Universidade da Beira Interior
Departamento de Informática

Segurança Informática - 2020/2021: *CHALLENGE-ACCEPTED*

Elaborado por:

41059 - Rúben Guilherme
41308 - Rúben Santos
41719 - Gonçalo Domingos
41853 - Paulo Duarte
41872 - Rui Barata

Orientador:

Professor Doutor Pedro Ricardo Morais Inácio

22 de maio de 2021

Conteúdo

Lista de Figuras

Resumo

Este projeto teve como objetivo o desenvolvimento de uma plataforma de desafios criptográficos que permite aos utilizadores publicarem desafios assim como resolvê-los.

Iniciámos o desenvolvimento deste sistema começando por planear a sua arquitetura e o modelo da sua base de dados. Concluído isto, prosseguimos com o desenvolvimento do sistema em JavaFX.

O resultado final é um sistema que tem diversas funcionalidades, nomeadamente: um sistema de login que encripta as passwords dos utilizadores, publicar desafios criptográficos com vários algoritmos diferentes, resolver desafios, uma página com os desafios que o utilizador tentou resolver e ainda um perfil com os exercícios publicados por cada utilizador.

Acrónimos

CSS *Cascade Style Sheets*

SQL *Structured Query Language*

HTML5 *HyperText Markup Language 5*

PHP *Hypertext Preprocessor*

UI *User Interface*

MD5 *Message-Digest* 5

PK *Primary Key*

FK *Foreign Key*

IDE *Integrated Development Environment*

JDK *Java Development Kit*

Capítulo

1

Introdução

1.1 Enquadramento

Este documento serve para explicar extensivamente do que se trata o nosso projeto e como este foi implementado.

Este projeto foi desenvolvido no âmbito da unidade curricular de Segurança Informática da licenciatura em Engenharia Informática lecionada pelo professor Pedro Inácio.

O tema de projeto que nos foi dado para desenvolvermos foi *CHALLENGE-ACCEPTED*: Um Sistema de Publicação de Desafios Criptográficos com o intuito de demonstrarmos principalmente o que foi dado, na primeira parte do plano curricular de Segurança Informática, em Criptografia.

O desenvolvimento deste projeto é bastante importante devido à temática que este aborda, uma vez que a Criptografia é um pilar importantíssimo no âmbito da proteção da informação.x

1.2 Objetivos

O objetivo deste projeto é a implementação de um sistema de publicação de desafios criptográficos que permita que diferentes utilizadores publiquem e resolvam desafios. Estes desafios podem ser mensagens cifradas com diferentes cifras ou encontrar determinados valores de hash.

1.3 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** – apresenta o projeto, o enquadramento para o mesmo, os seus objetivos, a abordagem ao projeto e a respetiva organização do documento.

2. O segundo capítulo – **Estado da Arte** – apresenta o estado da tecnologia e os projetos relacionados com este.

3. O terceiro capítulo – **Tecnologias Utilizadas** – descreve as tecnologias e as ferramentas utilizadas pelos programadores na implementação do projeto.

4. O quarto capítulo – **Engenharia de Software** – retrata os requisitos funcionais e não funcionais da plataforma, mostrando os diagramas de casos de uso e de atividade e ainda o modelo relacional da base de dados.

5. O quinto capítulo – **Testes e Validação** – mostra todos os testes efetuados com o intuito de garantir que todas as funcionalidades da aplicação se encontram disponíveis para o utilizador.

6. O sexto capítulo – **Conclusões e Trabalho Futuro** – apresenta as conclusões tiradas deste projeto, o que faltou fazer e ainda o que seria interessante ter feito.

Capítulo

2

Estado da Arte

2.1 Introdução

Neste capítulo vamos falar de algumas ferramentas parecidas com as deste trabalho. Na secção 2.2 mostramos alguns trabalhos que encontrámos que são parecidos ao nosso e o último foi aquele no qual nos baseámos para o design da interface.

2.2 Trabalhos Relacionados

2.2.1 *CryptoClub*

O projeto *CryptoClub*[1], desenvolvido originalmente pela universidade de *Illinois* e agora mantido pela universidade de *Chicago,* é um website cujo objetivo é desenvolver as capacidades criptográficas e as matemáticas associadas às anteriores. Contém vários desafios e várias ferramentas com cifras clássicas.

2.2.2 *DiscoverCrypt*

No *website Discover Crypt*[2], temos alguns desafios de cifra com vários níveis de dificuldade. O projeto foi criado para os utilizadores que têm interesse em aprender mais sobre cifras e especialmente para os que querer seguir uma carreira ou ter um hobby na área da criptografia. Foi criado por um estudante do liceu de *Portland*.

