

Departamento de Engenharia de Eletrónica e Telecomunicações e  
de Computadores  
Licenciatura em Engenharia Informática e de Computadores

**Trabalho Prático**

Programação em Dispositivos MóveisSemestre de Inverno 2018/2019Docente: Paulo Pereira

Trabalho realizado pelo grupo LI52D-03:

Bárbara Pires nº43105

Francisco Farinha nº 42153

José Pereira nº 42934

<Dezembro de 2018>

**Índice**

[1. Introdução 1](#_Toc529830842)

[1.1 Funcionalidades para o futuro 1](#_Toc529830843)

[2. Formulação do Problema 3](#_Toc529830844)

[2.1 Requisitos e funcionalidades 3](#_Toc529830845)

[2 Solução Proposta 6](#_Toc529830848)

[3 Conclusões 17](#_Toc529830857)

# Introdução

O objetivo deste trabalho é desenvolver um sistema de informação que permita gerir e guardar toda a informação associada ao processo de submissão e revisão de artigos científicos de múltiplas conferências realizados por investigadores.

Começou-se por desenvolver o modelo EA (Entidade Associação). Visto que este corresponde à estrutura conceptual na nossa base de dados, foi dada uma especial importância a que este fosse estável e compreensível. Isto levou a que sofresse alterações ao longo do tempo à medida que se discutiam certas funcionalidades e regras de negócio para que se comporte como requisitado.

Após essa fase inicial de estruturação da base de dados foram desenvolvidos mecanismos de interação com a mesma que asseguram a sua consistência e providenciam as funcionalidades requisitadas.

Finalmente foram desenvolvidos testes para garantir que estes mecanismos não só funcionavam de forma correta nesse momento, mas que se em algum momento fossem alterados seria fácil verificar a sua funcionalidade.

## Funcionalidades para o futuro

Em relação a alguns pedidos ou funcionalidades mencionadas no enunciado que não foram implementadas, pois estão fora do âmbito da unidade curricular Sistemas de Informação 2 e do que é pretendido desenvolver, gostávamos de falar brevemente sobre possíveis soluções.

A primeira funcionalidade está descrita no enunciado assim: “Inicialmente as conferências convidam os investigadores a submeterem artigos científicos”. Isto poderia ser implementado com um sistema de e-mail que expõe um serviço de envio de e-mails. Os administradores do sistema de informação desenvolvido para gerir conferências, especificavam o e-mail do destinatário e o sistema de e-mail tratava de enviar o e-mail, com a informação do convite mais um botão que ao ser clicado redirecionava o destinatário para uma página de registo no sistema.

A segunda funcionalidade seria o envio de e-mails informativos sobre o estado da submissão aos autores que são designados como responsáveis. Uma possível implementação seria na aplicação a ser desenvolvida na segunda parte do trabalho em C#, sempre que o estado de uma submissão seja alterado, é lançado um evento com informação sobre o artigo em questão, e consequentemente era executado qualquer função *hander* registada como *listener* no evento, que iria obter o e-mail do responsáveis do artigo e com o uso auxiliar do sistema de e-mail previamente referido, enviava o tal e-mail informativo.

# Formulação do Problema

O primeiro passo no desenvolvimento de software é a análise de requisitos, onde examinamos o enunciado do problema mais profundamente de forma a conseguirmos retirar requisitos funcionais e, possivelmente, não funcionais do sistema que vamos desenvolver.

A secção 2.1 trata-se de uma enumeração aos requisitos e funcionalidades esperados deste projeto. Na secção 2.2 será descrito o problema. Na secção 2.3 será analisado o problema, como por exemplo as regras de negócio.

## 2.1 Requisitos e funcionalidades

Além do modelo de dados (conceptual e relacional), incluindo todas as restrições de integridade, os requisitos pedidos pelo enunciado são:

(a) Criar o modelo físico;

(b) Remover o modelo físico;

(c) Preenchimento inicial da Base de dados;

(d) Inserir, remover e atualizar informação de um utilizador;

(e) Inserir, remover e atualizar informação de uma submissão;

(f) Atualizar uma conferência;

(g) Atribuir um papel a um utilizador registado;

(h) Listar os revisores compatíveis com uma revisão;

(i) Atribuir um revisor a uma revisão;

(j) Registar uma revisão;

(k) Calcular a percentagem de submissões aceites;

(l) Alterar o estado de todas as submissões, considerando um limite de corte

passado como parâmetro ou, em caso de ser NULL, considerando o valor registado no

sistema;

(m) Testar cada um dos requisitos anteriores.

