Universidade da Beira Interior
Departamento de Informática

Software da Escola Profissional de Hotelaria de Manteigas

Elaborado por:

Diogo Santos - a45842
Nelson Fortuna - a45429
Simão Farias - a45464
Tiago Barreiros - a46118

Orientador:

Professora Doutora Paula Prata

Agradecimentos

Queremos agradecer a todas as identidades, que direta ou indiretamente, estiveram envolvidas neste trabalho desenvolvido no âmbito da unidade curricular, Programação Orientada a Objetos (POO).

Como tal começamos por agradecer à nossa orientadora Prof. Doutora Paula Prata, por suscitar e inserir em nós, alunos, o interesse em saber mais, de nos tornar mais hábeis, proporcionando-nos todas as aprendizagens e métodos, necessários, para a realização deste projeto. De igual modo, realçar as aprendizagens obtidas do primeiro ano, como, por exemplo, a escrita de documentos em LaTeX, lecionada na unidade curricular Laboratórios de Programação (LP), regida, no nosso ano, pelo Prof. Doutor Pedro M. Inácio, a quem devemos bastante reconhecimento e somos igualmente gratos.

Agradecemos também aos nossos colegas, e amigos, com o quais pudemos trocar ideias e um especial agradecimento às nossas famílias que estão neste momento a investir e a acreditar em nós. Da nossa parte um muito obrigado!

Resumo

Este projeto, consistiu em desenhar e implementar uma aplicação, em Java, que permita gerir os recursos de uma escola de formação profissional, que oferece cursos numa determinada área, no nosso caso Hotelaria. A escola oferece vários cursos, que têm um certo número de horas, uma data de início e de fim, um horário, professores que lecionam os cursos e alunos que os frequentam.

Além da gestão de cursos, alunos e professores, neste trabalho, pretende-se que seja possível produzir estatísticas sobre o desempenho de cada escola, bem como manter a persistência dos dados armazenados.

Conteúdo

Palavras-Chave

Software, Escolas, Hotelaria, Estatísticas, Java.

Acrónimos

CC Cartão de Cidadão

EI Engenharia Informática

IDE *Integrated Development Environment*

LP Laboratórios de Programação

POO Programação Orientada a Objetos

UBI Universidade da Beira Interior

UC Unidade Curricular

Capítulo

1

Introdução

1.1 Âmbito e Enquadramento

Encontra-se a analisar um relatório, elaborado por alunos do 2.º ano da Licenciatura Engenharia Informática (EI) da Universidade da Beira Interior (UBI).

Foi confecionado no contexto da Unidade Curricular (UC) Programação Orientada a Objetos (POO) da UBI, desenvolvido presencialmente e à distância, através de pesquisas pessoais.

Abrange duas áreas muito importantes do conhecimento, EI e Gestão, pois desenvolvemos todo um programa que gere uma escola, os seus dados e consequentemente as suas estatísticas.

É importante abordar estes conceitos, entender como o analógico pode ser transformado no digital, como podemos tratar os dados e desenvolver toda uma aplicação em torno deles.

1.2 Motivação

Como referido anteriormente, este projeto une Programação com Gestão, permite aos utilizadores usufruir de um misto de funcionalidades essenciais no nosso dia a dia e em diferentes contextos, hoje gerimos uma escola, amanhã quem sabe uma fábrica.

Além disso, este trabalho foi processado no âmbito das metas definidas pela Prof. Doutora Paula Prata, como gerente da UC Programação Orientada a Objetos, como tal consideramos essa a nossa principal motivação.

A professora teve um papel fundamental, pois despertou em nós alunos, interesse e facultou as aprendizagens necessárias, apesar de ter existido uma

pesquisa extra da nossa parte, como forma de nos superarmos.

1.3 Objetivos

Este trabalho baseia-se no desenvolvimento de várias funções, implementadas em linguagem Java, de cariz matemático, bem como a sua criação e execução em ambiente de linha de comandos.

O principal objetivo do projeto consiste em desenhar e implementar uma aplicação, em Java, que permita gerir os recursos de uma escola de formação profissional, que oferece cursos numa determinada área, no nosso caso Hotelaria. A escola oferece vários cursos, que têm um certo número de horas, uma data de início e de fim, um horário, professores que lecionam os cursos e alunos que os frequentam.

