



## DBSLP - Database System Learning Plataform

Versão 1.0.0

2023

# Sumário

1. Introdução .....	3
1.1 Finalidade.....	3
1.2 Escopo .....	3
2 Descrição geral do sistema .....	4
2.1 Sobre o Problema .....	4
2.2 Missão .....	4
2.3 Principais envolvidos e suas características .....	4
2.3.1 Usuários do sistema .....	4
2.3.2 Desenvolvedores do Sistema .....	5
3 Descrição geral da arquitetura de desenvolvimento .....	6
4 Visão de Casos de Uso.....	7
4.1 Realizações de casos de uso.....	7
4.1.1 Carregar estado das modelagens.....	7
4.1.2 Carregar um Banco de Dados .....	8
4.1.3 Conectar em uma Sala .....	8
4.1.4 Desconectar de uma Sala .....	9
4.1.5 Executar SQL Scripts.....	9
4.1.6 Executar um SGBD .....	10
4.1.7 Modelar Sistemas de Banco de Dados.....	10
4.1.8 Salvar as modelagens como PNG.....	11
4.1.9 Salvar o Banco de Dados .....	11
4.1.10 Salvar o estado das modelagens .....	12
4.1.11 Selecionar uma sala.....	12
4.1.12 Terminar um SGBD .....	13
4.1.13 Verificar a presença de anomalias nas Tabelas .....	14
4.2 Protótipos .....	14

5 Visão Lógica .....	20
5.1 Modelagem.....	20
5.2 Laboratório.....	31
6 Visão de Processos.....	32
7 Visão de Implementação .....	33
8 Visão de Implantação .....	34
9 Testes .....	35
10 Problemas conhecidos .....	36
REFERÊNCIAS .....	37
ANEXOS .....	38
ANEXO I – Diagrama de Classes .....	39

## **1. Introdução**

### **1.1 Finalidade**

Este documento apresenta os requerimentos do sistema e quais decisões técnicas foram tomadas para atendê-las, servindo como guia de implementação para os engenheiros e desenvolvedores.

### **1.2 Escopo**

Este documento abrange: (i) a definição da missão ou objetivo geral do projeto e suas expectativas (ii) a definição dos requerimentos ou funções (iii) protótipos de alta fidelidade de sua interface (iv) quais decisões técnicas foram tomadas para implementá-los, isso é, a sua arquitetura (v) quais as limitações conhecidas (vi) quais as expectativas para o futuro do projeto.

## 2 Descrição geral do sistema<sup>1</sup>

### 2.1 Sobre o Problema

Há diversas ferramentas para aprendizado de banco de dados, as quais geralmente tem um objetivo específico, como modelagem, verificação dos scripts elaborados, etc. Assim, quando essas são especializadas e autossuficientes, esses sistemas se tornam de difícil integração, o que afeta a experiência dos usuários, pois acabam precisando transicionar entre diversas plataformas para solucionar as suas necessidades.

### 2.2 Missão

O DBSLP (*Database System Learning Platform*) é uma plataforma web distribuída que tem como objetivo auxiliar o aprendizado de sistemas de banco de dados relacionais. Esta plataforma provê um sistema que integra modelagem e execução banco de dados relacionais nativamente no *browser*. Além disso, possibilita a criação de sessões colaborativas *real-time*, possibilitando que discentes se conectem tanto com outros discentes quanto com os docentes, o que melhora a experiência de aprendizado.

Em suma, DBSLP possibilita que o seguinte seja feito diretamente em browser

- Modelagem de bancos de dados relacionais;
- Execução de um banco de dados;
- Criação e uso de sessões colaborativas;

### 2.3 Principais envolvidos e suas características

#### 2.3.1 Usuários do sistema

O sistema será usado por discentes, docentes ou interessados em sistemas de banco de dados relacionais, que se supõe que sejam acostumados com tecnologia e computadores.

---

<sup>1</sup> Essa seção é não-normativa, ou seja, apenas informacional quanto à estrutura do sistema.

O acesso ao sistema é feito preferencialmente por computadores, mas também por dispositivos móveis.

### 2.3.2 Desenvolvedores do Sistema

As áreas de atuação no projeto podem ser divididas em:

- 1) Arquitetos, responsáveis pela definição dos requisitos do sistema e a elaboração da sua arquitetura/estrutura;
- 2) Desenvolvedores front-end, responsáveis pela implementação da aplicação do cliente;
- 3) Desenvolvedores back-ends responsáveis pela implementação dos servidores que proveem os recursos necessários para o funcionamento do sistema;

### 3 Descrição geral da arquitetura de desenvolvimento

O modelo de arquitetura usado foi o *View Model 4 + 1*, que apresenta o sistema por meio de views (visualizações), em que cada view representa o mesmo sistema com detalhes ou ênfases diferentes (HU, 2023).

Conforme Hu (2023), o sistema pode ser apresentado por meio de quatro + 1 visualizações diferentes que são construídas sobre os casos de usos, consistindo de:

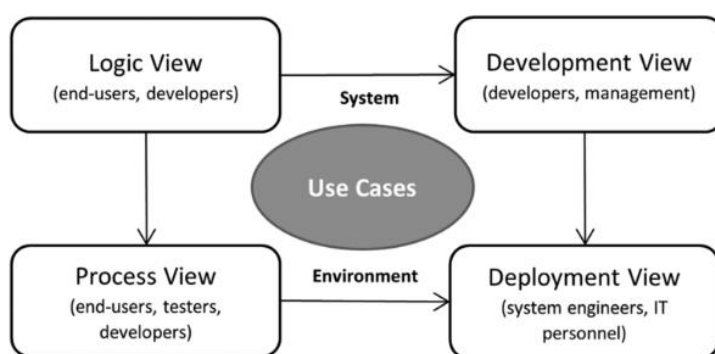
(i) visualização lógica (*logic view*), que especifica a modelagem de objetos em sistemas orientados a objetos;

(ii) visualização de processo (*process view*), que especifica como as entidades concorrem e/ou cooperam pelos recursos seja de forma síncrona ou assíncrona.

(iii) visualização de desenvolvimento (*development view*), que especifica a organização estática do software durante o desenvolvimento, focando nos elementos de software de alto nível;

(iv) visualização de implantação (*deployment view*), que especifica toda a infraestrutura dos hardwares do sistema que executam os softwares e serviços implementado, ou seja, a topologia;

(v) visualização de casos de usos (+1 view), que especifica os requerimentos funcionais e provê as informações fundamentais para todas as outras visualizações;



Fonte: Hu (2023, p.289).

Figura 1 Representação de uma arquitetura 4 + 1 *View Model*.

