

UNIVERSIDADE DE COIMBRA  
FACULDADE DE CIÊNCIAS E TECNOLOGIAS  
DEPARTAMENTO DE ENGENHARIA INFORMÁTICA  
ENGENHARIA DE SOFTWARE

## **Architecture and Design of Software**

**VERSION 0.4**

**WEB DASHBOARD PARA O GIT**

# **EDGESOFT**

**André Duarte  
Diana Pereira  
Gonçalo Pinto  
Henrique Pereira  
João Costa  
João Ferreiro  
Joel Pires**

**2014214845  
2013150712  
2014202176  
2011154576  
2013144034  
2014197760  
2014195242**

**14 de Outubro de 2016**

# Conteúdo

<b>1</b>	<b>TABELA DE REVISÕES</b>	<b>2</b>
<b>2</b>	<b>ARQUITETURA E DESIGN DO SOFTWARE</b>	<b>3</b>
2.1	PADRÃO DE ARQUITETURA . . . . .	3
2.2	DIAGRAMA DA ARQUITETURA DE ALTO NÍVEL . . . . .	4
2.3	DESIGN DETALHADO . . . . .	5
2.3.1	SERVIDOR . . . . .	5
2.3.2	CLIENTE . . . . .	6
2.3.3	BASE DE DADOS . . . . .	6
2.4	TECNOLOGIAS USADAS . . . . .	7

# 1 TABELA DE REVISÕES

Versão	Data	Autor	Descrição
0.4	14/10/2016	Joel & Henrique	Revisão do conteúdo dos diagramas
0.3	14/10/2016	Joel & Henrique	Revisão do conteúdo dos diagramas
0.2	13/10/2016	Joel & Henrique	Criação dos diagramas
0.1	13/10/2016	Joel & Henrique	Estrutura inicial do documento

## 2 ARQUITETURA E DESIGN DO SOFTWARE

### 2.1 PADRÃO DE ARQUITETURA

O padrão arquitetural escolhido foi o **MVC (Model View Controller)** uma vez que é o mais adequado e usado para aplicações web. A própria framework escolhida para o desenvolvimento da aplicação (Meteor) facilmente se adapta a este modelo. Esta arquitetura permite-nos dividir um grande problema em vários problemas menores e de menor complexidade, excelente para aplicar modelo de processos adotado (SCRUM). O facto de haver uma divisão entre camadas faz com que possamos trabalhar independentemente em diferentes partes da aplicação sem afetar o “workflow” (aspeto relevante quando uma equipa tem diferentes “skill sets”). O reaproveitamento, limpeza, alteração e manutenção do código também fica facilitado.