2.2.3 *LearnOCaml*

A plataforma *LearnOCaml*[3] é uma plataforma educativa para aprender a linguagem de programação *OCaml*. Foi-nos introduzida na Unidade Curricular de Programação Funcional e decidimos nos basear no seu design pois considerámos que era intuitivo para o utilizador.

Figura 2.1: *Design* do *LearnOcaml*

Capítulo

3

Tecnologias e Ferramentas Utilizadas

3.1 Introdução

Neste capítulo serão mencionadas todas as tecnologias e ferramentas empregues na elaboração da aplicação referente à proposta *Challenge-Accepted*.

3.2 Tecnologias Utilizadas

Destacam-se tecnologias destinadas ao *front-end* da aplicação, tais como, *CSS* e o *JavaFX*, e as restantes, como o *Java* e *SQLite* destinadas ao *backend* da mesma.

JavaFX

JavaFX [4] é uma plataforma de desenvolvimento de software multimédia baseada em *Java* para a criação e disponibilização de *Rich Internet Application* que pode ser executada em diferentes dispositivos.

Java

Java [5] é uma linguagem de programação orientada a objetos lançada pela primeira vez pela *Sun Microsystems*. A linguagem *Java* é compilada para um *bytecode* que é interpretado por uma máquina virtual denominada *Java Virtual Machine*.

SQLite

SQLite [6] é um biblioteca em linguagem C que implementa um mecanismo de base de dados *SQL* mais pequeno, rápido, de alta confiabilidade e com diversos recursos disponíveis.

3.3 Ferramentas Utilizadas

Para auxiliar e facilitar a resolução da presente aplicação foram utilizadas ferramentas como o *NetBeans*, que permitiu desenvolver o software na linguagem Java, o *Scene Builder* que tratou da questão dos *layouts* e o *GitHub*.

NetBeans

NetBeans [7] é um ambiente de desenvolvimento integrado e de código aberto que permite desenvolver software em linguagens como Java, JavaScript, JavaFX, HTML5, PHP entre outras. Também dispõe da possibilidade de conexão com o *GitHub*, o que se revelou muito útil no decorrer do trabalho desenvolvido.

Scene Builder

Scene Builder [8] é uma ferramenta de *layout* visual que permite aos seus utilizadores criar interfaces para as suas aplicações *JavaFX* sem recorrer a codificação. O resultado é um arquivo FXML que pode ser combinado com um projeto *Java*, permitindo assim, ligar as componentes UI à lógica da aplicação.

GitHub

GitHub [9] é uma plataforma de hospedagem de código e arquivos com controlo de versão e colaboração. Foi muito útil na resolução da plataforma permitindo assim trabalhar em simultâneo e remotamente.

3.4 Conclusões

De um modo geral, a grande parte das tecnologias e ferramentas utilizadas no trabalho foram escolhidas devido à ampla compatibilidade que possuem entre si, facilitando, deste modo, todo o processo de desenvolvimento da plataforma.

Capítulo

4

Engenharia de Software

4.1 Introdução

Neste capítulo são retratados os requisitos da plataforma, funcionais ou não funcionais, bem como uma modelação do *software* onde serão representados diagramas de casos de uso e de atividade relevantes no âmbito da aplicação e por último uma representação do modelo relacional da base de dados utilizada.

4.2 Requisitos Funcionais

Os requisitos funcionais correspondem a declarações de serviços que o sistema em questão deve fornecer, como deve reagir a entradas específicas e como este se deve comportar em situações particulares.

Requisito Funcional:

1. O sistema deve interagir com a base de dados para validar credenciais de login, armazenar os dados de registo e a criação e armazenamento de desafios e as estatísticas de cada utilizador.

2. O sistema deve permitir que um utilizador se registe inserindo o seu *username*, o seu email e a respetiva *password*.

3. Após ser efetuado o login, o sistema deve conter uma página inicial com um excerto referente à criptografia, a possibilidade de acerder ao per-

fil do utilizador em questão, criar desafios, consultar desafios e sair da aplicação.

4. O sistema deve permitir que, quando o utilizador seleciona o seu ícone, seja-lhe exibido o seu perfil onde terá a possibilidade de consultar todos os desafios que criou, bem como as suas estatísticas referentes à resolução de outros desafios.