## 4 Visão de Casos de Uso

Os requisitos funcionais são:

- RF1: Prover ferramentas para modelagem conceitual e lógica;
- RF2: Prover mecanismo para sessões compartilhadas para a ferramenta de modelagem;
- RF3: As ferramentas de modelagens devem ser salváveis;
- RF4: As ferramentas de modelagens devem possibilitar exportação como Imagem;
- RF3: Prover um SGBD para execução de script SQL;

Os requisitos não-funcionais são:

- RNF1: A aplicação deve ser multiplataforma (web e mobile);
- RNF2: Deve ser acessível ao usuário a documentação do sistema;
- RNF3: A aplicação deve ser SPA (Single Page Application);

No Diagrama 1 está representado os casos de uso do sistema. Esse diagrama representa os requisitos funcionais como funcionalidades ou usos do sistema, sendo a sua realização dividida principalmente em três grandes pacotes: (i) lab, que consiste do ambiente para execução de SQL Scripts (ii) modeling, que consiste do ambiente para modelagem conceitual e lógica e (iii) hooks, que consistem dos serviços secundários ou complementares.

### 4.1 Realizações de casos de uso

#### 4.1.1 Carregar estado das modelagens

##### 1 Objetivos

Permitir ao usuário carregar um documento de modelagem salvo previamente.

##### 2 Sumário

O usuário pode carregar um documento previamente salvo (serializado), de forma que a sua sessão é restaurada.

##### 3 Atores

Usuário



#### 4 Fluxo Normal

- a. Usuário clica no menu “Arquivos”;
- b. Usuário clica no botão “Carregar Modelagem”;
- c. O modal “Carregar Arquivos” é exibido;
- d. Usuário clica no botão “Selecionar Arquivos”;
- e. Usuário seleciona o arquivo JSON de seu computador;
- f. Usuário clica no botão “Carregar”;
- g. O arquivo é desserializado e é usado como novo estado da modelagem.

#### 5 Fluxo Alternativo

- a. Um formato de arquivo inválido é selecionado.

#### 4.1.2 Carregar um Banco de Dados

##### 1 Objetivos

Permitir ao usuário carregar um banco de dados salvo previamente.

##### 2 Sumário

O usuário pode carregar um banco de dados previamente salvo (serializado), de forma que a sua sessão é restaurada.

##### 3 Atores

Usuário.

#### 4 Fluxo Normal

- a. Usuário clica no botão “Sistema Gerenciador de Banco de Dados”;
- b. O modal “Sistema Gerenciador de Banco de Dados” é exibido;
- c. Usuário clica no botão “Selecionar Arquivos”
- d. Usuário seleciona o arquivo BINARY de seu computador
- e. Usuário clica no botão “Carregar”
- f. O arquivo é desserializado e é usado como novo estado do banco de dados.

#### 5 Fluxo Alternativo

- b. Um formato de arquivo inválido é selecionado.

#### 4.1.3 Conectar em uma Sala

## 1 Objetivos

Possibilitar ao usuário se conectar em uma sala de sessão compartilhada.

## 2 Sumário

O usuário pode entrar em uma sessão compartilhada a partir de seu token.

## 3 Atores

Usuário.

## 4 Fluxo normal

- a. Usuário clica no botão “Gerenciar sessões”;
- b. Usuário digita o token da sessão alvo;
- c. Usuário clica em “Entrar na Sala”.

### 4.1.4 Desconectar de uma Sala

## 1 Objetivos

Possibilitar que o usuário se desconecte de uma sala de sessão compartilhada.

## 2 Sumário

O usuário pode se desconectar de uma sala de sessão compartilhada, não sincronizando o seu documento com o de outros participantes.

## 3 Atores

Usuário.

## 4 Fluxo normal

- a. Usuário clica no botão “Gerenciar sessões”;
- b. Usuário clica no botão “Desconectar”.

### 4.1.5 Executar SQL Scripts

## 1 Objetivos

Permitir ao usuário executar SQL Scripts.

## 2 Sumário

O usuário pode inserir SQL Scripts que serão executados por um SGBD, gerando um resultado, possivelmente.

### 3 Atores

Usuário.

### 4 Fluxo normal

- a. Usuário digita SQL Statements em um campo de texto;
- b. Usuário clica no botão “Executar Script”;
- c. Usuário recebe um feedback do sucesso ou não da execução;
- d. Usuário recebe o resultado do Script, se algum.

### 5 Fluxo Alternativo

- a. Ocorre erro na execução do script, não sendo mostrado algum resultado.

#### 4.1.6 Executar um SGBD

##### 1 Objetivos

Permitir ao usuário a execução de um processo de SGBD.

##### 2 Sumário

O usuário pode começar a execução de um processo de SGBD, que é necessário para execução de SQL Scripts.

### 3 Atores

Usuário.

### 4 Fluxo normal

- a. Usuário clica no botão “Sistema Gerenciador de Banco de Dados”;
- b. O modal “Sistema Gerenciador de Banco de Dados” é exibido;
- c. Usuário clica no botão “Executar”;
- d. Um novo SGBD é alocado e é executado.

#### 4.1.7 Modelar Sistemas de Banco de Dados

##### 1 Objetivos

Permitir ao usuário ferramentas para modelagem conceitual e lógica de banco de dados relacionais.

## 2 Sumário

O usuário pode modelar conceitualmente e logicamente conforme os modelos Entidade-Relacionamento e relacional, respectivamente.

## 3 Atores

Usuário.

## 4 Fluxo normal

- a. Usuário clica no botão “Modelagens”;
- b. O aplicativo de modelagem é exibido.

### 4.1.8 Salvar as modelagens como PNG

#### 1 Objetivos

Permitir ao usuário que salve uma modelagem específica (conceitual ou lógica) como PNG.

#### 2 Sumário

O usuário pode salvar uma modelagem específica (conceitual ou lógica) como PNG.

#### 3 Atores

Usuário.

#### 4 Fluxo normal

- a. Usuário clica no Menu “Arquivo”;
- b. Usuário clica no botão “Salvar Como”;
- c. Usuário clica no botão “PNG”.
- d. O modelo atual (conceitual ou lógico) é salvo como PNG.

### 4.1.9 Salvar o Banco de Dados

#### 1 Objetivos

Permitir ao usuário que salve (serialize) o estado atual do banco de dados.

## 2 Sumário

O usuário pode salvar (serializar) o estado atual do banco de dados, de forma que possa regenerá-lo (desserializar) posteriormente.

## 3 Atores

Usuário.

## 4 Fluxo normal

- a. Usuário clica no botão “Sistema Gerenciador de Banco de Dados”;
- b. O modal “Sistema Gerenciador de Banco de Dados” é exibido;
- c. Usuário clica no botão “Salvar”
- d. O estado do banco de dados atual é salvo como BINARY.

### 4.1.10 Salvar o estado das modelagens

#### 1 Objetivos

Permitir ao usuário que salve (serialize) o estado atual das modelagens (conceitual e lógica).

#### 2 Sumário

O usuário pode salvar (serializar) o estado atual das modelagens (conceitual e lógica), de forma que possa regenerá-lo (desserializar) posteriormente.

