



SERVIÇO PÚBLICO FEDERAL · MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE VIÇOSA · UFV
CAMPUS FLORESTAL

Projeto Integrador: Prova de Conceito - Equipe 2

Autores (as):

Sabrina Bruni de Souza Faria - 5081

Luiz Guilherme de Sá Gontijo - 4839

Breno Júnio Oliveira Gomes - 5085

Sumário

1. Visão Geral:	3
2. Descrição Técnica:	4
2.1. Figma :	4
2.2. FL Studio :	4
2.3. Java :	5
2.4. VsCode :	5
2.5. LibGDX :	6
2.6. MySQL :	6
2.7. Aiven :	6
3. Desenvolvimento:	7
4. Organização do Projeto:	9
5. Código Desenvolvido:	10

1. Visão Geral:

O projeto do grupo consiste em um jogo educativo de cartas que visa introduzir a alunos do 2º ano do fundamental I dispositivos computacionais diversos, ensinando-lhes as variedades e utilidades da tecnologia em seu redor.

O jogo terá três tipos de desafio, todos envolvendo cartas: No primeiro, deve-se associar pares de cartas cujas imagens remetam ao mesmo dispositivo computacional; No segundo, associar pares de dispositivos com funcionalidades similares; No terceiro, escolher uma carta cujo dispositivo seja útil para resolver um determinado contexto.

Antes de resolver os desafios, o aluno deverá escolher seu nome e sua turma, para que, após resolvê-los, seja registrada num banco de dados em nuvem a partida que ele jogou, persistindo, de cada desafio, o tempo que ele levou para completar, a quantidade de erros cometidos e a pontuação obtida.



Figura 1: Visão Geral das Ferramentas.

Para atingir esse objetivo, serão utilizadas diferentes ferramentas. Para modelar a interface gráfica e elaborar os efeitos sonoros do sistema, serão utilizados, respectivamente, as aplicações Figma e FL Studio. Uma vez desenhadas as telas serão programadas através da biblioteca libGDX, em linguagem Java, no editor de código VsCode.

A libGDX conta com diversos recursos para desenvolvimento front-end de jogos, sendo necessário começar um projeto libGDX através de um programa inicializador de projetos fornecido pelos próprios desenvolvedores da biblioteca. Ele já prepara a organização do ambiente de desenvolvimento, separando as pastas de recursos, de códigos-fonte, além de oferecer templates de ciclos de vida de um jogo,

dos quais seguiremos o *ApplicationListener*, que gera a classe main, se estendendo de *Game*. A partir disso, pode-se programar o projeto com código Java, construindo pontes nos lugares corretos do ciclo de vida do jogo entre interface de usuário e classes de entidade.

Finalmente, ao fim de um jogo serão registradas as estatísticas da partida realizada pelo usuário. Isso será feito através das classes do pacote java.sql, nativo à linguagem, e da implementação do padrão de projeto DAO. Essa implementação estabelecerá uma conexão com um banco de dados MySQL em nuvem, hospedado através da ferramenta Aiven, que fornece 5GB de espaço gratuitos em um único banco, justamente a fim de atender a projetos pequenos e incipientes.

2. Descrição Técnica:

2.1. Figma :

Figma é um editor gráfico de vetor e prototipagem de projetos de design baseado principalmente no navegador web, com ferramentas offline adicionais para aplicações desktop para GNU/Linux, macOS e Windows.

O Figma será utilizado pelo time de Design de Jogo para criar os arquivos de imagem representando os elementos visuais do jogo, garantindo consistência visual e agilidade no processo de prototipagem.

Os componentes gráficos e telas do sistema foram modelados e projetados nesta plataforma, que permite a exportação direta dos elementos visuais para o ambiente de desenvolvimento. A disponibilidade multiplataforma do Figma também facilita a integração das contribuições da equipe, assegurando uma iteração contínua e eficiente ao longo do desenvolvimento do jogo.

Documentação: <https://help.figma.com/hc/pt-br>

Tutorial: https://www.youtube.com/watch?v=L7wLofT_BVc

2.2. FL Studio FL STUDIO:

FL Studio é uma estação de trabalho de áudio digital (DAW) projetada para produção musical, gravação, edição, mixagem e masterização. Disponível para Windows e macOS, ele oferece um ambiente completo para criação de música, com suporte a sequenciamento por padrão de padrões e linha do tempo, sintetizadores integrados, gravação de áudio multipista e uma ampla gama de efeitos e plugins. O software é amplamente utilizado para produção de música eletrônica, hip-hop e outros gêneros, oferecendo uma interface intuitiva e suporte para MIDI, VSTs e automação de parâmetros.

O FL Studio permite a composição de sons personalizados, que são integrados ao sistema por meio de arquivos de áudio exportados diretamente para os formatos compatíveis com o ambiente de desenvolvimento. A flexibilidade oferecida pela plataforma permite ajustes rápidos e refinamentos contínuos das

trilhas sonoras, garantindo que a identidade sonora do jogo possa evoluir de forma iterativa durante o desenvolvimento do jogo.

