

Requisitos e Capacidades do Produto

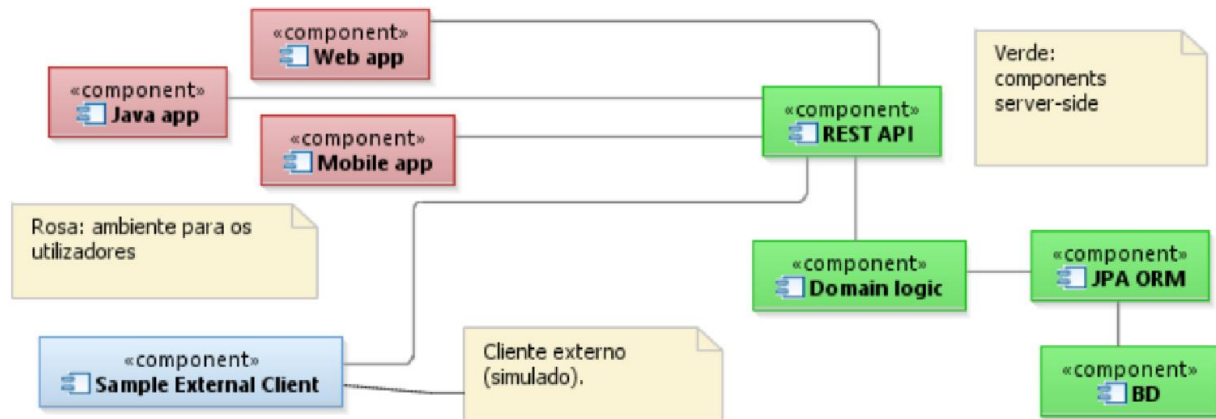
Projeto: **Note Keeper**
Preparado por: **Gonçalo Pinto - 65716**
Data: **05-07-2017**

Conteúdo do Relatório

Solução técnica.....	2
Arquitetura Pretendida	2
Arquitetura Efectuada.....	2
API de integração	3

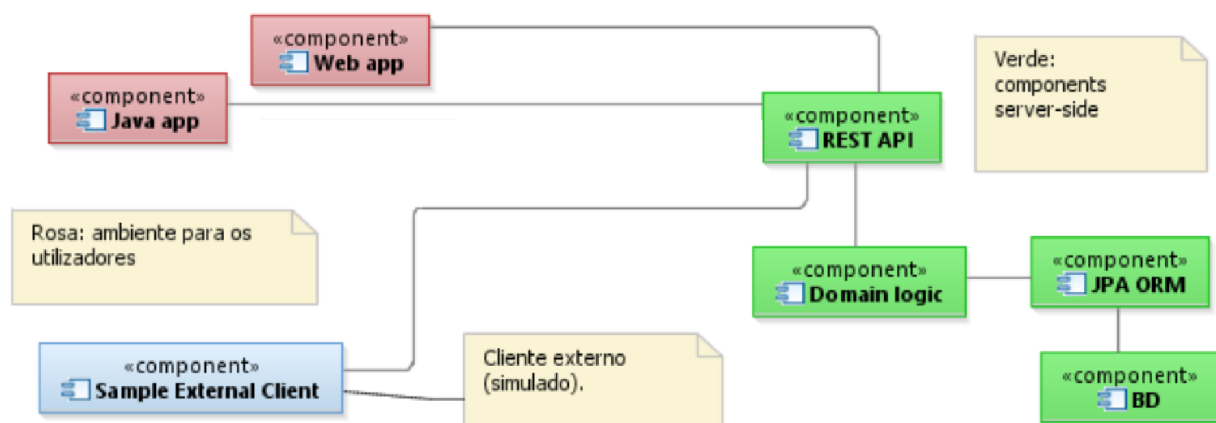
Solução técnica

Arquitetura Pretendida



- Web + Java App: Feita num projecto Java Web Maven App, a componente web utiliza xhtml para as suas páginas.
- BD: A base de dados usada foi Java Databases, a mesma encontra-se conectada ao projecto.
- RestAPI: Existe uma RestAPI que devolve numa página os dados da base de dados em formato XML.
- JPA: É usado neste trabalho para tratar da persistência entre a BD e o Trabalho/Interface.
- Sample External Client: Foi criado um RestClient que contém um URL, através deste URL é possível ir buscar o serviço do trabalho/BD.

