

**Documentação do projeto MOBILE SENAI APP 2020.**

Esta é uma amostra descritiva do projeto contendo o seu corpo técnico resumido para o funcionamento, manutenção e flexibilização do serviço.

**SENAI SÃO PAULO DE INFORMÁTICA**

**2020**

**AUTOR**

*Lucas Simplicio de Oliveira*

**TURMA**

*Técnico em Desenvolvimento de Sistemas (manhã)*

## Sumário

Equipe .....	3
Introdução .....	4
Metodologias.....	4
Tecnologias utilizadas .....	5
Arquitetura.....	5
Dicionário de dados .....	6
Documentação da API .....	7

## Equipe

*Lucas Simplício de Oliveira*

## Introdução

Aqui estarão documentadas as implantações, metodologias e nomenclaturas propriamente ornamentadas para o visionamento do projeto. Toda e qualquer alteração adotada antes e/ou durante o projeto, deverá ser exposta neste documento, há exceção somente para possíveis alterações simplórias feitas na parte visual do projeto, também como pequenos scripts executados na parte lógica, cujos quais não agreguem ao MVP e/ou não interfiram parcialmente no resultado final do desenvolvimento.

O projeto será uma aplicação mobile, as funcionalidades e parte lógica da aplicação serão transmitidas via API.

- Será usado durante o desenvolvimento o *Xamarin* para programar e configurar a aplicação internamente no sistema Android.
- Para o serviço externo de dados, a API, será usada a tecnologia da *Microsoft ASP .NET Core 3.1*
- A parte de design está indefinida e cabe alterações num futuro próximo.

### Resumo do contexto

A empresa WTower, especializada em gerência de estádios de futebol, está se organizando para uma competição internacional que ocorrerá localmente sediada em seus estádios.

Eles sugerem que o espectador se informe sobre as estatísticas das partidas de futebol utilizando um simples aplicativo, consultando os placares dos jogos, jogadores, número de gols e etcetera.

A informação é um passo essencial para manter o espectador atento, por isso, foi pedido que, em conformidade com o plano de negócio, fosse desenvolvida uma solução viável em um espaço curto de tempo.

## Metodologias

Ainda pode ter mudanças em relação ao desenvolvimento, mudanças relacionadas às metodologias deverão ser escritas aqui.

Não houve preferência por quaisquer metodologias para o time de desenvolvimento. O desenvolvimento atuará em etapas, com metas e objetivos bem estabelecidos desde o início, e sujeito às adaptações.

# Tecnologias utilizadas

As tecnologias que serão utilizadas no desenvolvimento do projeto serão especificadas abaixo.

Ferramentas do ambiente de desenvolvimento

Visual Studio 2019 || Visual Studio Code || Android Studio

## Arquitetura<sup>1</sup>

API + Serviço externo (XAMARIM)

## Frameworks, bibliotecas e extensões.

- ASP .NET Core API – Versão 3.1 (C#)

- Entity Framework Core – ???

HTML5, CSS3 & JS

Xamarin - ????

Bibliotecas e serviços

Jquery - ???

Bootstrap - ???

## Dicionário de dados

Atributo	classe	tipo	Tamanho	Null	Tabela
Id	Determinante	INT		PK	selecao
nome	Simples	VARCHAR	255	false	selecao
bandeira	Simples	BYTE		true	selecao
uniforme	Simples	BYTE		true	selecao
escalacao	Simples	VARCHAR	10	true	selecao

Atributo	classe	tipo	Tamanho	Null	Tabela
Id	Determinante	INT		PK	jogador
nome	Simples	VARCHAR	255	false	jogador
nascimento	Simples	DATETIME		false	jogador
posicao	Simples	VARCHAR	255	false	jogador
qtdefaltas	Simples	INT(0)		false	jogador
qtdecartoesamarelo	Simples	INT(0)		false	jogador
qtdecartoesvermelho	Simples	INT(0)		false	jogador
qtdegols	Simples	INT(0)		false	jogador
informacoes	Simples	TEXT		false	jogador
numerocamisa	Simples	INT		false	jogador
foto	Simples	BYTE		true	jogador
selecaoid	Simples	INT		FK(selecao)	jogador