Além da gestão de cursos, alunos e professores, pretende-se que seja possível produzir estatísticas sobre o desempenho de cada escola, bem como manter a persistência dos dados armazenados e elaborar alguns extras.

Como objetivos secundários, vitais para a nossa vida académica, este trabalho permite uma maior aptidão no uso da linguagem Java, promove o trabalho em equipa, de forma a adquirir métodos, experiência e um maior à vontade no confronto com uma coletividade, empresa ou entidade referenciada, neste caso uma reconhecida professora universitária.

1.4 Abordagem

Após a escolha do tema pretendido, reunimo-nos presencialmente para discutir o trabalho, trocámos ideias e criámos uma lista de objetivos, objetivos esses com o principal intuito de agilizar todo o processo. Após esclarecermos algumas dúvidas pertinentes junto da Prof. Doutora Paula Prata, as nossas ideias começaram a ganhar uma nova imagem e o trabalho uma nova dimensão. Contudo, estávamos cientes do grande desafio que nos esperava, portanto, criámos um servidor na plataforma *Discord*, para facilitar todo o processo e permitir a ágil comunicação entre todos.

Dado o meio de comunicação estabelecido, procedemos à sua realização, aprimoramento e conclusão. O relatório foi o ultimo ponto a ser concluido.

1.5 Organização do Documento

De modo a refletir o trabalho que foi feito, este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo – **Introdução** – refere o projeto que nos foi proposto, o seu âmbito e Enquadramento, a Motivação para a sua realização, os objetivos e a forma como foram abordados.

2. O segundo capítulo – **Estado da Arte** – descreve as ferramentas utilizadas na realização deste projeto;

3. O terceiro capítulo – **Desenvolvimento** – apresenta as dependências que existem no programa e os detalhes de implementação;

4. O quarto capítulo – **Testes e Validação** – explicita todos os Testes e Aprimoramentos realizados ao programa;

5. O quinto capítulo – **Conclusões e Trabalho Futuro** – como o nome indica, apresenta uma breve conclusão.

Capítulo

2

Estado da Arte e Trabalhos Relacionados

2.1 Introdução

Neste capítulo o objetivo é descrever as ferramentas que foram utilizadas ao longo deste trabalho. Na secção 2.2, **Estado da Arte**, está explícita a importância que damos às ferramentas que usamos no decorrer do trabalho.

2.2 Estado da Arte

Para a concretização deste trabalho utilizamos o *Integrated Development Environment* (IDE) usado nas aulas práticas, *NetBeans* IDE, o *Discord* como meio de comunicação e o *Overleaf* para a escrita e embelezamento do relatório técnico.

O *NetBeans* IDE foi utilizada para desenhar e implementar o projeto em linguagem Java. Esta tem várias características importantes sendo algumas delas as seguintes:

- Capacidade de auto-completar código;
- Ferramentas para depuração do programa;
- Informa o programador de alguns erros de sintaxe que possam existir.

Recorremos ao uso deste IDE, pois todos nos sentiamos à vontade com a sua utilização.

A plataforma *Overleaf* foi importante na redação do relatório final. Esta também tem características que se destacam, entre as quais:

- Capacidade de auto-completar funções;
- Compila o trabalho várias vezes, com o objetivo de estar sempre tudo atualizado e não se perder nada.
- Na eventualidade de existirem erros, indica os mesmos.

O *Discord* foi o meio de comunicação escolhido, pois como pode observar na figura 2.1, permite-nos uma vasta organização e esse é um fator crucial.

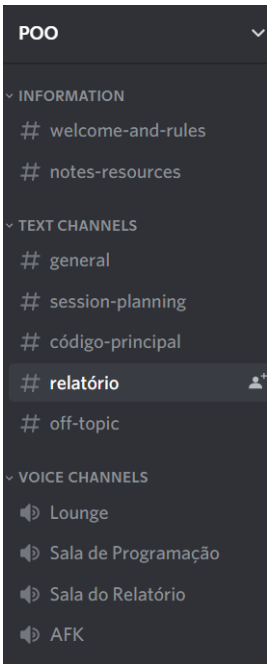

Figura 2.1: Representação do servidor criado no *Discord*.