Arquitetura Efectuada



Não realizei a parte “Mobile app”, para tal inicialmente pensei em utilizar páginas HTML5/CCS3 (BootStrap). Porém com os poucos dias e para conseguir uma melhor integração, usei as páginas por defeito .xhtml. A Interface corre em aplicações móveis, porém apenas de forma estática. A Interface corre em Computador a 100%, isto é, não de forma estática, está totalmente integrada com o código.

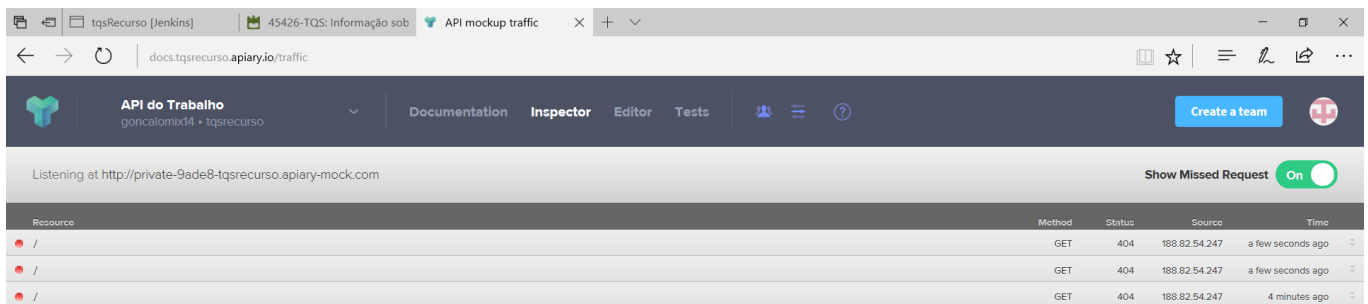
API de integração

O trabalho utiliza tanto XML como JSON, porém esta parte não ficou totalmente completa. Apenas está completo a parte do Rest que indica uma página com os dados da base de dados, estes dados são mostrados em XML.

O site apiary.io e o meu repositório no GitHub estão interligados, o link de ligação: [goncalomix14/tqsRecurso/apiary.apib](https://github.com/goncalomix14/tqsRecurso/apiary.apib)

Comecei a descrever o processo no site apiary.io (<http://docs.tqsrecurso.apiary.io/#>) , porém como para ter um melhor “tutorial” precisaria de mais funcionalidades no trabalho.

