****

**INSTITUTO POLITÉCNICO DE LEIRIA**

MatchPlanner

RELATÓRIO DE projeto de sistemas de informação

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 de **Professor Sílvio Mendes**

Daniel Batista Nº2171836

Diogo Alpendre nº2170324

Projeto de sistemas de infromação

4/1/2019

**DECLARAÇÃO**

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

O estudante 1,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Leiria, .... de ............... de ...............

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 de (**nome do orientador**)

**DECLARAÇÃO**

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

O estudante 2,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Leiria, .... de ............... de ...............

# Agradecimentos

**(Opcional)**

**Um especial agradecimento ao Professor Sílvio Priem Mendes.**

# Resumo

**[RELATÓRIO DE PROJETO - MATCHPLANNER]**

**[AUTORES]**

**Daniel Batista Nº 2171836**

**Diogo Alpendre Nº 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: xxxx, yyyy, zzzz

# Índices

O relatório deve conter o índice principal, o índice de figuras e o índice de tabe

# Introdução

Este projeto consiste numa aplicação de gestão de eventos. Onde é possível criar eventos, publicações e comentários. Os perfis que os utilizadores podem criar são de Solo e Team.

Neste relatório vai conter as nossas metodologias de organização de tempo, as ferramentas usadas para a elaboração do projeto.

# 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. 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 sub-tarefas da mesma, entre outros.

# Arquitetura do Sistema

A arquitetura do sistema é composta por 3 componentes: uma aplicação web Yii2 Framework, uma API REST e uma aplicação móvel Android. A aplicação móvel comunica com a aplicação web através da API REST, criada juntamente com a aplicação web (módulo na aplicação). A aplicação Android foi desenvolvida com a linguagem Java com o IDE Android Studio.

# Gestão do Projeto

Gantt Chart

Identificação das principais atividades

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

Execução

* Cartões
* Alunos
* Horas – homem

Controlo

* Cronograma
* Horas – homem

Lições aprendidas

Deliverables

Milestones

Equipa de projeto e funções respetivas

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

# 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.

# Desenho

Modelo de Dados

# Implementação

Explicar o processo de desenvolvimento do software.

## 7.1- Aplicação WEB

A aplicação web começou por começarmos através das fichas que foram resolvidas nas salas de aula.

Na nossa aplicação web tem como inicio o planeamento do que nos vamos fazer, a aplicação tem como objetivo final a criar eventos. O evento modelo em que guarda o nome, data de inicio e de fim, descrição e quem o criou. Este modelo (o evento) é o modelo mais usado, o que deve ser mais completo. Como o evento não se pode criar sozinho (ser uma ação do sistema), usamos com relacionamento o User(utilizador em português) em relacionamento, isto é, uma criação apenas pode ser criada com o utilizador. O utilizador tem duas vertentes (ambas em modelos separados alem do que é criado com a framework), a vertente SoloProfile(ou perfil pessoal) que este um perfil de apenas uma pessoa, um utlizador comum que simplesmente quer usar a aplicação em inscrição única, e a vertente TeamProfile(ou Perfil de equipa) que é a criação de uma equipa de jogo na aplicação, esta so pode ter ate 6 elementos por equipa. O post (ou publicação em português) é um modelo que esta relacionado com Evento com a relação de 1 para N[[1]](#footnote-2). O post guarda o titulo do post, conteúdo, tags e a hora de criação to post. O comment (ou comentário em português) é um modelo que esta relacionado com Post com a relação 1 para N. O comment guarda conteúdo, tags e quando foi criado.

No fim de se planear o que íamos fazer de projeto e de fazermos os mockup’s para as paginas web, de acordo com o nosso planeamento foi criar a base de dados. A nossa base de dados esta criada em cadeira, ou seja, para criar um “Comentário” tempos que criar um “Post” e para criar um post temos criar um “Evento” e se quisermos criar um Evento temos que ter um “Utilizador”, resumindo isto quer dizer que a nossa tabela tem muitas chaves estrangeiras [[2]](#footnote-3).

Para começar-mos a criar os modelos, na framework Yii2 criamos um template yii-advanced. Este template em relação ao template basic entre muitas funcionalidades faz com que seja possível criar users.

Os modelos foram gerados com um ferramenta chamada Gii, esta ferramenta faz parte da própria Framework, esta serve para criar modelos e tudo que é comum no sites e na Framework, por exemplo, criar vistas e CRUD[[3]](#footnote-4). (Framework, s.d.). A geração é muito fácil de se criar, os modelos é usar a opção Model Generator e os CRUD é usar o CRUD Generator.

Para criarmos o dois tipos de user (solo e team) no fim de geramos o código base tivemos existem campos em que o utilizador não podia preencher, pois podia preencher mal e fazer com que a aplicação não funcionasse em condições. Um dos casos era de colocar o id em que o userprofile estava relacionado. Para que isso acontecesse tivemos que importar o ID do utilizador. De inicio não foi uma tarefa fácil, pois não tínhamos muito bem a noção de como o MVC funcionava, com algum apoio e pesquisa, reparamos que era usado uma função da yii 2 que indicava o utilizador que estava ligado logo era fácil de importar. Como o relacionamento era de 1 para 1 (ou seja um user apenas pode ter um profile ou é de solo ou é team) decidimos para ser mais fácil de criar tudo para a frente devido ao relacionamento os IDs do team e do solo são exatamente iguais ao user, para preencher o surname e o firname no caso do solo é usado text input no formulário, já o Birthdate utilizamos uma widget para facilitar e o utilizador introduzir a data no formato correto que neste caso é ano-mês-dia.