5. O sistema deve distinguir os tipos de desafios existentes.

6. O sistema deve permitir que, quando o tipo de desafio escolhido for "Cifra", apresente os parâmetros título, descrição, mensagem a encriptar, tipo de cifra a utilizar e password ao utilizador, referente ao respetivo desafio a ser criado.

7. O sistema deve permitir que, quando o tipo de desafio escolhido for "Hash", terá de apresentar os parâmetros título, descrição, mensagem a encriptar, tipo de *hash* a utilizar ao utilizador, referente ao respetivo desafio a ser criado.

8. O sistema deve permitir cancelar a criação de um desafio.

9. O sistema deve permitir que o utilizador consulte os desafios criados.

10. O sistema deve exibir os desafios numa tabela interativa, onde constam o seu *ID*, o título e o tipo.

11. Ao clicar num dos desafios, o sistema deve permitir que o utilizador resolva o desafio em questão conforme o seu tipo.

12. O sistema deve permitir que, caso o desafio esteja errado o utilizador tenha a hipótese de o tentar resolver novamente.

13. O sistema deve atualizar a percentagem de acerto do utilizador conforme as suas falhas ou acertos.

14. O sistema deve assegurar que não seja criado um desafio sem que os campos estejam todos preenchidos corretamente.

15. O sistema deve atualizar a percentagem de acerto do utilizador conforme as suas falhas ou acertos.

4.3 Requisitos Não Funcionais

Os Requisitos não funcionais retratam restrições e propriedades dos serviços, bem como, funções oferecidas pelo sistema.

Requisito Não Funcional:

1. O sistema deve exigir autenticação de utilizadores.
2. O sistema deve guardar os dados inseridos pelos utilizadores num sistema de gestão de base de dados, garantindo assim a sua segurança e integridade.
3. Os dados mais importantes referentes aos utilizadores, como as passwords, deverão ser encriptados através do *MD5* de um número aleatório concantenado com a *password* e encriptado com a *SHA-265*.
4. O sistema deve garantir que as *passwords* têm no mínimo seis caracteres, entre os quais estejam presentes no mínimo um número, uma letra minúscula e uma maiúscula.
5. O sistema deve assegurar que o registo de um utilizador não será concluído com sucesso caso o endereço de e-mail inserido já esteja presente na base de dados.
6. O sistema deve responder num curto espaço de tempo a qualquer interação do utilizador.
7. O sistema deve possuir uma interface de utilizador simples, eficaz e de fácil utilização.

4.4 Diagramas de Casos de Uso

Neste setor, serão retratados os casos de uso que ilustram as interações entre o sistema, a base de dados, o seu ambiente e os seus utilizadores.

Casos de uso de um utilizador:

1. **Efetuar registo** - No caso de ainda não estar registado na aplicação, o utilizador terá que se registar para assim usufruir das funcionalidades presentes;
2. **Iniciar sessão** - O utilizador terá que proceder à sua autenticação recorrendo às suas credencias de *login*;
3. **Visualizar excerto sobre Criptografia** - Após iniciar sessão na aplicação, na página inicial o utilizador terá ao seu dispor um excerto relativo a Criptografia;
4. **Consultar perfil** - No menu, o utilizador tem à sua disposição a possibilidade de aceder ao seu perfil;
5. **Consultar estatísticas** - No perfil, o utilizador pode consultar a sua percentagem de acerto baseada na resolução de desafios;
6. **Consultar desafios criados** - Dispõe da possibilidade de consultar os desafios que criou;
7. **Criar desafio** - Cada utilizador tem a oportunidade de criar desafios para os outros utilizadores;
8. **Consultar desafios** - Todos os utilizadores podem consultar os desafios que estão na aplicação;
9. **Resolver desafio** - Ao clicar em determinado desafio, o utilizador tem a hipótese de o resolver;
10. **Sair da aplicação** - Ao clicar no botão "Sair", a aplicação encerra;

