



**UNIVERSIDADE FEDERAL DE VIÇOSA**  
**UFV - Campus Florestal**

**ARQUITETURA DE SOFTWARE**  
Gláucia Braga e Silva

**Prova de Conceito**

Álvaro Gomes Da Silva Neto - 5095  
Danilo Matos De Oliveira - 5073  
Jéssica Cristina Carvalho - 4686  
Ítallo Winícios Ferreira Cardoso - 5101  
Manuel Ferreira Ribeiro Di Simões - 5091  
Rodrigo Dos Santos Miranda - 5090

Florestal - MG  
2024

# 1. Introdução

Neste documento, apresentamos uma proposta de arquitetura para nosso jogo de cartas 2D, meticulosamente delineada com a seleção criteriosa de frameworks, linguagens e bibliotecas para simplificar o desenvolvimento do software proposto.

A construção da aplicação será baseada na engine libGDX, que oferece um ambiente robusto e versátil para o desenvolvimento de jogos. A escolha da libGDX é respaldada por sua ampla aceitação na indústria de desenvolvimento de jogos, proporcionando eficiência na criação e na manipulação de elementos gráficos e lógicos. A libGDX permite o gerenciamento de cartas, animações e interações, essenciais para criar uma experiência envolvente e educativa.

No que diz respeito ao backend da aplicação, optamos por integrar o Spring Boot com um banco de dados MySQL. Essa combinação proporciona uma estrutura sólida que gerenciará a lógica do jogo, o armazenamento de dados dos usuários e as interações complexas que ocorrem no servidor. A escolha do Spring Boot garante um desenvolvimento ágil e eficiente, enquanto o MySQL oferece um sistema de gerenciamento de banco de dados confiável e escalável.

Para facilitar os testes locais, utilizaremos o Docker. Essa abordagem permitirá que configuramos rapidamente um ambiente isolado e consistente para o backend, assegurando que as dependências e configurações do MySQL sejam replicáveis em diferentes máquinas. O uso do Docker foi para simplificar o processo de desenvolvimento e também garantir que todos os membros da equipe trabalhem em um ambiente padronizado.

## 2. Pré-Requisitos

O jogo será desenvolvido utilizando a linguagem de programação Java e o SGBD MySQL. Sendo assim, antes de definirmos a arquitetura, precisamos instalá-los.

### 2.1 Java

Para desenvolver aplicativos e jogos com a engine libGDX, é essencial ter o Java Development Kit (JDK) instalado no seu sistema. Certifique-se de seguir este guia para instalar o JDK 17 ou 21, pois a engine requer, no mínimo, a versão 17 do JDK para funcionar corretamente.

#### 2.1.1 Verificar a Versão do Java Atual (Opcional)

Antes de começar, é aconselhável verificar se você já possui o JDK instalado e qual a versão atual do Java no seu sistema. Isso pode ser feito abrindo um terminal ou prompt de comando e digitando o seguinte comando:

```
java -version
```

Isso mostrará a versão atual do Java instalada no seu sistema. Se a versão for inferior a 17, continue com a instalação.

#### 2.1.2 Baixar o JDK 17 ou Superior

Para baixar a versão 17 ou superior do JDK, siga estas etapas:

1. Acesse o site oficial da Oracle ou de outro provedor confiável de JDK, como o OpenJDK.
2. Procure pela versão 17 ou superior do JDK para download.
3. Selecione o pacote de instalação adequado para o seu sistema operacional (Windows, macOS ou Linux). Certifique-se de escolher a versão adequada para a arquitetura do seu sistema (32 bits ou 64 bits).
4. Clique no link de download e siga as instruções para baixar o JDK.

#### 2.1.3 Instalar o JDK

A instalação do JDK pode variar dependendo do seu sistema operacional, mas geralmente envolve seguir as instruções de instalação padrão. Aqui estão algumas orientações gerais:

- **Windows:** Execute o arquivo de instalação baixado e siga o assistente de instalação. Certifique-se de adicionar o JDK ao PATH do sistema durante a instalação.
- **macOS:** Abra o arquivo de instalação baixado e siga as instruções na tela. O JDK será instalado no diretório `/Library/Java/JavaVirtualMachines/`.
- **Linux:** Dependendo da distribuição, você pode instalar o JDK usando o gerenciador de

pacotes. Por exemplo, no Ubuntu, você pode usar o seguinte comando:

```
sudo apt-get install openjdk-17-jdk
```

### 2.1.4 Verificar a Instalação

Após a instalação, abra um terminal ou prompt de comando e execute novamente o seguinte comando para verificar a versão do Java:

```
java -version
```

## 2.2 MySQL

Este guia abordará a instalação do MySQL, um sistema de gerenciamento de banco de dados amplamente utilizado. Além disso, você também aprenderá a configurar o MySQL usando Docker, facilitando a criação de um ambiente de desenvolvimento isolado e a inicialização com um script de inicialização ([init.sql](#).)