#### 3 Atores

Usuário.

#### 4 Fluxo normal

- a. Usuário clica no menu “Arquivo”.
- b. Um menu é exibido.
- c. Usuário clica no botão “Salvar Como”
- d. Um menu é exibido.
- e. Usuário clica no botão “JSON”.

### 4.1.11 Selecionar uma sala

## 1 Objetivos

Permitir ao usuário acessar uma sala de sessão colaborativa.

## 2 Sumário

Permite ao usuário entrar em uma sala de sessão colaborativa a partir de uma chave. Gerar uma chave de acesso à sessão criada.

## 3 Atores

Qualquer entidade que quer ser um Host de uma sessão.

## 4 Fluxo normal

- a. Usuário clica no botão “Gerenciador de sessões”;
- b. O modal “Gerenciador de sessões” é exibido;
- c. Usuário clica no botão “Iniciar nova sessão”;
- d. É exibido no modal a chave de acesso a sessão criada.

## 5 Fluxos alternativo

- a. A criação da sessão falha;
- b. O usuário já está em uma sessão (ou como host ou como guest);

### 4.1.12 Terminar um SGBD

## 1 Objetivos

Permitir ao usuário que termina a execução de um SGBD.

## 2 Sumário

O usuário pode terminar a execução de um SGBD, de forma que não será mais possível executar SQL Scripts.

## 3 Atores

Usuário.

## 4 Fluxo normal

- a. Usuário clica no botão “Sistema Gerenciador de Banco de Dados”;
- b. O modal “Sistema Gerenciador de Banco de Dados” é exibido;
- c. Usuário clica no botão “Parar”;

- d. A execução do SGBD é terminada e esse é liberado.

#### 4.1.13 Verificar a presença de anomalias nas Tabelas

##### 1 Objetivos

Permitir ao usuário verificar a presença de anomalias nas Tabelas.

##### 2 Sumário

O usuário pode verificar a presença de anomalias nas Tabelas do Banco de Dados.

##### 3 Atores

Usuário.

##### 4 Fluxo normal

- Usuário clica no botão “Buscador de Anomalias”;
- Um modal é exibido;
- Usuário seleciona uma tabela da lista de Tabelas;
- Uma lista de hints (sugestões) é exibida, indicando quais colunas da Tabela tiveram linhas que sugerem dependências implícitas/funcionais.

