Engenharia de Software

2019-2020

Plataforma de Outsourcing de Tarefas

Iteração 1

1 Enunciado

A *startup Tasks for Joe* (T4J) dedica-se a facilitar e promover o contacto entre pessoas que trabalham por conta própria (*freelancers*) e organizações que pretendem contratar alguém externo (*outsourcing*) para a realização de determinadas tarefas. Para promover e suportar o seu negócio, a T4J pretende desenvolver uma plataforma informática que, por uma lado, permita que qualquer organização interessada possa registar-se na plataforma de forma a poder publicar tarefas e gerirem elas próprias o processo de adjudicação dessas tarefas a *freelancers*; e por outro lado, permita que os *freelancers* acedam facilmente a essas tarefas e possam candidatar-se à realização das mesmas.
Desde já, perspetiva-se que a plataforma seja acedida por vários utilizadores com diferentes papeis, tais como:

- Administrativos: são colaboradores da T4J afetos à gestão da plataforma e, em particular, por realizar na plataforma várias atividades de suporte ao negócio, tais como, definir áreas de atividade (e.g. IT, Marketing, Design), definir categorias de tarefas (e.g. desenvolvimento aplicações web, desenvolvimento de aplicações móveis) e especificar competências técnicas requeridas para a realização de tarefas.
- Gestor de Organização: pessoa indicada como gestor da organização aquando do registo da organização na plataforma. Assume-se que é um colaborador dessa organização, sendo responsável por especificar na plataforma outros colaboradores dessa mesma organização;
- Colaborador de Organização: pessoa registada na plataforma como sendo alguém que atua em representação de uma determinada organização. Entre outras responsabilidades, cabe-lhe especificar tarefas para posterior publicação pela organização respetiva;
- *Freelancers*: pessoas que se propõem a realizar as tarefas publicadas pelas organizações.

As interações dos utilizadores supramencionados devem ser precedidas de um processo de autenticação. A utilização da plataforma por outras pessoas é restrita ao registo de organizações. Aquando deste registo é obrigatório requerer o nome da organização, o seu número de identificação fiscal (NIF), o endereço postal, um contacto telefónico, um endereço web, um endereço de correio eletrónico (email) e os dados do colaborador responsável pelo registo (nome, função, contacto telefónico, endereço de email). Após o registo da organização, os seus colaboradores pode aceder imediatamente à plataforma.
Tanto uma área de atividade como uma competência técnica caracterizam-se através de um código único, uma descrição breve e outra mais detalhada respetivamente. Uma competência técnica caracteriza-se ainda por ser referente a uma dada área de atividade. Por outro lado, cada categoria de tarefa

caracteriza-se por um identificador interno (automático), uma descrição, a área de atividade em que se enquadra (apenas uma) e uma lista de competência técnicas tipicamente requeridas para a realização de tarefas dessa categoria. Algumas destas competências têm caracter obrigatório e outras apenas são desejáveis.

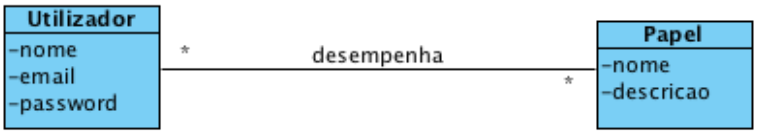
Cada tarefa caracteriza-se por ter uma referência única por organização, uma designação, uma descrição informal e outra de carácter técnico, uma estimativa de duração e custo bem como a categoria em que a mesma se enquadra. Enquanto a mesma não for publicada, o acesso à tarefa é exclusivo aos colaboradores da organização respetiva.

No desenvolvimento deste sistema a equipa de desenvolvimento deve: (i) adotar boas práticas de identificação de requisitos e de análise e design de software OO; (ii) implementar o núcleo principal do software em Java; (iii) adotar normas de codificação reconhecidas e (iv) reutilizar o componente de gestão de utilizadores existente na T4J baseado em identificador de utilizador (i.e. email) e palavra-passe (cf. documentação).

A designação comercial da plataforma e outros dados que venham a ser relevantes devem ser especificados por configuração aquando da sua implantação.

2 Gestão de Utilizadores

O componente de gestão de utilizadores foi desenvolvido por outra equipa de desenvolvimento que também é responsável pela sua manutenção/evolução. Este componente visa satisfazer um conjunto de requisitos/funcionalidades genéricas e comuns a várias aplicações.
Os conceitos principais abordados neste componente estão representados no modelo de domínio apresentado de seguida.



De acordo com o mesmo, um "*Utilizador*" desempenha vários "*Papel*" (i.e. funções), sendo que o mesmo "*Papel*" também pode ser desempenhado por vários "*Utilizador*".
Este componente disponibiliza as suas funcionalidades aos outros sistemas através de um ponto único de acesso (i.e. uma classe de software) denominado "*AutorizacaoFacade*". Desta forma, a estrutura interna do componente pode ser alterada sem afetar as aplicações que estão a usar este componente.
Assim sendo, a documentação relevante com vista à sua reutilização consiste na descrição das classes "*AutorizacaoFacade*" e "*SessãoUtilizador*" (cf. diagrama de classes). A primeira disponibiliza os seguintes métodos:

- *"boolean registaPapelUtilizador(String strPapel)"*: permite definir a existência de um novo papel/função de interesse para a aplicação;
- *"boolean registaPapelUtilizador(String strPapel, String strDescricao)"*: semelhante ao método anterior, com a diferença de permitir associar uma descrição a esse papel/função;
- *"boolean registaUtilizador(String strNome, String strEmail, String strPassword)"*: permite definir a existência de um novo utilizador que pode usar a aplicação;
- *"boolean registaUtilizadorComPapel(String strNome, String strEmail, String strPassword, String strPapel)"*: semelhante ao método anterior, com a vantagem de associar imediatamente um papel/função a esse utilizador;
- *"boolean registaUtilizadorComPapeis(String strNome, String strEmail, String strPassword, String[] papeis)"*: semelhante ao método anterior, com a diferença de poder associar imediatamente mais do que um papel/função a esse utilizador;
- *"boolean existeUtilizador(String strId)"*: permite saber se existe algum utilizador no sistema que seja identificável pelo identificador (strId) especificado;
- *"SessaoUtilizador doLogin(String strId, String strPwd)"*: permite autenticar um utilizador com vista à utilização da aplicação. O resultado é uma sessão de utilizador (i.e. classe *SessaoUtilizador*);
- *"SessaoUtilizador getSessaoAtual()"*: retorna (se existir) a sessão de utilizador ativa;

- *"void doLogout()"*: termina a sessão de utilizador que estiver ativa no momento.

Uma sessão de utilizador (*SessaoUtilizador*) disponibiliza os seguintes métodos:

- *"boolean isLoggedIn()"*: indica se a sessão possui efetivamente um utilizador autenticado com sucesso ou não;
- *"boolean isLoggedInComPapel(String strPapel)"*: indica se o utilizador afeto à sessão possui o papel/função especificado como parâmetro;
- *"String getNomeUtilizador()"*: retorna o nome do utilizador afeto à sessão;
- *"String getIdUtilizador()"*: retorna o identificador do utilizador afeto à sessão;
- *"String getEmailUtilizador()"*: retorna o email do utilizador afeto à sessão.

Por fim, salienta-se que o identificador dos utilizadores é o meio mais adequado para que as aplicações que usam este componente relacionem os utilizadores do sistema com classes/objetos específicos do domínio de aplicação.