



INSTITUTO POLITÉCNICO DE LEIRIA

MATCHPLANNER

RELATÓRIO DE PROJETO DE SISTEMAS DE
INFORMAÇÃO

DANIEL BATISTA N°2171836
DIOGO ALPENDRE N°2170324

PROJETO DE SISTEMAS DE INFROMAÇÃO

4/1/2019

Relatório de Projeto em Sistemas de Informação para cumprimento dos requisitos necessários à realização da prova de apresentação de projeto do Curso Técnico Superior Profissional (TeSP) de **Programação de Sistemas de Informação** realizado sob a orientação dos **Professores Sílvio Mendes, Nélon Marques e Romeu Paz.**

DECLARAÇÃO

Declaro que este Relatório se encontra em condições de ser apreciada (o) pelo júri a designar.

O estudante 1 – Daniel Correia Batista,

A handwritten signature in black ink, reading "Daniel Batista". The signature is written in a cursive style with a large initial 'D' and 'B'.

Leiria, 4 de fevereiro de 2019

DECLARAÇÃO

Declaro que este Relatório se encontra em condições de ser apreciada (o) pelo júri a designar.

O estudante 2 – Diogo Cruz Alpendre,



Leiria, 4 de fevereiro de 2019

Agradecimentos

O nosso especial agradecimento aos professores das UCs de PlatSI e Sistemas de Informação, professores Sílvia Priem Mendes e Romeu Paz, e aos restantes professores envolvidos no desenvolvimento deste projeto.

Resumo

[RELATÓRIO DE PROJETO - MATCHPLANNER]

[AUTORES]

Daniel Batista N° 2171836

Diogo Alpendre N° 2170324

O projeto MatchPlanner é um trabalho de âmbito académico, com a finalidade de projeto final de curso, e tem como temática a gestão de eventos. O projeto final é composto por 3 projetos desenvolvidos durante o 1º semestre do 2º ano do curso: um serviço web, uma aplicação móvel Android e uma API REST.

A aplicação web pode ser usada pelo utilizador no seu computador ou no seu smartphone.

A API REST foi desenvolvida para ser consumida pela aplicação móvel para poder ter acesso à base de dados da aplicação web, e poder enviar e receber dados da mesma.

A aplicação móvel foi desenvolvida como uma forma de o utilizador ter uma maneira mais prática de usar a aplicação.

PALAVRAS-CHAVE: Yii2, MVC, PHP, Android, Java, Codeception

Índices

Índice geral

Agradecimentos	v
Resumo	vi
Índices.....	vii
1. Introdução	1
2. Metodologia.....	2
3. Arquitetura do Sistema	4
4. Gestão do Projeto.....	6
5. Análise.....	12
6. Desenho.....	13
7. Implementação.....	14
7.1 - Aplicação WEB	14
7.2 APLICAÇÃO ANDROID	18
7.3 - REST API	18
8. Testes	19
9. Conclusão e trabalho futuro.....	20
10. Bibliografia	21
11. Anexos	22

Índice de figuras

Figura 1 - Listas do quadro criado no Trello com as tarefas nas diferentes listas	2
Figura 2 – Frontend aplicação web	4
Figura 3 - Backend Aplicação Web	4
Figura 4 - API REST com lista JSON de utilizadores	5
Figura 5 - Cronograma com as diferentes etapas de projeto	8
Figura 6 - Tarefas inseridas no Microsoft Project e os seus prazos previstos	9
Figura 7 - Registo de horas de trabalho por elemento	10
Figura 8 - Modelo de dados para implementação da base de dados	13
Figura 9 – Excerto de um controller.....	16
Figura 10 - Excerto de um model	17
Figura 11 - Excerto de uma vista.....	17
Figura 12-Resoltados da execução de testes	19
Figura 13 - Aplicação web (frontend)	22
Figura 14 – Aplicação web de administração (Backend)	22

1. Introdução

O MatchPlanner é um projeto que consiste numa aplicação de gestão de eventos, onde é possível criar contas de utilizadores com perfis associados, e os utilizadores podem criar eventos, publicações e comentários.

Neste relatório vai ser relatado a nossa metodologia de planeamento e as ferramentas usadas para desenvolver as componentes do projeto, bem como detalhes da implementação, e testes.

2. Metodologia

Durante todo o desenvolvimento do projeto, foi utilizada a metodologia Scrum para planeamento e gestão do mesmo. Cada sprint foi adaptado à semana (de terça a segunda), e a divisão das tarefas foi feita com a ferramenta Trello e o Microsoft Project.

O Trello é uma ferramenta gratuita, que usámos para criar um quadro para o nosso projeto, onde também criámos várias listas com cartões onde escrevemos as tarefas e o seu estado de desenvolvimento. Nesses mesmos cartões também inserimos a data prevista para a conclusão da tarefa, e subtarefas da mesma, entre outros. Durante os diferentes estados de conclusão, cada tarefa passa em diversas listas, desde a lista On Hold, para a lista Current Sprint (tarefas a realizar durante o sprint), e depois passa para a lista In Progress quando começa a ser desenvolvida. Após a sua conclusão, passa para a lista Done.

