

---

# PROGRAMAÇÃO ORIENTADA A OBJETOS

---

Trabalho Prático  
Universidade Autónoma de Lisboa



## **Docente**

Raul Dionísio

## **Discentes**

Nuno Silva nº20160645  
Valeriano Filipe Calolégio nº20160937

31 DE JANEIRO DE 2018

## Índice

Introdução.....	2
Resumo do trabalho.....	3
Resumo do trabalho em inglês.....	4
Lista de Quadros e Gráficos .....	5
Lista de fotografias .....	7
Desenvolvimento .....	10
Conclusão .....	11
Bibliografia .....	12
Anexos .....	<b>Erro! Indicador não definido.</b>

# Introdução

Neste projeto tínhamos como objetivo modelizar um sistema de informação tendo como referência uma farmácia, nós iremos apresentar um projeto em que temos um sistema de informação que nos permite guardar funcionários, os seus respetivos registos de atividade e algumas informações sobre estes, como o nome, número de funcionário, senha, género, permitindo assim que tenhamos um sistema de login antes que se possa aceder ao sistema de gerenciamento.

Temos quatro módulos, gestão de pessoal, gestão de recursos, contabilidade e gestão de stock, cada módulo tem um conjunto de funcionalidades e informações usadas nas interfaces gráficas

## Resumo do trabalho

Neste projeto, temos quatro módulos, a gestão de pessoal, gestão de recursos, stock e a contabilidade. Ao iniciar-se o programa é pedido o usuário(número de funcionário) e a sua respetiva senha, se os dados estiverem corretos entra-se no sistema de gerenciamento, o estado do funcionário passa a ativo, é guardado no registo de atividade a data de entrada/saída, o número de funcionário e o seu nome.

Se os dados estiverem incorretos, aparecerá no ecrã uma mensagem a dizer que o usuário ou a senha estão incorretos, no sistema de gerenciamento temos quatro botões para a gestão de pessoal, gestão de recursos, contabilidade e gestão de stock, bem como nome, número e o departamento a que o funcionário pertence.

Cada funcionário tem restrições aos módulos a que têm acesso por exemplo, um funcionário do departamento de contabilidade não pode ter acesso ao módulo de gestão de pessoal, caso um funcionário tente aceder a um módulo a que não tenha permissão aparecerá no ecrã uma mensagem a dizer, acesso negado.

O número de empregado, a senha, o nome, sexo, cargo e se está ativo ou não é guardado num ficheiro csv.

## Resumo do trabalho em inglês

In this project, we have four modules, personnel management, resource management, stock and accounting. At the start of the program the user (employee number) and his / her password are requested, if the data is correct, it enters the management system, the employee status becomes active, it is stored in the activity record the date entry / exit, the employee number and your name.

If the data is incorrect, a message will appear on the screen saying that the user or password is incorrect, in the management system we have four buttons for personnel management, resource management, accounting and stock management, as well as name, number and the department to which the employee belongs.

Each employee has restrictions on the modules they have access for example, an accounting officer can not have access to the personnel management module, if an employee tries to access a module that does not have permission a message will appear on the screen saying , access denied.

The employee number, password, name, gender, job title and whether it is active or not is stored in a csv file.

