



OnExam

Relatório Final

Síntese

Aplicação para criar e realizar testes ou exames.

Inês Graça

Nº 2219129, TGPSI – 19J

Índice

1 – Introdução	2
2 – Tecnologias e recursos.....	2
3 – Implementação.....	2
3.1 Cronograma de desenvolvimento do projeto	2
3.2 Aspetos técnicos do desenvolvimento do projeto	3
4 – Conclusão.....	6
5 – Bibliografia	6

1 – Introdução

O meu projeto *OnExam* consiste numa aplicação que permita criar e realizar exames. O objetivo do meu projeto é poder ajudar muitos avaliadores a realizarem provas em formato digital, uma vez que, a meu ver, as opções existentes não têm as ferramentas necessárias e/ou são pagas.

Irei falar das tecnologias utilizadas e dos recursos e ferramentas necessários, do desenvolvimento e implementação do projeto e, por fim, do balanço final do trabalho realizado.

2 – Tecnologias e recursos

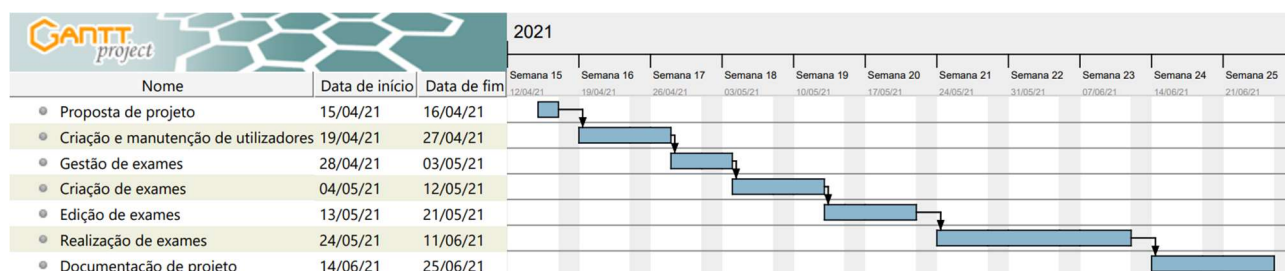
Para o desenvolvimento deste projeto, utilizei a linguagem C#, base de dados relacional em SQL Server e interface do utilizador em Windows Forms.

Relativamente aos recursos e ferramentas necessários, utilizei um computador a correr Windows 10, com Visual Studio 2019, SQL Server Management Studio 18 e gestor de versão com Git.

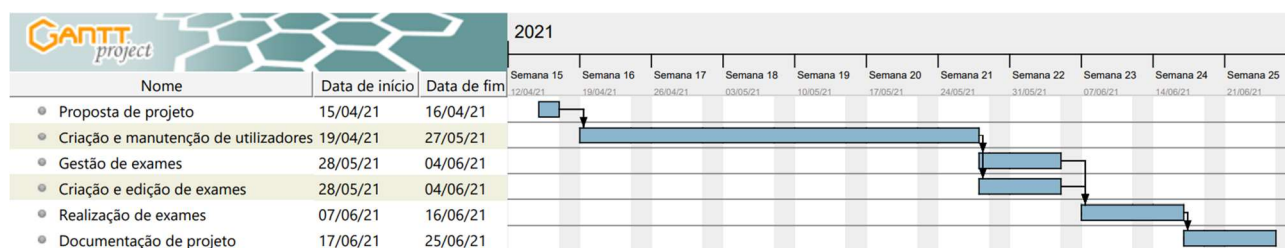
3 – Implementação

3.1 Cronograma de desenvolvimento do projeto

Antes de começar a implementação do projeto, tinha realizado um cronograma de Gantt para planear o desenvolvimento do projeto. No entanto, durante a realização do projeto, o planeamento sofreu algumas alterações.



i. Cronograma de Gantt pré-desenvolvimento



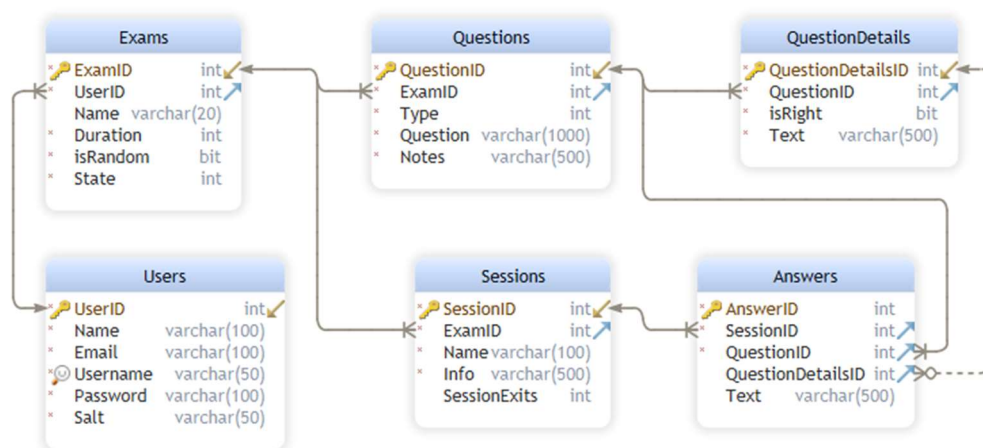
ii. Cronograma de Gantt pós-desenvolvimento

