

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

Projeto Final (Pg)
SCC541 - Laboratório de Bases de Dados

Caio Cesar Aguiar Alarcon - 7241109
Victor Luiz de França - 9790781

Prof. Dr. Caetano Traina Jr.
Igor Alberte R. Eleutério

São Carlos
6 de Julho de 2022

1. Introdução

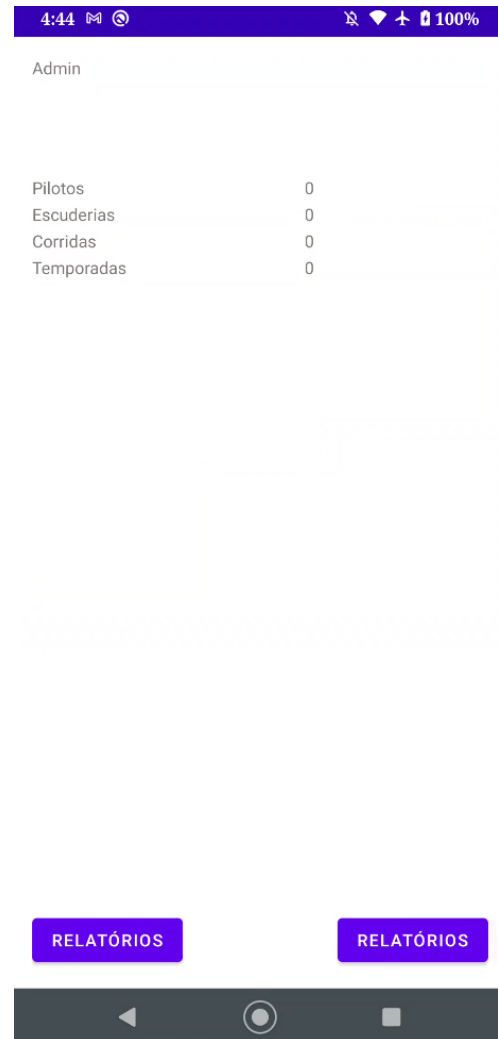
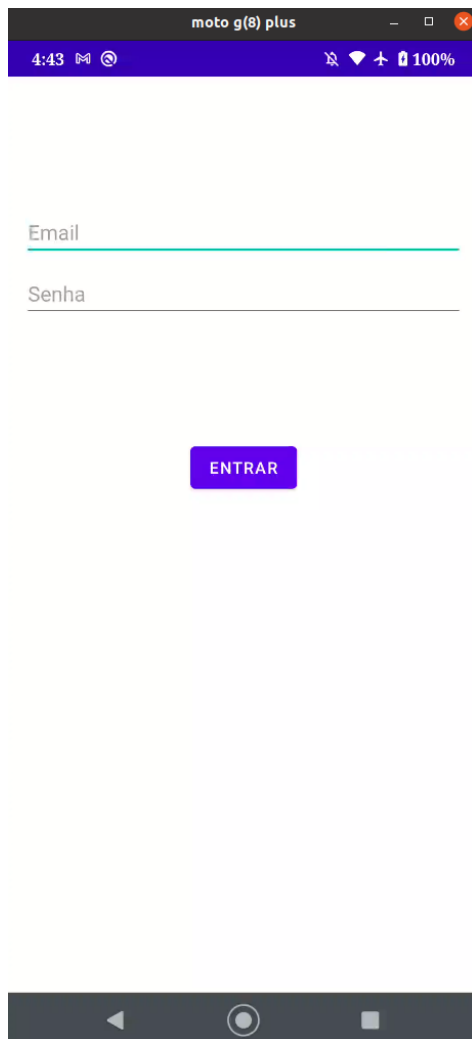
O objetivo deste projeto final é, usando os conhecimentos adquiridos, produzir um projeto funcional, contendo três telas, uma de login, uma tela de overview que apresente informações referentes aos respectivos usuários logados e, ainda, uma tela de relatórios contendo informações úteis para cada tipo de usuário.

2. Desafios encontrados

A ideia original do desenvolvimento da aplicação era criar um aplicativo Android usando a linguagem de programação Kotlin. Porém, as configurações de segurança do Android dificultam ao máximo o uso de acesso direto ao banco de dados por meio da rede. Neste sentido, o mais indicado é usar uma API como intermediária entre o banco de dados e a aplicação. Houve dificuldade também para criar alguns índices para consultas. Além disso, a tabela de usuários não foi preenchida com todos os usuários, apenas com o administrador para testes, mas não foi possível testar usando o spring boot, apenas testamos usando o pgAdmin.

3. Passo a passo do que foi feito

Primeiramente as telas foram criadas para login e overview. A tela de relatórios seria criada só depois de obtermos os dados a serem exibidos.