#### 4.2 Protótipos

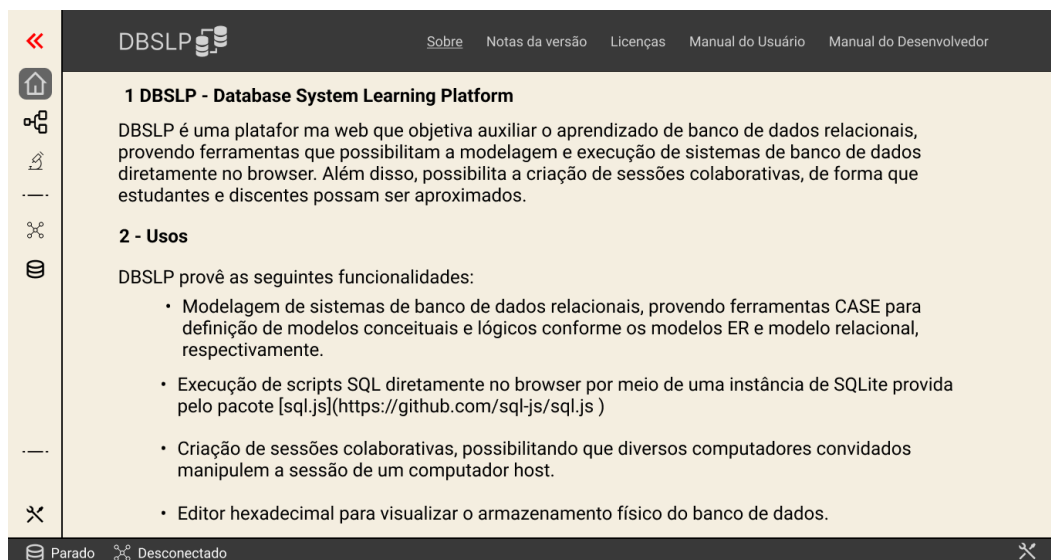


Figura 2 Desktop-Home-Page

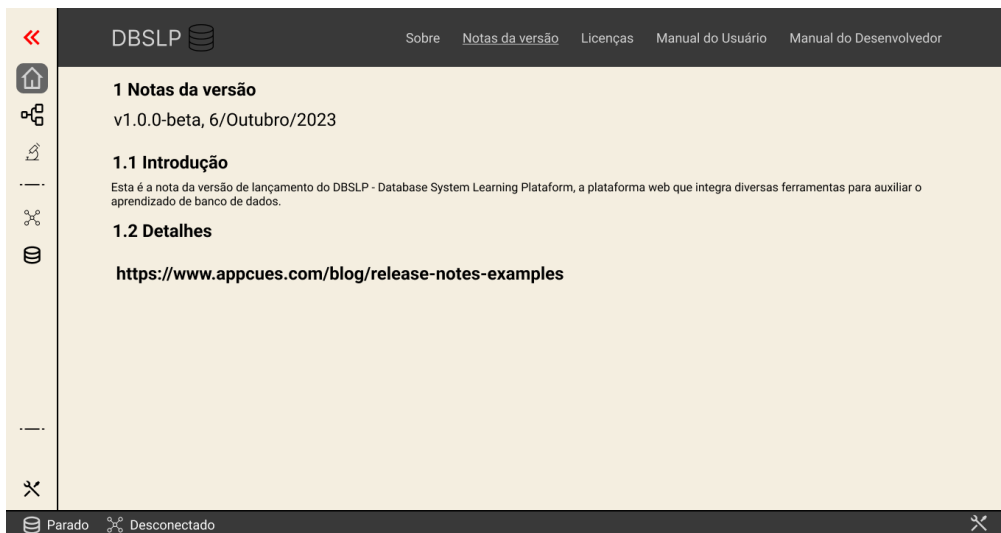


Figura 3 Desktop-Home-Page-2

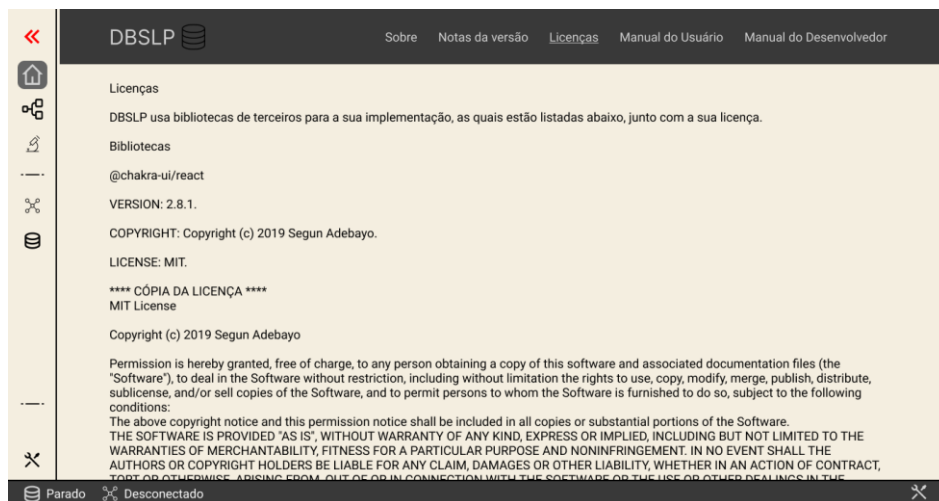


Figura 4 Desktop-Home-Page-3

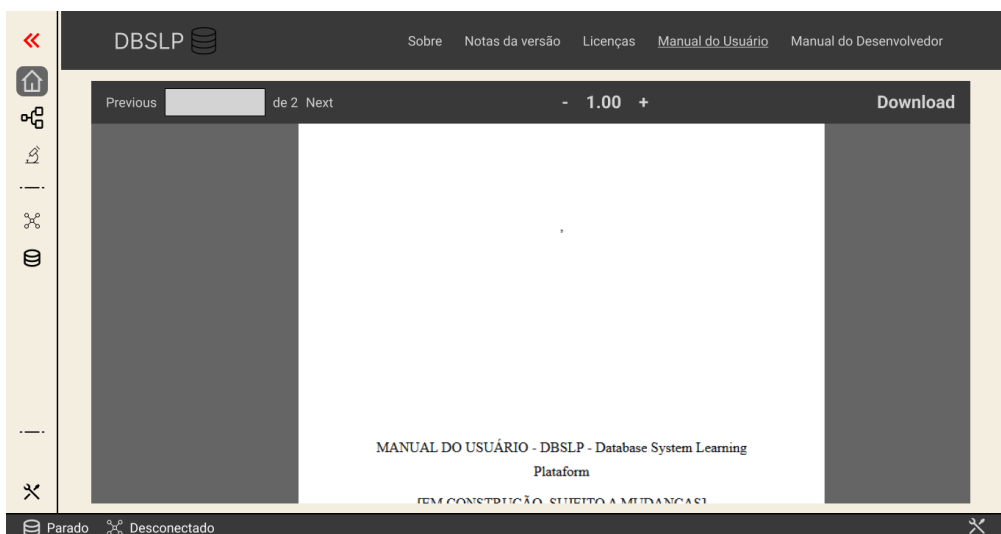


Figura 5 Desktop-Home-Page-4



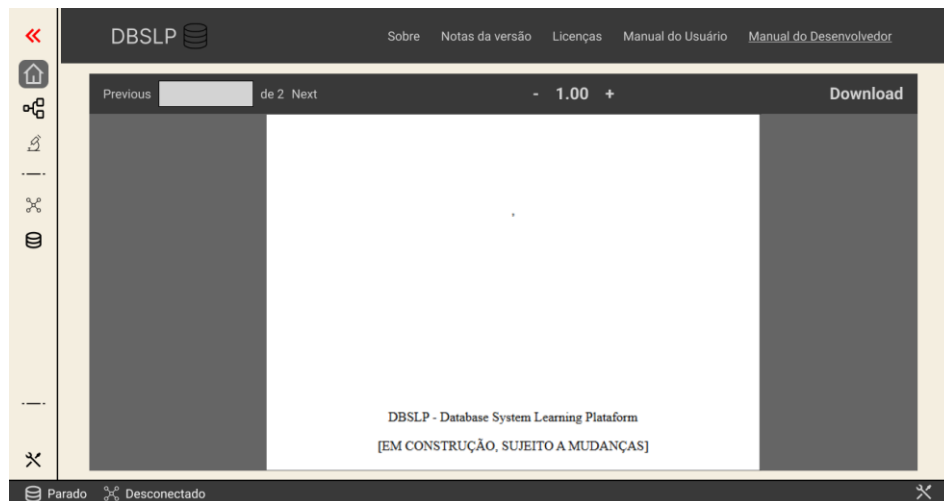


Figura 6 Desktop-Home-Page-5