# Lista de Quadros e Gráficos

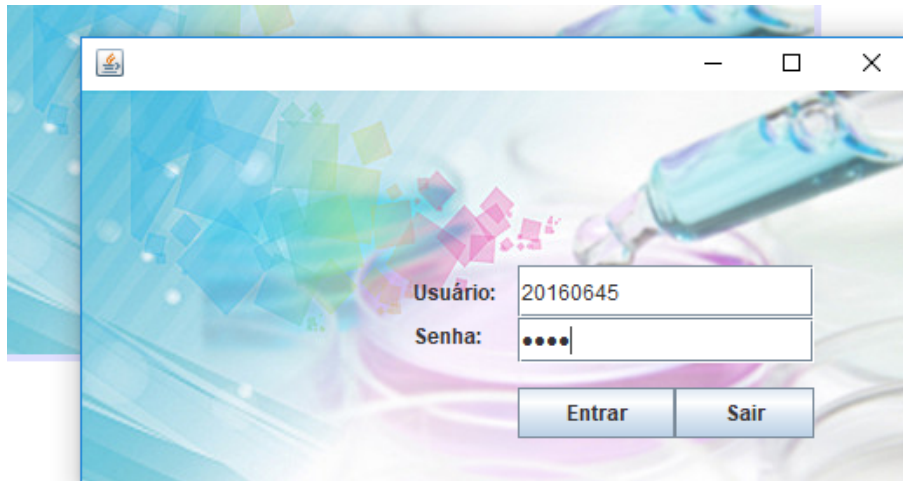
1. TelaLogin
  - 1.1. JButtonSair
  - 1.2. JButtonEntrar
  - 1.3. JLabelUsuario
  - 1.4. JLabelSenha
  - 1.5. JPasswordFieldSenha
  - 1.6. JPasswordFieldUsuario
  - 1.7. JLabelFundoLogin
2. TelaPrincipal
  - 2.1. JMenuBar2
  - 2.2. JMenu3
  - 2.3. JMenuItemTerminarSessao
  - 2.4. JMenuItemAjuda
  - 2.5. JMenuItemManual
  - 2.6. JMenuItemEquipe
  - 2.7. JButtonGestaoPessoal
  - 2.8. JButtonContabilidade
  - 2.9. JButtonGestaoRecursos
  - 2.10. JButtonGestaoStock
  - 2.11. JPanelInformacoesGerais
  - 2.12. JLabel1
  - 2.13. JLabelNome
  - 2.14. JLabelNFuncionario
  - 2.15. JLabelDepartamento
  - 2.16. JFormattedTextFieldNome
  - 2.17. JFormattedTextFieldNFuncionario
  - 2.18. JFormattedTextFieldDepartamento
3. TelaGestaoPessoal
  - 3.1. JMenuBar1
    - 3.1.1. JMenuItemSair
    - 3.1.2. JMenuItemTerminarSessao
  - 3.2. JButtonNovo
  - 3.3. JButtonEditar
  - 3.4. JButton1
  - 3.5. JPanelNovo
    - 3.5.1. JLabelNome
    - 3.5.2. JLabelGenero
    - 3.5.3. JLabelDepartamento
    - 3.5.4. JLabelSenha

- 3.5.1.5. JLabelNFuncionario
- 3.5.1.6. JButtonGuardar
- 3.5.1.7. JButtonCancelar
- 3.5.1.8. JTextFieldNfuncionario
- 3.5.1.9. JTextFieldNome
- 3.5.1.10. JTextFieldGenero
- 3.5.1.11. JTextFieldSenha
- 3.5.1.12. JComboBoxDepartamento