Documentação: <https://www.image-line.com/fl-studio-learning/fl-studio-online-manual/>

Tutorial: <https://www.youtube.com/watch?v=yiRIOeouS2k>

2.3. Java :

Java é uma linguagem de programação orientada a objetos e de tipagem estática, executada em uma Máquina Virtual Java (JVM), permitindo a portabilidade entre diferentes sistemas operacionais. Suas principais aplicações incluem o desenvolvimento de software corporativo, aplicações web, sistemas móveis (especialmente Android) e aplicações distribuídas, com suporte nativo a multithreading e gerenciamento automático de memória por meio de coleta de lixo (Garbage Collector).

O Java será utilizado pelos Desenvolvedores Júnior e pelos Desenvolvedores Sênior para a programação das classes do Back-End, responsáveis pela lógica e funcionamento interno do jogo.

Como o projeto será desenvolvido em Java, é preciso ter o JDK instalado tanto na máquina do desenvolvedor quanto na máquina do cliente, mais precisamente o JDK 17.

Documentação: <https://docs.oracle.com/en/java/>

Tutorial: <https://www.youtube.com/watch?v=EpXYPB1rv4w>

2.4. VsCode :

Visual Studio Code (VS Code) é um editor de código-fonte leve e poderoso, projetado para desenvolvimento de software, depuração, versionamento e colaboração em tempo real. Disponível para Windows, macOS e Linux, ele oferece um ambiente robusto para programadores com suporte a múltiplas linguagens de programação, incluindo extensões para autocompletar, linting e depuração. O editor é amplamente utilizado para desenvolvimento web, back-end e outras aplicações, oferecendo uma interface personalizável e uma vasta biblioteca de extensões. O software é conhecido por sua flexibilidade, com integração de ferramentas e ambientes de desenvolvimento como Docker, Kubernetes e ambientes remotos.

O VsCode será utilizado para programar todo código Java desenvolvido no projeto, tanto pelos desenvolvedores Sênior quanto Júnior.

Documentação: <https://code.visualstudio.com/docs>

Tutorial: https://www.youtube.com/playlist?list=PLgBAHVu7jSqAuR8lE3ChhX_BcuzCe8GGk

2.5. LibGDX :

LibGDX é uma estrutura de aplicação de desenvolvimento de jogos livre e de código aberto, escrito em Java, com alguns componentes de C e C++.

O LibGDX será utilizado pelo Desenvolvedor Front-End para a programação das telas do jogo. Ele será responsável por utilizar as imagens criadas no Figma e os áudios criados no FL Studio para a animação do jogo e a interação com o usuário. Também é onde são definidos vários outros detalhes do jogo, como botões, por exemplo.

A máquina do cliente não precisa ter o libGDX instalado, somente o Java. Para que o cliente possa executar o jogo basta abrir o arquivo .jar disponibilizado pelo desenvolvedor. O desenvolvedor precisa ter o libGDX instalado na máquina para a execução do jogo e para gerar o executável .jar.

Documentação e Tutoriais:

<https://libgdx.com/>

<https://chatgpt.com/share/6712f42f-1e7c-800c-90ee-6684fd6396d5>

<https://www.youtube.com/playlist?list=PLrnO5Pu2zAHKAJjRtTLAXtZKMSA6JWnmf>

2.6. MySQL :

MySQL é um sistema de gerenciamento de banco de dados relacional (SGBD) baseado em SQL, amplamente utilizado para armazenar, gerenciar e consultar grandes volumes de dados. Disponível para várias plataformas, como Linux, Windows e macOS, o MySQL é conhecido por sua velocidade, escalabilidade e confiabilidade, sendo amplamente utilizado em aplicações web, incluindo sistemas de gerenciamento de conteúdo (CMS) e plataformas de e-commerce. Ele suporta transações, integridade referencial, replicação de dados, e oferece uma interface de linha de comando, além de APIs para diversas linguagens de programação, como PHP, Java e Python.

O MySQL será utilizado pelos Projetistas de Banco de Dados para a criação de comandos SQL para manipulação de banco de dados, como: SELECTs, CREATEs, INSERTs, entre outros.

O MySQL WorkBench precisa estar instalado na máquina do projetista para que os comandos possam ser executados e testados. A máquina do cliente não precisa de qualquer instalação relacionada ao banco de dados.

Documentação: <https://dev.mysql.com/doc/>

Tutorial: https://www.youtube.com/playlist?list=PLHz_AreHm4dkBs-795Dsgvau_ekxg8g1r

2.7. Aiven aiven :

Aiven é uma plataforma de serviços gerenciados na nuvem que oferece infraestrutura de banco de dados e serviços de streaming, como PostgreSQL, MySQL, Apache Kafka, Redis e outros. Projetado para simplificar a implementação e o gerenciamento de bancos de dados e serviços de dados, o Aiven suporta múltiplos provedores de nuvem, como AWS, Google Cloud e Azure, permitindo alta disponibilidade, escalabilidade e segurança. Ele oferece automação de backups, atualizações, monitoramento e gerenciamento de clusters, facilitando a criação e operação de aplicações baseadas em dados sem a necessidade de gerenciar a infraestrutura subjacente.

