similares, devem ser usadas sintaxes similares, e vice-versa
* A interface deverá prever e prevenir possíveis erros por parte do utilizador

Fiabilidade/Confiabilidade

Desempenho

Suportabilidade

* A UI do widget deverá ser pensada para ser usada como widget em várias páginas Web, pelo que o seu aspeto deve poder variar (cf. Anexo 2) conforme a aplicação Web hospedeira
* O widget deve funcionar pelo menos nos navegadores Firefox (desktop) e Chrome (desktop).

Restrições de design

* Adotar-se-á um processo de desenvolvimento de software iterativo e incremental em que os requisitos irão ser progressivamente especificados ao longo do semestre e em que devem ser feitas entregas/demonstrações frequentes do software ao cliente/docente, que emite opinião (i.e. feedback sobre o software desenvolvido) que deve servir para melhorar o software.
* Documentação arquitetural completementar à apresentada na secção 2

Restrições de implementação

* A base de dados será acedida através duma API REST
* O widget deve ser desenvolvido em HTML, JavaScript, AJAX, JSON/XML e CSS.
* Qualquer outra biblioteca ou tecnologia a usar deve ser antecipadamente acordada com o cliente/docente.
* Existe uma base dados única denominada “Datum”
* Lugares” liga-se a “Datum” através de Entitiy Framework (EF)
* “Cancela” liga-se a “Datum” através de Entitiy Framework (EF)
* “Visita” acede à informação via “Cancela”
* Excetuando o Widget, todas as partes do sistema devem ser desenvolvidas em ASP.Net e em particular ASP.Net MVC e ASP.Net Web API
* Deve ser usado preferencialmente o SGBD MS SQL Server
* O Widget desenvolvido na iteração 1 também deve considerar os sensores disponíveis através de “Cancela”

Restrições de interface

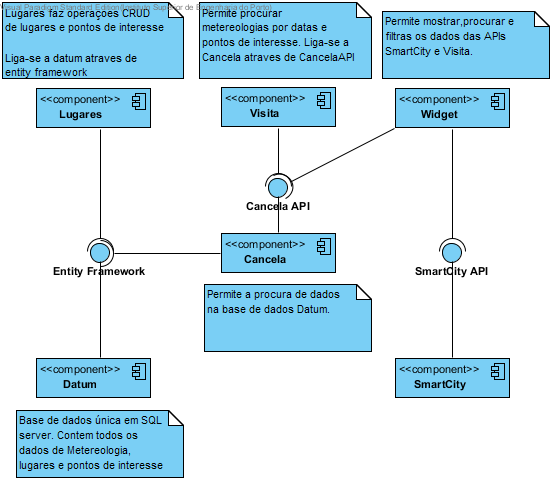
* A interface deve ser dinâmica em função dos dados recolhidos da API

Restrições físicas

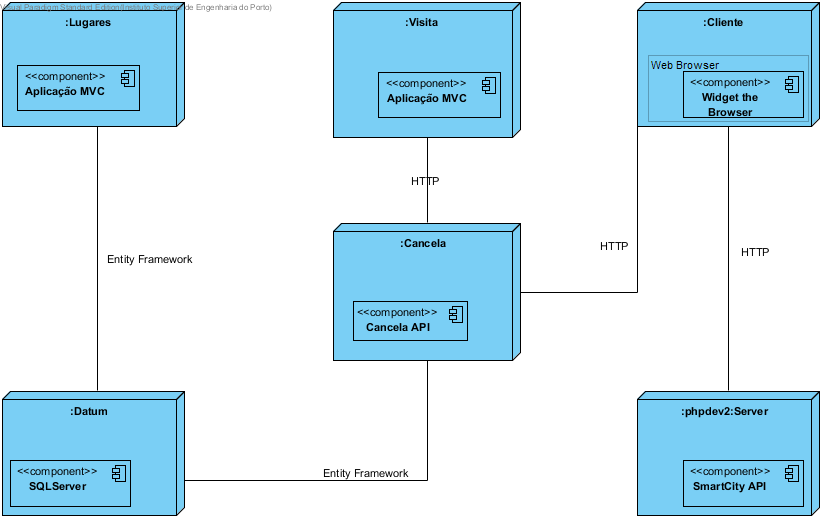
Decisões Arquiteturais

* Cada faceta é um elemento individual
* Cada input da faceta é um elemento individual
* O codígo javascript é separado em Utils.js(funções genéricas) Objects.js(objetos utilizados) e Smartcity.js (widget)

