**Paradigmas da Programação**

**Trabalho Prático – descrição das funcionalidades**

**Realizado:**

**Manuel Garnel -1131325**

**Arcénio Uate -1171351**

1. **Contextualização do trabalho**

O trabalho prático proposto na disciplina *Paradigmas da Programação* (*PPROG*) consistiu no desenvolvimento de uma aplicação, em *Java*, que implementasse as funcionalidades definidas no caso de uso **UC3-**Registar Perfil de Autorização e UC-6 Aceder Área Restrita que está a ser desenvolvido em *Engenharia de Software* (*ESOFT*). Porém, deve-se realçar que se realizaram algumas omissões no âmbito dos atributos de cada objeto e de algumas classes com as quais havia algum tipo de relação com as classes principais destes casos de uso. Convém realçar que foi realizado um esforço para minimizar o número dessas situações.

De forma a dar resposta ao problema apresentado, foi criado um projeto,

através do ambiente integrado de desenvolvimento *NetBeans*, que se encontra dividido

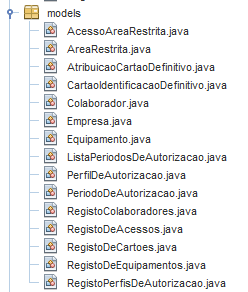
nos *packages* seguintes:

. *models*, o qual contém as classes necessárias à implementação das funcionalidades pretendidas;

. *ui*, que contém as classes responsáveis pela criação da interface gráfica da aplicação.

-controllers, que contém os controladores das classes que surgem se encontram nos models.

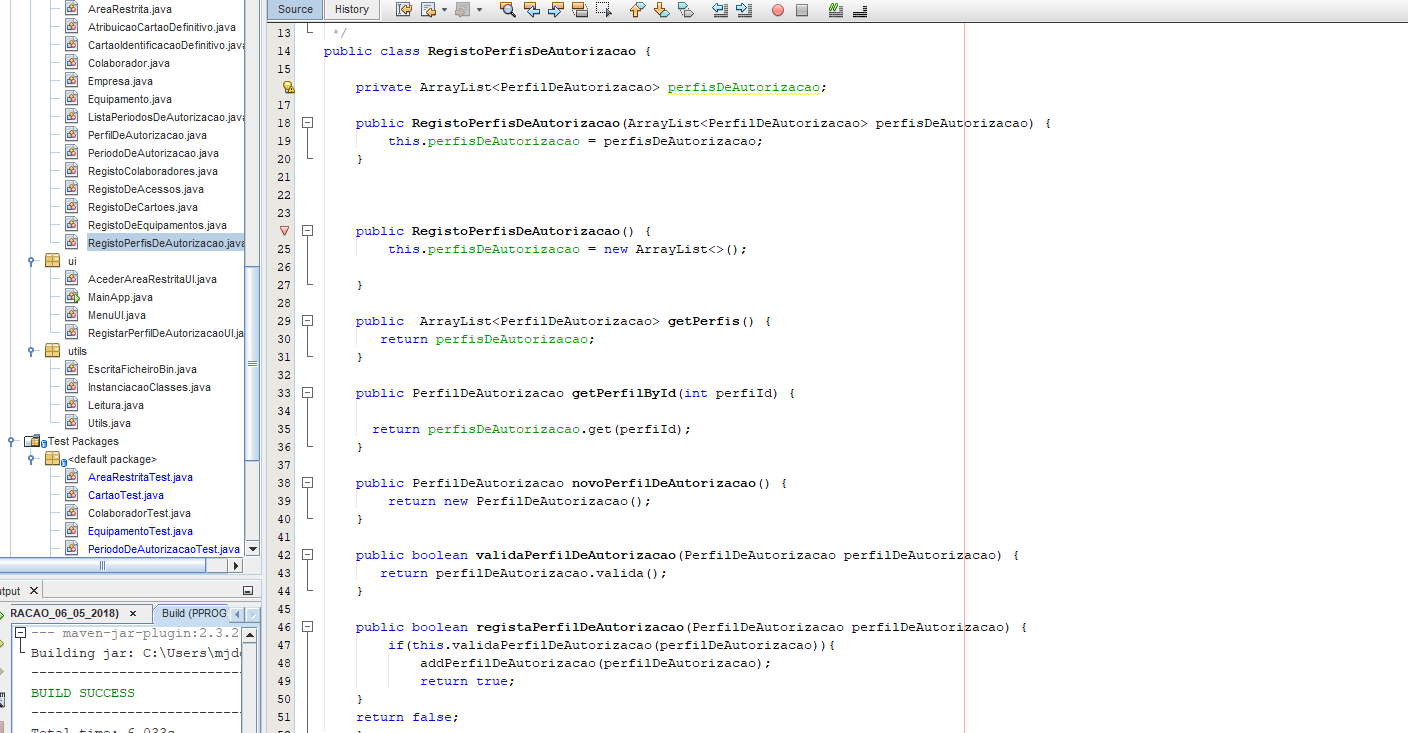
No que diz respeito ao package models podemos encontrar as classes descritas na figura seguinte.



No que diz respeito aos controllers, a figura seguinte apresenta a lista dos mesmos.