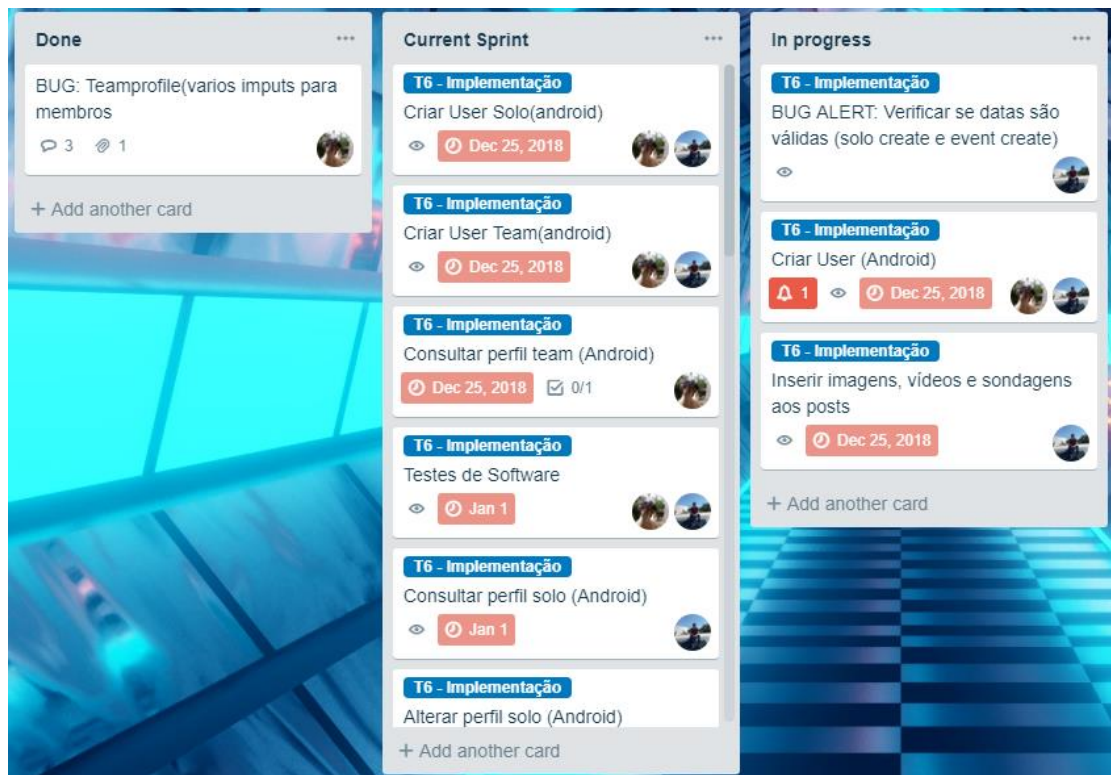


Figura 1 - Listas do quadro criado no Trello com as tarefas nas diferentes listas

As horas de trabalho foram registadas numa folha de Excel Online, disponibilizada pelo professor Nelson Marques, onde ficam registadas as horas totais de trabalho por semana (sprint) e totais até ao momento desde o início do projeto.

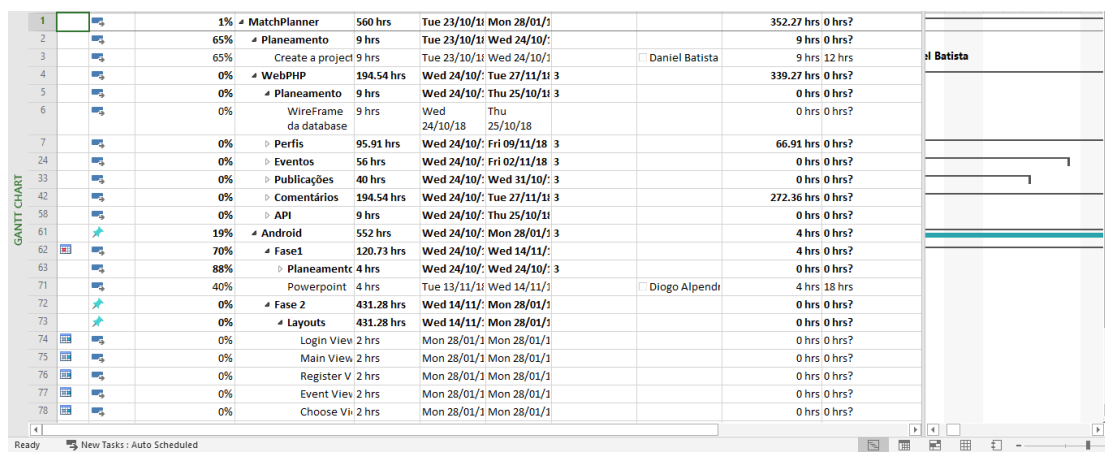


Figura 2 - Tarefas divididas no Microsoft Project

TeSP PSI PL1 09					
Ficheiro Editar Ver Inserir Formatar Dados Ferramentas Suplementos Ajudar A atualizar... o foi efetuada há 5 dias					
Tarefa					
	A	B	C	D	E
1	Tarefa	Aluno	Data	Semana	Duração
2	T6 - Implementação	Diogo Alpendre	22/11/2018	10	1.5
3	T6 - Implementação	Diogo Alpendre	26/11/2018	11	3
4	T6 - Implementação	Diogo Alpendre	20/11/2018	10	1.5
5	T2 - Análise concorrential	Diogo Alpendre	21/11/2018	10	0.2
6	T6 - Implementação	Daniel Batista	22/11/2018	10	1
7	T6 - Implementação	Diogo Alpendre	23/11/2018	10	3
8	T6 - Implementação	Daniel Batista	24/11/2018	10	2
9	T2 - Análise concorrential	Diogo Alpendre	25/11/2018	10	0.2
10	T6 - Implementação	Daniel Batista	26/11/2018	11	1
11	T6 - Implementação	Daniel Batista	26/11/2018	11	1
12	T6 - Implementação	Daniel Batista	19/11/2018	10	2
13	T6 - Implementação	Daniel Batista	20/11/2018	10	2
14	T7 - Gestão de projeto	Daniel Batista	27/11/2018	11	3
15	T6 - Implementação	Daniel Batista	28/11/2018	11	1
16	T6 - Implementação	Diogo Alpendre	28/11/2018	11	2
17	T6 - Implementação	Diogo Alpendre	28/11/2018	11	2.5
18	T6 - Implementação	Diogo Alpendre	28/11/2018	11	1
19	T6 - Implementação	Diogo Alpendre	29/11/2018	11	1
20	T6 - Implementação	Diogo Alpendre	1/12/2018	11	1.5

Figura 3 Tarefas e horas de trabalho registadas numa folha de excel

3. Arquitetura do Sistema

A arquitetura do sistema é composta por 3 componentes:

- Uma aplicação web desenvolvida na framework Yii2;

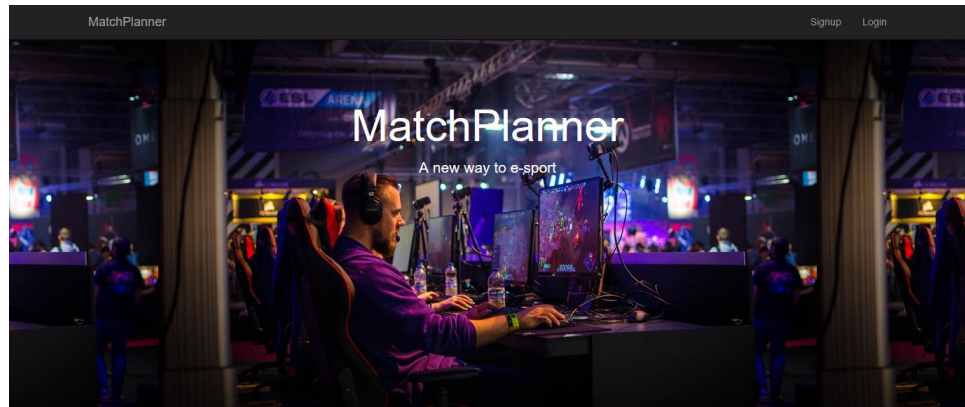


Figura 4 – Frontend aplicação web

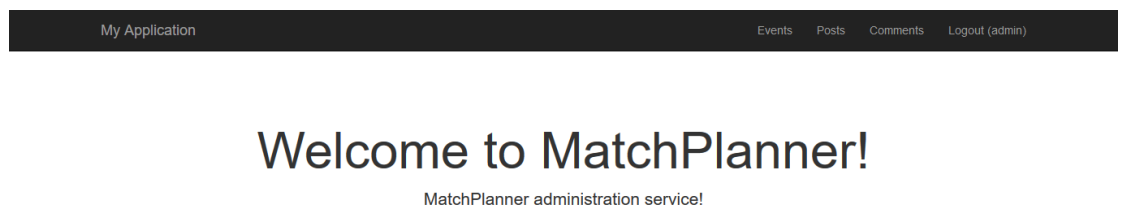


Figura 5 - Backend Aplicação Web

- Uma API REST desenvolvida em Yii2;

