



RELATÓRIO DE DEFINIÇÃO DE ARQUITETURA

HEALTH +

LABORATÓRIO DE
GESTÃO DE PROJETO

6 DE JUNHO DE 2014

FACULDADE DE ENGENHARIA
DA UNIVERSIDADE DO PORTO

ANTÓNIO GARCEZ
MÁRIO AGUIAR
PEDRO ALVES
RICARDO TEIXEIRA
VÍTOR SILVA



HEALTH +

RELATÓRIO

DE DEFINIÇÃO DE ARQUITETURA

LABORATÓRIO

DE GESTÃO DE PROJETO

FACULDADE DE ENGENHARIA

DA UNIVERSIDADE DO PORTO

ÍNDICE

ÍNDICE.....	2
01 GLOSSÁRIO	4
02 INTRODUÇÃO	5
03 FERRAMENTAS E TECNOLOGIAS.....	6
3.1. FERRAMENTAS	6
3.2. TECNOLOGIAS	7
04 ARQUITETURA LÓGICA.....	10
04 ARQUITETURA FÍSICA	12
06 BIBLIOGRAFIA	14

01 GLOSSÁRIO

Sigla/Nome	Definição
GIT	É um sistema de gestão de versões e de código. Cada projeto em GIT funciona como um repositório, havendo várias soluções <i>online</i> com sistema GIT que oferecem um repositório remoto.
IDE	Integrated Development Environment é um tipo de aplicação que reúne variadas ferramentas e funcionalidades que focam na agilização do processo de desenvolvimento de <i>software</i> .
.NET	O .NET é um ambiente de execução de <i>software</i> que abriga diversas linguagens como C# ou Visual Basic.NET.
C#	É uma linguagem de programação orientada a objetos, é parte integrante do da plataforma .NET, tendo muitas das características do C++ mas também de outras linguagens como JAVA.
Javascript	Javascript é uma linguagem de programação orientada a objetos sendo usada essencialmente por <i>browsers</i> , correndo localmente no pc do utilizador. O desenvolvimento nesta linguagem é abstraído de diversas funções e características de outras linguagens populares, por ser uma linguagem de um nível mais alto.
HTML	HTML é uma linguagem de marcação utilizada para produzir páginas na Web, tendo como versão atual o HTML5.
CSS	CSS ou Cascading Style Sheets, é uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de estruturação, como HTML ou XML.
AJAX	É o uso metodológico de tecnologias como Javascript, para tornar páginas Web mais interativas com o usuário, utilizando solicitações assíncronas de informações.
HTTP	É um protocolo de comunicação utilizado para sistemas de informação, distribuídos e colaborativos, sendo ele a base para a comunicação de dados da Web.
REST	REST é acrónimo para Representational State Transfer, esta tecnologia usa quatro métodos HTTP para executar diferentes operações, sendo estas GET, POST, PUT e DELETE.

02 INTRODUÇÃO

Este documento tem como foco enquadrar o projeto Health+ num contexto técnico, analisando fatores como *user stories*, que são apresentadas no Relatório de Especificação de Requisitos, produzindo soluções que permitem obter uma visão quer a nível de *software* como de hardware sobre os sistemas envolvidos.

Sendo a Health+ um projeto que requer a existência de diversos sistemas, como uma aplicação mobile e um *website* convencional, é necessária a utilização de diferentes tecnologias e ferramentas para desenvolver com eficiência e qualidade que são pretendidas.

Com efeito, neste relatório é realizada uma análise às diferentes tecnologias a serem utilizadas, às diferentes características num nível de *software* e de hardware, e sobre as decisões chave tomadas no desenho da arquitetura.

03 FERRAMENTAS E TECNOLOGIAS

3.1. FERRAMENTAS

De seguida são analisadas as ferramentas a utilizar, focando as vantagens de cada uma e alguns pontos de diferenciação em relação a outras ferramentas.