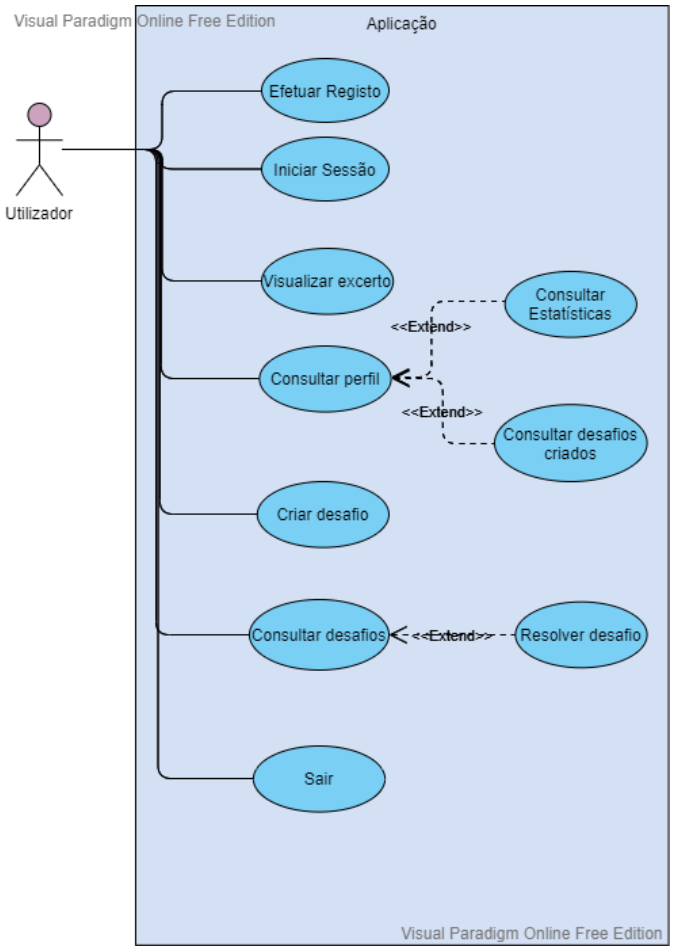

Figura 4.1: Diagrama de casos de uso associado a um utilizador.

4.5 Diagramas de Atividade

Esta secção retrata os diagramas de atividade mais relevantes no contexto da aplicação. Estes descrevem as atividades envolvidas no processo ou processamento de dados. O diagrama de atividade seguinte (Figura 4.2) , representa todos os passos efetuados pelo utilizador para efetuar o seu registo na aplicação.