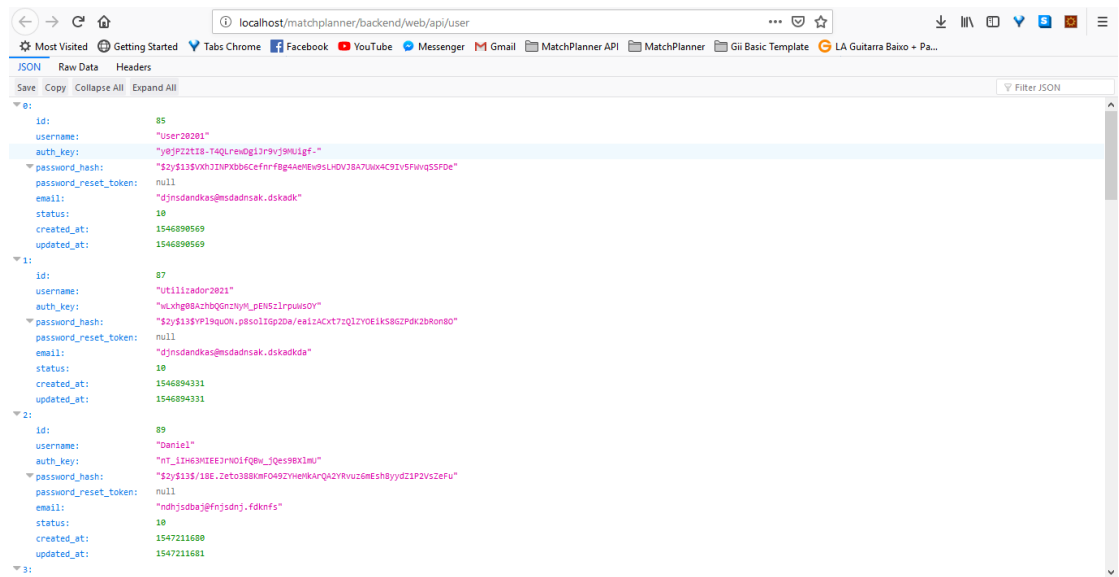
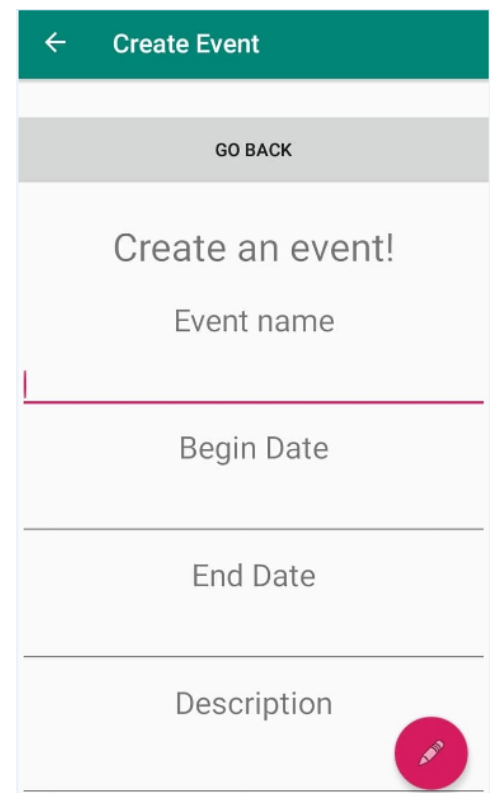
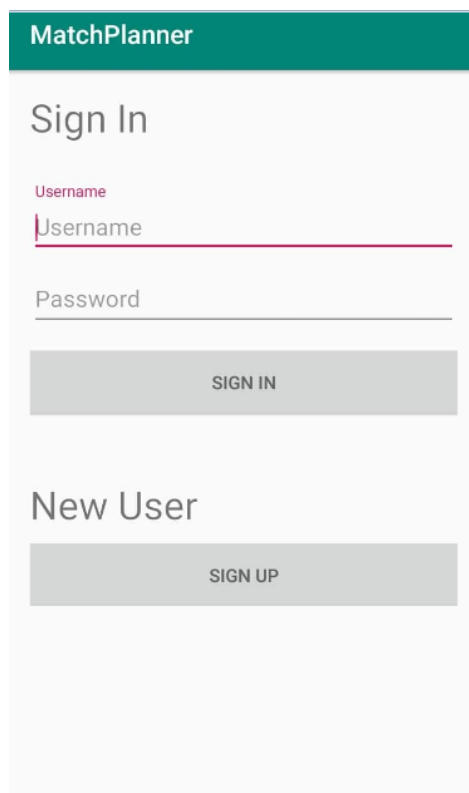


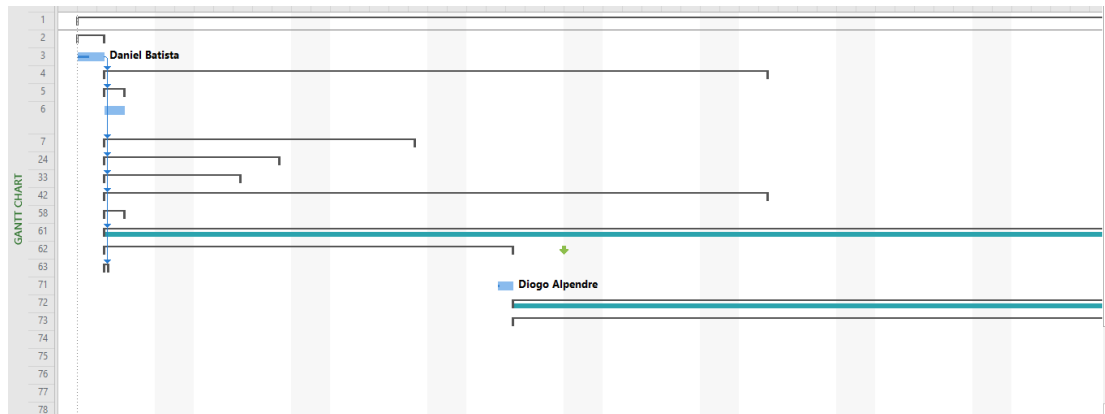
Figura 6 - API REST com lista JSON de utilizadores

- Uma aplicação móvel Android.