- **Bitbucket:** O Bitbucket é o ponto de controlo de versões do projeto, servindo de repositório central de ficheiros relacionados com as soluções a desenvolver. Tendo esta ferramenta a opção de utilizar GIT ou Mercurial [1], como sistema de revisão, optou-se por utilizar GIT, pela facilidade de utilização e pela experiência da equipa com o sistema. Esta ferramenta permite ainda a criação de um repositório privado gratuitamente, ao contrário de outras ferramentas similares, como o GitHub, que só permitem serem públicas, tornando o processo de desenvolvimento mais seguro e privado.
- **Appcelerator Studio:** Para desenvolver as soluções mobile será utilizado o Appcelerator Studio. Produzido especificamente para Appcelerator Titanium, é imperativo o uso deste IDE que simplifica o uso desta tecnologia.
- **Microsoft Visual Studio 2013:** Produzido pela Microsoft, é um IDE que permite ao utilizador ter uma experiência de desenvolvimento fluida e integrada, tendo a opção de criar várias soluções para o mesmo projeto. O Visual Studio foca-se principalmente em ambientes de produção para linguagens. NET fazendo com que o processo de desenvolvimento nos mesmos, como neste projeto, seja realizado de uma forma mais eficaz para a equipa, havendo diversas ferramentas que automatizam muitos dos processos necessários.
- **Sublime Text 3:** O Sublime Text é um editor de texto com inúmeras opções de personalização do ambiente de trabalho ao utilizador, contendo opções de marcação de sintaxe, gestão de indentações e formatação de código para

uma extensa lista de linguagens que pode ser aumentada recorrendo ao uso de extensões que o Sublime Text permite adicionar, sendo estas extensões encontradas em repositórios da comunidade que suporta este editor.

- **Microsoft Windows 7, Microsoft Windows 8:** Visto que o .NET está intrinsecamente relacionado com o projeto, optar-se-á por utilizar um sistema operativo Microsoft, nomeadamente o 7 e o 8, para produzir as soluções, tanto por um ponto de vista de gestão de performance, como a disponibilidade de ferramentas para os sistemas operativos da empresa.

3.2. TECNOLOGIAS

Neste capítulo são apontadas algumas das características das tecnologias a serem usadas e as suas relações.

- **ASP.NET MVC:** O ASP.NET é uma *framework* específica para a criação de soluções Web, que para além da facilidade de utilização da *framework*, por se enquadrar nos paradigmas do .NET a que muitos utilizadores estão habituados, permite utilizar muitas das bibliotecas disponibilizadas pelo sistema operativo Windows. Neste projeto utilizar-se-á esta *framework* recorrendo ao uso da linguagem de programação C# [2].

O padrão de desenho MVC - Model View Controller- é parte integrante da ideologia do ASP.NET, onde existe uma separação das soluções em camadas, a camada do Modelo, representado pelo modelo de domínio apresentado no Relatório de Especificação de Requisitos, a camada da Visão, que é a parte da interface mostrada ao utilizador final, e a camada dos Controladores, que serve como ponte entre as duas camadas anteriores, gerindo a informação introduzida e produzida pelas camadas.

- **MySQL:** O gerenciamento da base de dados é deixado a cargo do MySQL, que utiliza a linguagem SQL para permitir ao utilizador efetuar operações na base de dados [3]. Esta tecnologia tem vantagens como ter um excelente

desempenho e documentação, suportar praticamente todas as plataformas e ser compatível com bastantes linguagens como o C# que contém bibliotecas compatíveis com o MySQL.

- **Entity Framework:** Esta framework é do tipo ORM, que oferece aos desenvolvedores um mecanismo automatizado para aceder e armazenar os dados na base de dados e trabalhar com os resultados. Suporta vários tipos de bases de dados como SQL server, Oracle, DB2 e MySQL. O Entity é uma ferramenta para o armazenamento de dados de objetos de domínio para bases de dados de forma automática, sem muita programação. A ferramenta ajuda a manter a base de dados separada do domínio, fazendo com que o produto seja mais sustentável e extensível, automatizando ainda operações do tipo CRUD (Create, Read, Update & Delete) [4].
- **HTML5:** Esta tecnologia é uma evolução do tradicional HTML, contendo inúmeras novidades a nível de possibilidade de estruturação de código, adiciona ainda opções de automatização para adicionar conteúdo multimédia. O HTML5 é considerado como o novo standard para colocar informação online.
- **CSS3:** Tal como o HTML5, o CSS3 é uma evolução do CSS, trazendo novos fatores de personalização de páginas Web, é essencial para a simplificação e unificação da forma como se produz visualmente páginas Web.
- **jQuery:** jQuery é uma biblioteca de Javascript que permite ao utilizador ter processos como pedidos AJAX extremamente simplificados em relação ao processo normal. É ainda uma biblioteca com uma curva de aprendizagem bastante rápida e de fácil utilização.
- **Appcelerator Titanium:** O Appcelerator Titanium é uma tecnologia *open-source* que traduz código desenvolvido em Javascript para código nativo mobile, inclusive a geração para diferentes sistemas operativos em

