|  |
| --- |
| XY-Inc |
| Backend as a Service |
| Arquitetura e Tecnologias |

|  |
| --- |
| Marianna Spirandelli  22/06/2016 |

Conteúdo

[Introdução 3](#_Toc454354807)

[Descrição e Requisitos 4](#_Toc454354808)

[Arquitetura 6](#_Toc454354809)

[Representação Gráfica: 6](#_Toc454354810)

[Descrição: 6](#_Toc454354811)

[Tecnologias 7](#_Toc454354812)

[Front-end: 7](#_Toc454354813)

[Back-end: 7](#_Toc454354814)

[Banco de Dados 7](#_Toc454354815)

# Introdução

Esse documento visa apresentar a arquitetura e tecnologias que serão utilizada no desenvolvimento do sistema XY-Inc.

# Descrição e Requisitos

Temos o seguinte cenário. Um desenvolvedor mobile está desenvolvendo uma app que precisa de um backend simples, mas ele não tem conhecimento para desenvolver esse backend.

A Zup quer resolver o problema desse desenvolvedor. A idéia é que através de uma tela simples o desenvolvedor consiga criar um novo “Modelo” e inserir dados (instâncias do “Modelo”) para serem consumidos pela app mobile.

Um caso de uso de exemplo seria:

Desenvolvedor cria um novo “Modelo” com o nome “Products"

Desenvolvedor configura os atributos do modelo: name:string, description:text, price:decimal, category:string

A partir desse cadastro um recurso REST para gerenciamento desse modelo com os seguintes métodos devem estar disponíveis:

GET /products - Lista todos os produtos

GET /products/{id} - Busca um produto por id

POST /products - Cria um novo produto

PUT /products/{id} - Edita um produto

DELETE /products/{id} - Deleta um produto

Com isso o desenvolvedor consegue criar uma app mobile que utiliza esse conteúdo sem desenvolver um backend. Claro que existem vário outros detalhes necessários para uma solução completa, mas acredito que esses requisitos são suficientes para um MVP.

O que deve ser feito:

Pensar na melhor arquitetura para a solução, pensando em evolução, escalabilidade, etc..

Descrever brevemente a arquitetura proposta

Desenvolver o backend dessa versão inicial. Se quiser desenvolver o frontend também fique a vontade, mas seria um plus

O código-fonte deve ser disponibilizado na sua conta do Github, em um repositório com o nome xy-inc, juntamente com as instruções para execução e testes da aplicação (arquivo README)

Considerações importantes:

Você pode utilizar qualquer tecnologia que julgar necessário

Serão avaliados: organização do código, simplicidade da solução (KISS), conhecimentos da linguagem/framework utilizado, cobertura de testes, extensibilidade, manutenibilidade.

# Arquitetura

## Representação Gráfica:



## Descrição:

Basicamente a arquitetura pensada, teria uma camada Rest (Rest API) para poder atender qualquer tipo de cliente que deseje utilizar a aplicação, sendo necessário apenas ao cliente interpretar o JSON retornado pela aplicação, tornando qualquer expansão e/ou reutilização de serviços simples.

O NodeJs é uma importante ferramenta para a escalabilidade do sistema, por isso foi escolhido dentre outros fatores positivos da sua utilização, em conjunto com o Express e o MongoDb por ser uma base de dados NoSQL, permitindo uma maior flexibilização na criação dos conteúdos e dos schemas.

A Autenticação e Autorização seria realizada por um Middleware na camada do Express, através de validação do Token enviado junto à requisição do cliente.

# Tecnologias

## Front-end:

* HTML5;
* Bootstrap (Framework CSS);
* AngularJs (Framework Javascript);
* Bower (Gerenciador de pacotes);
* Grunt (Automatização de tarefas - EX.: Minificar .js)

## Back-end:

* NodeJs;
* Express (server side framework);
* JWT-SIMPLE (modulo para gerar JSON Web Tokens);
* Mongoose (DB Driver)

## Banco de Dados

* MongoDb.