4. Gestão do Projeto

Gantt Chart



Identificação das principais atividades

- Inserir utilizadores e autenticá-los quando se faz o login;
- Criar eventos;
- Criar publicações e comentários;

Planeamento

- Tarefas

- Efetuar o registo de utilizadores e criar um perfil associado (pessoal ou de equipa);
- Consultar utilizadores e os seus perfis;
- Atualizar dados dos utilizadores e perfis;
- Apagar utilizadores (inativação da conta para outros utilizadores ainda poderem consultar informação dos mesmos). Ex: Publicações;
- Criar eventos;
- Consultar eventos;
- Atualizar dados dos eventos;
- Apagar eventos;
- Criar publicações nos eventos;
- Ver publicações;
- Atualizar publicações;
- Apagar publicações;
- Criar comentários nas publicações;
- Ver comentários;
- Atualizar comentários;
- Apagar comentários;
- Inserir imagens nas publicações;
- Enviar pedido de amizade a um utilizador;

- Cronograma

	T1	T2	T3	T4	T5	T6
Estudo do problema						
Proposta da temática						
Análise concorrencial						
Requisitos:						
- Funcionais;						
- Não funcionais;						
Organizados por role						
Classificação por importância / prioridade (baixo, média e alta)						
Elementos adicionais						
- Diagrama de classes						
- Mockups						
- Casos de uso ou user stories						
Modelo de dados						
Converter os requisitos ou casos de uso ou user stories em atividades (Trello)						
Implementação						
Escrita do relatório de projeto						

Figura 7 - Cronograma com as diferentes etapas de projeto

Execução

- Cartões

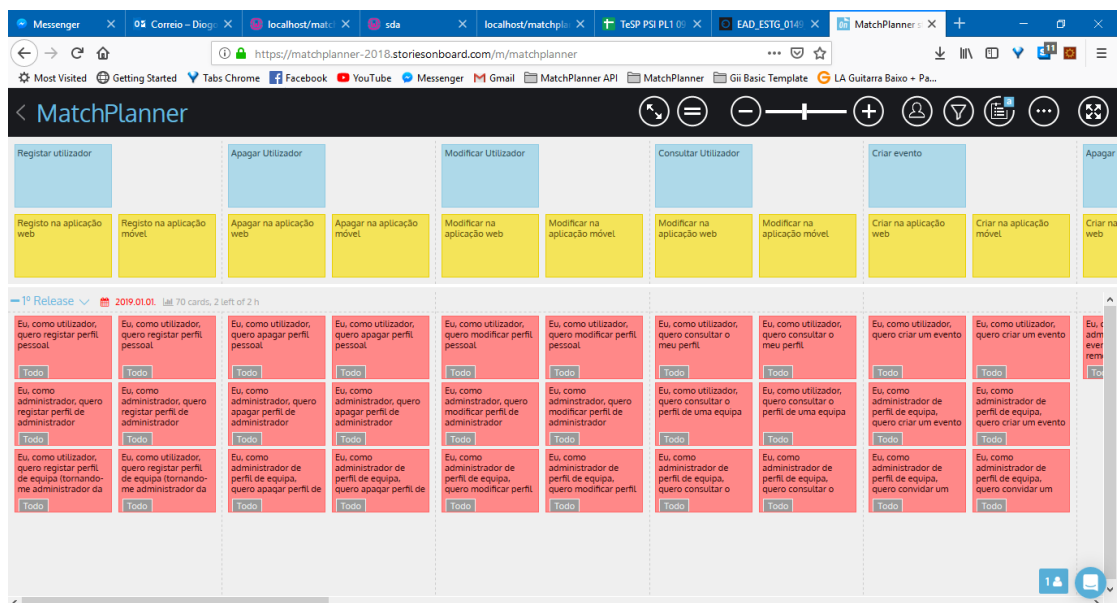


Figura 8 - User Stories

- Horas – homem

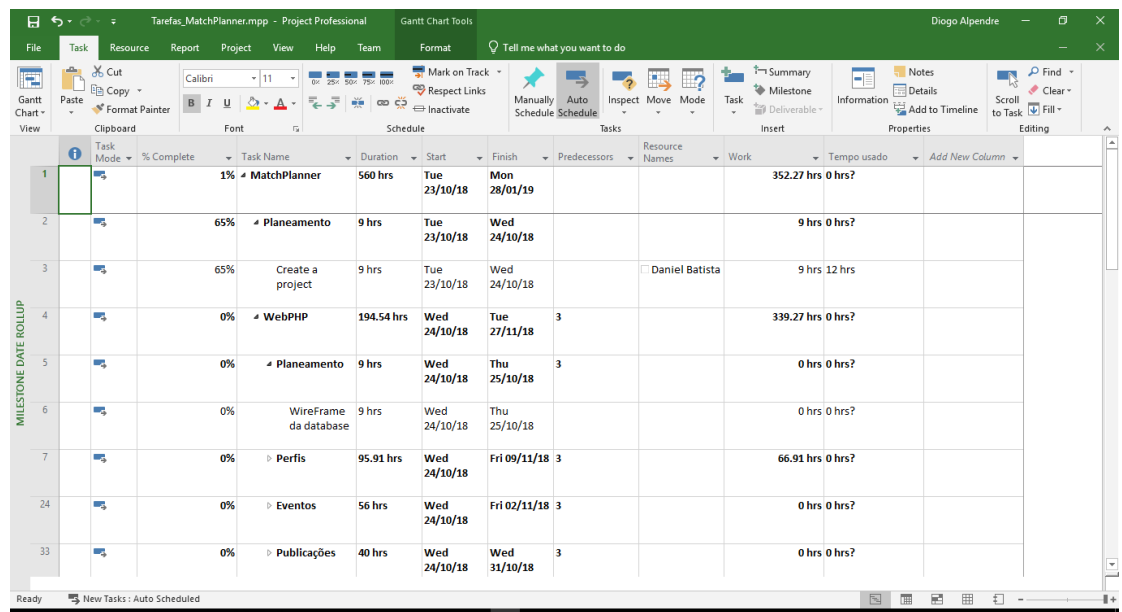


Figura 9 - Tarefas inseridas no Microsoft Project e os seus prazos previstos

Controlo

- Cronograma

1	SUM de Duração	Aluno			
2	Semana		Daniel Batista	Diogo Alpendre	Total geral
3		0		0	0
4	10		7	3.9	10.9
5	11		6	12,5	18,5
6	12		7	7,5	14,5
7	13		6	6,5	12,5
8	14		7	3,5	10,5
9	15			10	10
10	16			9	9
11	17		7	4	11
12	Total geral	0	40	26.10	5.12