### 2.2.1 Instalação do MySQL sem Docker

1. Acesse o site oficial do MySQL ([MySQL Downloads](#)) e escolha a versão do MySQL apropriada para o seu sistema operacional (Windows, macOS ou Linux).
2. Faça o download do instalador e execute-o.
3. Siga as instruções do instalador para configurar o MySQL. Você será solicitado a definir uma senha para a conta root do MySQL durante o processo de instalação.
4. Após a conclusão da instalação, o MySQL estará em execução e você poderá acessá-lo usando um cliente MySQL, como o MySQL Workbench ou o MySQL Command Line Client.

## 2.2.2 Instalação do MySQL com Docker

Se preferir usar o Docker para configurar o MySQL, siga estas etapas:

1. Certifique-se de que o Docker esteja instalado no seu sistema. Você pode baixá-lo em [Docker Get Started](#).
2. Crie um arquivo chamado Dockerfile (sem extensão) em um diretório de sua escolha e adicione o seguinte conteúdo:

```
# Use a imagem base oficial do MySQL
FROM mysql:8

# Defina as variáveis de ambiente para configurar o MySQL
ENV MYSQL_ROOT_PASSWORD=dummpassword
ENV MYSQL_USER=user
ENV MYSQL_PASSWORD=dummpassword
ENV MYSQL_DATABASE=poc

# Exponha a porta 3306 para permitir conexões externas
EXPOSE 3306

# Copie o arquivo init.sql para a pasta /docker-entrypoint-initdb.d/
COPY ./scripts/init.sql /docker-entrypoint-initdb.d/
```

3. Dentro do mesmo diretório, crie uma pasta chamada scripts e coloque o arquivo init.sql com as instruções de criação das tabelas necessárias para a prova de conceito.
4. Abra um terminal ou prompt de comando na pasta onde você criou o Dockerfile e os arquivos init.sql.
5. Execute o seguinte comando para construir a imagem do Docker do MySQL

```
docker build -t mysql .
```

6. Após a conclusão da construção da imagem, você pode iniciar um contêiner MySQL com o seguinte comando:

```
docker run --detach -p 3306:3306 mysql
```

O MySQL estará em execução dentro do contêiner, e você poderá acessá-lo na porta 3306 do seu sistema. Isso conclui o guia de instalação do MySQL, tanto de forma convencional quanto com o uso do Docker.

## 2.3 LibGDX

A LibGDX é uma poderosa framework de desenvolvimento de jogos 2D e 3D em Java, ideal para criar jogos multiplataforma. Ela oferece um conjunto abrangente de ferramentas e bibliotecas, facilitando o desenvolvimento e a implementação de gráficos, som, entrada de usuário e física. Ele pode ser obtido pelo link <https://libgdx.com/wiki/start/project-generation>, onde também pode ser encontrada sua documentação.

