

UFCD 5425

Trabalho de Programação em *JAVA* (com *SWING* e *MySQL*)

```
243 int main(){
244     FILE fp;
245     ClearConsoleToColors(15, 1);
246     int i;
247     SetConsoleTitle("Project - Programming Code Abstract Screen");
248     int choice;
249     char ch = 'a';
250     while(1){
251         printf("1. Find Out the Day\n");
252         printf("2. Print all the day of month\n");
253         printf("3. Add Note\n");
254         printf("4. EXIT\n");
255         printf("ENTER YOUR CHOICE : ");
256         scanf("%d",&choice);
257         system("cls");
258         switch(choice){
```

Formador: Gonalo Ferreira

Curso: Jovem+Digital - Linguagens de Programação: Programação em JAVA

Data: Setembro de 2025



INSTITUTO DO EMPREGO
E FORMAÇÃO PROFISSIONAL



CENTRO DE EMPREGO E FORMAÇÃO PROFISSIONAL DO PORTO

UFCD 5425 - Projeto em *Java*: Gestão de Registos

Designação da UFCD: Projeto de tecnologias e programação de sistemas de informação

Código: 5425

Carga horária: 50 horas

Objetivo: Implementar um projeto em *Java* que dê uma visão global do ciclo de desenvolvimento, desde a definição até à entrega.

Enunciado do Trabalho

O formando deve desenvolver uma aplicação em *Java* para gerir registos à sua escolha (clientes, produtos, tarefas, etc).

A aplicação deve ser feita em **janela única** com *CardLayout*, clara, funcional e fácil de usar.

Núcleo obrigatório (até 17 valores)

1. **Login simples** com utilizadores pré-inseridos (em memória ou em BD).

Valorizado se lido da **BD** com tabela *users*. - 2 valores

2. **Dashboard** inicial com relatório simples (total de registos e últimos 5 inseridos) - 2 valores

3. **Listagem em tabela** (*JTable*) com ordenação por coluna, pesquisa por texto e contagem de registos. - 4 valores

4. **CRUD completo** (Criar, Ler, Editar, Apagar) - 5 valores

⑩ Ao Apagar, pedir **confirmação por popup** (*JOptionPane*).

⑩ Mensagens de **feedback** (“Registo inserido com sucesso”).

5. **Exportar CSV** da listagem. - 1 valores

6. **Textos PT↔EN** guardados em **mapas internos**, com botão para alternar em tempo real. - 2 valor

7. **Sair da aplicação** com *popup* de confirmação ou na janela. - 1 valor

Extras (até 3 valores)

Cada formando deve implementar **pelo menos 2 extras**.
Quem fizer mais pode acumular até ao limite de 3 valores.

- ⑩ E01. Importar CSV com relatório de linhas inválidas
- ⑩ E02. Menu de clique direito na tabela (Ver, Editar, Apagar, Exportar linha)
- ⑩ E03. Paginação na tabela (ex. 20 por página)
- ⑩ E04. Validação avançada nos formulários (mensagens junto ao campo)
- ⑩ E05. Ligação real a *MySQL/MariaDB* com *prepared statements* e scripts *SQL*
- ⑩ E06. Hash de password em *SHA-256* com *salt*
- ⑩ E07. Ficheiro de configuração . *properties* externo (opcional, profissionalização)
- ⑩ E08. Relatório gráfico simples na *UI*
- ⑩ E09. Registo de atividades em tabela *audit_log* e lista no *dashboard*
- ⑩ E10. Ficheiro de *log* em disco com erros principais

Base de dados sugerida (para *login* valorizado)

```
DESC users;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
username	varchar(50)	NO		NULL	
password	varchar(50)	NO		NULL	
name	varchar(50)	NO		NULL	
role	varchar(20)	NO		'user'	


Mini guias para simplificar

Configuração embutida

```
final class AppConfig{  
    static final String dbHost="localhost";  
    static final String dbUser="aluno";  
    static final String dbPass="aluno";  
    static final String dbName="gestao";  
}
```

Textos PT↔EN embutidos

```
final class I18n{  
    static java.util.Map<String,String> pt=java.util.Map.of(  
        "title","Gestão de Registos",  
        "login","Entrar",  
        "user","Utilizador",  
        "pass","Palavra-passe"  
    );  
    static java.util.Map<String,String> en=java.util.Map.of(  
        "title","Records Management",  
        "login","Login",  
        "user","Username",  
        "pass","Password"  
    );  
}
```



(Basta alternar o mapa activo ao clicar no botão PT↔EN e atualizar os rótulos).

Entrega

- ⑩ Projeto em . *zip*
- ⑩ *Scripts SQL* (se usarem BD)
- ⑩ Guia rápido com instruções, funcionalidades feitas, limitações e trabalho futuro
- ⑩ Será efectuada uma apresentação do trabalho posteriormente

Avaliação - 0 a 20 valores

- ⑩ **Núcleo obrigatório** - até 17 valores
- ⑩ **Extras** - até 3 valores (mínimo 2 extras escolhidos)
- ⑩ **Nota final**: soma até 20

Plano recomendado de execução (10 passos)

1. Criar janela principal com *CardLayout*.
2. Implementar *Login* simples (pré-inserido, depois opcionalmente BD).
3. Criar modelo da entidade (ex. Cliente).
4. Fazer listagem em *JTable* (ordenar + pesquisar).
5. Criar formulário de inserção.
6. Criar edição e apagar com *popup* de confirmação.
7. *Dashboard* com contagens e últimos registos.
8. Exportar *CSV* da listagem.
9. Textos PT↔EN com botão de alternância.
10. Escolher pelo menos 2 extras e concluir.

Bom trabalho!