Figura 10 - Cronograma de controlo de horas de trabalho

- Horas – homem

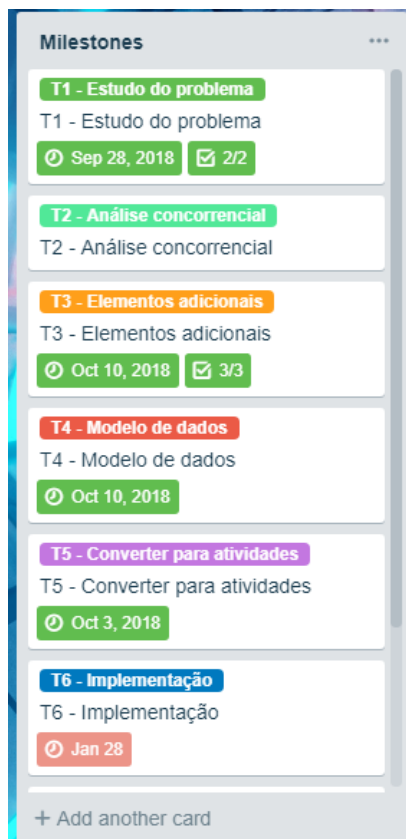
	A	B	C	D	E	F
1	Tarefa	Aluno	Data	Semana	Duração	Descrição
2	T6 - Implementação	Diogo Alpendre	22/11/2018	10	1.5	Criação de layouts das atividades
3	T6 - Implementação	Diogo Alpendre	26/11/2018	11	3	Validação do login caso user faça logout após signup
4	T6 - Implementação	Diogo Alpendre	20/11/2018	10	1.5	DateTimePicker na vista create event
5	T2 - Análise concorrencial	Diogo Alpendre	21/11/2018	10	0.2	Requisitos funcionais colocados na lista On Hold
6	T6 - Implementação	Daniel Batista	22/11/2018	10	1	Implementação de CRUDs e AutoID
7	T6 - Implementação	Diogo Alpendre	23/11/2018	10	3	Criação de Date and hour picker
8	T6 - Implementação	Daniel Batista	24/11/2018	10	2	Criação de eventos compatíveis
9	T2 - Análise concorrencial	Diogo Alpendre	25/11/2018	10	0.2	Classificação da importância e prioridade das tarefas
10	T6 - Implementação	Daniel Batista	26/11/2018	11	1	Criação do Date Picker no form userprofile
11	T6 - Implementação	Daniel Batista	26/11/2018	11	1	Criação do Dropdown no campo "sex" form userprofile
12	T6 - Implementação	Daniel Batista	19/11/2018	10	2	(aula) Resolver bugs na aula da parte web
13	T6 - Implementação	Daniel Batista	20/11/2018	10	2	(aula) implementação da formula de puxar automaticamente o "User_id"
14	T7 - Gestão de projeto	Daniel Batista	27/11/2018	11	3	(aula) Gestão de trello e Google sheets - Colocação das tarefas corretas
15	T6 - Implementação	Daniel Batista	28/11/2018	11	1	(aula) Proteção sobre o a criação de user
16	T6 - Implementação	Diogo Alpendre	28/11/2018	11	2	(aula) Ver perfil
17	T6 - Implementação	Diogo Alpendre	28/11/2018	11	2.5	Ver eventos criados na main page
18	T6 - Implementação	Diogo Alpendre	28/11/2018	11	1	Criar layouts aplicação android
19	T6 - Implementação	Diogo Alpendre	29/11/2018	11	1	Criar layouts aplicação android
20	T6 - Implementação	Diogo Alpendre	1/12/2018	11	1.5	Criar layouts aplicação android

Figura 11 - Registo de horas de trabalho por elemento

Lições aprendidas

- Apesar de uma notória melhoria da gestão de tempo, ainda tem de ser melhorada;

Milestones



Equipa de projeto e funções respetivas

- Daniel Correia Batista (Developer, scrum master)
- Diogo Cruz Alpendre (Developer)

5. Análise

Temática do Projeto

A temática escolhida para projeto foi a gestão de eventos. Consequentemente, implementámos funcionalidades como a possibilidade de criar páginas de eventos, onde os utilizadores que para ele são convidados pode fazer publicações e escrever comentários nessas mesmas publicações.

Objetivos principais

- Cumprir os requisitos funcionais e não funcionais definidos;
- Desenvolver os nossos conhecimentos nas linguagens e ferramentas usadas para desenvolver o projeto;

Análise de requisitos e funcionalidades dos sistemas a desenvolver

Sendo o tema do projeto a gestão de eventos, os requisitos foram pensados a poderem resolver as necessidades dos utilizadores. Portanto, com influência dos eventos da plataforma Facebook, decidimos permitir ao utilizador criar eventos, onde nos mesmos os participantes podem criar publicações e os demais inserem comentários nas mesmas publicações. Também foi proposto um sistema de pedidos de amizade.

6. Desenho

Modelo de Dados

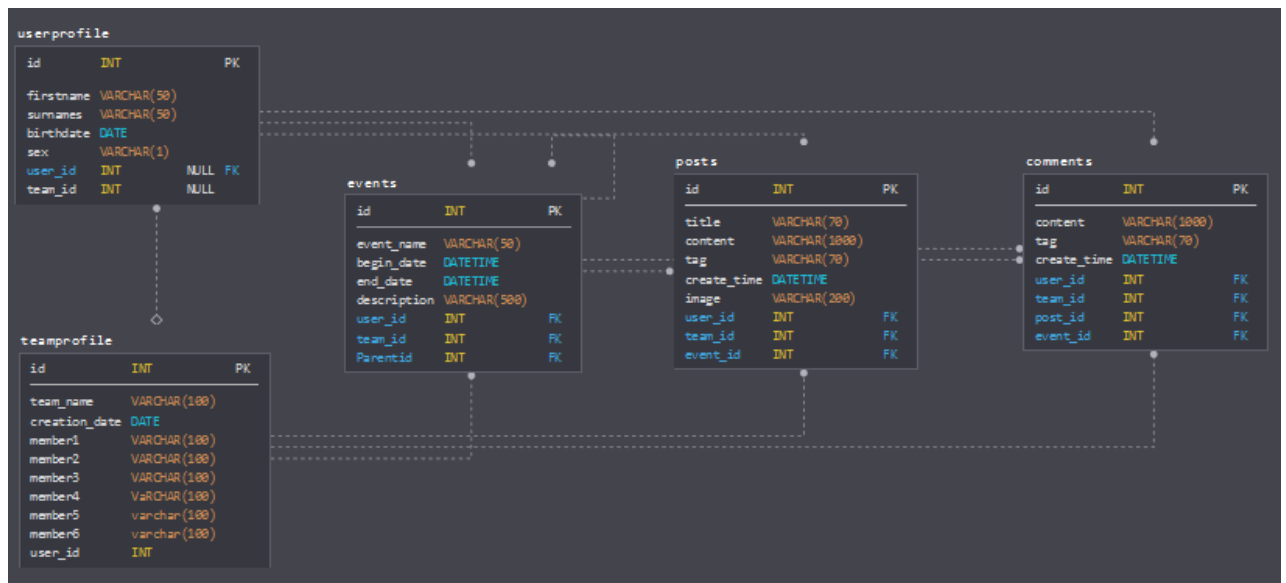


Figura 12 - Modelo de dados para implementação da base de dados

7. Implementação

7.1 - APLICAÇÃO WEB

O trabalho a desenvolver na nossa aplicação web baseou-se no planeamento feito. Teve como objetivo final permitir registo e autenticar utilizadores, e os utilizadores poderem criar eventos, publicações nos eventos e comentários nas publicações, e ainda possibilitar imagens nas publicações e enviar pedidos de amizade aos diversos utilizadores, e também poder aceitar esses mesmo pedidos de amizade.