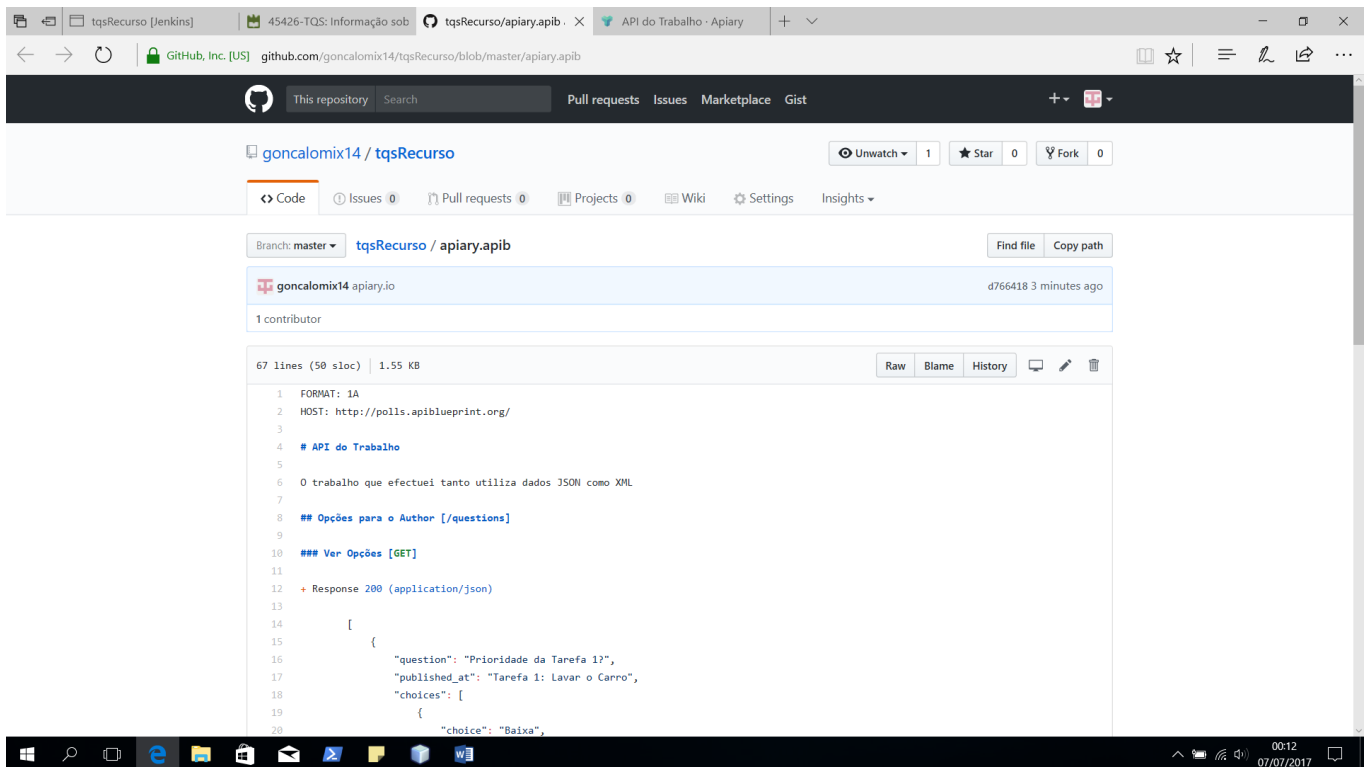
Com o apiary.io é possível usar este link <http://private-9ade8-tqsrecurso.apiary-mock.com> (apenas para o meu trabalho) e comunicar entre os GET e POST (XML e JSON). Porém como referido, não ficou totalmente finalizada esta parte, por isso será sempre dada a mensagem de erro.



Resource	Method	Status	Source	Time
/	GET	404	188.82.54.247	a few seconds ago
/	GET	404	188.82.54.247	a few seconds ago
/	GET	404	188.82.54.247	4 minutes ago



Caso o professor tenha interesse em consultar esta secção, como esta parte está integrada com o repositório GitHub, através de lá consegue aceder, terá uma página deste género:



The screenshot shows a web browser window displaying a GitHub repository page. The repository is named 'goncalomix14 / tq\$Recurso'. The file 'apiary.apib' is selected, showing its content. The file is 67 lines long (50 sloc) and 1.55 KB. The content is an API specification in API Blueprint format. The browser's address bar shows the URL 'github.com/goncalomix14/tq\$Recurso/blob/master/apiary.apib'. The Windows taskbar is visible at the bottom, showing the time as 00:12 on 07/07/2017.

```
1 FORMAT: 1A
2 HOST: http://polls.apiblueprint.org/
3
4 # API do Trabalho
5
6 O trabalho que efectuei tanto utiliza dados JSON como XML
7
8 ## Opções para o Autor [/questions]
9
10 ### Ver Opções [GET]
11
12 + Response 200 (application/json)
13
14 [
15   {
16     "question": "Prioridade da Tarefa 1?",
17     "published_at": "Tarefa 1: Lavar o Carro",
18     "choices": [
19       {
20         "choice": "Baixa",
```