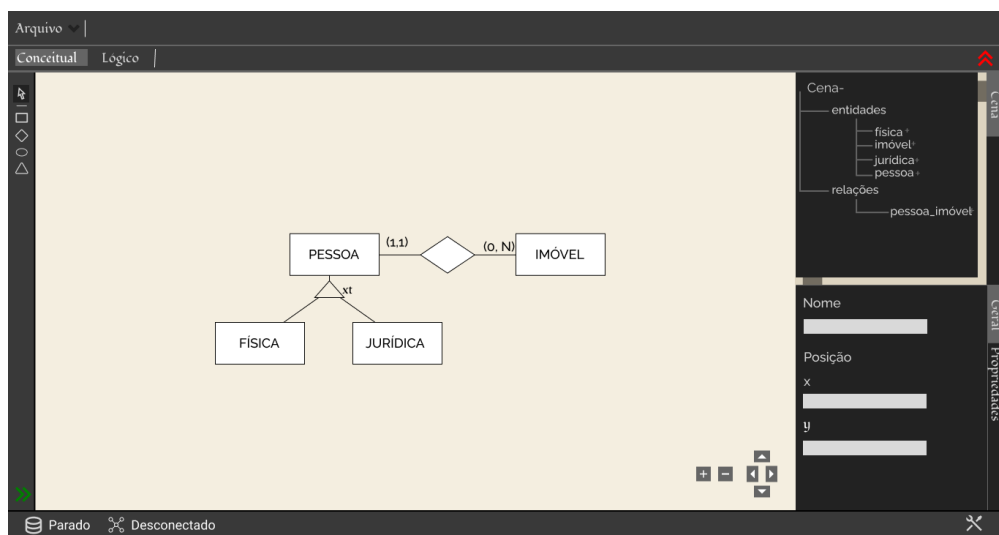


Figura 7 Desktop-Modelagem-Conceitual

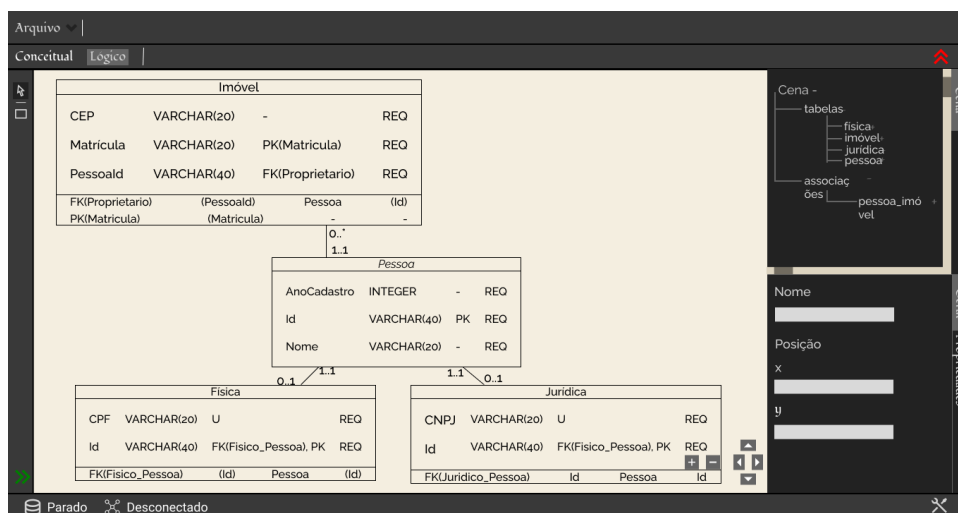


Figura 8 Desktop-Modelagem-Logica

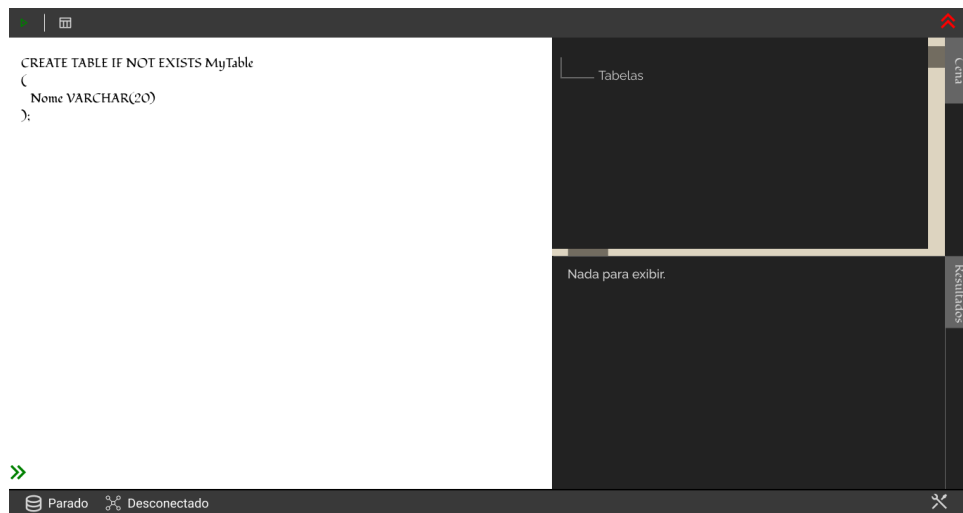


Figura 9 Desktop-SQL-Lab.



Figura 10 Desktop-Modal-Anomalias.

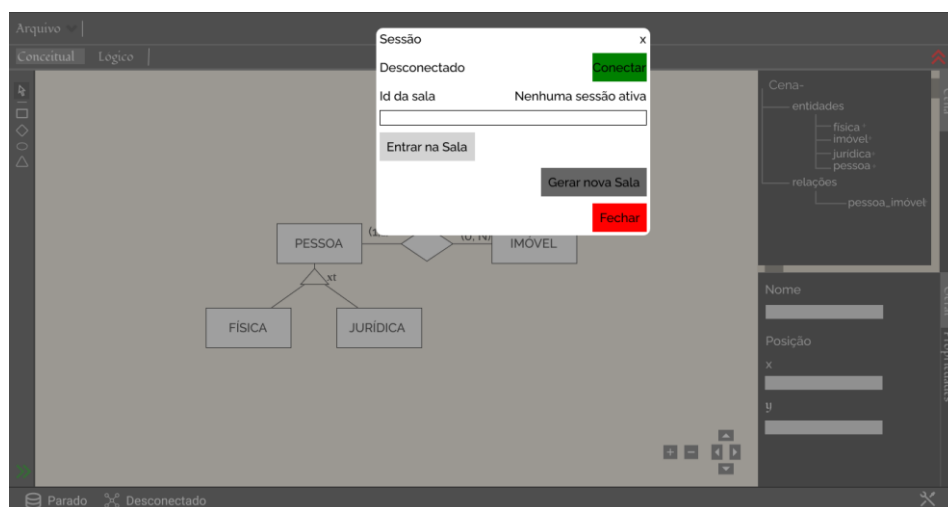


Figura 11 Desktop-Modal-Sessão.

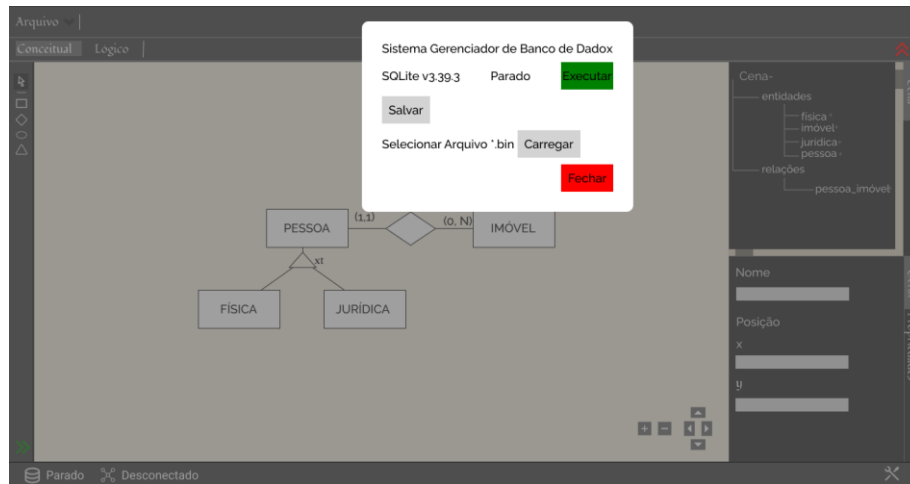


Figura 12 Desktop-Modal-SGBD.