A aplicação web foi feita com a framework Yii2, usada para o desenvolvimento da aplicação, está construída com base no modelo MVC, tendo portando 3 componentes principais: modelos, vistas e controladores. Como servidor, foi usado o apache, incluído no stack Wampserver.

Após o planeamento, ficou decidido o número de entidades presentes no projeto. E, para ser possível realizar as operações CRUD para cada entidade, foram criados através da ferramenta Gii (gerador de modelos, controladores, módulos, entre outros) da Yii2. O primeiro controlador com que trabalhamos foi o controlador User. O controlador User já estava presente na framework, pois é um controlador próprio da mesma. O mesmo realiza o registo de um utilizador, login com autenticação e as restantes funções de CRUD. Após análise do funcionamento da aplicação, decidiu-se que, em caso de o utilizador decidir apagar a sua conta, a mesma não seria eliminada da base de dados, mas sim desativada, para que os demais utilizadores possam ver os eventos criados pelo mesmo. Como extensão da conta de utilizador, o mesmo terá de optar por duas vertentes de utilizador: perfil “SoloProfile” (perfil pessoal) que identifica unicamente informação de um único indivíduo (nome próprio, apelidos, data de nascimento e género), ou o perfil “TeamProfile” (perfil de equipa) que guarda informação de uma equipa de *e-sports* (nome da equipa e nome dos elementos (máximo de 6 elementos)). Após o registo de utilizador e do perfil associado, os utilizadores podem criar eventos. Como cada evento está associado ao utilizador que o cria, foi feito um relacionamento de 1-N entre as entidades User e Event, respetivamente. Dentro dos eventos, os utilizadores podem realizar publicações (posts), e os posts estão relacionados com os eventos e com os utilizadores que os criam. Cada publicação (modelo Post) guarda o seu título, conteúdo, tag e a hora de criação do

mesmo. Cada comentário possui o seu conteúdo e a sua tag. Possui um relacionamento de 1-N com as publicações, ou seja, uma publicação pode ter diversos comentários.

No fim do planeamento dos requisitos funcionais do projeto e de fazermos os mockups e modelo de dados, foi criada a base de dados. A nossa base de dados tem todas as tabelas relacionadas entre si, ou seja, um evento tem de ser criado por um utilizador, uma publicação de ser criada num evento existente e os comentários nas publicações e eventos existentes.

No desenvolvimento da aplicação web, foi utilizada a template advanced da Yii2. A framework Yii2 possuiu dois tipos de template, basic e advanced. Optámos pelo advanced pois era a que nos permitia fazer registo de utilizadores e criar roles para a aplicação, entre outros.

A aplicação é composta pelo lado frontend, usado pelos utilizadores, e pelo backend, usado para administração da aplicação.

O utilizador, ao criar o seu perfil associado, não pode preencher certas informações, como o ID da sua conta, que está lá para relacionar o perfil à conta. Esse ID é automaticamente preenchido pelo sistema. De início foi complicado perceber como o fazer, pois não tínhamos muito bem a noção do funcionamento da Yii2, mas após uma breve pesquisa, encontrámos um método que devolvia o ID do utilizador que está autenticado, podendo assim dar esse valor à chave estrangeira que representa o ID do utilizador. Como o perfil é único para a conta, ou seja, possui um relacionamento de 1 para 1 decidimos por o ID do perfil igual ao ID da conta de utilizador.

Na página principal do perfil (à qual chamámos inicialmente de “operations” mas mudámos para “Account Feed”), é possível visualizar os dados da conta e do perfil, através de botões na página.

Para atualizar os utilizadores e os perfis deve-se ir à visualização do mesmo e carregar no botão para atualizar.

Para apagar o utilizador deve-se usar o botão de apagar também na “view” da conta e perfil, e, como foi referido anteriormente, caso o utilizador decida apagar a sua conta, a mesma não é apagada da base de dados, mas sim desativada.

Para criar um evento o utilizador tem à sua disposição um botão na página principal. Ao clicar no botão, irá ser mostrado ao mesmo o menu de criação de evento, onde é necessário inserir os dados do mesmo (nome, datas de início e fim) e descrição. No menu foi usado o widget `datetimepicker` desenvolvido pela `kartik` (também usado para no menu para criar perfis pessoais). Para validação das datas, foram usadas funções da classe `DateTime` (as datas do evento possuem horas). Se os dados estiveram válidos, é mostrado o evento com os dados respetivos (`event/view`). Dentro da vista de cada evento, é possível criar publicações e comentários, e os mesmo ficam listados dentro do evento.

Ao voltar para a página principal da aplicação, os eventos criados pelo utilizador encontram-se listados com o acesso ao mesmo feito por botões respetivos a cada evento.