# Lista de fotografias

## 1. Login

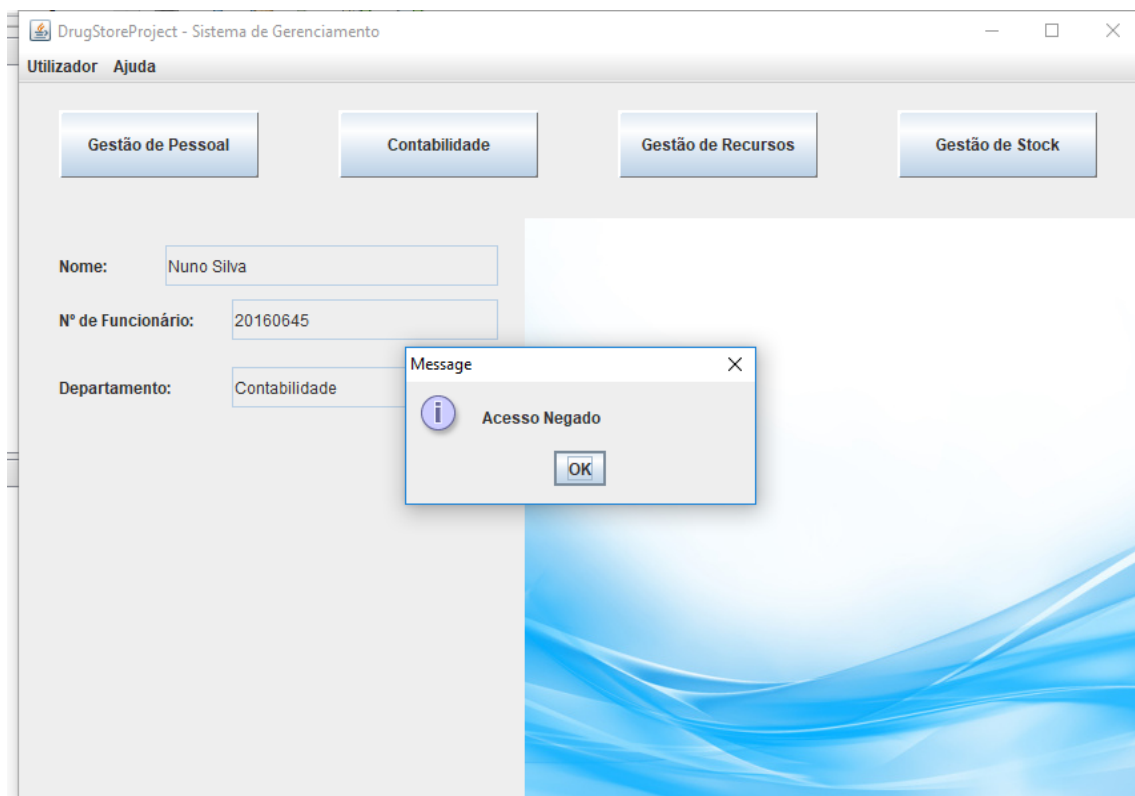
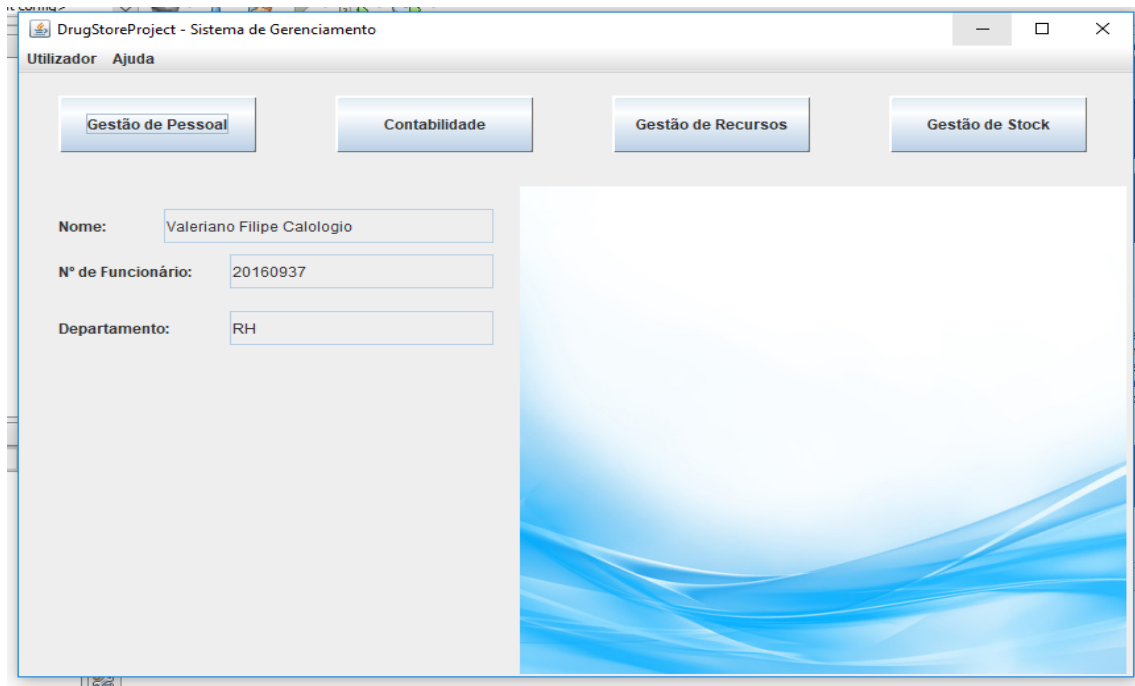
Foi desenvolvido antes de tudo uma interface gráfica chamada TelaLogin que valida os dados inseridos (procurando no registoHistorico.csv) escrevendo a no ficheiro de registo de Atividades.