## 2.4 Spring Boot

1. Navegue até o diretório do projeto.
2. Execute o seguinte comando Maven:

```
mvn spring-boot:run
```

3. O projeto começará a rodar em background, usando o próprio PC como server

## 2.5 Gradlew

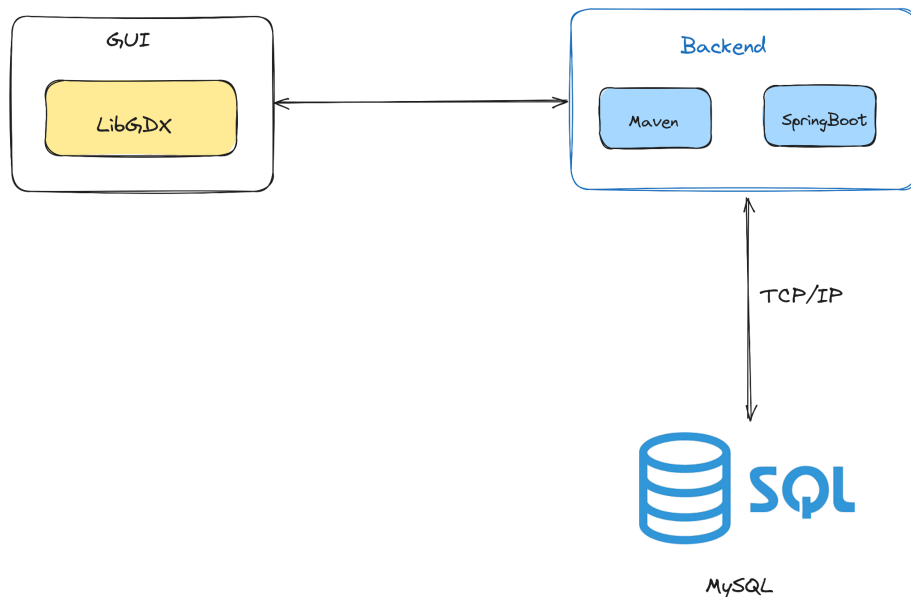
1. Em outra Aba do terminal, navegue até o diretório inicial, onde está o arquivo gradlew e execute o seguinte comando:

```
./gradlew clean build
```

```
./gradlew run
```

### 3. Arquitetura

Nesta seção, exploraremos a arquitetura do projeto de jogo desenvolvido com LibGDX, que é organizada em dois módulos distintos: **"backend"** e **"gui"**. Cada módulo desempenha um papel fundamental no desenvolvimento e funcionamento do jogo, e a estrutura de diretórios reflete a organização lógica das funcionalidades.



#### 3.1 Módulo "backend"

O módulo **"backend"** abrange a lógica central do jogo, incluindo:

- **Gerenciamento de Dados:** Responsável por armazenar e recuperar informações relacionadas ao jogo.
- **Interação com o Banco de Dados:** Processa consultas e atualizações no banco de dados, garantindo a persistência dos dados.
- **Processamento de Regras de Negócios:** Implementa as regras que governam a mecânica do jogo.
- **Serviços Essenciais:** Fornece funcionalidades críticas que suportam a operação do jogo, como autenticação e gerenciamento de sessões.

```
//utilizando os serviços do backend
public class ApiService {
    private final String BASE_URL = "http://localhost:8080/usuarios";
    public void createUser(String nome, String email) {
        String json = "{\"nome\":\"" + nome + "\",\"email\":\"" + email
+ "\"}";
        HttpRequest request = new HttpRequest(Net.HttpMethods.POST);
        request.setUrl(BASE_URL);
        request.setContent(json);
        request.setHeader("Content-Type", "application/json");
    }
}
```

```

Gdx.net.sendHttpRequest(request, new HttpListener() {
    @Override
    public void handleHttpResponse(HttpResponse response) {
        if (response.getStatus().getStatusCode() == 200) {
            System.out.println("Usuário criado com sucesso: " +
response.getResultAsString());
        } else {
            System.out.println("Erro ao criar usuário: " +
response.getStatus());
        }
    }
    @Override
    public void failed(Throwable t) {
        System.out.println("Falha na requisição: " +
t.getMessage());
    }
    @Override
    public void cancelled() {
        System.out.println("Requisição cancelada.");
    }
}

```

No nosso cenário, cadastramos um item como exemplo:

```

mysql> select * from usuario;
+-----+-----+-----+
| id | email                                | nome |
+-----+-----+-----+
| 6 | DaniloEujogoLOL@example.com | Danilo |
+-----+-----+-----+
1 row in set (0,00 sec)

```

## 3.2 Módulo "gui"

O módulo "gui" concentra-se na interface gráfica do jogo, gerenciando:

- **Elementos Visuais:** Controla a aparência dos gráficos, animações e outros componentes visuais do jogo.
- **Controles de Usuário:** Implementa a interação do jogador com o jogo, incluindo botões, menus e controles de movimento.
- **Apresentação Geral da Experiência do Jogador:** Organiza a interface de maneira que proporcione uma experiência fluida e intuitiva para os usuários.

Essa arquitetura modular não apenas facilita o desenvolvimento e a manutenção do jogo, mas também permite que diferentes equipes trabalhem em paralelo em cada módulo, aumentando a eficiência do projeto.