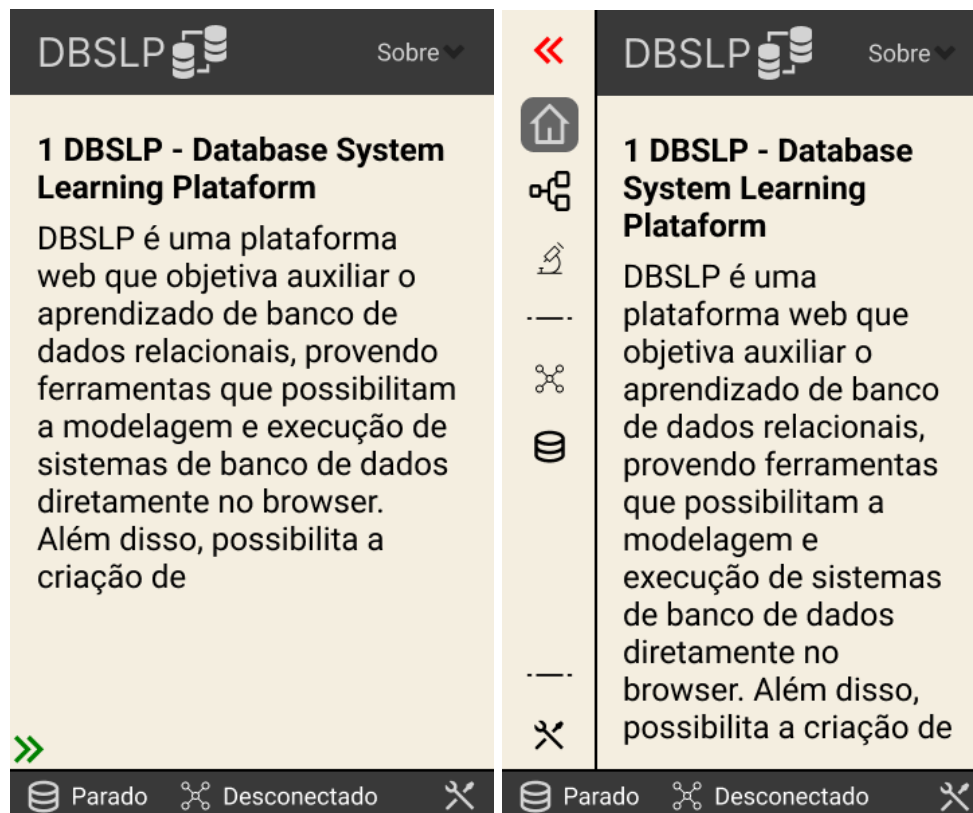


Figura 13 Mobile-Home

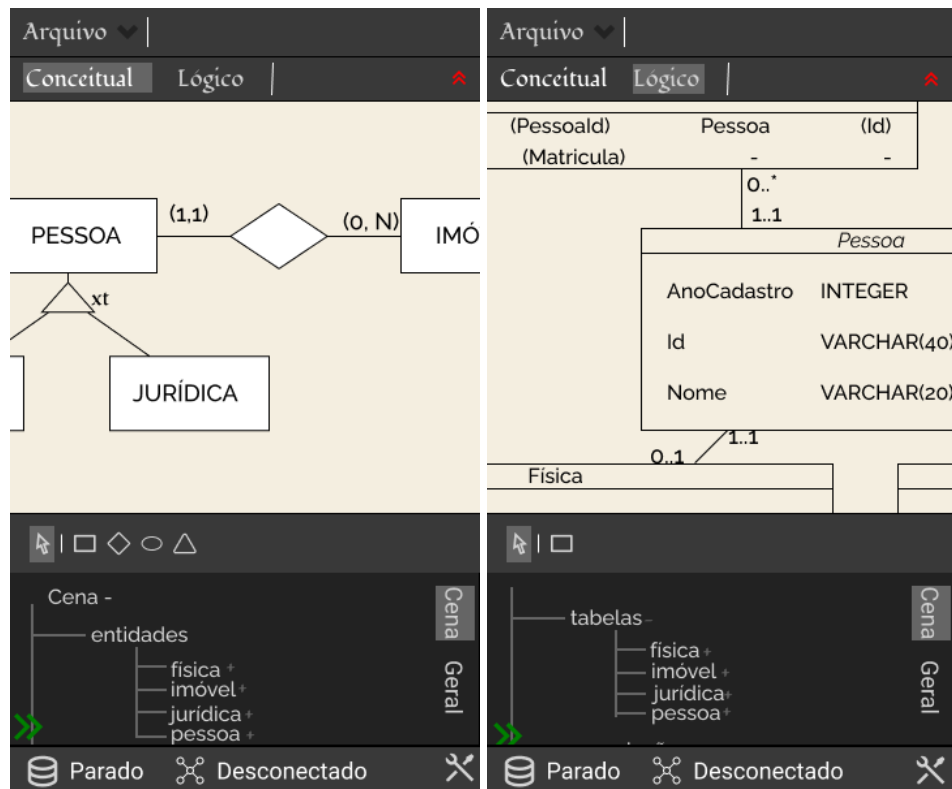


Figura 14 Mobile-Modelagem

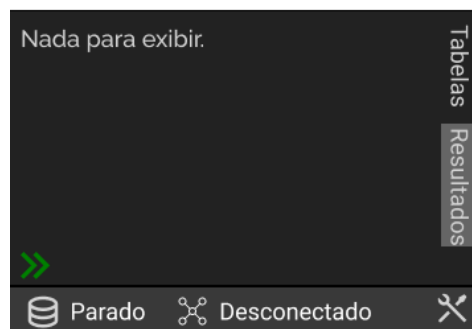


Figura 15 Mobile-SQL-lab.

## 5 Visão Lógica

### 5.1 Modelagem

No ANEXO I (p.39) está o Diagrama de Classes. Nos Diagramas 2-5 estão as diversas partes desse diagrama comentadas.

As classes foram usadas para representar unidades de trabalho para as ferramentas de modelagem lógica e conceitual. Note que não foi usado algum método estático, seguindo um estilo “procedural”. Isso foi feito porque funções não são serializáveis como JSON e a sincronização dos documentos entre os usuários usa JSON, de forma que se optou pelo uso de métodos estáticos para as rotinas comuns.

No Diagrama 2 está representado as classes mais básicas. São aquelas que representam as propriedades gráficas (CanvasDetails, Vertex, Dimension e Rect) e a classe Cardinalidade, que é usada para representar a relação entre duas figuras/entidades.

No Diagrama 3 está representado as classes principais associadas à modelagem conceitual e lógica (Entidade, Relação, Atributo, Generalização, Tabela, Coluna, etc.).

No Diagrama 4 está a classe Conexão e seus associados, que é usada para representar uma conexão entre duas figuras, que é um conjunto de segmentos.

No Diagrama 5 está a classe Store e seus associados, que são usados para representar o estado da modelagem em seu nível mais alto<sup>2</sup>

---

<sup>2</sup> A classe Store foi apresentada apenas para representar a estrutura do objeto usado para representar o documento, pois esse objeto tem estrutura equivalente a essa classe, contudo é instanciado sem ser por meio dessa.

