

# MARCO AVALIATIVO 1

COMPUTAÇÃO PARA DISPOSITIVOS MÓVEIS

## ALUNOS

- Salatiel Costa Bairros (201611865);
- Rhuan Dornelles (201521884).

## AGENDA TELEFÔNICA

Software de agenda segundo os requisitos descritos no PDF do exercício. Além do código enviado por e-mail, fonte está disponível em um repositório público do GitHub, disponível no link [https://github.com/SalatielBairros/Agenda\\_MA1](https://github.com/SalatielBairros/Agenda_MA1).

## ACTIVITIES

Segue abaixo a descrição de cada uma das Activities implementadas.

### Alterar

Responsável por alterar ou deletar um registro de contato cadastrado.

Além dos métodos e eventos básicos para realizar a manipulação dos dados, foi implementada nesta Activity o método *callNumber*, ainda que o mesmo não esteja sendo usado nela. O propósito disso era testar a chamada de um método implementado em uma Activity através de outra e também deixar em aberto uma futura implementação da realização de chamadas por dentro do cadastro do contato.

### ListaContatos

Exibe a listagem de todos os contatos cadastrados.

### Main

Responsável pelo cadastro de novos contatos.

### Preferences

Realiza o cadastro e alteração das preferências de usuário.

## CLASSES E INTERFACE ADICIONAIS

Para o desenvolvimento deste projeto foram criadas classes para auxiliar na estruturação do mesmo, buscando reaproveitar código e organizar os dados utilizados.

### BaseController

Classe com as quatro operações básicas de banco de dados: inserir, alterar, ler e excluir. Seus métodos esperam como parâmetro alguma implementação da interface *IEntityModel*, contendo as informações a serem manipuladas.

### AgendaContext

Contexto do banco de dados do aplicativo. Possui os métodos para iniciar e atualizar o SQLite e informações sobre o banco criado.

### IEntityModel

Interface que contém os métodos básicos para classes do tipo *Model*, ou seja, classes que representam entidades do banco de dados.

### BaseModel

Classe abstrata contendo as informações relacionadas ao *id* da entidade do banco de dados. Implementa a interface *IEntityModel*.

### ContatoModel

Classe que representa a tabela “*contato*” do banco de dados. Herda da classe *BaseModel* e implementa os métodos da interface *IEntityModel*.

### UserPreferencesViewModel

Semelhante a classe *ContatoModel*, porém não realiza a implementação de métodos vindos da *IEntityModel* nem herda da *ContatoModel*, visto que não representa uma entidade do banco de dados, apenas o XML de preferências de usuário. Devido a isso, possui o sufixo *ViewModel*.

## DIFICULDADES ENCONTRADAS

A principal dificuldade encontrada para o desenvolvimento foi a configuração do ambiente, visto que qualquer coisa diferente faz com que o Gradle não consiga atualizar os pacotes instalados e a solução mais rápida para um projeto pequeno é criar um outro e copiar os arquivos de código.

Uma segunda dificuldade, consideravelmente menor, encontrada foi o cuidado necessário com contexto entre as Activities, pois facilmente o aplicativo trava, caso o contexto não esteja correto ao se chamar métodos de uma Activity em outra.

## POSSÍVEIS MELHORIAS

Em adendo ao que foi solicitado nesta documentação, cremos ser interessante também listar algumas possíveis melhorias neste projeto e em sua estrutura para um desenvolvimento futuro.

1. Separar as implementações em diferentes módulos. Por exemplo, utilizar um módulo apenas para regras de negócio e outro apenas para acesso ao banco;
2. Utilizar um ORM;
3. Maior uso de interfaces, especialmente caso novas tabelas surgissem no projeto, permitindo a generalização de métodos e reaproveitamento de código.
4. Utilização de componentes e estruturas de layout mais modernos fornecidos pela biblioteca do Android.