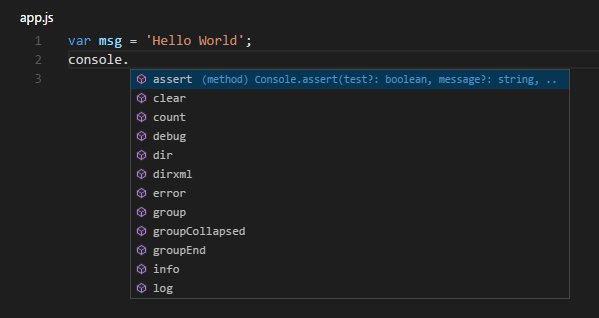
**Arquitetura de Sistemas**

**Licenciatura Engenharia Informática**

**Título**: *Relatório* *Projeto 2*



**Data de Entrega:** 11 de novembro

**Autores:**

Joel Coelho

Rui Cruz 1160823

***Índice***

***[Introdução](#_Toc529659736)*** [3](#_Toc529659736)

[***Modelo de Domínio*** 4](#_Toc529659737)

[***Diagrama de Componentes*** 5](#_Toc529659738)

[***Diagrama de Packages*** 5](#_Toc529659739)

[***Diagrama de Nós*** 6](#_Toc529659740)

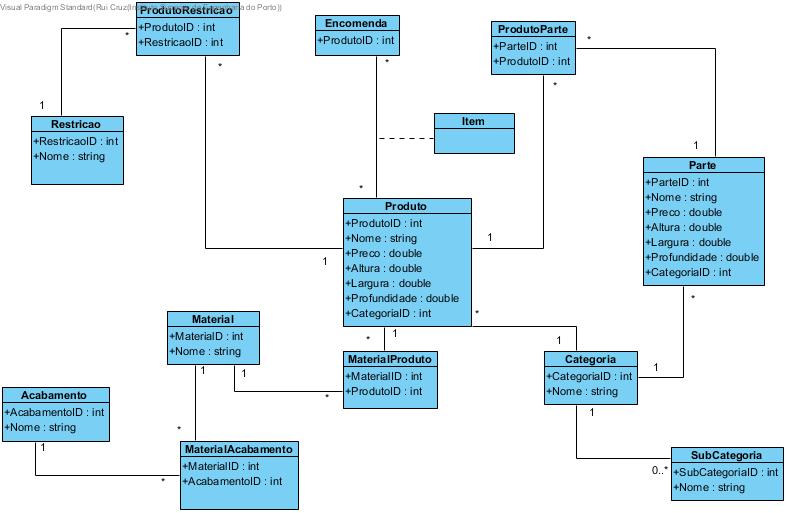
[***Conclusão*** 7](#_Toc529659741)

[***Bibliografia*** 8](#_Toc529659742)

## ***Introdução***

No âmbito da unidade curricular de ARQSI (Arquitetura de Sistemas) do curso de Engenharia Informática do Instituto Superior de Engenharia do Porto, foi-nos proposto a realização de um projeto com um conjunto de requisitos. Um desses requisitos é a elaboração de um relatório onde estejam as explicações para a documentação criada neste mesmo projeto.

## ***Modelo de Domínio***



Um produto é caraterizado por um ProdutoID e possui vários atributos como nome, preço, dimensão (altura, largura e profundidade) e uma categoria.

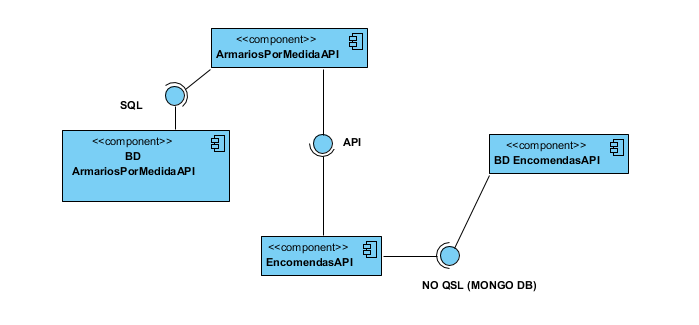
Uma categoria tem um CategoriaID e uma subcategoria (caraterizada por SubCategoriaID e nome).

Um produto tem vários materiais e um material tem vários acabamentos. Para exprimir estas ligações foram utilizadas “tabelas auxiliares” como MaterialProduto e MaterialAcabamento. A primeira tabela tem como chaves primárias MaterialID e ProdutoID, já a segunda é MaterialID e AcabamentoID.