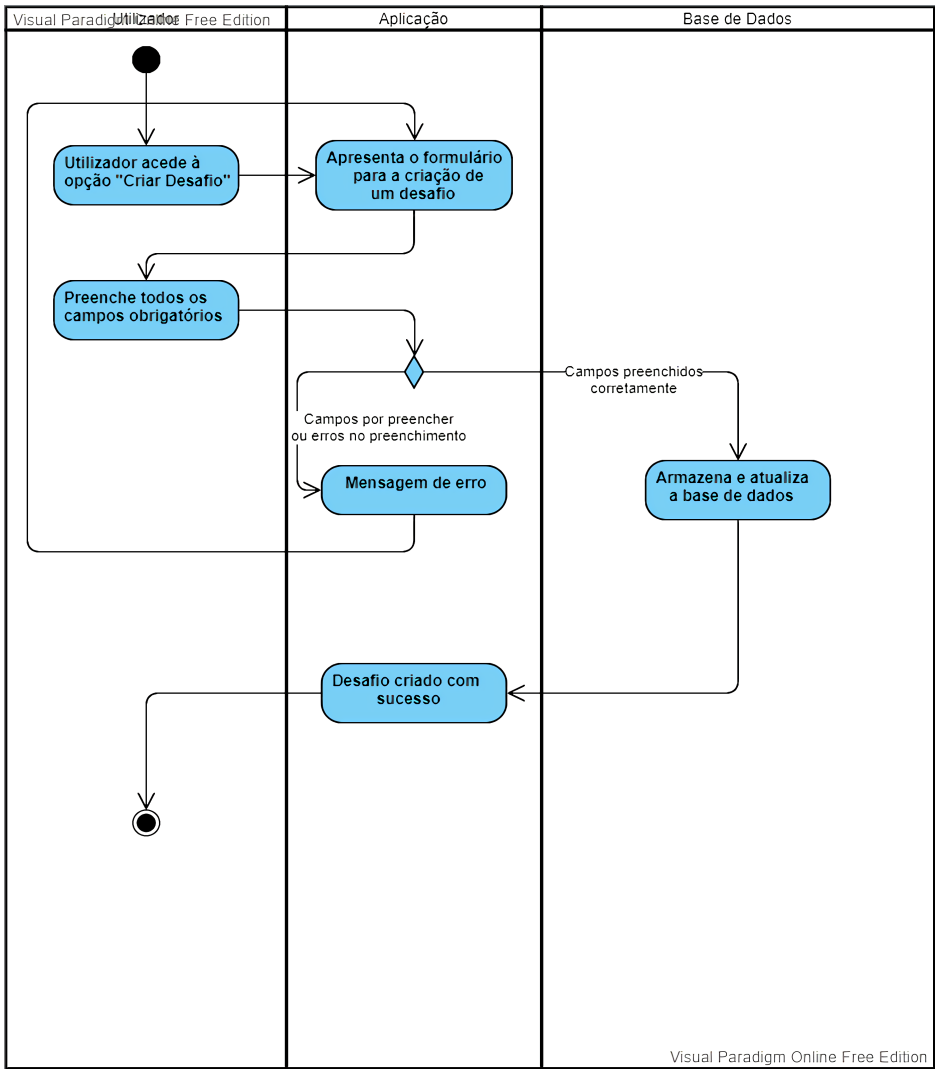

Figura 4.2: Diagrama de atividade que retrata a criação de um desafio.

Caso tenha intenção de resolver desafios, o utilizar necessita de consultar aqueles que se encontram presentes na aplicação e selecionar o que deseja resolver. Após selecionado, será exibido ao utilizador todos os detalhes relevantes para a resolução do mesmo e um campo para responder com a solução que pondere correta.

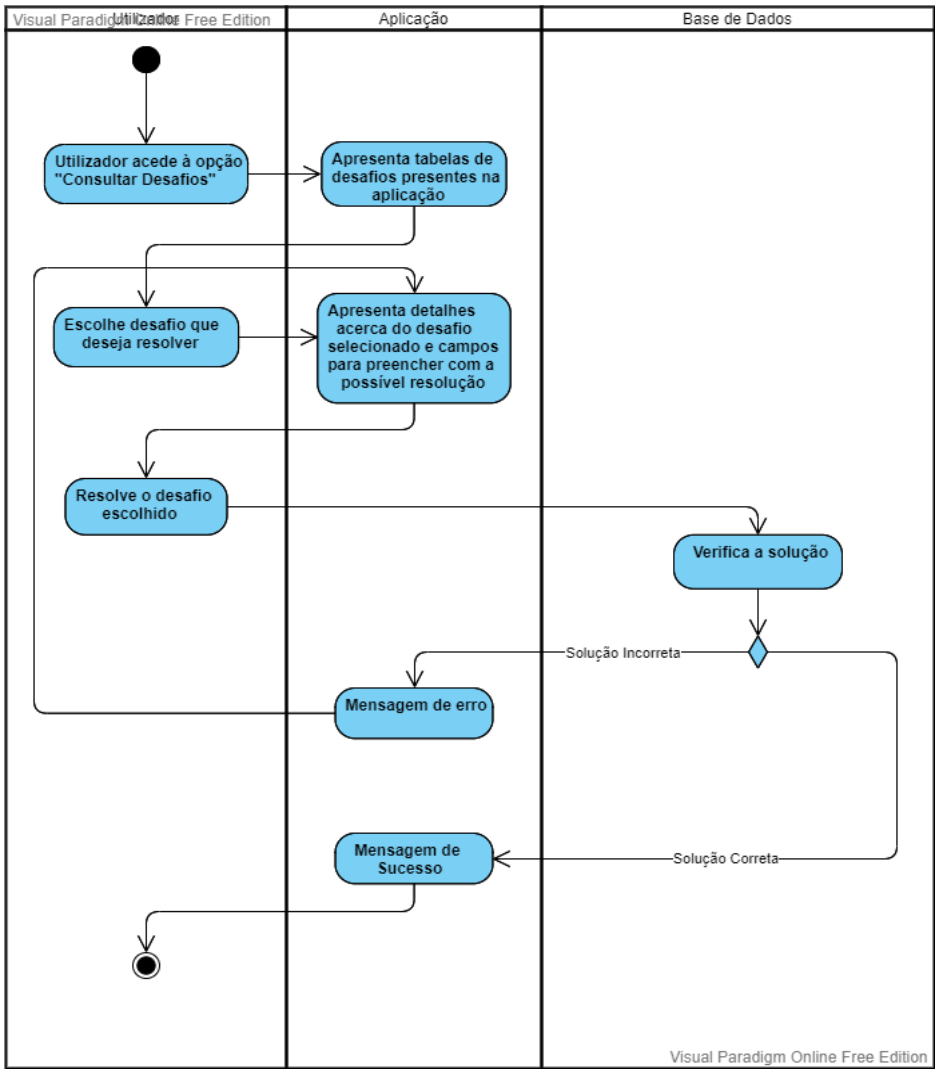

Figura 4.3: Diagrama de atividade que retrata a resolução de um desafio.