O Aiven será utilizado pelos Desenvolvedores Sênior Back-End para a criação do host em nuvem do banco de dados MySQL. A conexão com esse host será gerenciada por meio de um DAO (Data Access Object), que será criado também pelo Sênior Back-End. O DAO utilizará os comandos SQL criados pelos Projetistas para manipular o banco de dados em nuvem.

Para que o cliente execute o jogo não é necessário qualquer instalação, a máquina precisa somente estar conectada à Internet.

Documentação: <https://aiven.io/docs/>

Tutorial: <https://www.youtube.com/watch?v=VhIY7kkAw7w>

3. Desenvolvimento:

Para o desenvolvimento da prova de conceito foi escolhido desenvolver um jogo da memória, visto que possui vários elementos em comum com o projeto que podem ser reutilizados no desenvolvimento oficial.

O jogo da memória inicialmente possui uma tela de menu com botão de iniciar e caixa de texto para o usuário digitar seu nome. A tela seguinte é onde começa o jogo em si, sendo distribuídas pela tela cartas com duas faces que giram ao clicar. Além disso, o jogo possui três níveis de dificuldade, aumentando uma linha e uma coluna de cartas a cada nível.

Também foram criadas telas para quando o usuário perder e para quando ele passar de nível. O usuário perde o jogo quando excede o tempo limite do nível sem terminar de relacionar todas as cartas, voltando para a tela de menu logo em seguida da tela de *game over*. O usuário ganha a cada vez que completa um nível dentro do tempo limite, passando para o próximo nível logo em seguida da tela de vitória.

Além disso, foram criadas classes DAO para o acesso ao banco de dados web. Os dados são salvos quando a aplicação é fechada, quando o usuário ganha todos os níveis ou quando perde um deles. Em ambos os últimos casos, o jogo é reiniciado, podendo ser inserido um novo usuário. O jogo também possui música e sons durante todo o seu ciclo de vida, deixando-o mais divertido e agradável.



Figura 2: Tela de Menu.



Figura 3: Tela nível 1.



Figura 4: Tela de conclusão de nível.



Figura 5: Tela de *game over*.

4. Organização do Projeto:

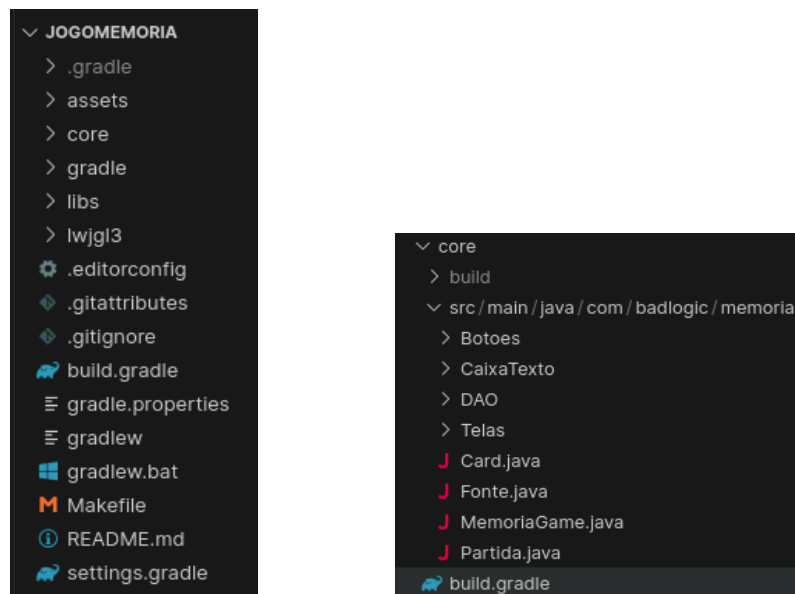


Figura 6 e 7: Capturas de tela da organização das pastas do projeto.

A organização dos arquivos do projeto é elaborada pelo próprio inicializador de projetos da libGDX: O código é programado à partir da pasta-pai *core*, sendo que na pasta *assets* ficam recursos como imagens e sons, e na pasta *main*, os pacotes e arquivos Java de fato escritos por nós. Nela, organizamos as telas e o DAO em pacotes próprios para eles, assim como elementos de front-end, como botões e caixas de texto. Entretanto, como isso é apenas uma prova de conceito, muitas coisas não precisaram de pastas, pois estão sozinhas, como a classe *Card* e a classe de entidade *Partida*. Além destas pastas, há também a pasta *libs*, externa à pasta *core*. Ela foi adicionada manualmente para conter um arquivo de configuração da biblioteca JDBC, necessária para a realização da conexão com o banco de dados.

As outras inúmeras pastas e arquivos no projeto são referentes aos recursos e configurações da biblioteca libGDX, responsáveis pela construção do programa para que tudo possa funcionar perfeitamente. Dentre esses arquivos temos, por exemplo, builds e settings para gerenciamento de dependências, e arquivos responsáveis pelas configurações mais externas (*1wjg13*), como tamanho da janela e ícones.

5. Código Desenvolvido:

Disponível em:

<https://github.com/ProjetoIntegradorUFV2024/Equipe-2Ano/tree/ProvaDeConceito>