diferentes linguagens, como em Objective-C (iOS) ou Java (Android) [5]. O Appcelerator Titanium oferece ainda outras vantagens como criar experiências de utilizador atraentes com serviços na *cloud*, como notificações *push* e *check-ins*. Esta ferramenta destaca-se da concorrência como por exemplo PhoneGap ou AppMobi que usam uma *webview* num *container* com código HTML e Javascript, dando uma sensação de aplicação nativa ao utilizador. No entanto esta arquitetura fica muito longe do nível de performance e de experiência de utilizador que uma solução nativa com interfaces compostas de elementos nativos ao sistema pode oferecer.

04 ARQUITETURA LÓGICA

Nesta secção é exposta a arquitetura lógica dos diferentes sistemas, isto é, quais as relações entre as variadas componentes a nível de *software* que estão encarregues a cada sistema, particularizando-se em interface que contém as componentes referentes à visualização da informação, a lógica de negócio que trata de relacionar as componentes do modelo de domínio a um nível mais elevado de compreensão por módulos, e o acesso a dados e dados, que simbolizam as tecnologias utilizadas para gerir a informação nos sistemas.

Como é possível observar no diagrama abaixo, existe uma diferenciação entre todas as interfaces do sistema, não havendo essa total diferenciação na lógica de negócio, isto pois o Sistema Mobile e o Sistema Web Cliente contêm praticamente as mesmas funções. Em relação aos dados verifica-se que existe uma unificação entre todos os sistemas, onde os mesmos encontram-se ligados a um único servidor central responsável por gerir a informação.