4.6 Modelo Relacional da Base de Dados

O modelo relacional contêm um maior nível de detalhe e informação devido a expor todos os atributos das tabelas existentes na base de dados, os respetivos tipos e todas as Primary Key *(PK)* e Foreign Key *(FK)* presentes.

Recorreu-se a três tabelas para corresponder às necessidades da aplicação. A tabela *"Users"* guarda todas as informações dos utilizadores existentes. O atributo *UserID* é a *PK* desta tabela.

Pelo contrário, a tabela *"Exercise"* armazena todos os detalhes referentes

aos desafios que são criados na aplicação. Cada uma contêm um criador, sendo assim esta associação possível devido à *FK UserID*.

De outro modo, a tabela *"Attempts"* regista o número de tentativas que um determinado utilizador necessitou para resolver um certo desafio. Relaciona-se com a tabela de *"Users"* através da *FK UserID* e com a tabela *Exercise* devido à *FK ExerciseID*.

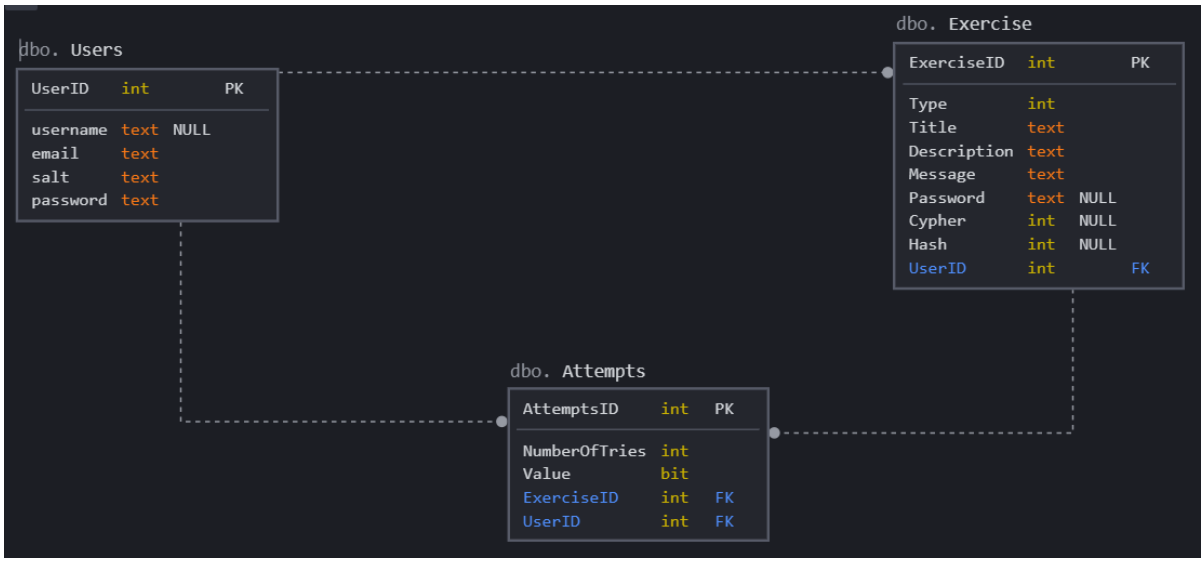

Figura 4.4: Modelo Relacional da base de dados.

4.7 Conclusões

Este capítulo representa todas as fases para alcançar a aplicação desejada no trabalho "*CHALLENGE-ACCEPTED*", de forma a garantir que todas as funcionalidades estão operacionais e que os dados são armazenados de forma coesa, consistente e segura, evitando possíveis falhas no sistema.

Capítulo

5

Execução e Desenvolvimento

5.1 Introdução

Este capítulo retrata todo o processo de desenvolvimento e execução do trabalho. Isto é as depêndencias, os detalhes de implementação e o procedimento de instalação.

5.2 Dependências e Detalhes de Implementação

Dependências são as relações entre as tarefas que é preciso fazer para acabar um projeto. Estas são melhor ilustradas com um diagrama de *Gantt* (5.1).

O projeto foi desenvolvido com a versão 8 do *Java Development Kit* (JDK).

Para iniciarmos o trabalho tivemos que criar um projeto no *GitHub*, criar a base de dados e criar a folha de *CSS* para termos uma aplicação uniforme.

Posto isto, avançámos para o desenvolvimento da aplicação. Aqui nada poderia ser feito sem primeiro termos o ecrã de registo e de inicio de sessão prontos, pois estes dão acesso ao resto da aplicação. Após termos isto feito pudemos fazer a página inicial, que nos permite ir para o feed, criar um desafio e por fim visitar o nosso perfil. Apesar das 3 funcionalidades poderem ser acedidas a partir da página principal o perfil só podia ser feito a partir do criar desafio estar pronto, pois no perfil mostramos todos os desafios criados pelo utilizador.