2.3 Conclusão

Neste capítulo pretendíamos que fosse feita uma abordagem às aplicações utilizadas e fossem enunciadas algumas das características a que recorremos na elaboração do trabalho.

Capítulo

3

Desenvolvimento

3.1 Introdução

Este capítulo apresenta dependências que existem e os detalhes de implementação do programa. O capítulo divide-se do seguinte modo:

- a secção 3.2, **Dependências**, explica dependências do programa;
- a secção 3.3, **Detalhes de Implementação**, introduz uma breve descrição das funções do programa;

3.2 Dependências

O programa apresenta algumas dependências como os *import´s* utilizados:

- `import java.io.*` - para guardar os ficheiros, fazer os *try catch ect* e exceções;
- `import myinputs.Ler` - usado para ler *strings/chars/int* do utilizador;
- `import java.util.*` - usado para os *arraylist* que usamos muito no trabalho;
- `import java.time.*` - usado para fazer a *data ect.*

3.3 Detalhes de Implementação

Começamos pela criação de um objeto "**Pessoa**"e as suas subclasses "**Professor**"e "**Aluno**":

- **Pessoa**- tem como construtor `public Pessoa(int CC, String Nome, String Morada, int phone)`, *getters, setters* e os métodos *toString, clone* e *equals*;
- **Professor**- tem como construtor `public Professor(Pessoa x)`, *getters, setters* e os métodos *toString, clone* e *equals*;
- **Aluno**- tem como construtor `public Aluno(Pessoa b)`, *getters, setters* e os métodos *toString, clone* e *equals*.

Temos também, as classes "**Escola**", "**Curso**","**GerirProfessor**", "**GerirAluno**", "**GerirCursos**"e "**Dia**":

- **Escola**- tem como construtor `public Escola(String NomeEscola, String Morada)`, *getters, setters* e o método *toString*;
- **Curso**- tem como construtores `Curso(int codigo, String nome)` e `Curso(int codigo, String nome, ArrayList<Dia> ListaDias, ArrayList<Professor> ListaP, ArrayList<Aluno> ListaA)`, *getters, setters* e os métodos *toString, clone* e *equals*;
- **GerirProfessor**- contem o seu *menu*, bem como as funções associadas aos professores;
- **GerirAluno**- contem o seu *menu*, bem como as funções associadas aos alunos;
- **GerirCursos**- contem o seu *menu*, bem como as funções associadas aos cursos.
- **Dia**- tem como construtor `Dia(LocalDate Dia, int HoraInicio, int HoraFim)`, *getters, setters* e o método *toString*;

A classe "**GerirEstatisticas**", contem as seguintes estatísticas:

1. NºAlunos que moram numa determinada cidade;
2. Cursos lecionados por mais que um professor;
3. Curso(s) mais frequentados;

4. Curso com mais horas;
5. Professor com mais horas;
6. Aluno inscrito em mais cursos.

Por fim, mas não menos importante, focamos no embelezamento do trabalho através da criação de todo um *menu* esteticamente mais apreciável, para corrigir e prevenir possíveis erros dados pelo utilizador.

3.4 Conclusão

Em suma, referimos todos os objetos, *import´s*, construtores, funções e métodos criados nesta aplicação, bem como as estatísticas fornecidas no trabalho.

Capítulo

4

Testes e Validação

4.1 Introdução

Este capítulo indica todas as Fases, Testes e Aprimoramentos ligados ao desenvolvimento da aplicação, melhorias e alguns detalhes. Está estruturado da seguinte forma:

- a secção 4.2 descreve a **Especificação de Testes** na criação das funções;
- a secção 4.3 fala sobre a **Execução dos Testes**, onde detalhamos alguns erros obtidos e a sua resolução;
- a secção 4.4 refere-se à **Análise dos Resultados e Aprimoramentos**, ou seja, algumas melhorias acrescentadas ao trabalho, incluindo pequenos pormenores em relação à estética.