Telas de login e overview com acesso a tela de relatórios

Algumas funções foram criadas usando o pgAdmin para a tela de overview. Além disso, foi criada a tabela de usuários. Então foi criado uma classe seguindo o padrão de projeto Repositório para acessar o banco de dados por meio de uma api do java chamada JDBC, porém esta api nunca funcionou:

```
bose | Q_ | [x] Regex | S
formula: Accessing hidden method Landroid/view/View;->computeFitSystemWindows(Landroid/graphics/Rect;Landroid/graphics/Rect;)Z (greylist, reflection, allowed)
formula: Accessing hidden method Landroid/view/ViewGroup;->makeOptionalFitsSystemWindows()V (greylist, reflection, allowed)
org.postgresql.util.PSQLException: Something unusual has occurred to cause the driver to fail. Please report this exception.
at org.postgresql.Driver.connect(Driver.java:277)
at java.sql.DriverManager.getConnection(DriverManager.java:580)
at java.sql.DriverManager.getConnection(DriverManager.java:218)
at com.example.formula1.PostgresConnection.<init>(PostgresConnection.kt:14)
at com.example.formula1.data.LoginDataSource.<init>(LoginDataSource.kt:11)
at com.example.formula1.ui.login.LoginViewModelFactory.create(LoginViewModelFactory.kt:19)
at androidx.lifecycle.ViewModelProvider.get(ViewModelProvider.kt:171)
at androidx.lifecycle.ViewModelProvider.get(ViewModelProvider.kt:139)
at com.example.formula1.ui.login.LoginActivity.onCreate(LoginActivity.kt:38)
at android.app.Activity.performCreate(Activity.java:7822)
at android.app.Activity.performCreate(Activity.java:7811)
at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1328)
at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:3430)
at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:3598)
at android.app.servertransaction.LaunchActivityItem.execute(LaunchActivityItem.java:83)
at android.app.servertransaction.TransactionExecutor.executeCallbacks(TransactionExecutor.java:135)
at android.app.servertransaction.TransactionExecutor.execute(TransactionExecutor.java:95)
at android.app.ActivityThread$H.handleMessage(ActivityThread.java:2164)
at android.os.Handler.dispatchMessage(Handler.java:107)
at android.os.Looper.loop(Looper.java:241)
at android.app.ActivityThread.main(ActivityThread.java:7582) <1 internal line>
```

O erro obtido ao tentar usar o jdbc

Depois de alguns dias tentando resolvemos desistir e usar uma API externa feita com o Spring boot. Usando esta tecnologia foi possível conectar ao banco de dados disponibilizado no localhost, porém não tivemos tempo de criar os endpoints relativos às consultas SQL. O problema dessa vez foi criar as classes para receber os valores correspondentes às consultas.

```
Actor
springframework.beans.factory.BeanCreationException: Error creating bean with name 'userRepository' defined in com.bd.f1.repository.UserRepository defined in @EnableJpaRepositories declared on F1Application:
at org.springframework.beans.factory.support.AbstractAutowiredCapableBeanFactory.initializeBean(AbstractAutowiredCapableBeanFactory.java:186) ~[spring-beans-5.3.21.jar:5.3.21]
at org.springframework.beans.factory.support.AbstractAutowiredCapableBeanFactory.doCreateBean(AbstractAutowiredCapableBeanFactory.java:420) ~[spring-beans-5.3.21.jar:5.3.21]
at org.springframework.beans.factory.support.AbstractBeanFactory.lambda$doGetBean$0(AbstractBeanFactory.java:336) ~[spring-beans-5.3.21.jar:5.3.21]
at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:216) ~[spring-beans-5.3.21.jar:5.3.21]
at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:333) ~[spring-beans-5.3.21.jar:5.3.21]
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:288) ~[spring-beans-5.3.21.jar:5.3.21]
at org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:936) ~[spring-beans-5.3.21.jar:5.3.21]
at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:918) ~[spring-context-5.3.21.jar:5.3.21]
at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:583) ~[spring-context-5.3.21.jar:5.3.21]
at org.springframework.boot.web.servlet.context.ServletWebServerApplicationContext.refresh(ServletWebServerApplicationContext.java:147) ~[spring-boot-2.7.1.jar:2.7.1]
at org.springframework.boot.SpringApplication.refresh(SpringApplication.java:734) ~[spring-boot-2.7.1.jar:2.7.1]
at org.springframework.boot.SpringApplication.refreshContext(SpringApplication.java:389) ~[spring-boot-2.7.1.jar:2.7.1]
at org.springframework.boot.SpringApplication.run(SpringApplication.java:309) ~[spring-boot-2.7.1.jar:2.7.1]
at org.springframework.boot.SpringApplication.run(SpringApplication.java:1205) ~[spring-boot-2.7.1.jar:2.7.1]
at com.bd.f1.F1ApplicationKt.main(F1Application.kt:10) ~[main/:na]
Caused by: java.lang.NoClassDefFoundError: Could not create proxy for public abstract com.bd.f1.repo.UserRepository findDataBySqlById(java.lang.Long):
```

Erro obtido no springboot

4. Conclusão

Quando se trata de lançar aplicações no mundo real, não basta conseguir usar os SGDB'S disponíveis, também é um grande desafio integrar os dados com a aplicação de modo a se obter um sistema funcional. Além disso, por mais amigável que seja o ambiente de desenvolvimento do spring boot, por ele permitir conexão ao banco de dados de maneira praticamente transparente, sendo necessário apenas configurar alguns arquivos, associar as queries à funções pode ser um pouco confuso, principalmente com queries mais elaboradas.