Finalizando o ecrã de criação de desafios e o ecrã do feed, pudemos acabar o ecrã de resolver um desafio terminado assim a aplicação permitindo fazer o relatório.

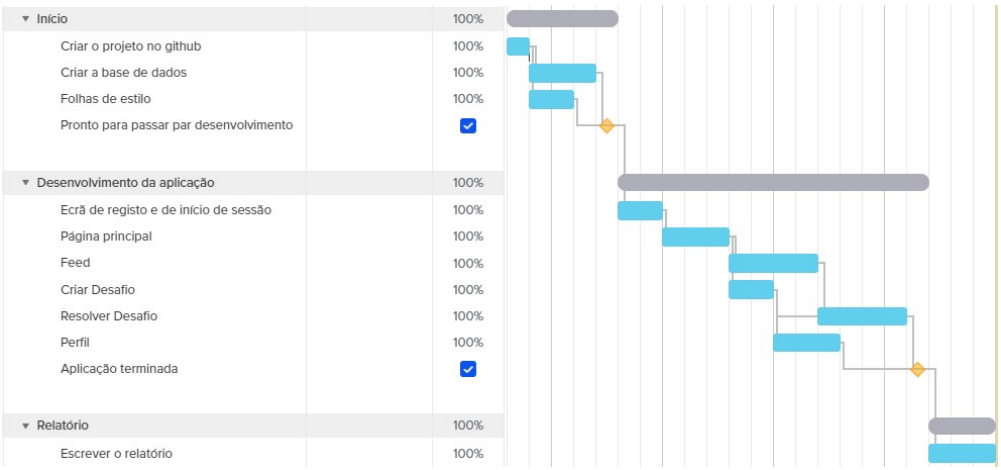

Figura 5.1: Diagrama de *Gantt*

5.3 Procedimento de Instalação

Para escrever e compilar o trabalho decidimos usar o NetBeans. Este permite assim compilar e executar o código dentro do *Integrated Development Environment* (IDE).

5.4 Conclusão

Este capítulo fala de como executar a aplicação e da forma que abordamos todas as fases do trabalho retratadas no capítulo 4, as dependencias do projeto e das tarefas desenvolvidas.

Capítulo

6

Testes e Validação

6.1 Introdução

Neste capitulo irão ser tratados todos os testes realizados no âmbito a garantir que todas as funcionalidades da aplicação se encontrem disponíveis e operacionais, tendo em consideração a visão do utilizador para que este não obtenha uma má experiência.

6.2 Especificação dos Testes

Os testes realizados têm como fim comprometer o funcionamento da aplicação, a integridade dos dados e a experiência do *user*.

1. **Registo de uma conta.** Sendo a primeira interação com a aplicação a criação de uma conta, tratou-se de ser uma das primeiras parcelas a serem realizados testes. O utilizador ao inserir, nos devidos campos, o seu nome, o email e a sua *password*, importante referir que esta seja uma *password* forte pois esta será avaliada,e após clique o botão "Registar"será criada uma conta, caso o utilizador se registe com sucesso, será redirecionado para o *feed* onde encontrará uma página a descrever o propósito da aplicação caso contrário, seja por falta de campos, uso de uma password fraca ou credenciais já existentes, é devolvido um erro auto-explicativo e a conta não será registada.

2. **Iniciar Sessão.** Um utilizador com uma conta registada, após inserir as suas credenciais nos devidos campos, ao clique o botão "Iniciar Sessão"será redirecionado para o *feed* onde encontrará uma página a descrever o propósito da aplicação. No entanto caso utilizador insira uma

conta que não esteja presente na base de dados ou deixe os campos em branco, é lhe devolvido um erro.

3. **Sair da aplicação.** Seja por qualquer motivo, caso utilizador deseje terminar a aplicação, é lhe apresentado um botão "Sair"em todas as instâncias da aplicação onde, ao ser clicado, trará o encerramento da mesma sem que comprometa o funcionamento da base de dados.

4. **Criação de Desafios.** Encontrando-se na página de criação de desafios, o utilizador deverá escolher o tipo de desafio, cifra ou *hash*, preencher os devidos campos e só após clicar no botão "Criar". O utilizador também terá disponível a opção de cancelar, ao clicar no botão "Cancelar", onde será redirecionado para o feed acabando por não criar o desafio em questão. Na presença de algum campo em branco, é lhe devolvido uma mensagem de erro.