4.2 Especificação de Testes

O objeto "**Pessoa**", permiti-nos dar uso a uma das aprendizagens obtidas nas aulas teóricas, a herança, usada sobre os objetos "**Professor**"e "**Aluno**". Após criarmos o que seria a "base"de todo o projeto, procedemos à criação da classe "**Curso**"e uma classe temporária "Data", a qual usámos para realizar testes enquanto não chegávamos a uma ideia definitiva acerca do horário. A seguir as classes "**GerirProfessor**", "**GerirAluno**", "**GerirCursos**"e "**GerirEstatisticas**"foram criadas para realizarmos testes a nível da *main*.

No final, criámos a atual classe "Dia"que veio substituir a "Data"e aventuramo-nos nos testes de aprimoramento.

4.3 Execução dos Testes

Realizámos testes em todos os objetos, nos seus argumentos, nos *menus*, criámos novas funções e políticas na gestão da escola. Criámos estatísticas, modificámos construtores, procedemos à realização de um horário e testes no armazenamento dos dados.

4.4 Análise dos Resultados e Aprimoramentos

Observámos que no número de aluno e de professor podíamos usar uma variável *static* que incrementava cada vez que instanciávamos (género contador). Criámos uma função que verifica se os professores estão a lecionar mais de um curso ao mesmo tempo, outra que calcula as horas de um curso, após criação do horário, bem como calcular o professor que deu mais horas.

Quando removemos um professor, e se ele for o único a lecionar um curso, pergunta se queremos adicionar um professor ou eliminar o curso e quando removemos um aluno, temos de atualizar a lista do curso. Colocámos os professores e os alunos a serem alterados pelo seu número, comparamos o Cartão de Cidadão (CC) que queremos introduzir com os já existentes.

Fizemos aprimoramentos no horário e colocámos o mesmo por ordem. Quando queremos renovar o horário para um determinado curso, é apagado o *arraylist* que existe e é chamada novamente a função.

Em termos estéticos criamos uma "cadeia"de *menus* e usamos *Case Sensitive*. Finalmente, chegámos a um consenso em relação ao horário e decidimos criar a class "**Dia**", esta permitia aprimorar todas as classes anteriormente criadas, tornando desnecessária a existencia de um objeto "data", pois elabora um *arraylist* dias com as horas das aulas e os dias que decorrem.

4.5 Conclusão

Após esgotarmos todo o tipo de testes, aprimorarmos as suas funcionalidades e corrigirmos os erros, obtivemos o resultado final. Apresentamos um programa funcional e cumpre os objetivos pretendidos, flexível a novas alterações, ou seja, novos pedidos, sempre com o conceito de otimização.

Capítulo

5

Conclusões e Trabalho Futuro

5.1 Conclusões Principais

Em conclusão, com estes trabalho aprendemos bastante e pudemos colocar à prova todas as aprendizagens obtidas nesta UC. Ganhamos uma maior aptidão para trabalhar em equipa, elaborar relatórios técnicos, bem como a agilidade necessária para programar e proceder à sua pesquisa.

Nada disto seria possível sem o conhecimento obtido na UC Programação Orientada a Objetos, as aulas mostraram se objetivas e auto-contidas, o que permitiu criar uma ideia clara sobre o longo trabalho que tinhamos pela frente e como iriamos proceder à sua realização. Felizmente conseguimos resolver, com sucesso, todos os erros que captá-mos, produzimos um programa que faz exatamente o que é pedido, de forma eficaz, contida, evitando erros do utilizador e ajudando o mesmo com avisos, caso se encontre um pouco perdido.

A escrita em LaTeX, permitiu-nos rever as competências adquiridas no primeiro ano, em torno da escrita e embelezamento de Relatórios Técnicos, definida como meta na UC Laboratórios de Programação.Com base no trabalho, aprendemos bastante acerca da linguangem Java e a sua independencia no desenvolvimento de pequenos projetos e *software*.

5.2 Trabalho Futuro

O nosso objetivo, para já, foi cumprir as metas propostas pela Prof. Doutora Paula Prata, mas num possível trabalho futuro, seria interessante explorar uma componente gráfica, de forma a conseguir implementar um trabalho

mais interativo e atraente, através de uma interface ou até mesmo ligar o projeto a uma base de dados.

Tendo em conta os critérios de avaliação propóstos, defendemos que foi uma justa proposta para os 4 valores pretendidos, trabalho acrescido não se justificava por 20% da nota.