Fases de implementação do projeto:

- Criação e manutenção de utilizadores – esta fase consistiu na criação de menus de registo, login e perfil do utilizador, e o seu funcionamento (também incluindo a parte de base de dados). Originalmente tinha previsto demorar uma semana a tratar desta fase, mas acabou por consumir muito mais tempo devido a dificuldades com o hashing de passwords e por estar a trabalhar maioritariamente em aula, acabando por demorar quase 4 semanas de aulas.
- Gestão de exames – nesta fase está incluído o menu de visualização de exames, e operações possíveis a realizar (criar, abrir e eliminar). Tinha previsto demorar 4 dias nesta fase e, apesar de ter iniciado mais tarde que o previsto, esta fase consistiu em 6 dias.
- Criação e edição de exames – esta fase consistia em adicionar, abrir, editar e eliminar os detalhes de um exame (como o nome, duração e estado) e as suas perguntas. Originalmente tinha previsto realizar a criação e depois a edição de exames (em fases separadas) com um total de 14 dias. Porém, foi mais simples desenvolver estas fases em simultâneo, enquanto desenvolvia também a fase anterior, tendo afinal demorado também 6 dias.
- Realização de exames – nesta fase inclui-se os menus e operações para a entrada num exame para o realizar, responder às perguntas que este possui e ainda a visualização posterior das respostas dadas num exame. Estava previsto demorar 15 dias nesta fase, mas revelou-se mais simples que o esperado e demorou apenas 8 dias.

3.2 Aspetos técnicos do desenvolvimento do projeto

Para este projeto, o modelo de dados utilizado foi o seguinte:



Este modelo não sofreu alterações em termos de relações, apenas ajustes. Em primeiro lugar, as denominações de tabelas e campos foram alteradas para inglês para facilitar a relação com as denominações dadas em código. Para além disso, foram adicionados campos: para o salt da password em *Users*, para o nome e estado em *Exams* e para o número de acessos ao exterior em *Sessions*. Ainda foi eliminado um campo de texto em *QuestionDetails* e um de booleano em *Answers*.

Todos os comandos SQL foram realizados através de stored procedures.

A aplicação inicia-se por um menu inicial onde é possível escolher (através de opções) registar, fazer login e realizar um exame.

Para realizar o registo, são necessárias várias informações do utilizador, como o nome, email, username e password. Ao registar, estas informações são inseridas na tabela *Users*, sendo que é realizado o hashing da password com SHA-256, sendo assim guardado também o salt.

No login, o utilizador pode iniciar sessão com o email ou username (campo unique) e a password. Nesta operação procura-se o utilizador correspondente ao email/username dado e comparada a password dada com a original.

Ao iniciar sessão, o utilizador acede a um menu onde consegue ver e gerir os seus exames. É feita uma consulta à tabela *Exams* por todos os exames do utilizador. Isto corresponde a um requisito: um login que permite a um utilizador ver e gerir as suas avaliações, visíveis apenas para si.

Neste menu é possível adicionar, abrir e eliminar exames, aceder ao seu perfil e sair. No perfil é permitido alterar as informações dadas no registo, sendo atualizadas as informações na tabela *Users* ao guardar alterações.

Ao criar um exame, é inserido um novo registo na tabela *Exams* e é aberto um menu em modo MDI. Quando se abre um exame já existente, é realizada uma consulta para conseguir os detalhes do exame (da tabela *Exams*) e questões (da tabela *Questions*), sendo depois aberto o mesmo menu em modo MDI, com os detalhes do exame e janelas correspondentes às questões.

É possível adicionar, editar e eliminar questões de um exame, podendo estas ser de seleção de opção/opções ou de resposta escrita. As perguntas de escolha necessitam ainda de um registo na tabela *QuestionDetails* por cada opção, para guardar se é a opção correta e o texto da opção.

Todas as alterações a um exame são guardadas ao fechar o menu. Estas funcionalidades correspondem assim a outros 2 requisitos: conseguir criar um exame que permita várias questões, e que estas possam ser de vários estilos.

