

Base de Dados

Encontros

Relatório Intermédio

Turma 4, Grupo 406

Mestrado Integrado em Engenharia Informática e
Computação

Álvaro Miranda
up201603694@fe.up.pt

Joana Ferreira
up201705722@fe.up.pt

Maria Soares
up201605010@fe.up.pt

Março 2019

Conteúdo

| | | |
|---|---------------------|---|
| 1 | Tema | 2 |
| 2 | Diagrama de Classes | 3 |
| 3 | Descrição | 4 |

Capítulo 1

Tema

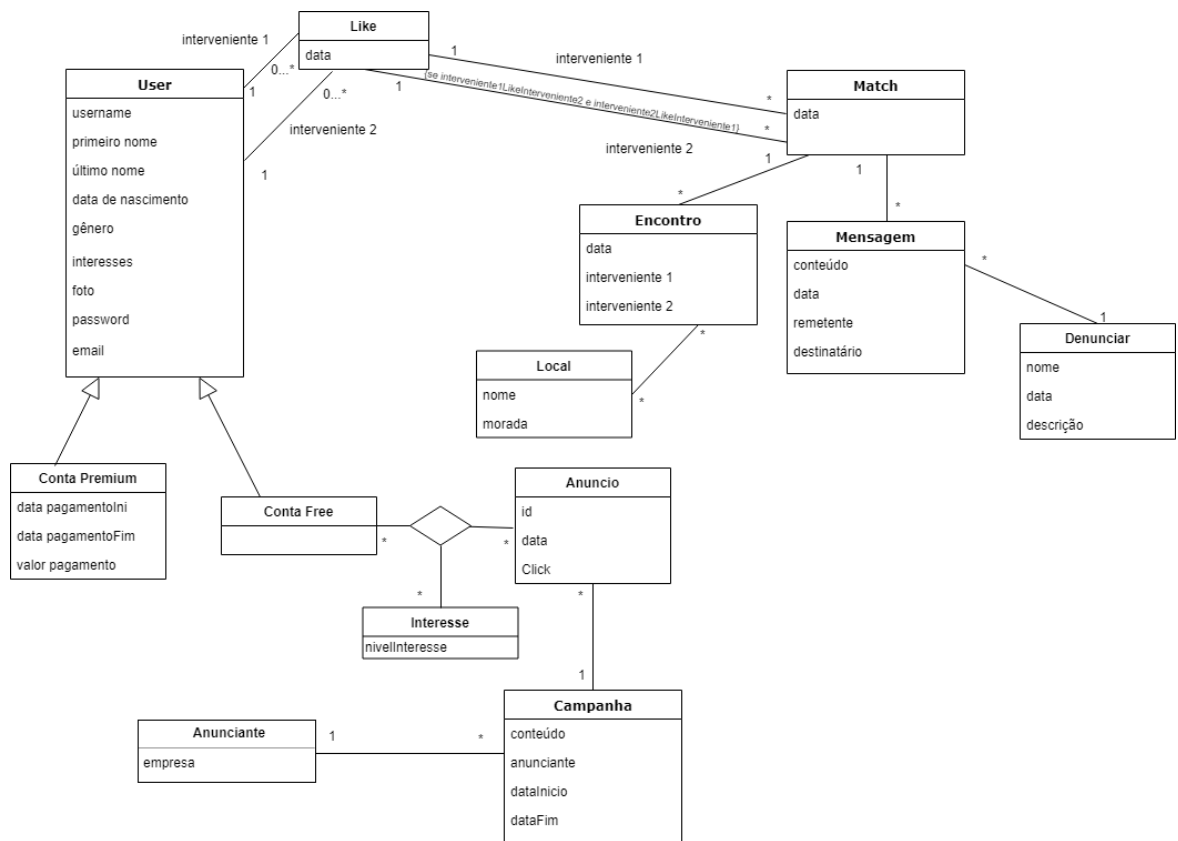
Neste trabalho pretende-se criar uma base de dados que permita guardar os dados associados a uma aplicação de encontros. A aplicação terá um UX muito semelhante ao Tinder, ou seja, haverá um **Match** entre dois utilizadores, sempre trocarem um like entre si.

Assim, é necessário, em primeiro lugar, guardar as informações relativas aos *users*, como *username*, *email*, *interesses*, etc. Cada **User** poderá ser **Free** ou **Premium**, para o qual terá de se guardar informações de pagamento. Todos os utilizadores não pagantes terão de visualizar anúncios publicitários, selecionados de acordo com o seu perfil. Deste modo, teremos também de guardar informações relativas às empresas parceiras e às campanhas que cada uma terá ativa, assim como ao interesse de cada utilizador (que se pode alterar ao longo do tempo, visto que é necessário repetir anúncios) em cada anúncio de forma a otimizar a publicidade apresentada.

Para cada **Match** é necessário guardar uma *data*. Após esta data, os users terão a possibilidade de trocar mensagens e de as denunciar, caso sejam contra as regras da plataforma. Por fim, se houver encontros entre os intervenientes, caso queiram partilhar esta informação a plataforma guarda-la-á de forma a otimizar a experiência para futuros clientes.

Capítulo 2

Diagrama de Classes



Capítulo 3

Descrição

Cada **User** é **Premium** ou **Free**, daí a classe **User** ser uma generalização destas duas.

A cada **User** vai ser apresentado o mesmo anúncio várias vezes, e para cada anúncio, o utilizador dará um nível de interesse, por cada vez. Assim conseguimos criar uma ligação ternária, devendo-se a um **User** com conta **Free**, ao seu nível de interesse e ao anúncio mostrado. Por sua vez, cada **Anuncio** pertencerá a apenas uma só **Campanha**, que pertencerá a apenas um só **Anunciante**. Contudo cada anunciante poderá ter várias campanhas e cada **Campanha** poderá ter vários anúncios, justificando-se assim a criação de relações um para muitos.

Um **Match**, é feito através de um **constraint**, e realiza-se entre dois **User** e para tal, é necessário que o interveniente 1 dê um **Like** no interveniente 2 e vice-versa.

Para cada **Match**, haverão várias mensagens e nenhum ou vários encontros (multiplicidade um para muitos). Por último, cada encontro poderá ter vários locais associados (multiplicidade muitos para muitos).