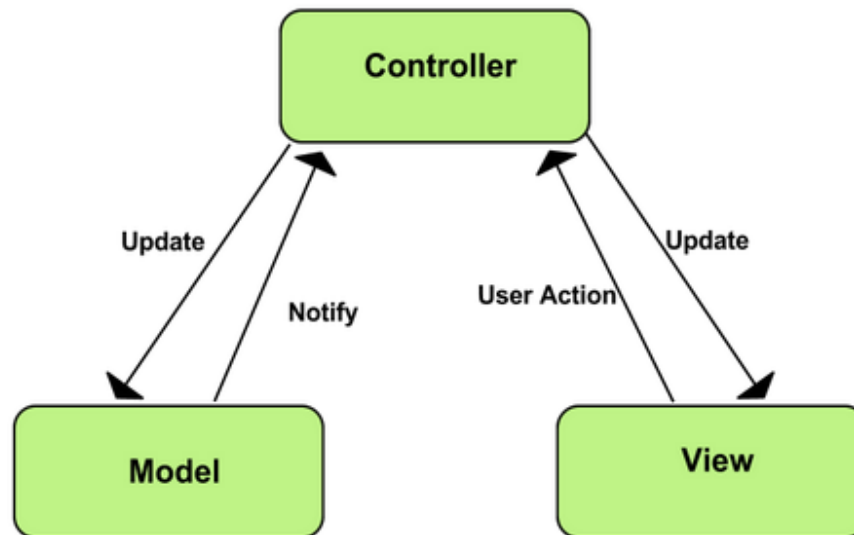


Figura 1: Diagrama do MVC

## 2.2 DIAGRAMA DA ARQUITETURA DE ALTO NÍVEL

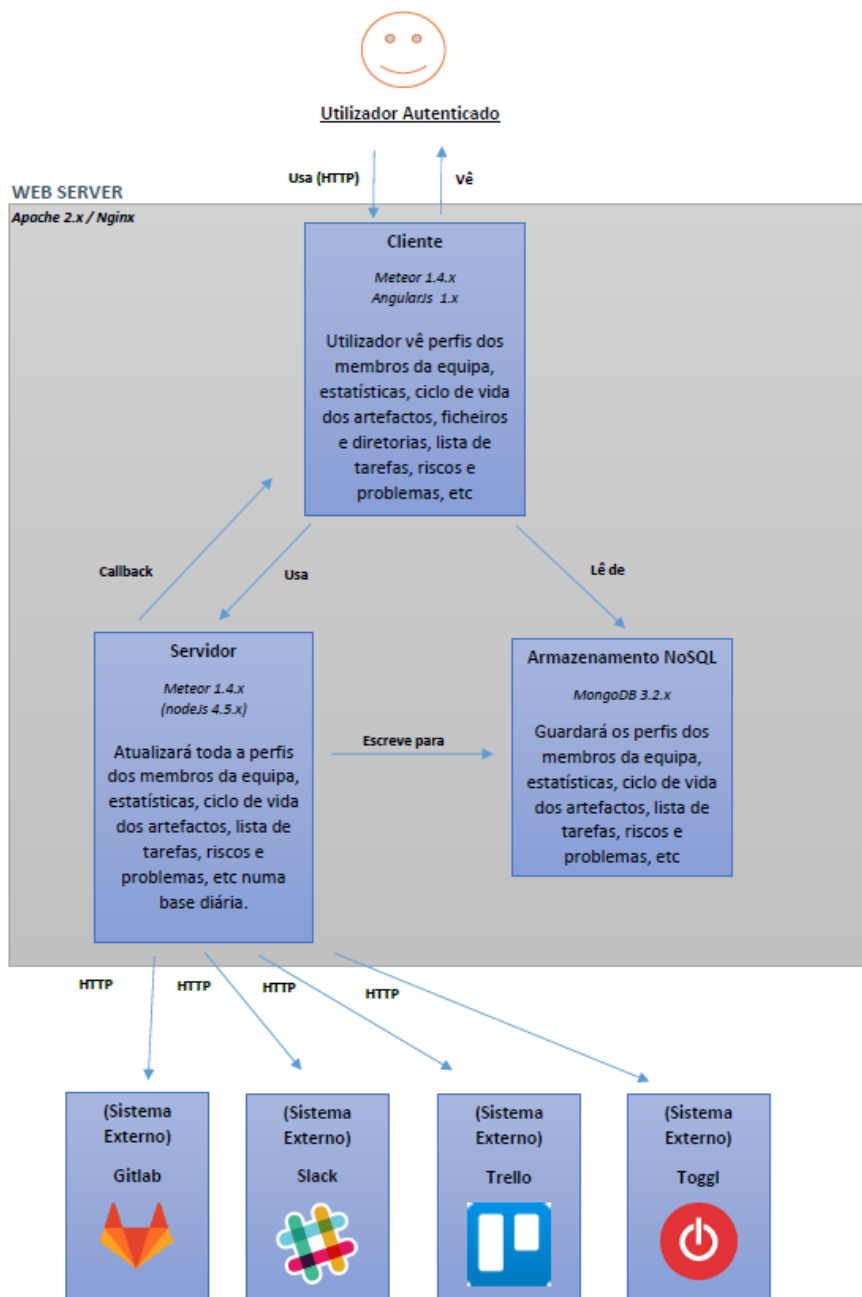


Figura 2: Diagrama da Arquitetura de Alto Nível

## 2.3 DESIGN DETALHADO

### 2.3.1 SERVIDOR

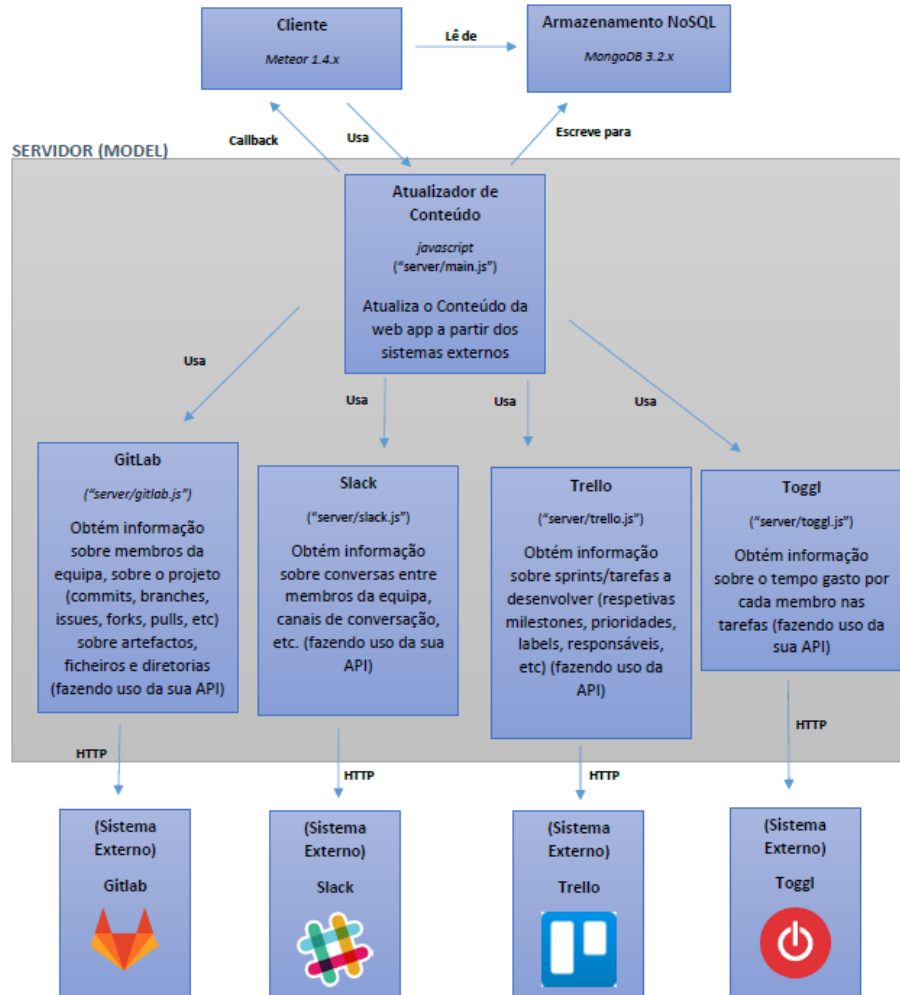


Figura 3: Design Detalhado do Servidor

### 2.3.2 CLIENTE

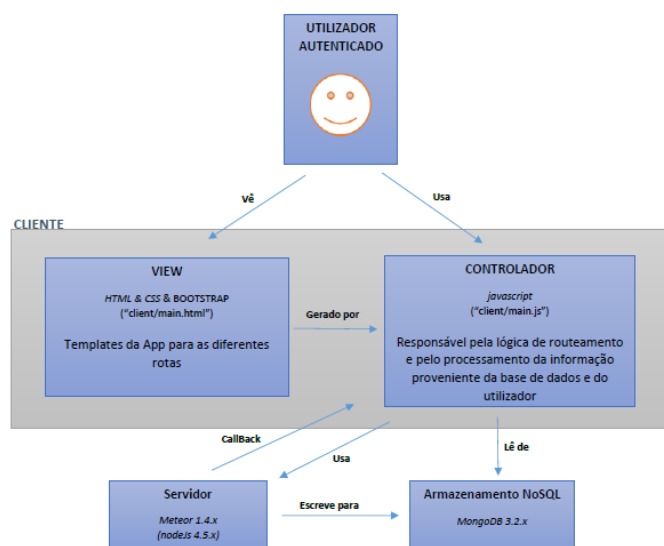