Atributo	classe	tipo	Tamanho	Null	Tabela
Id	Determinante	INT		PK	jogo
selecaocasa	Simples	INT		FK(selecao)	jogo
selecaovisitante	Simples	INT		FK(selecao)	jogo
placarcasa	Simples	INT(0)		false	jogo
placarvisitante	Simples	INT(0)		false	jogo
data	Simples	DATETIME		false	jogo
estadio	Simples	VARCHAR	255	false	jogo

Atributo	classe	tipo	Tamanho	Null	Tabela
Id	Determinante	INT		PK	usuario
nome	Simples	VARCHAR	255	false	usuario
email	Simples	VARCHAR	255	false(UNIQUE)	usuario
apelido	Simples	VARCHAR	255	false(UNIQUE)	usuario
foto	Simples	BYTE		false	usuario
senha	Simples	VARCHAR	255	false	usuario

## Documentação da API

O conteúdo que será mostrado á seguir representa as rotas que estarão disponíveis para o tráfego de dados, inserção e modificação.

- As requisições entre as pontas de acesso se limitam ao GET, PUT e POST.
- Alguns retornos possuem mensagens de erro padronizadas e parâmetros fixos.
- Não há uso do protocolo HTTPS, todas as requisições haverão de ser feitas pelo protocolo padrão HTTP.
- Houve o uso de formulários para a inserção de arquivos estáticos para a API.
- Todas as requisições estão assíncronas.
- Todas as requisições estão pré-configuradas em JSON.
- Uso de autenticação JWTBearer. O token válido possui trinta (30) minutos de duração.
- Uso do Swagger, com todas as requisições de controle devidamente documentadas, o endereço é: <http://servidor/swagger>.

<http://servidor/jogadores/{param}> GET

Faz a busca de jogadores pelo nome ou faz a listagem de todos os jogadores ordenados pelo número da camisa e posição caso a propriedade seja “all”.

Param (string) = nome do jogador ou all

Retorna uma lista ou vazio.

<http://servidor/jogadores/selecao/{param}> GET

Faz a busca de jogadores que pertencem á uma seleção específica. Caso a propriedade seja “all”, é feita uma listagem de seleções por ordem alfabética com todos os seus jogadores em sequência.

Param (string) = nome da seleção ou all

Retorna uma lista ou vazio.

<http://servidor/selecoes> GET

Requisição simples que faz a listagem de todas as seleções em ordem alfabética. Não há parâmetros para esta requisição.

Retorna uma lista ou vazio.

<http://servidor/partidas> GET

Requisição simples que faz a listagem de todas as partidas, ordenadas pela data mais recente até as mais velhas. Não há parâmetros para esta requisição.

Retorna uma lista ou vazio.

<http://servidor/partidas/{param}> GET

Requisição que traz todas as partidas de uma determinada data.

Param (string) = a data da partida buscada (dd-mm-yyyy)

Retorna uma lista ou vazio.

<http://servidor/partidas/pesquisa/{param}> GET

Requisição que traz as partidas que estão relacionadas ao nome de um estádio ou seleção.

Param (string) = nome do estádio ou nome da seleção

Retorna uma lista ou vazio.

<http://servidor/usuario/cadastrar> [FromForm] POST

Cadastro de usuário através de formulário e com envio de arquivo estático \*imagem\* para a API. (inserir no formulário nome, apelido, email e senha como texto, o campo foto será inserido como arquivo).

Retorna mensagens de erro avisando de conflito caso o apelido ou email postulante já esteja cadastrado. Retorna confirmação de cadastro.



http://servidor/usuario PUT (AUTENTICADO)

Alteração de informações do usuário pelo próprio utilizador. É possível alterar o Nome, Senha, E-mail e Apelido.

Retorna uma confirmação de que a alteração foi concluída com sucesso.

Retorna erro de Apelido e/ou E-Mail existente.

http://servidor/usuario GET (AUTENTICADO)

Consulta do usuário autenticado sobre os seus dados, retornando as configurações do próprio usuário.

Retorna as informações do usuário logado.

http://servidor/login POST

Faz o login do usuário pelo E-Mail (ou apelido) e senha. Gera um token de acesso.

(inserir no objeto “info:credencial” + “senha:credencial”)

Retorna mensagens de erro avisando que as credenciais estão incorretas.

Retorna um token de acesso para o usuário.