



Sistema de rastreio de tempo em tarefas



Trabalho feito por:

Vasco Cardoso 23438

Nuno Azevedo 21631

31/1/2021

Trabalho submetido por:

Nuno Azevedo 21631



Índice

1. Introdução-----	3
2. Objetivos-----	3
3. Implementação:	
3.1. Arquitetura da solução (Diagrama de Classes)-	4
3.2. Especificação das classes-----	5
3.3. Métodos-----	6
3.4. Descrição dos Ficheiros Criados-----	7
3.5. Outras informações relevantes-----	7
4. Conclusão -----	8



Introdução:

Este projeto consiste no desenvolvimento de um sistema, em linguagem Java, de rastreio de tarefas para os freelancers poderem anotar o trabalho por eles realizado diariamente, assim como cobrar o valor adequado aos seus clientes.

Objetivos:

No programa para o Utilizador comum é possível:

- Criar um utilizador
- Fazer Login
- Ver e editar o Perfil
- Criar uma nova tarefa
- Ver todas as tarefas
- Terminar um tarefa
- Editar uma tarefa
- Criar um novo projeto
- Ver todos os projetos
- Adicionar outros Utilizadores a um projeto
- Aceitar convites para um projeto
- Eliminar um Utilizador de um Projeto
- Imprimir um relatório Mensal

Como Manager é possível:

- Fazer tudo como utilizador comum
- Adicionar novos Utilizadores
- Listar utilizadores
- Imprimir o relatório mensal de algum Utilizador
- Editar o perfil de um Utilizador
- Ver o histórico de ações de um utilizador
- Tornar um utilizador comum num manager

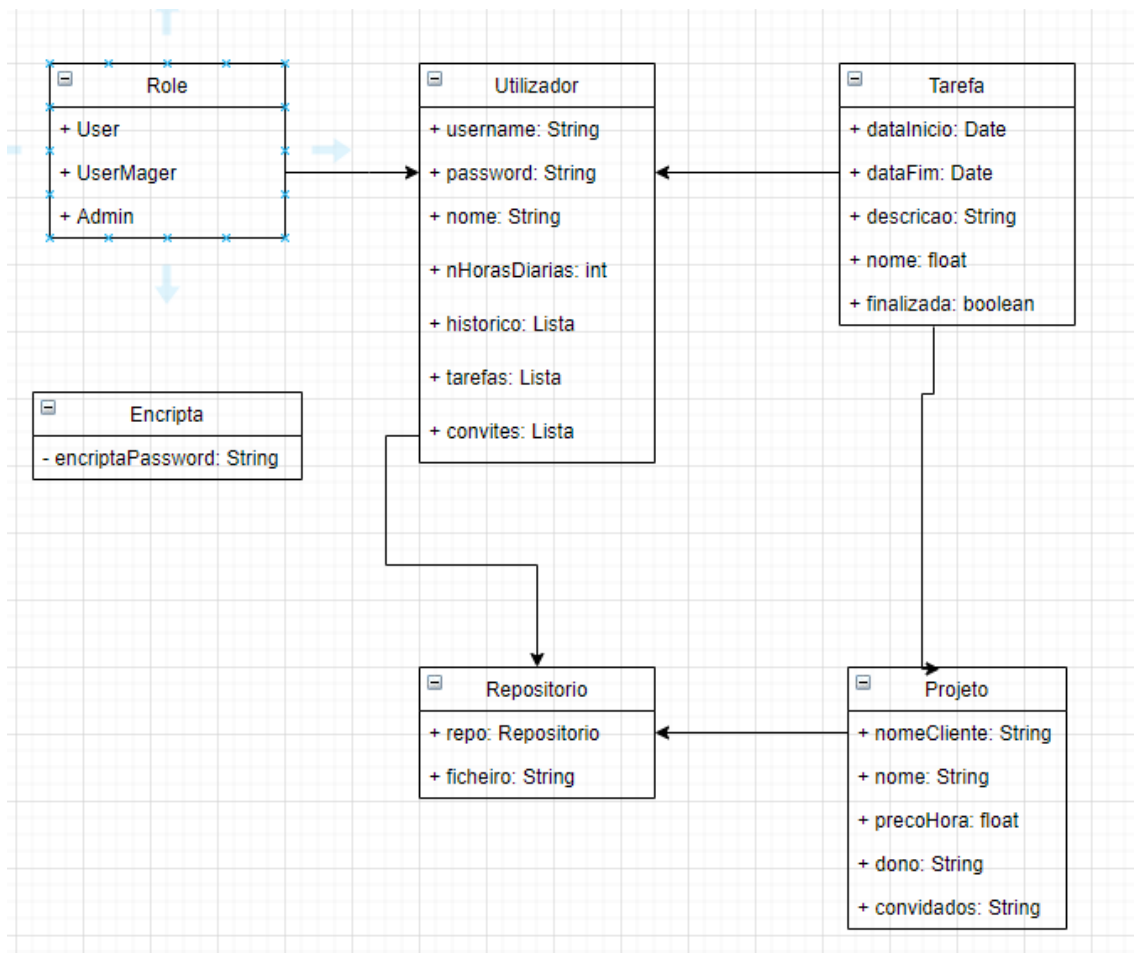
Como Admin é possível:

- Fazer tudo que um manager pode
- Tornar um utilizador (comum ou manager) em admin
- Tirar permissões a um utilizador



Implementação:

Arquitetura da solução (Diagrama de Classes)





Especificação das classes

Role:

- Enum, para diferenciar as permissões do utilizador.

Encripta:

- Classe para guardar a função de encriptar a password.

Tarefa

- Classe para estruturar informação de uma tarefa como a sua data inicial final, se está terminado, nome e descrição

Projeto:

- Classe que trata do processamento de informação relativo aos projetos, possuir nome (único), nome de cliente, nome de utilizador do dono, preço relativo, lista de tarefas e lista de convidados.

Utilizador:

- Classe que processa as informações sobre o utilizador como username, password, nome, horas diárias, uma lista de histórico das suas ações, uma lista de tarefas e uma lista de convites para projetos.

Repositório:

- Classe principal que trata das informações gerais do projeto como logins registos, etc. Possui uma lista de projetos, uma lista de utilizadores, e um utilizador que será o “utilizador logado”, ou seja, o utilizador que tem sessão iniciada para mais fácil processamento de informação.



Métodos (identificação da classe e explicação do método)

Classe – Encripta

Método – encriptaPassword(String password)

Este método é chamado sempre antes de realizar alguma ação com a password, como login, registo ou edição de perfil, que usando o “MessageDigest”, lê uma String, e devolve outra String desta vez encriptada.

Classe – Repositório

Método – Login

Este método é chamado sempre que se faz o login, recebe o nome de utilizador e a password inserida, já encriptada, que depois compara com o resto dos registos no sistema, e devolve se foi bem-sucedido ou não.

Classe – Repositório

Método – getTarefasRelatorioUtilizador

Este método recebe como parâmetro uma data inicial, data final um boolean, e um nome de utilizador que são usados para verificar as tarefas presentes no utilizador, dependendo do que foi passado, por exemplo, a data inicial e a data inicial da tarefa devem sempre ser compreendidas entre as datas passadas, a tarefa deve estar realizada ou não dependendo do boolean, e devem todas pertencer ao utilizador passado, que por fim será útil para o balanço



Classe – Repositório

Método – getTarefasRelatorioProjeto

Este método é igual ao método de utilizador, mas é relativo ao projeto passado como parâmetro em vez do utilizador.

Descrição dos Ficheiros Criados

Foi criado apenas um ficheiro “repositorio.txt”, que guarda a informação toda presente na classe Repositório, e que depois é lido quando a aplicação é iniciada de novo.

Outras informações relevantes

Os utilizadores com permissões Administrador podem ver as informações de todos os outros utilizadores, e podem editar informações acerca do seu perfil, como alterar permissões username e password.

Os utilizadores com permissões UserManager podem apenas ver e editar informações sobre o utilizador.

O utilizador pode apenas criar projetos, tarefas consultar balanços e editar o seu próprio perfil.

Todos os utilizadores são tratados como Freelancers, sendo que os utilizadores não “User” podem editar e ver informações sobre outros.



Conclusão:

Concluindo, com a realização deste trabalho, feito através dos conhecimentos aprendidos e aplicados nas aulas da Unidade Curricular de programação 2, ficamos a entender melhor como funciona a linguagem Java e como se faz uma Interface gráfica. Este trabalho foi um bom aprendizado para melhorar em trabalhos futuros.