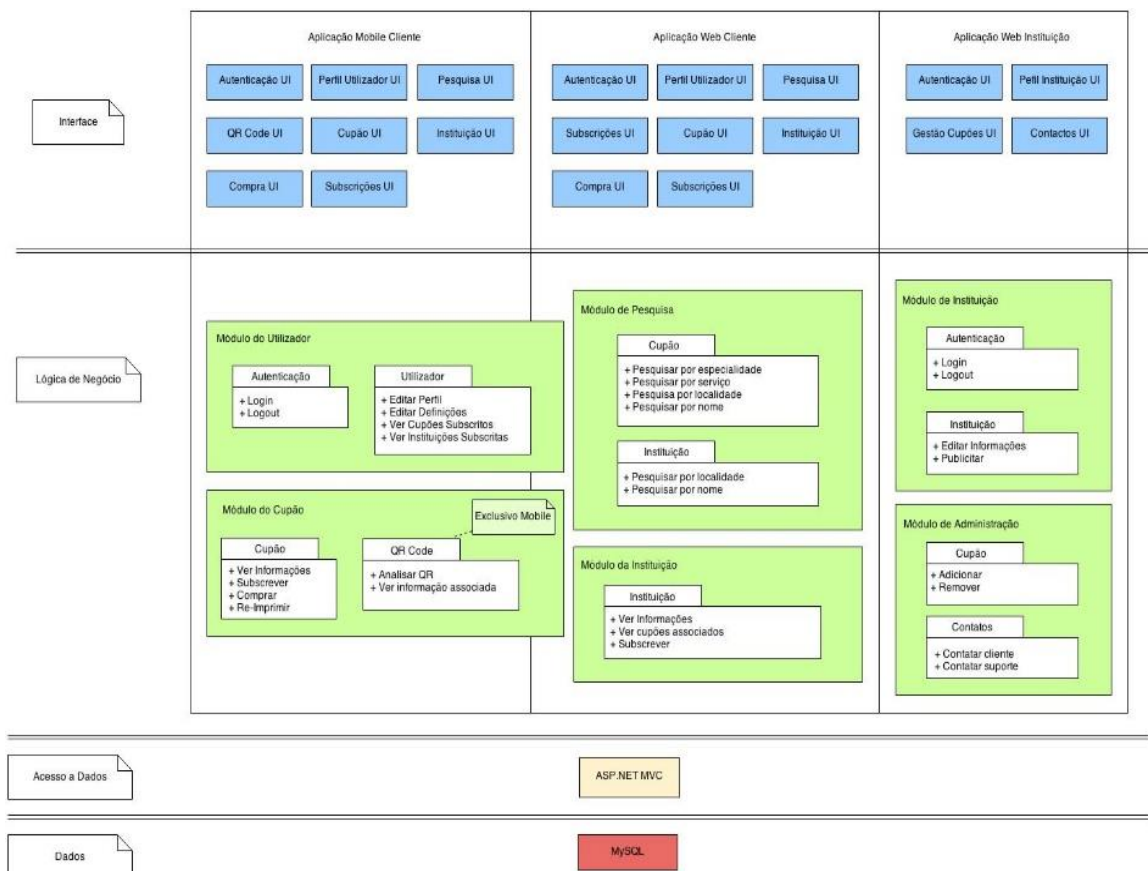


Figura 1- Arquitetura Lógica

04 ARQUITETURA FÍSICA

Esta secção tem como objetivo fornecer uma visão geral sobre a arquitetura física das soluções. São apresentados dois tipos de diagramas, um diagrama de instalação que oferece uma análise a nível de *hardware* e um diagrama de componentes que relaciona diferentes componentes a nível de *software*.

O diagrama de instalação mostrado de seguida decompõe o projeto em vários sistemas, existindo um servidor central responsável por interligá-los, uma base de dados para armazenar informação permanentemente, e finalmente os sistemas do utilizador, o PC e o Mobile. Todos os sistemas, exceto a base de dados, comunicam entre si através do protocolo HTTP num ambiente REST, garantindo uniformização nas mensagens trocadas.

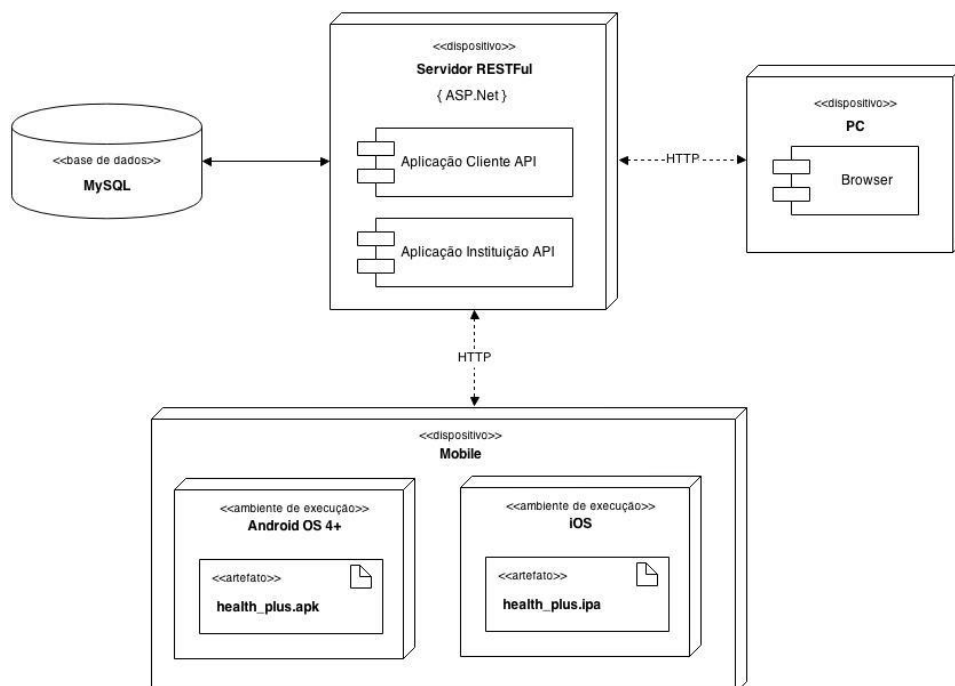


Figura 2- Diagrama de Instalação

Abaixo encontra-se o diagrama de componentes que expõe as necessidades de cada módulo dos sistemas, isto é, para cada módulo existem relações com outros módulos que indicam se o componente requer ou cede informação.

Como caso de exemplo, no sistema Web Instituição pode-se verificar que o Administrador carece de autenticação, como tal este o seu módulo correspondente requer informação do módulo Autenticação, que por sua vez fornece informação ao módulo Administrador.