5. ***Hash* sem *password*.** Quando selecionado o tipo de desafio *hash* é esperado que a *Text Box* da *password* desapareça.

6. **Consulta de Desafios.** Nesta fase para além de serem efetuados testes para que o utilizador possa resolver o desafio fazendo duplo clique num desafio á sua escolha, também foram realizados testes de forma a que o utilizador não consiga ver os desafios que ele próprio criou ou que já tenha resolvido, ou caso não haja exercícios na base de dados seja então apresentada uma mensagem "Não existem eventos".

7. **Desafios Criados.** Foram realizados testes no perfil com finalidade de ser apresentado apenas os desafios que o utilizador que está logado na aplicação criou. Caso o utilizador não tenha criado nenhum desafio, é então apresentado uma mensagem "Ainda não criou desafios".

8. **Percentagem de Acerto.** Talvez uma das estatísticas mais importantes para um *hardcore user* e por causa disso, é uma das atenções nesta fase de testes. Será sempre iniciado a "0%"e nunca será menor que 0.

9. **Redirecionamento entre atividades.** Uma das partes mais importantes da aplicação, sendo realizados inúmeros testes para que a aplicação não *crash* durante a transição de atividade.

10. **Structured Query Language *(SQL) Injections*.** Sendo este projeto direcionado a uma cadeira de Segurança Informática, testes como *SQL Injections* seriam uma prioridade nesta fase.

6.3 Possíveis Falhas na Segurança

Devido á implementação da aplicação ser baseada em *JavaFX* e á nossa falta de experiência com esta tecnologia, um dos problemas que observamos foi a má gestão de recursos, ou seja, caso o atacante por exemplo, inicie a atividade "Consultar Desafios"e logo de seguida a atividade "Perfil"repetidamente, a aplicação irá consumir mais memória até que eventualmente a esgote e acabe por dar *crash*.

6.4 Conclusão

Em suma, após a realização dos testes descritos nos setores acima, poderá ser afirmado que a aplicação encontra-se operacional, com as funcionalidades solicitadas implementadas.

Capítulo

7

Conclusões e Trabalho Futuro

7.1 Conclusões Principais

Concluímos assim que foram atingidos todos os objetivos do trabalho, visto termos implementados todos os desafios de cifra e de hash que foram sugeridos no enunciado. Conseguimos também implementar um sistema de registo/login com a password a ser cifrada antes de guardar e a funcionalidade extra de adicionar 15 segundos de espera nas tentativas falhadas de resolver um exercício de cifra.

7.2 Trabalho Futuro

No futuro, gostaríamos de implementar o resto das funcionalidades extra que nos foram sugeridas no relatório além de melhorar a interface do utilizador, com mais opções de customização.

Seria interessante implementar também alguns desafios de cifras mais simples, como a de *César* ou a de *Vigenère*, para aqueles utilizadores que não têm ainda bases na área da criptografia, como os projectos enunciados na secção 2.2.

Bibliografia

[1] University of Illinois/University of Chicago. CryptoClub. [Online] https://www.cryptoclub.org/. Último acesso a 16 de Maio de 2021.

[2] Estudante de South Portland Highschool. Discover Crypt, Cryptology Game. [Online] http://highschool.spsd.org/crypt/about.html. Último acesso a 16 de Maio de 2021.

[3] OCaml Software Foundation. LearnOCaml's Github. [Online] https://github.com/ocaml-sf/learn-ocaml. Último acesso a 15 de Maio de 2021.

[4] Oracle. JavaFX: Getting Started with JavaFX. [Online] https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784. Último acesso a 17 de Maio de 2021.

[5] Oracle. O que é o Java? [Online] https://www.java.com/pt-BR/download/help/whatis_java.html. Último acesso a 18 de Maio de 2021.

[6] SQLite. What Is SQLite? [Online] https://www.sqlite.org/index.html. Último acesso a 18 de Maio de 2021.

[7] Apache NetBeans. About Apache NetBeans. [Online] https://netbeans.apache.org/about/index.html. Último acesso a 18 de Maio de 2021.

[8] Oracle. JavaFX Scene Builder. [Online] https://www.oracle.com/java/technologies/javase/javafxscenebuilder-info.html. Último acesso a 18 de Maio de 2021.

[9] GitHub Guides. What is GitHub? [Online] https://guides.github.com/activities/hello-world/. Último acesso a 18 de Maio de 2021.