Para além disto ainda foram implementados:

* Calcular a média de um artigo
* Verificar se um autor é o único autor de qualquer artigo
* Verificar se um autor é o único autor responsável de qualquer artigo
* Submeter/atualizar o ficheiro associado à submissão

## 2.2 Descrição do problema

O problema que apresentado é o desenvolvimento de um sistema de informação, que permita gerir o processo de submissão e revisão de artigos científicos realizados por investigadores numa conferência, como foi referido na introdução. Neste contexto, uma conferência é caracterizada pelo seu acrónimo, nome e ano de realização. Um artigo científico pode ter vários autores e um autor pode submeter diversos artigos científicos na mesma conferência. O processo de submissão de um artigo armazena um resumo do artigo, a data de submissão, um identificador dentro da conferência, um estado (“em revisão”, “aceite”, “rejeitado”) e o respetivo ficheiro do artigo científico que é obrigatório existir. Cada ficheiro submetido é guardado pelo sistema, dando a possibilidade de ver o histórico dos diversos ficheiros associados a uma submissão de um artigo. Para cada submissão fica registado pelo menos um autor (podendo ser mais) que se afirma como um dos responsáveis, quer isto dizer que pretende receber os e-mails informativos sobre o estado da submissão.

Para cada artigo submetido são atribuídos um ou mais revisores. Um revisor pode rever um ou mais artigos, seguindo a regra de que o artigo não pode ter sido submetido por autores da mesma instituição que o revisor. Após finalizada a revisão de um artigo, o revisor atribui uma nota de 0 a 100 ao artigo juntamente com um texto da revisão. A revisão dos artigos também está limitada por uma data estabelecida pela conferência que tem de ser posterior à data de submissão de artigos. Um artigo/submissão pode ser aceite ou não (isto é rejeitado). Para que um artigo seja aceite é necessário que a média das notas de todos os revisores seja  
superior ou igual à nota mínima estabelecida pela conferência para efeitos de aceitação dos  
artigos.

Um utilizador pode ser simultaneamente autor e revisor ou só autor ou só revisor. Cada utilizador é identificado pelo seu email. Para além disso, para cada utilizador é necessário guardar a informação do seu nome assim como a informação sobre a sua instituição. Para cada instituição é necessário guardar o nome da instituição, morada, país entre outras informações. Um utilizador pode ainda ter o papel de presidente da conferência, não podendo rever artigos que entrem em conflito com as regras anteriormente definidas. Deve ficar registada a data de registo de um utilizador na conferência, independentemente do papel inicial desse registo (autor, revisor ou presidente da conferência).

## 2.3 Análise do problema e funcionalidades

As regras de negócio são asserções que definem ou restringem algum aspeto do negócio em questão, garantidas por código. As **regras de negócio** extraídas da descrição do problema são:

* Não podem existir duas conferências com o mesmo nome nem com o mesmo nome no mesmo ano.
* A submissão de artigos científicos pode ser realizada até uma data limite estabelecida pela conferência. Os autores apenas podem atualizar o ficheiro submetido até esta data.
* Para que um artigo seja aceite é necessário que a média das notas de todos os revisores seja superior ou igual à nota mínima estabelecida pela conferência.
* A nota mínima de aceitação de artigos é definida dinamicamente, ficando registada posteriormente.
* Os autores e revisores têm que se registar na conferência
* Um revisor não pode rever artigos que tenham sido submetidos por autores da mesma instituição, incluindo os artigos da sua autoria.
* Após finalizada a revisão de um artigo, o revisor atribui uma nota de 0 a 100 ao artigo juntamente com um texto da revisão
* A revisão dos artigos tem que ser realizada até uma data limite estabelecida pela conferência (posterior à data de submissão de artigos)
* Um artigo tem de ter pelo menos um autor responsável, para quem devem ser enviados os e-mails informativos sobre o estado da submissão

# Solução Proposta

Neste capítulo é apresentado o modelo conceptual e as restrições de integridade, modelo relacional e o modelo físico, em conjunto com a explicação mais detalhada de cada funcionalidade pedida.