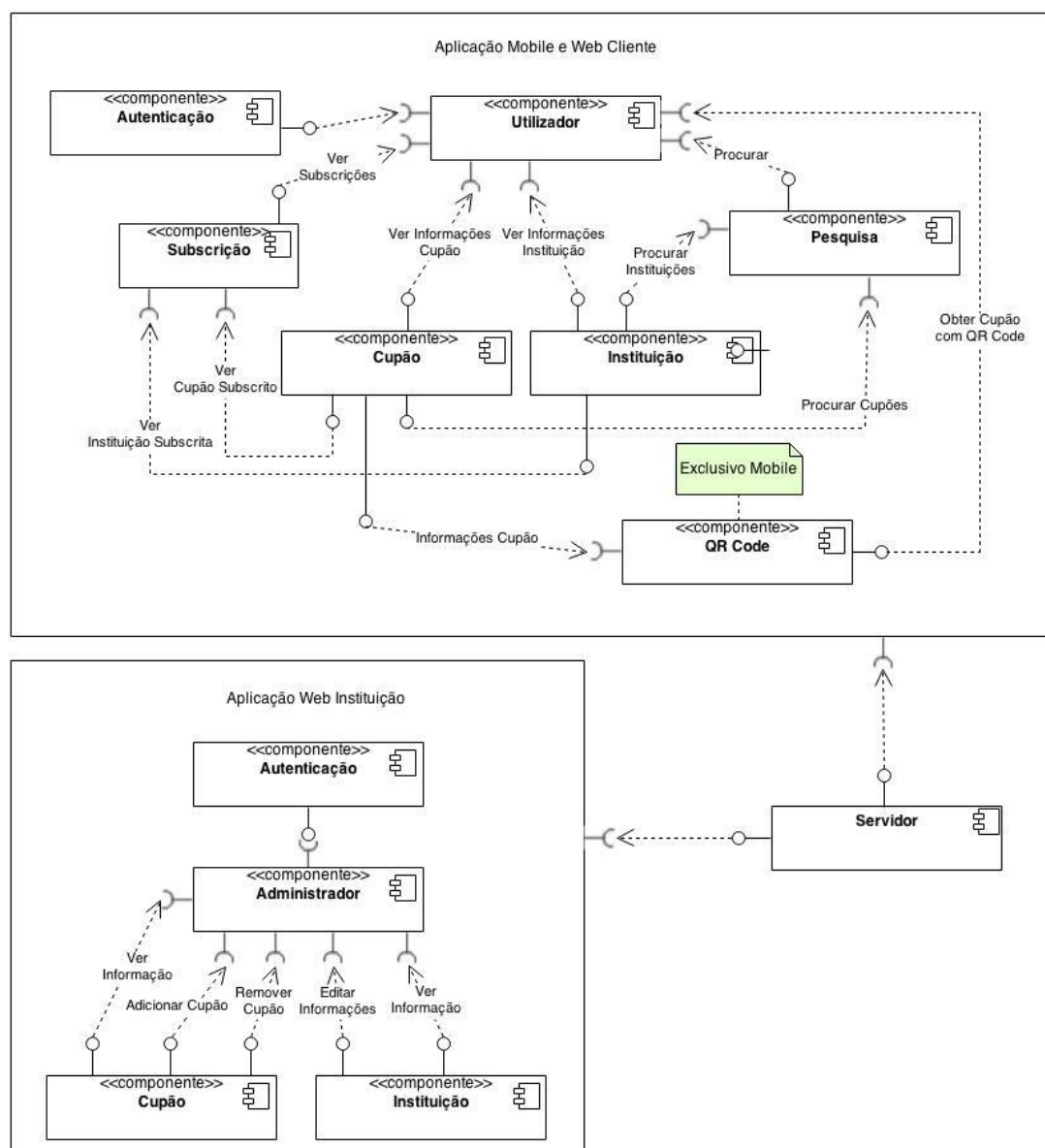


Figura 3 - Diagrama de Componentes

06 BIBLIOGRAFIA

[1] Wikipédia : “Bitbucket”, 2014 [Online] Disponível em <http://en.wikipedia.org/wiki/Bitbucket> (Março 2014)

[2] MSDN : “ASP.NET Overview”, 2012 [Online] Disponível em <http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx> (Março 2014)

[3] Wikipédia : “MySQL”, 2014 [Online] Disponível em <http://pt.wikipedia.org/wiki/MySQL> (Março 2014)

[4] Microsoft Virtual Academy: “Fundamentos do Entity Framework”, 2013 [Online] Disponível em <http://www.microsoftvirtualacademy.com/training-courses/fundamentos-do-entity-framework-4#?fbid=7Hs7mApCzlr> (Março 2014)

[5] Appcelerator “Titanium Mobile Development Environment”, 2014 [Online] Disponível em <http://www.appcelerator.com/titanium/> (Março 2014)