No menu do exame é ainda possível editar os detalhes, como a duração e se as questões surgem numa ordem aleatória. Para além disto, o nome está predefinido para o ID do exame, sendo possível alterá-lo (iniciado sempre pelo username, para não existirem erros por vários utilizadores terem exames com o mesmo nome).

Para um exame poder ser realizado, necessita de estar ativo. O estado de um exame é inicialmente colocado como inativo, sendo possível ainda neste menu ativar o exame e posteriormente fechá-lo (o que não afeta exames que estejam a decorrer). Após o fecho do exame, é possível aceder às sessões realizadas (através de uma consulta à tabela *Sessions*), e às respostas dadas em cada sessão (vindas da tabela *Answers*). Isto corresponde a mais um requisito: apenas ser possível realizar um exame ativo, e conseguir visualizar as respostas depois de fechado.

Voltando ao menu principal, para realizar um exame são precisas algumas informações: o nome do exame (que tanto pode ser apenas o ID como o nome), o nome do que irá realizar o exame e informação extra que o avaliador tenha solicitado. Esta parte corresponde a mais um requisito pretendido: permitir que o exame a realizar seja indicado pelo ID ou pelo nome dado pelo utilizador.

Caso o exame seja encontrado e esteja ativo, é inserido um novo registo na tabela *Sessions*. O exame e questões são abertos num menu em modo MDI, em ecrã inteiro.

Ao iniciar o exame, o temporizador inicia a duração do exame e é permitida a resposta às questões. Se alguma aplicação exterior for acedida, é registado que se saiu da aplicação, aparecendo uma mensagem que não permite continuar a resposta às questões até ser fechada.

Apesar de não evitar o acesso a outras aplicações, que corresponderia da forma mais ideal a outro requisito (não conseguir aceder a nenhuma aplicação exterior), a única solução encontrada foi dar a informação de quantas vezes outras aplicações foram acedidas, podendo assim conseguir transmitir o quão segura e de confiança foi a realização do exame por essa sessão.

Quando o exame é terminado (por saída voluntária ou pelo fim da duração permitida), as respostas às questões são inseridas na tabela *Answers* e, se outras aplicações foram acedidas, o número de vezes de acesso ao exterior é atualizado na tabela *Sessions*.

Por último, a aplicação adapta-se ao idioma no Windows, sendo que poderá surgir em inglês ou português consoante a situação. Isto correspondia ao último requisito: permitir alteração de cores da aplicação e permitir alteração de idioma. Não existe, porém, opção para alterar as cores, uma vez que esta funcionalidade não era essencial, sendo mais pertinente a aplicação estar funcional.

4 – Conclusão

Eu considero que no geral o projeto tenha corrido bem. No entanto, gostaria de ter gerido o tempo durante a implementação do projeto de melhor forma.

Quando, logo no início, tive alguns problemas com o hashing de passwords, deveria ter despendido algum tempo fora de aula para tentar resolver os erros que ocorriam, ao invés de acabar por despendendo tanto tempo útil na parte de gestão dos utilizadores.

Apesar desta situação não ter sido nem de perto ideal, aprendi por experiência que problemas deste tipo devem ser deixados em segundo plano se não forem cruciais para o funcionamento esperado da aplicação. O hashing de passwords não fazia parte dos requisitos indicados, e por esta razão deveria ter ficado para o fim, quando as outras funcionalidades essenciais estivessem asseguradas.

No entanto, consegui recuperar o tempo perdido trabalhando mais em casa, e também tirando partido dos feriados existentes, e a aplicação ficou a funcionar como queria e como planeei.

Gostaria apenas de ter terminado o último requisito, relativamente à opção de alteração das cores da aplicação. Contudo, desde o princípio, tinha noção que daria muito trabalho e que ficaria para o fim caso existisse tempo, pois o projeto estar a funcionar devidamente é muito mais relevante do que um pequeno pormenor relativo a design.

Estou a considerar fortemente continuar com esta ideia para a Prova de Aptidão Profissional e assim aproveitar o trabalho já realizado, não tendo, no entanto, ainda planeado o desenvolvimento futuro desta ideia.

5 – Bibliografia

Estes foram os links que precisei de consultar:

- [C# - Check if application lost focus to a different application - Stack Overflow](#)
- [C# - How to hide only the Close \(x\) button? - Stack Overflow](#)
- [Compute SHA256 Hash In C# - C# Corner](#)
- [Timer Class \(System.Windows.Forms\) | Microsoft Docs](#)
- [Walkthrough: Localizing Windows Forms | Microsoft Docs](#)