**1. Introdução:**

Este Artefato apresenta a arquitetura proposta para o projeto web-app chamado Badala relativo a cadeira de Projeto Aplicado I da Universidade de Fortaleza. A arquitetura é apresentada através de um conjunto de visões que reunidas objetivam cobrir os principais aspectos técnicos relativos à implementação e implantação do projeto em questão. O intuito é capturar e formalizar as principais decisões tomadas com relação à arquitetura do sistema.

**1.1. Finalidade**

Este documento tem como objetivo documentar as partes mais significativas do ponto de vista da arquitetura da aplicação Badala. Por arquitetura, entende-se a estrutura ou o conjunto de estruturas do sistema, que compreendem:

(i) elementos de software;

(ii) propriedades externamente visíveis desses elementos; e

(iii) os relacionamentos entre os mesmos.

A arquitetura do sistema é formada por diversos componentes que se relacionam globalmente sob diferentes perspectivas. Assim, sua finalidade é definir um modelo arquitetural para ser aplicado ao desenvolvimento da aplicação, bem como reunir todas as informações necessárias ao controle das atividades de Arquitetura, oferecendo uma visão macro dos requisitos arquiteturais as quais representam características e interesses diferentes de diversos stakeholders sobre um mesmo objeto, ou seja, sobre o software desenvolvido.

**1.2. Definições, Acrônimos e Abreviações**

|  |  |
| --- | --- |
| **Termo** | **Significado** |
| IIS | Gerenciador do Serviços de Informações da Internet. Servidor padrão do Windows, será usado para testes do Software. |
| REST | Padrão arquitetural usado em serviços  Web. |
| C# | Linguagem de programação usada para o Backend da aplicação. |
| Angular 2/4 | Framework utilizado para desenvolver o Frontend da aplicação. |
| Entity Framework | ORM utilizado para trabalhar com dados no modelo de objetos. |
| MVC | Model-View-Controller, padrão de arquitetura do software. |
| ASP Core | Estrutura de software. |
| SPA | Single Page Application. |

**1.4. Visão Geral**

O restante do documento está organizado como segue. A Seção 2 descreve a representação global da arquitetura da aplicação. A Seção 3 apresenta as principais metas e restrições arquiteturais. As Seções de 4 a 8 apresentam as diferentes visões arquiteturais, sejam elas: visão de caso de uso, visão lógica, visão de processos, visão de implantação, visão de implementação e visão de dados, respectivamente.

.

**2. Representação Arquitetural**

O Badala é composto por quatro camadas:

(i) A camada de visão, construída com HTML5, CSS e Javascript, utilizando o padrão SPA e os frameworks Angular 2/4 e Bootstrap;

(ii) A camada de serviços, onde é disponibilizada uma API REST escrita em C#;

(iii) A camada de negócio, onde são construídos os objetos de negócio gerenciados pelo ASP Core;

(iv) A camada de dados, que realiza o mapeamento objeto-relacional no padrão do Entity Framework.

A camada de visão é executada inteiramente no navegador do cliente que, por sua vez, faz uma série de acessos assíncronos à camada de serviços com o objetivo de ler e manipular dados. O formato usado para a troca de dados é o JSON.

A API REST foi desenvolvida na plataforma ASP Core.

As entidades persistentes são construídas utilizando a implementação em Entity Framework e mapeadas para tabelas de o banco de dados PostgreSQL.

**3. Metas e Restrições da Arquitetura**

A decisão de escolha do padrão Single Page Application (SPA) foi permitir trazer todo o processamento da camada de visão para o cliente, balanceando a responsabilidade de execução entre cliente e servidor, o que melhora a experiência do usuário (UX) criando interfaces com usabilidade moderna, rápida, fluida e de fácil entendimento do usuário. Outro motivo foi que com SPA otimiza-se o consumo de banda, minimizando a quantidade de requisições ao servidor e o tráfego de rede, pois as cargas de dados são feitas por demanda.

A aplicação Badala foi desenvolvido inicialmente para informar aos usuários onde está ocorrendo eventos movimentados no perímetro, porém sua arquitetura foi projetada para permitir a inclusão de novas funcionalidades ao sistema, como um agregador de ofertas. Dessa maneira, a criação de uma API REST tem o objetivo de facilitar a evolução do sistema, permitindo a inclusão de novas funcionalidades de forma transparente entre as camadas.

**4. Visão de Casos de Uso**

Ver documento de Casos de Uso

**5. Visão Lógica**

Esta seção descreve as partes significativas do ponto de vista da arquitetura do modelo de design. Além disso, documenta as classes mais importantes, suas organizações nos pacotes de serviço e subsistemas, e a organização desses subsistemas em camadas.

**5.1 Visão Geral**

src é a pasta principal onde irá conter todo os arquivos de classe, arquivos de configuração e subpastas do projeto.

**5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura**

A seguir uma tabela contendo cada pacote da aplicação, suas respectivas descrições e exemplos de artefatos que compõem:

|  |  |  |
| --- | --- | --- |
| **Pacote** | **Descrição** | **Exemplos de Artefatos** |
| Src | Agrupa todas as DLL do Projeto | Badala  DAL |
| ClientApp | Agrupa todos os arquivos HTML e Typescript, assim como as dependências dos frameworks Angular2 e Bootstrap. | *home.component.ts* |
| DAL | DLL de conexão com o banco | ApplicationDbContext.ts |
| Constrollers | Agrupa os controllers do projeto. | AccountController.cs |