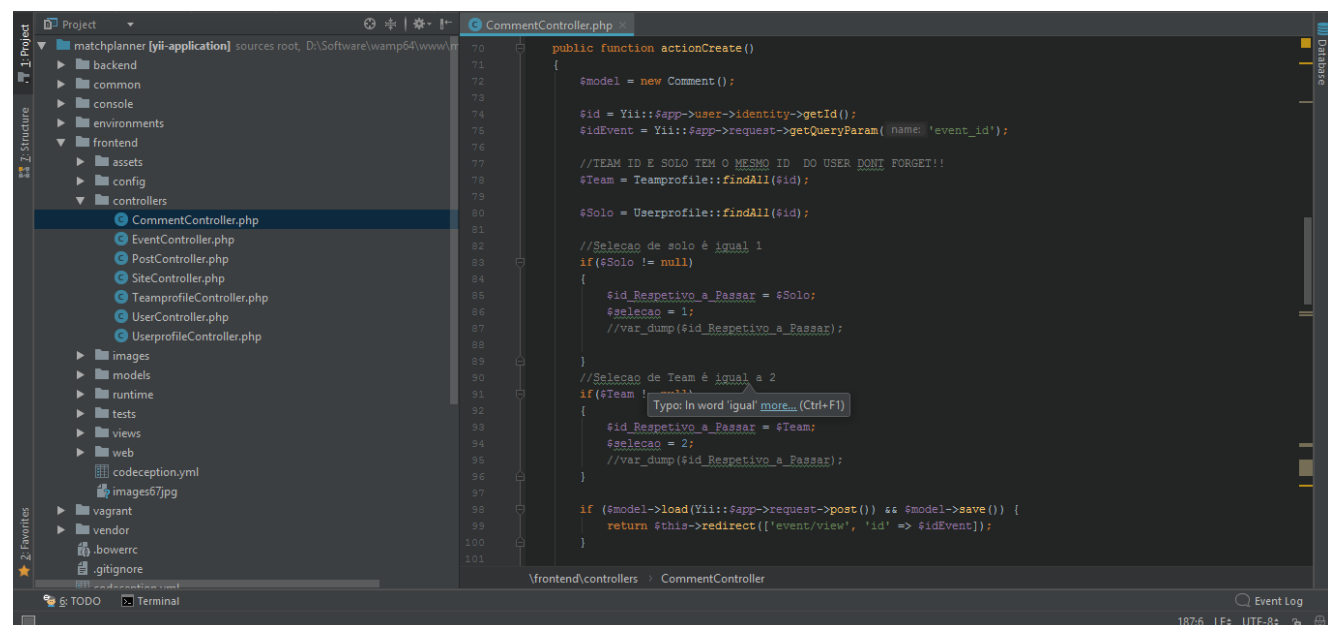


Figura 13 – Excerto de um controller

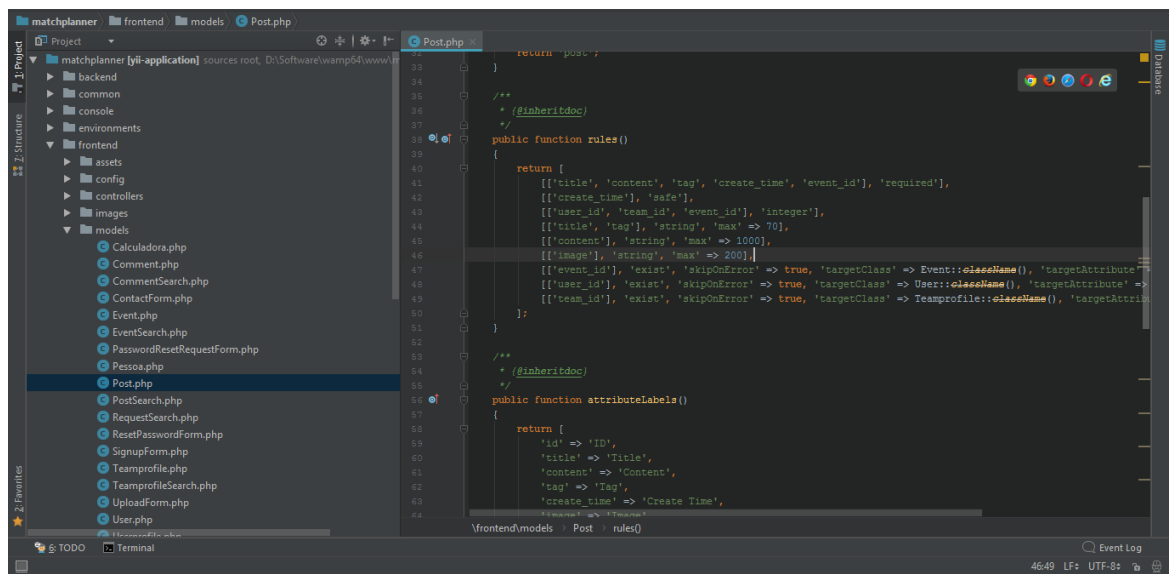


Figura 14 - Excerto de um model

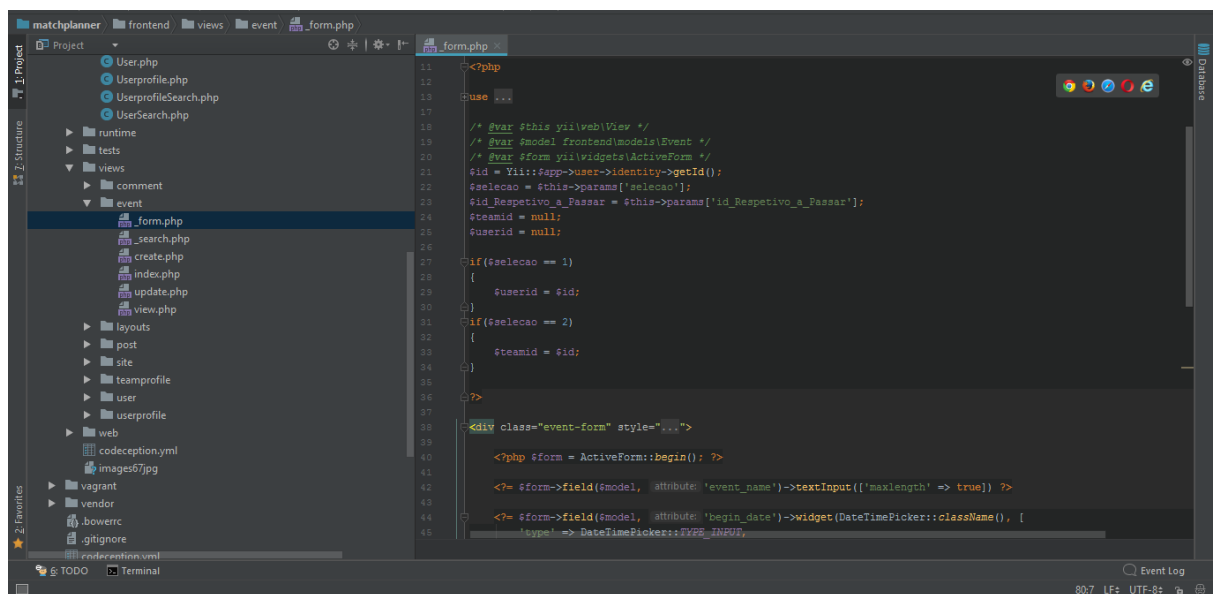


Figura 15 - Excerto de uma vista

7.2 APILICAÇÃO ANDROID

A aplicação Android foi desenvolvida no IDE Android Studio com a linguagem Java. Ao iniciar a aplicação, é mostrado ao utilizador o menu de login, onde também pode optar por registar um utilizar. Ao clicar no botão “New User” o utilizador terá de optar por escolher o tipo de perfil que deseja (solo ou team). Ao clicar no tipo de perfil respetivo, o utilizador faz o registo da sua conta de utilizador juntamente com o perfil integrado no mesmo formulário. Após registar o utilizador, o mesmo é direcionado à atividade onde poderá consultar, alterar e editar o seu utilizador. Na atividade principal o utilizador poderá também criar eventos, onde dentro deles poder criar publicações e comentários, e realizar o respetivo CRUD. A aplicação usa a API REST (descrita abaixo) para poder receber e alocar dados na base de dados da aplicação web.