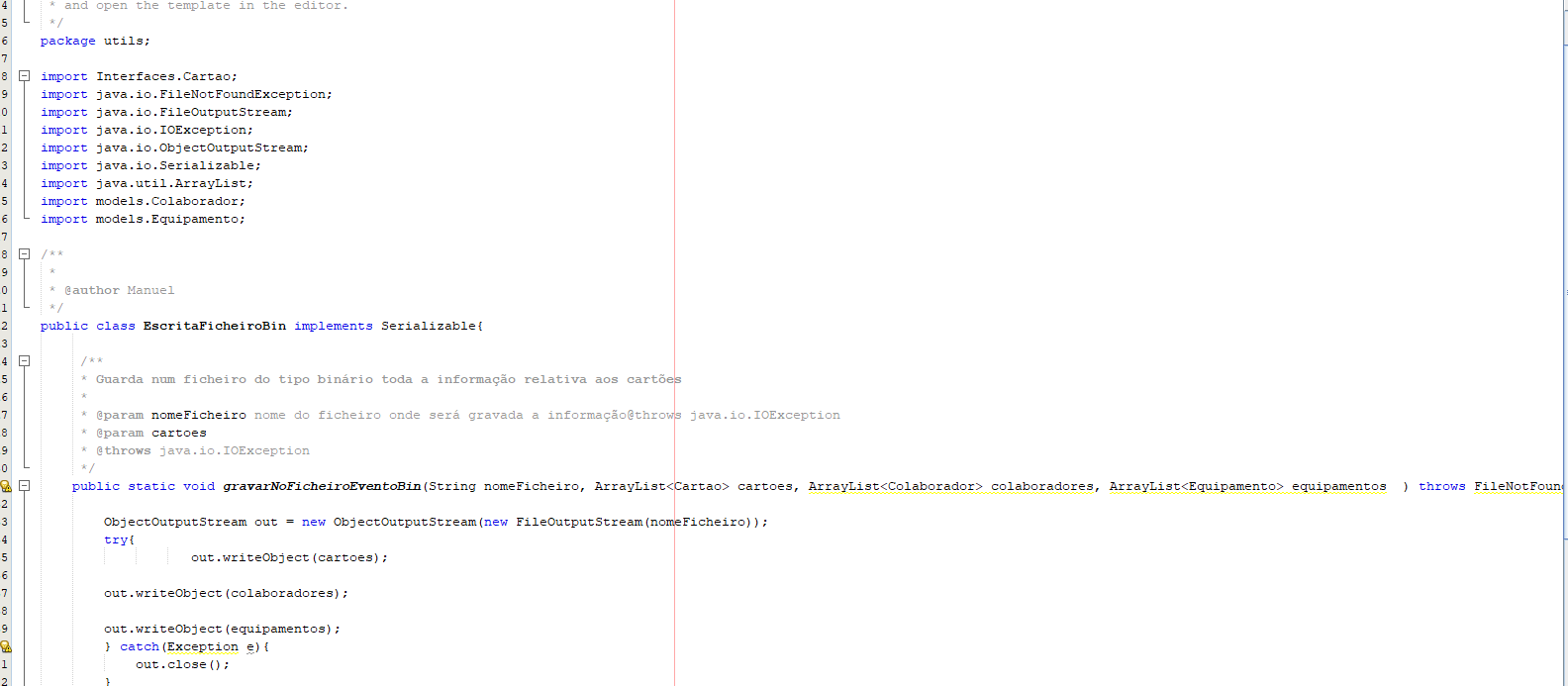
Relativamente ao package ui, existem as classes AcederAreaRestritaUI, MenuUI, e RegistarPerfilDeUtilizacaoUI.

**2.Considerações sobre as funcionalidades implementadas**

Relativamente às funcionalidades diretamente ligadas às UC3 e UC6, estão na sua maioria implementadas. A figura seguinte apresenta alguns métodos da classe “RegistoPerfilDeAutorizacao.class”.

A classe RegistarPerfilDeAutorizacaoController possui as funcionalidades necessárias para interligar classes como Empresa (que agrega informação sobre os registos de colaboradores, de equipamentos, etc), RegistoPerfisDeAutorizacao, PerfilDeAutorizacao, Equipamentos e RegistoDeEquipamentos.

A implementação da classe que instancie todas as classes necessárias também está devidamente realizada e, apesar de centrar-se nas classes chave do projeto, tais como Cartao , Colaborador, Equipamento, não foram descuradas as ligações existentes entre todas estas classes e as demais existentes. Esta classe permite instanciar a classe Cartao, AreaRestrita, Equipamento, registar um novo equipamento através da classe Empresa visto ser esta conhecedora do registo de equipamentos através da classe RegistoDeEquipamento. Também instancia o perfil de autorização e colaborador e apoia-se na classe Empresa para ter o registo de colaboradores atualizado.A atribuição de cartão também não foi posta de parte nesta classe.

A persistência de dados, assim como a sua leitura e escrita também está devidamente salvaguardada, A figura seguinte apresenta um extrato da classe desenvolvida.

Para o requisito apresentado no enunciado deste projeto, implementação de uma interface, optou-se por utilizar a consola. Esta apresenta uma interatividade muito ampla com o utilizador, permitindo explorar todas as funcionalidades pretendidas.

Quanto à instanciação das classes a partir da leitura de um conteúdo de ficheiro texto também está devidamente salvaguardada. A figura seguinte apresenta um extrato do código relativo à funcionalidade descrita na frase anterior.