3.1 – Decisões de compromisso

Para darmos suporte à funcionalidade: Instant Messaging pedida no enunciado escolhemos usar o serviço/tecnologia fornecida pelo Firebase, produto da Google de *cloud computing*, chamado de Cloud Firestore. Devido a esta escolha estamos comprometidos à infra-estrutura que nos é fornecida por este serviço.

Porém, é graças às funcionalidades deste serviço que o código que temos de desenvolver, particularmente para enviar e receber mensagens, e acesso offline às mensagens, é mais simples do que se usássemos outra tecnologia como o Realtime Database ou Cloud Messaging, que também são serviços disponíveis pelo Firebase, mas para o uso que nós queremos implica fazermos mais código porque temos mais responsabilidades.

**FirebaseDatabase**

Esta entidade foi criada com o propósito de tratar de todo o código de acesso ao FirebaseFirestore. Isto para que esta responsabilidade esteja localizada num sítio e não haja dependências espalhadas pelo código (controlo de depências)

**Room**

É nos pedido no enunciado que tenhamos suporte para o modo offline, e portanto precisamos de armazenar os dados vindos da API Github de forma persistente, e a sua atualidade tem de ser garantida diariamente. Este último requisito foi implementado recorrendo ao WorkManager (ter um link para a page que explica a utilização do WorkManager na wiki).

O Room é uma tecnologia ORM (Object Relational Mapping) que nos facilita o código de acesso a dados local (base de dados SQLite disponível nos dispositivos android), garantindo também que existe um mapeamento das tabelas da base de dados para os objetos em memória.

**WorkManager**

Esta

**Arquitetura da solução e preocupações de desenho**

Decidimos estruturar o código por “componentes”, em que alguns destes componentes são simplesmente os próprios componentes fornecidos pelo Android como por exemplo Activity ou ViewModel e WorkManager que fazem parte do Android Jetpack. Baseamos nos no artigo disponível no site Android Developers: Guide to App Arquitecture, e por isso representamos cada componente num package, demonstrado na figura a seguir:……

3.2 – Navegabilidade da aplicação

Numa primeira fase de desenvolvimento, consideramos usar um Navigation drawer que permitiria ao utilizador navegar para qualquer destino na nossa aplicação. Porém apercebemos nos mais tarde que o uso que lhe estávamos a dar não era o recomendado, pois não tínhamos uma aplicação com cinco ou mais destinos *top-level*.

Acabamos por escolher usar simplesmente a *app bar* também conhecida como *action bar*. A classe Toolbar implementa a nossa app bar e garante que a aplicação tem um comportameto consistente em vários dispositivos.

# Conclusões

Por falta de tempo não foi possível melhorar alguns aspetos em relação ao código produzido, em particular *error handling* que poderíamos ter usado o *throw* em vez do *raiserror*,já que este vai ficar *deprecated* e até é preferível usar o *throw* dentro do *catch clause*, como também seria possível dar melhor uso ao parâmetro *state* do *throw* ou *raiserror* para distinguir secções onde o erro pode ter ocorrido, facilitando depois a procura do erro e também dar mais contexto e informação a quem recebeu o erro. Outra possibilidade era ter um *test suite* mais completo que testava restrições das tabelas como apresentado nos slides das aulas

# Referências

|  |  |
| --- | --- |
| [1] | 10.Digit; How to Store Documents,Videos, music to SQL Server database; Disponível na WWW: URL:<https://www.youtube.com/watch?v=RxUrUK4O0vc> |
| [2] | [Database Testing Guide (Why, How, and What About Database Testing)](https://www.softwaretestinghelp.com/database-testing-process/)  <https://www.softwaretestinghelp.com/database-testing-process/>. |