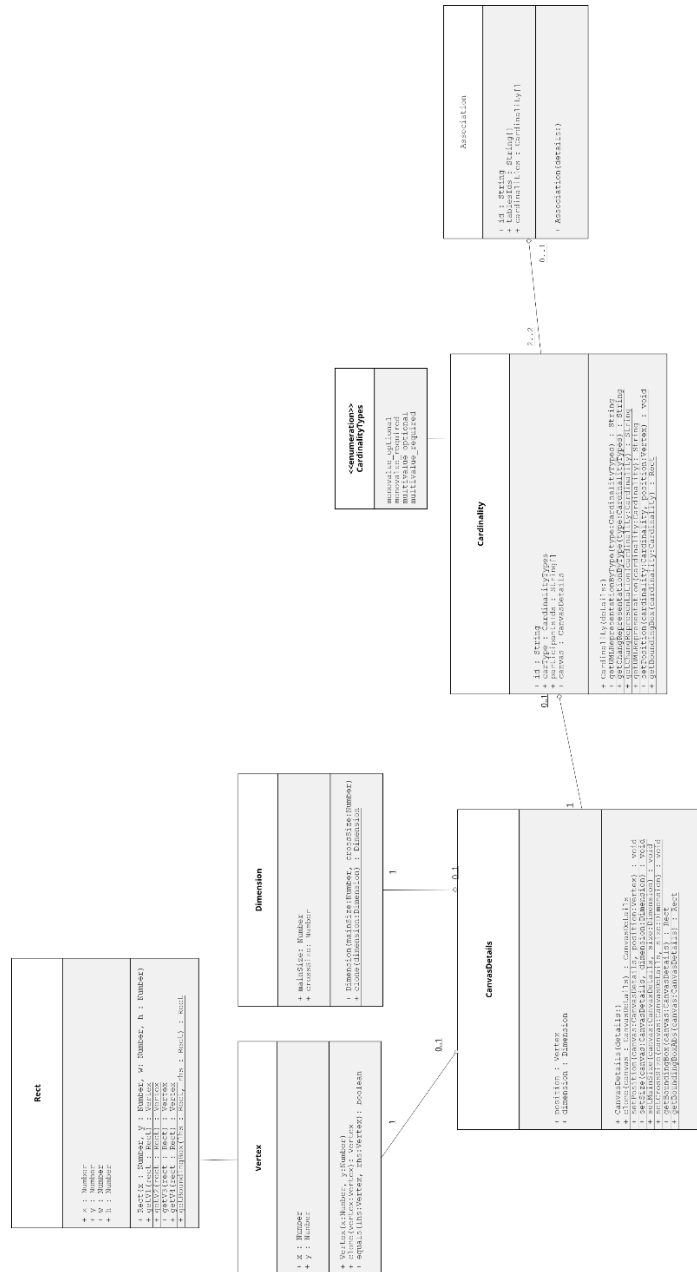


Diagrama 2 Diagramas de Clases Parte-1.



