

LEIC

Projeto e Seminário

Arnaldo Tema – nº 39342

Prof. Carlos Martins

Projeto - PlayersNet

Este documento visa descrever o projeto a ser desenvolvido no âmbito da cadeira de Projeto e Seminário do curso de Engenharia Informática e de computadores no Instituto Superior de Engenharias de Lisboa.

Introdução

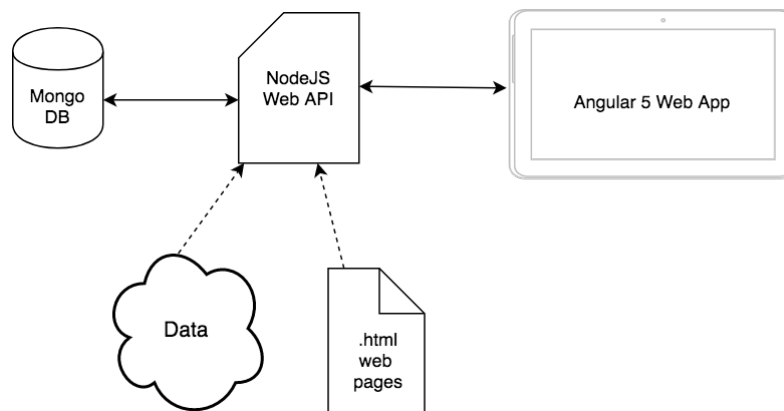
O Projeto consiste numa plataforma Web utilizada por atletas, treinadores e representantes de clubes desportivos (inicialmente apenas equipas de futebol, mas passível de ser utilizada para outras modalidades noutras versões).

A plataforma funciona como uma rede social desportiva onde os jogadores têm a possibilidade de apresentar o seu perfil/currículo de atleta de forma a tornar mais fácil a sua promoção perante entidades desportivas, permitindo também aos treinadores e respetivos clubes desportivos agilizar o processo de scouting e contratação/aquisição de jogadores nela inscritos.

Entregáveis e Arquitetura

Serei responsável por planejar e conceber toda a estrutura da plataforma, toda a stack de desenvolvimento do site e respetiva base de dados, nomeadamente:

- User Interface e Styling – Angular 5 / HTML 5 / Bootstrap / Materialize.
- Base de Dados da plataforma – MongoDB.
- API da plataforma – NodeJS.
 - Estabelece a comunicação entre a Base de Dados e a Aplicação Web em Angular 5.
 - Adquire informação (dados) de outros sites através de um Web Crawler.



Descrição e Objetivos

A plataforma assemelhar-se-á à famosa rede social de negócios – LinkedIN, mas focada no futebol.

Existem utilizadores, que podem ser: jogador, treinador ou clube, que se podem “seguir” mutuamente, recomendar, fazer publicações nas páginas dos utilizadores que seguem (com o consentimento do mesmo), gostar de publicações, comentá-las, podem comunicar entre si através de mensagens e ainda confirmar, através de um voto, qualidades que o utilizador tiver alegadas na sua página (novamente, tal como no LinkedIN).

Cada utilizador tem uma página configurável com os seus atributos principais. Para além da página de perfil (situada num separador intitulado “Perfil” e que será o separador principal), cada utilizador terá também o separador “Estatísticas”, “Media” e, eventualmente, “Carreira” (Dependendo do tipo de utilizador).

Para além das páginas dos utilizadores, haverá uma página principal (homepage/feed principal) onde se encontrará uma lista de “notícias” com base nas preferências do utilizador. Por exemplo, tratando-se de um utilizador do tipo “Clube”, sendo essa a escolha do clube, o feed (homepage) poderá apresentar uma lista dos jogadores do escalão “Sénior” e as suas atualizações marcantes (por exemplo, um jogador “Sénior” que tenha feito um hat-trick, ou um jogador “Sénior” que se tenha lesionado, etc.).

Estas notícias serão criadas pelos próprios utilizadores quando os mesmos atualizam os seus dados nas suas páginas. Quer sejam dados visíveis ou invisíveis para os outros utilizadores.

Em adição à informação introduzida pelos utilizadores, há a necessidade de ter informação presente no site de forma automática. Esta informação poderá ser fornecida pela Federação Portuguesa de Futebol, pela Associação de futebol, pela FIFA ou por crawlers (neste caso, um programa que lê páginas web de forma a popular a base de dados).