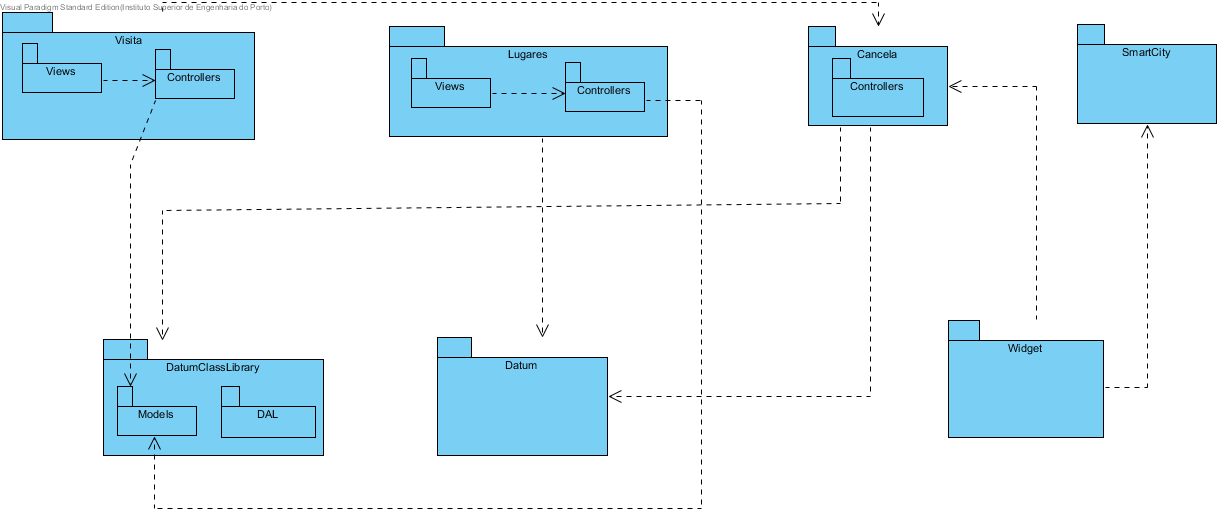
**Vista Lógica**

****

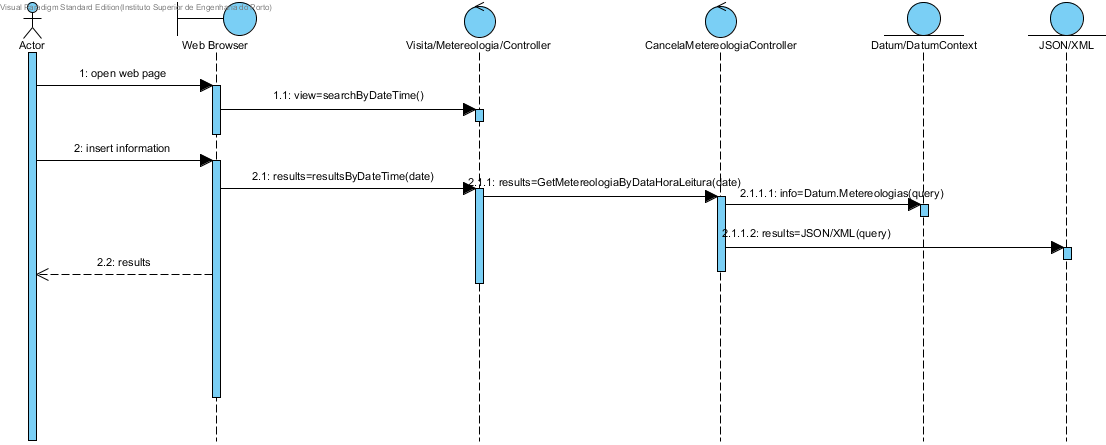
**Vista de Implantação**

****

**Vista de Implementação**

****

**Vista de Cenários**

****