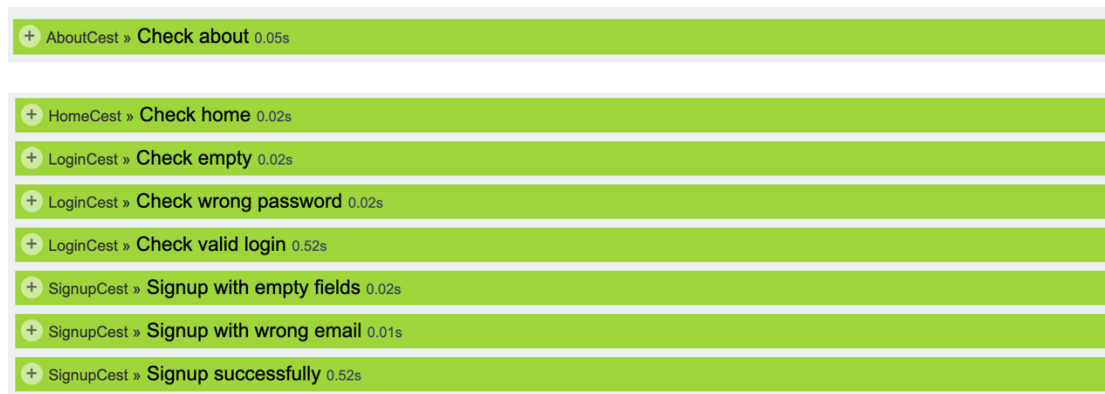
7.3 - REST API

Para estabelecer a comunicação entre a aplicação móvel e a aplicação web, foi criada uma API REST para ser possível ao utilizador realizar operações CRUD e armazenar os dados na base de dados da aplicação web, e possuir uma maneira mais prática de utilizar a aplicação web. Foi desenvolvida dentro de um módulo criado na aplicação web, onde estão controladores responsáveis por realizar as operações de crud e autenticação.

8. Testes

Durante o desenvolvimento da aplicação web, foi usada uma framework de testes chamada Codeception, para realizar testes unitários, funcionais e de aceitação.

- Resultados obtidos e correção de erros



+ AboutCest » Check about	0.05s
+ HomeCest » Check home	0.02s
+ LoginCest » Check empty	0.02s
+ LoginCest » Check wrong password	0.02s
+ LoginCest » Check valid login	0.52s
+ SignupCest » Signup with empty fields	0.02s
+ SignupCest » Signup with wrong email	0.01s
+ SignupCest » Signup successfully	0.52s

Figura 16-Resultados da execução de testes

9. Conclusão e trabalho futuro

Após todo o desenvolvimento do projeto, concluímos que conseguimos alcançar um nível razoável de cumprimentos dos nossos requisitos, e que temos que melhorar no desenvolvimento de aplicações móveis e renovar a nossa API a ficar mais simples e concisa. Percebemos também que houve uma melhoria à gestão de tempo efetuada, em relação ao 1º ano do curso.

10. Bibliografia

- <https://www.yiiframework.com/>
- <https://developer.android.com/>
- <https://pt.stackoverflow.com/>
- <https://codeception.com/>
- <https://www.yiiframework.com/doc/guide/2.0/en/rest-quick-start>

11. Anexos

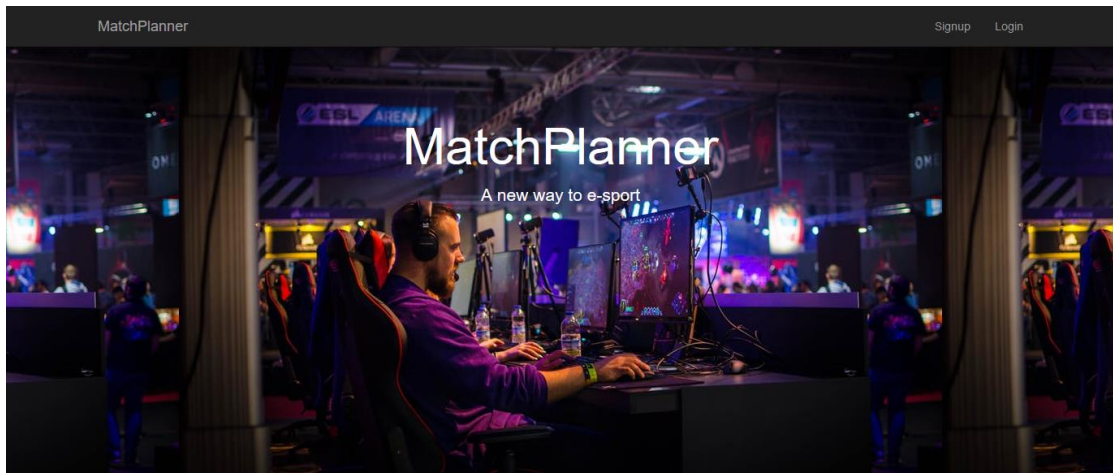


Figura 17 - Aplicação web (frontend)

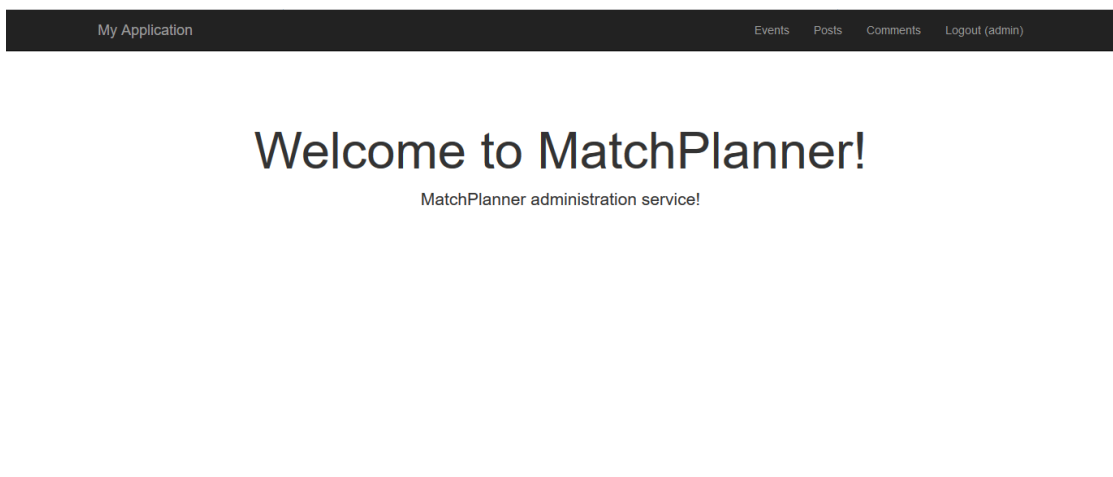


Figura 18 – Aplicação web de administração (Backend)