Para este projeto, não é relevante saber qual será a fonte de informação, pelo que o mesmo estará preparado para qualquer tipo de input nesse sentido, **sendo este mapeamento automático parte dos desenvolvimentos opcionais.**

O modelo EA da plataforma encontra-se no documento que acompanha este ficheiro. No entanto, e para dar uma noção mais concreta do mesmo, para além das entidades “Jogador”, “Treinador” e “Clube” já referidas, há também “Jogo”, “Competição” e “Evento” (associadas entre si através de outras entidades associativas).

A plataforma terá um sistema de autenticação e autorização e uma associação de cada conta a uma ou várias redes sociais (FaceBook, LinkedIN, Instragram, YouTube).

Desenvolvimento e Planeamento

De forma a tornar “ágil” o planeamento e gestão de desenvolvimentos das tarefas necessárias, este processo está a ser monitorizado pelo TFS VisualStudio Online. Assim, todas as tarefas são descritas e controladas.

O planeamento é subdividido em 3 categorias:

Feature - > **Backlog** -> **Task**

Esta metodologia de trabalho implica que cada uma destas categorias (denominadas por work items) tenham um propósito específico em contexto profissional, que facilita a entrega e integração de diversas componentes a um eventual cliente final.

Em contexto académico, vamos adquirir esta metodologia, porém adaptá-la às nossas necessidades.

Assim, as **features** têm uma lista de **backlogs**, e estes uma lista de **tasks**. Portanto, uma **feature** descreve uma componente de desenvolvimento, nessa componente, temos os **backlogs** que representam uma etapa (ou subcomponente) de desenvolvimento que pode ser testada e fechada independentemente das outras subcomponentes, e, por fim, cada **backlog** tem associado a si uma lista de **tasks**, que são as tarefas exatas associadas a essa subcomponente para que se possa concluir o desenvolvimento da mesma.

Posto isto, segue a stack de todos os desenvolvimentos do projeto:

- Criação das páginas em Angular 5 e integração das mesmas com a Web Api:

Feature	▼ 🏆 Pages
Product Backl...	Profile
Product Backl...	Stats
Product Backl...	Carreer
Product Backl...	Media
Product Backl...	Recommendations
Product Backl...	Skills
Product Backl...	Roster
Product Backl...	Staff
Product Backl...	Tryouts
Product Backl...	Badgers
Product Backl...	Home
Product Backl...	Login
Product Backl...	Create Account
Product Backl...	Game
Product Backl...	League (Competition)

- Criação de componentes genéricas para as páginas em Angular 5:

Feature	▼ 🏆 Page Components
Product Backl...	> 📄 Header (Nav-bar)
Product Backl...	📄 Footer
Product Backl...	📄 Chat
Product Backl...	📄 Notifications
Product Backl...	📄 Settings
Product Backl...	📄 Game Info
Product Backl...	📄 League Info
Product Backl...	📄 Following
Product Backl...	📄 News

- Criação da Web Api em NodeJS através de um gerador genérico de CRUD, também em NodeJS:

Feature	▼ 🏆 WEB API
Product Backl...	▼ 📄 CRUD Generator
Task	✅ Assert and assign names and endpoints
Task	✅ Add dynamic endpoints with queries

- Aquisição e instalação do servidor e suas dependências:

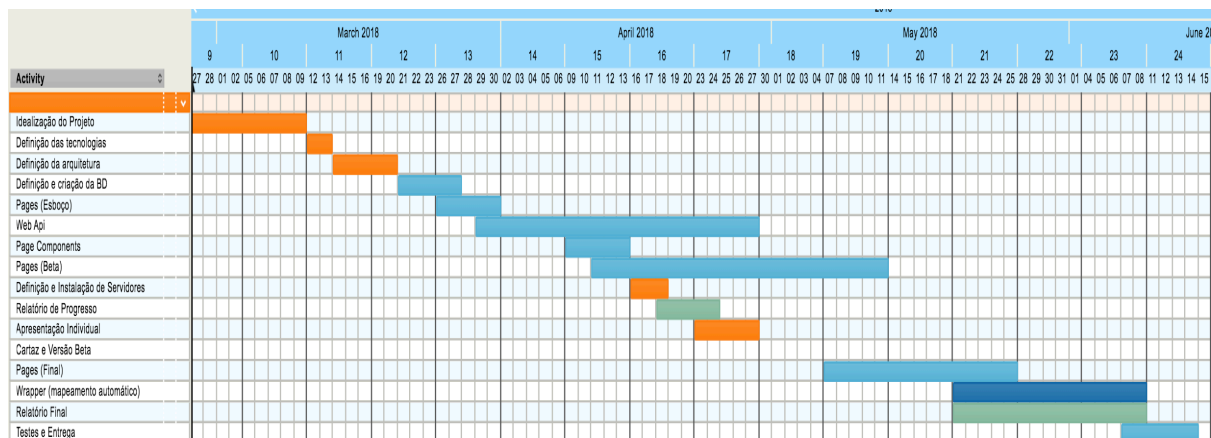
Feature	▼ 🏆 Environment Set UP (Production)
Product Backl...	▼ 📄 Production Setup
Task	✅ Buy VM
Task	✅ Install VM's dependencies
Task	✅ Buy SSL Certificates and Domains
Task	✅ Install /Associate SSL Certificates and Domains
Task	✅ Search for Helpers and Add-ons
Task	✅ Install Helpers and Add-ons