Figura 4: Design Detalhado do Cliente

### 2.3.3 BASE DE DADOS

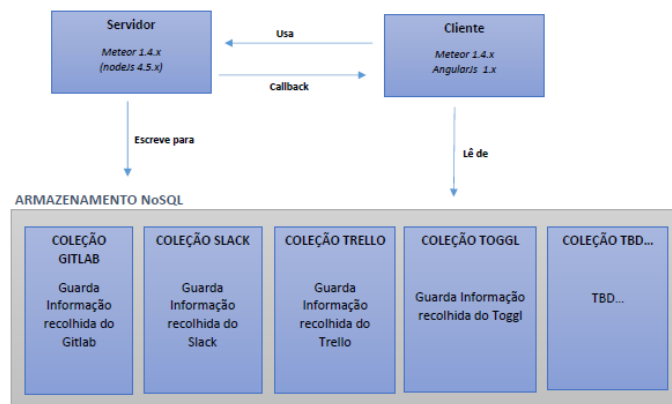


Figura 5: Design Detalhado da Base de Dados

## 2.4 TECNOLOGIAS USADAS

### Sistemas Operativos:

- Linux

### Tecnologias:

- Meteor
- nodeJS
- Angularjs
- MongoDB
- Bootstrap
- OAuth2
- Apache/Nginx

### Linguagens:

- Javascript
- HTML & CSS

### Interfaces:

- Http
- REST
- JSON

### API'S:

- Slack
- Trello
- Gitlab
- Toggl