Para Mostrar os utilizadores existe um botão na pagina principal que faz com que mostre o perfil, este botão tem na função para ir buscar o id do user logado e vai mostrar o utilizador, em que o form onde é mostrado foi feito através do Gii, a função que vai buscar o id do utilizador é a mesma que vai buscar o id do user normal pois estes são iguais.

Para Atualizar os utilizadores (solo e team) é usado a função do crud generator dedicado a cada um. Agora para atualizar o User (da framework) é usado a função que veio na Framework, esta escolha foi assim para não criar problemas.

Para Apagar o utilizador não é “apagado” é colocado em modo desativado. Nos decidimos colocar assim mesmo que a pessoa desista da conta e os eventos, comentários, posts não são, isto é as outras pessoas que estão no evento não vão perder dados que podem ser fundamentais.

Para criar um evento é preciso um utilizador na chave estrangeira, esta chave é preenchida com a mesma função que é usada no solo e team profile, os campos de inserção de titulo de evento, descrição de evento é preenchida num text input no formulário, agora as datas de inicio em fim também foi utilizado um widget da kartik em que é um datetimepicker a diferença entre este e o dos dosamigos (utilizado na criação dos solo e team profiles) é que este acrescenta hora à data, da mesma maneira que na criação dos perfiles é utilizado este widget como maneira de segurança, para com que o utilizador não introduza as datas de maneira errada.

Para Ver os eventos na operations é criada um botao para cada evento gerado isto é se o o utilizador quando acaba de preencher o formulário todo corretamente e clicar em submit é rederecionado para a pagina operations e aparece o botao com o nome do evento. Se o utilizador clicar neste, pode visualizar o detalhes do evento assim como os posts e comentários. Os detalhes do evento foram gerados no gii. Agora os posts e comentários foi com a utilização de foreach’s.

Para Atualizar foi utilzado a função que vei-o com o crud generator, esta função simplesmente escreve por cima dos dados alterados faz com que fique certo com o que o utlizador quer.

Para Apagar os eventos é necessário apagar tudo o que esta la dentro. Isto quer dizer que se tivermos um evento com post e comentários, temos que os apagar antes de apagar o evento, isto tem que ser assim pois exitiria dados com a chave estrageira que não era valida e criava erros na base de dados e a longo prazo a base de dados ficaria lenta com dados que não eram validos. Logo posto esta situação de optimização decidimos em criar uma função que apagava todos os posts e comentários.

Para criar um Post é nessesario existir um evento. Como é preciso ter um evento nos colocarmos um botão na vista do evento, assim é ter a certeza que o post é criado do sitio correto, pois este tambem tem campos em que o utilizador não preenche, um deles é o próprio id. Com a função de ir buscar o id assim se torna fácil. Agora para colocar na chave estrangeira do evento com ações básicas de Get e Post(enviando pelo URL os dados a passar).

Para serem vistos nos eventos os posts é colocado um foreach na vista de detalhes(view), a maneira de mostar-mos nos eventos formos buscar pela chave estrangeira em qual mostra se for igual ao id do evento, isto é se (post).id\_event == (event).id.

Para os atualizarmos foi utilzado a função que vei-o com o crud generator, esta função simplesmente escreve por cima dos dados alterados.

Para os apagarmos os posts é um bocado como a situação dos eventos temos que apagar os comentários para com que a base de dados não fique lenta, sendo assim criamos uma função que puxe todos os comentários com a chave estrageira igual ao id do post, e fazemos delete.

Os Comentarios para serem criados é colocado um botão no foreach dos posts que estam no evento. Estes também tem uma herarquia e que são criados da mesma maneira que fizemos nos Post, buscando os ids(neste caso são do event ,do post e user).

Para os conseguirmos ver nos posts é utilizado um foreach dentro do foreach dos posts. São puxados com uma ida há base de dados pela chave estrangeira, sendo assim se que todos os comentários que tem a chave estrangeira igual ao id do post são puxados.

Para conseguirmos dar update (atualizar) decidimos usar a função do curd generator, que nos faz um overwrite dos dados alterados.

Para eliminarmos os comentários simplesmente apenas o criador do comentário consegue nos usamos a função do Crud generador para fazer isso, pois estes não tem nada em relação que crie data sem chave estragueira.

Como foi feito a main view (operations) a main view tem como nome operations pois é onde o utilizador consegue ter acesso a tudo, desde criar um evento,atualizar o perfil ou apagar o mesmo.

## 7.2-APILICAÇÃO ANDROID

## 7.3-Aplicação REST API

# Testes

Este capítulo deve incluir:

* Desenho de testes
* Ferramentas de testes utilizadas
* Execução dos testes
* Resultados obtidos e correção de erros

# Conclusão e trabalho futuro

# Bibliografia

# Anexos

1. Relacionamento 1-N é quando uma das partes apenas pode ter uma e a outra pode ter as que quizer. [↑](#footnote-ref-2)
2. Chaves estrangeiras ou foreign keys são campos nas tabelas na base de dados que faz com exista relacionamento entre elas, estas chaves podem ser obrigatórias ou não. [↑](#footnote-ref-3)
3. CRUD – Create Read Update Delete (Criar, Ler, Atualizar, Apagar) são as quatro operações básicas em base de dados relacionais. [↑](#footnote-ref-4)