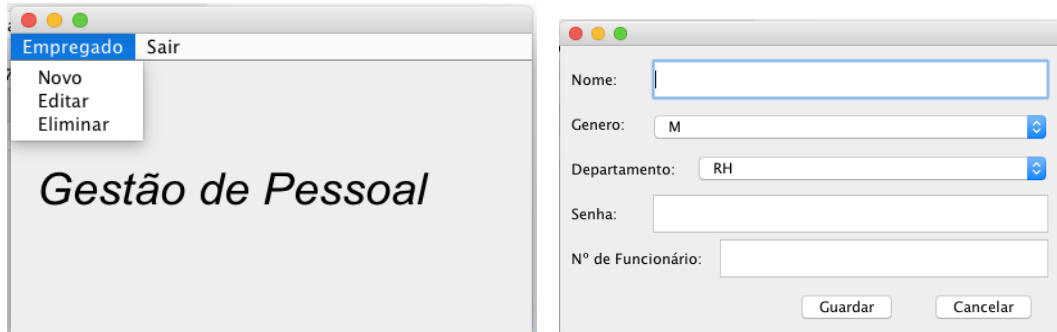
## 2. Tela principal (Sistema de gerenciamento)

Depois da tela de login e ser validado, é aberto a Tela Principal, onde é apresentado os módulos do sistema e informações do usuário ativo, seguindo uma regra de restrição de acesso aos módulos segundo o Departamento/Cargo/Módulo em que o usuário ativo pertence. O exemplo a baixo mostra usuário do modulo de Contabilidade tentando acessar o modulo de Gestão de Pessoal.



### 3. Gestão de pessoal

Através do mecanismo de registo é possível inserir dados para um novo usuário (Empregado), onde os dados simplesmente são adicionados no ficheiro de registo histórico.



#### 4. Registro Histórico

[illegible]

## 5. Registo de atividade

[illegible]

# Desenvolvimento

A entrada para o programa é feita no botão entrar, quando se carrega no botão entrar chama-se o evento como se pode ver na seguinte linha de código:

```
- private void jButtonEntrarActionPerformed(java.awt.event.ActionEvent evt) {
```

No evento encontra-se o código de validação de login:

```
- if (validarLogin(jTextFieldUsuario.getText(), jPasswordFieldSenha.getText())){
```

Se os dados forem aceites é aberta a tela principal:

```
-new TelaPrincipal().setVisible(true);
```

```
- dispose();
```

O início de atividade é feito através de um sistema try catch:

```
try{
```

```
    FileWriter registoAtiv = new FileWriter(registoAtividade.getName(), true);
```

```
    BufferedWriter entrada = new BufferedWriter(registoAtiv);
```

```
    entrada.newLine();
```

```
    entrada.write(new Date().toString()+";"+linha[0]+";"+linha[2]+"; Entrada");
```

```
    entrada.close();
```

```
}
```

```
catch(IOException ex){
```

```
    JOptionPane.showConfirmDialog(rootPane, "Foi impssivel registar Entrada!");
```

```
}
```

```
}
```

```
else{
```

```
    JOptionPane.showMessageDialog(rootPane, "Usuário ou Senha Incorrecta");
```

```
}
```

```
}
```

Quando se carrega no botão sair é feito um simples fecho de sistema:

```
- System.exit(0);
```

O evento da senha chama o evento do botão entrar

## Conclusão

Com este trabalho realizado em java (programação orientada a objetos) conseguimos ter uma melhor noção de como funcionam os sistemas de gerenciamento por detrás das pequenas empresas, conseguimos ganhar uma melhor noção das utilidades do java em termos de organização de código e organização de quadros, organização esta que faz com que o projeto seja facilmente perceptível, e que as alterações sejam facilmente realizadas.

# Bibliografia

<https://www.caelum.com.br/apostila-java-orientacao-objetos/interfaces/#interfaces>

<https://docs.oracle.com/javase/tutorial/java/landl/createinterface.html>