Um produto tem várias restrições e uma restrição pode pertencer a vários produtos. Por isso criamos a tabela ProdutoRestricao (ProdutoID e RestricaoID são as chaves primárias), de forma a exprimirmos esta relação de muitos para muitos.

Falta ainda exprimir a relação de muitos para muitos existente entre produto e parte e entre produto e encomenda. Para isso foram criadas as tabelas ProdutoParte (ProdutoID e ParteID) e Item.

## ***Diagrama de Componentes***

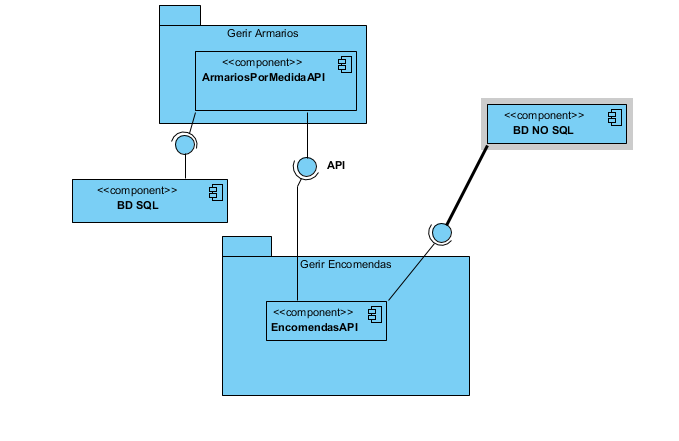


Na primeira parte do projeto, criamos a API ArmariosPorMedidaAPI que comunica com uma base de dados através da linguagem SQL.

Nesta segunda iteração, foi implementada a EncomendasAPI que utiliza um mecanismo NoSQL (Mongo DB).

De salientar que estas duas API’s (ArmariosPorMedidaAPI e EncomendasAPI) comunicam entre si, uma vez que existem dados que interessam a uma aplicação e que se encontra na outra.

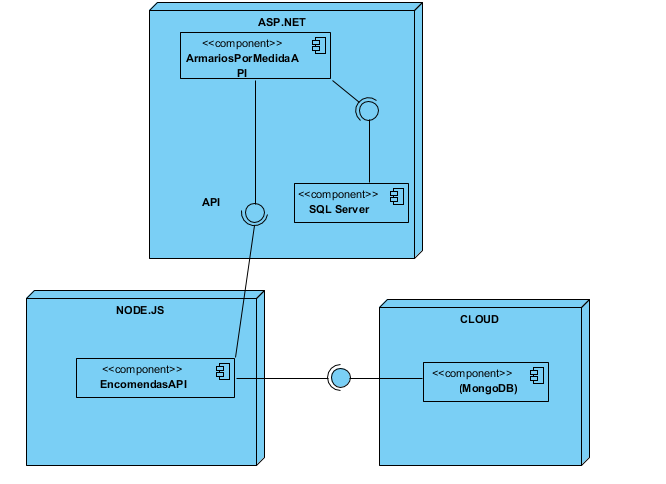
## ***Diagrama de Packages***



Na primeira parte do projeto, criamos o package Gerir Armarios onde se encontra ArmariosPorMedidaAPI que comunica com uma base de dados através da linguagem SQL.

Nesta segunda iteração, foi implementada a EncomendasAPI, localizado no package Gerir Encomendas que utiliza um mecanismo NoSQL (Mongo DB).

## ***Diagrama de Nós***



Na primeira parte do projeto, criamos o package Gerir Armarios onde se encontra ArmariosPorMedidaAPI que comunica com uma base de dados através da linguagem SQL. Esta API foi desenvolvida em APS.NET.

Nesta segunda iteração, foi implementada a EncomendasAPI, localizado no package Gerir Encomendas que utiliza um mecanismo NoSQL (Mongo DB). Esta aplicação foi desenvolvida em Node.js.

## ***Conclusão***

O projeto foi desenvolvido de forma cooperativa e solidária, uma vez que todos se empenharam de forma irrepreensível, tendo cada um realizado as tarefas que lhe foram atribuídas e auxiliando os colegas nas suas, sempre que necessário. Podemos também dizer, que ao longo do desenvolvimento do projeto, tentámos sempre implementar da forma mais eficiente possível, tentámos fazer mais com menos.

## ***Bibliografia***

* Material fornecido no Moodle da disciplina de Arquitetura de Sistemas (ARQSI).