- (Opcional) Desenvolvimento de um Web Crawler, englobado num wrapper, que lê páginas web (será apenas testado com o site zerozero.pt) de forma a popular a base de dados e obter informação automática dos jogadores que se inscrevem na plataforma.

As **tasks** são praticamente idênticas para cada **backlog**, daí estar apenas colapsado o separador *Team fetcher*.

Feature	▼ 🏆 Wrapper
Product Backl...	> 📄 League Fetcher
Product Backl...	▼ 📄 Team Fetcher
Task	✅ Team Achievements (Non-game related)
Task	✅ Team detail fetcher
Task	✅ Roster fetcher
Task	✅ Games fetcher
Product Backl...	> 📄 Game Fetcher
Product Backl...	> 📄 Achievement Fetcher
Product Backl...	> 📄 Player Fetcher

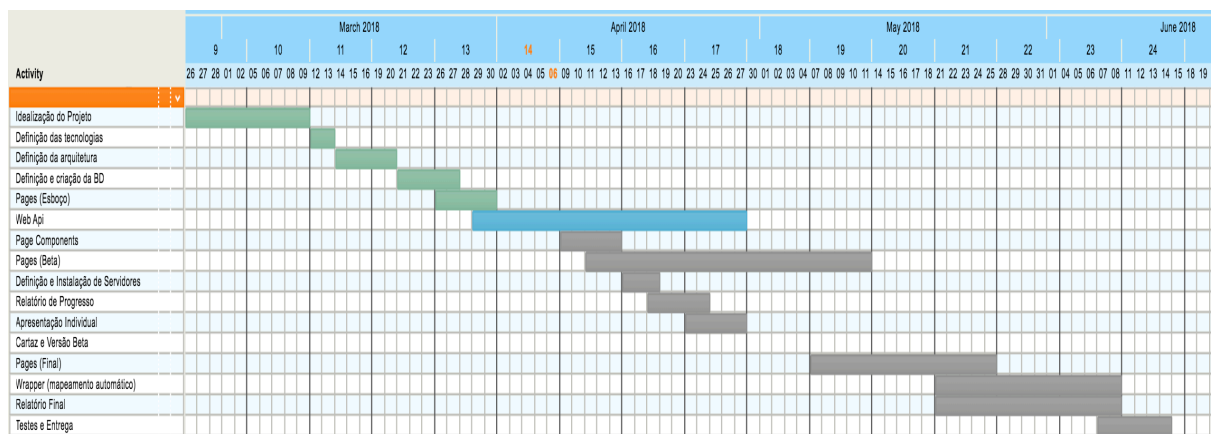
Planeamento e Calendarização:



Legenda:

Laranja: Definições de projeto, Azul claro: Desenvolvimento, Verde: Documentação, Azul escuro: Opcional.

Calendário Corrente:



Legenda:

Azul claro: Em desenvolvimento, Verde: Feito, Cinzento: Por fazer.

Referências

- Visual Studio Online e desenvolvimento “Agile”: <https://www.visualstudio.com/pt-br/team-services/agile-tools/> (28/03/2018)
- MongoDB: <https://www.mongodb.com/> (28/03/2018)
- Angular5: <https://angular.io/> (28/03/2018)
- Web Crawler: https://pt.wikipedia.org/wiki/Rastreador_web (28/03/2018)
- NodeJS: <https://nodejs.org/en/> (28/03/2018)