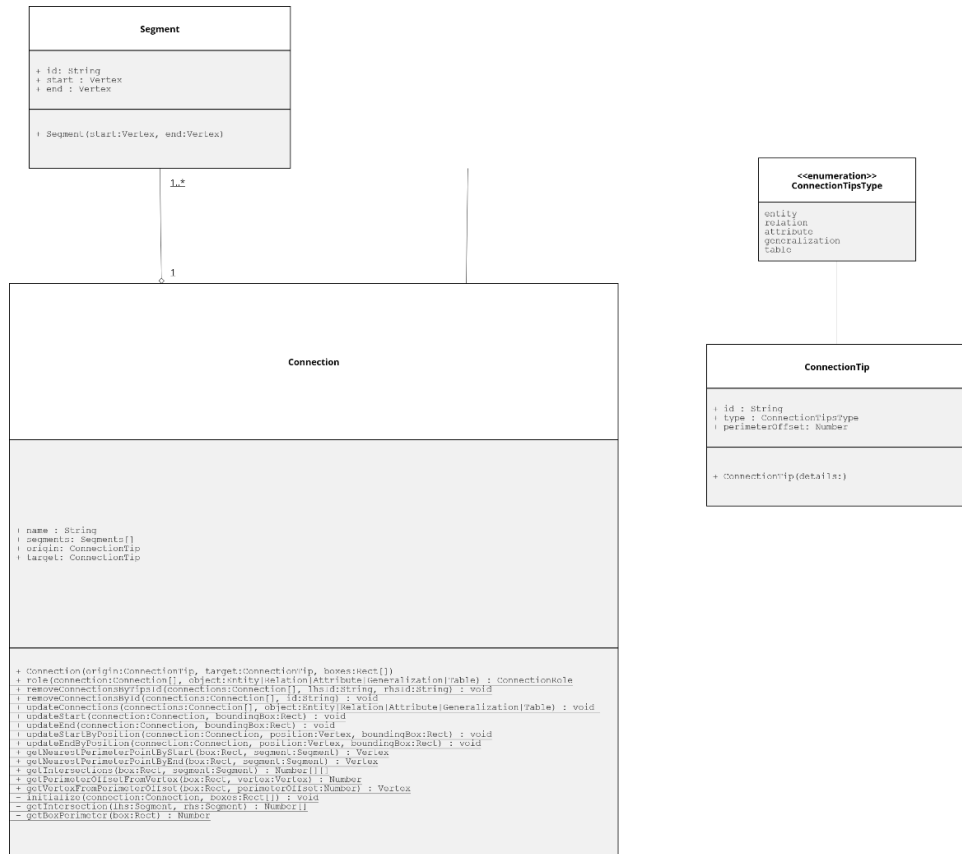


Diagrama 4 Diagramas de Classes Parte-3

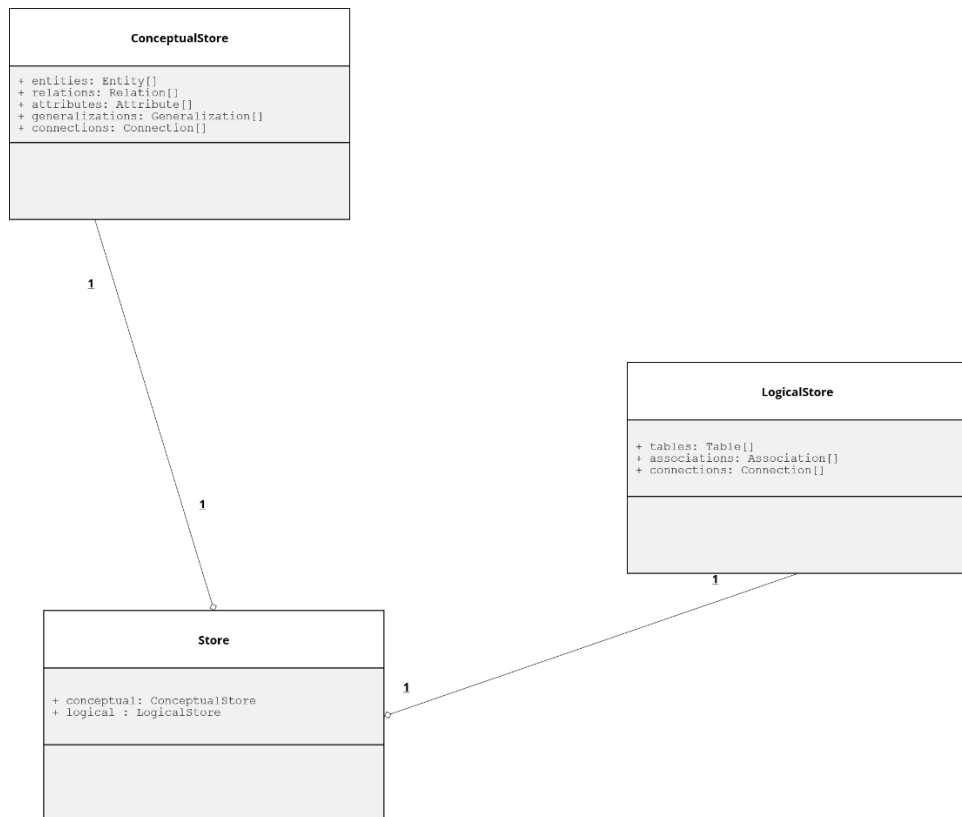


Diagrama 5 Diagramas de Classes Parte-4.



Para gerenciar o estado da interface das ferramentas de modelagem usou-se *statecharts (finite state machines)*, conforme representado no Diagrama 6.

A máquina usada foi do tipo paralela consistindo de duas regiões com estado estendido compartilhado: (i) conceitual, para a modelagem conceitual (ii) lógica, para modelagem lógica. Nas Tabelas 1-3 estão todos os eventos possíveis e a sua descrição.

Os eventos da Tabela 1 são independentes da região, enquanto que os eventos da Tabela 2 e 3 são acoplados, respectivamente, à região conceitual e à região lógica.

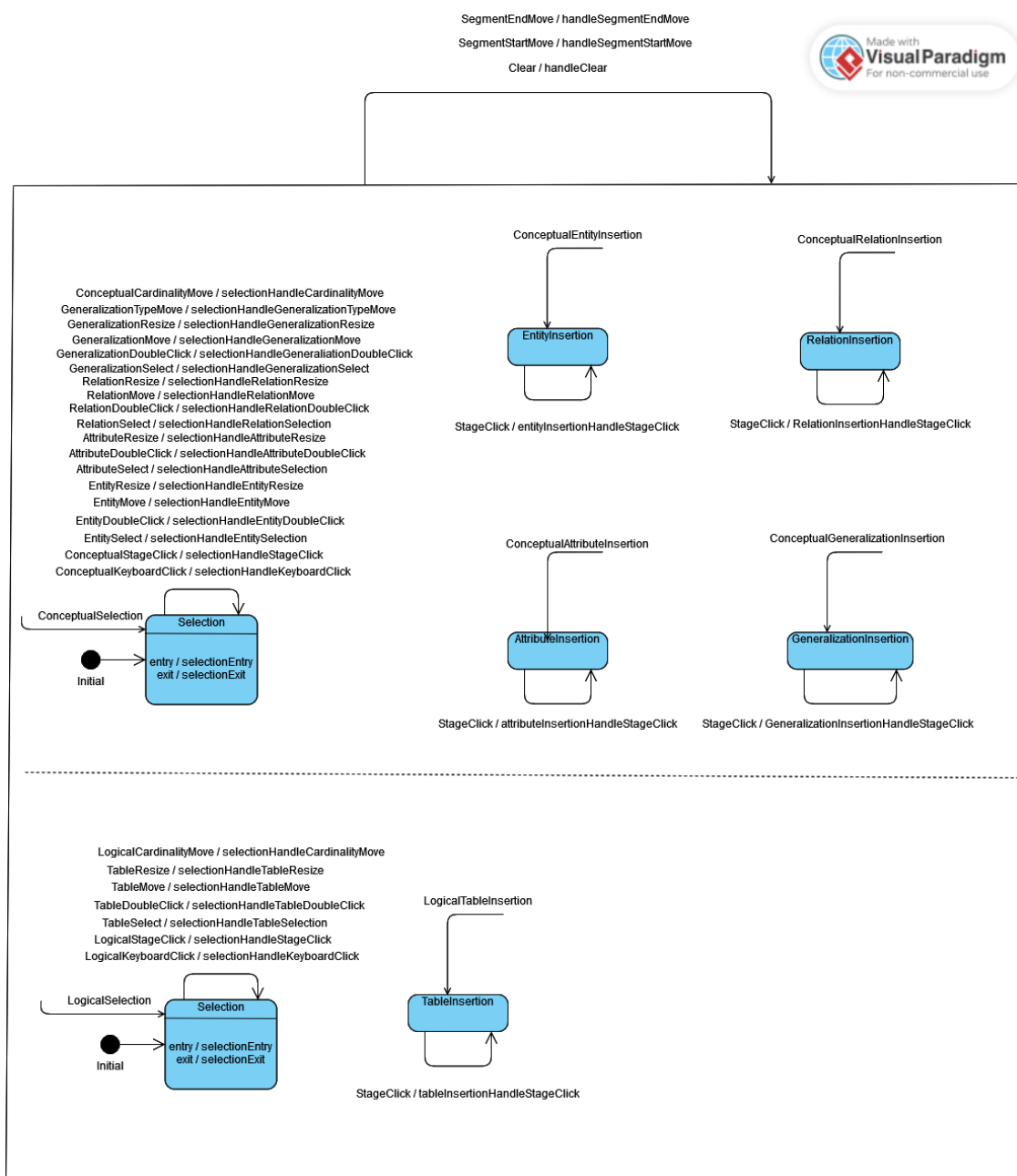


Diagrama 6 Diagrama de *finite state machine*.

Tabela 1 Eventos Gerais da Máquina de Estado.

<b>Nome</b>	<b>Descrição</b>
<b>CONCEPTUAL_SELECTION</b>	Modo de seleção para a modelagem conceitual.
<b>CONCEPTUAL_ENTITY_INSERTION</b>	Modo de inserção de entidade para a modelagem conceitual.
<b>CONCEPTUAL_RELATION_INSERTION</b>	Modo de inserção de relação para a modelagem conceitual.
<b>CONCEPTUAL_ATTRIBUTE_INSERTION</b>	Modo de inserção de atributo para a modelagem conceitual.
<b>CONCEPTUAL_GENERALIZATION_INSERTION</b>	Modo de inserção de generalização para a modelagem conceitual.
<b>LOGICAL_SELECTION</b>	Modo de seleção para a modelagem lógica.
<b>LOGICAL_TABLE_INSERTION</b>	Modo de inserção de tabela para a modelagem lógica.
<b>CLEAR</b>	Limpar todas as seleções do contexto.
<b>SEGMENT_START_MOVE</b>	Atualizar o vértice inicial de um segmento.
<b>SEGMENT_END_MOVE</b>	Atualizar o vértice final de um segmento.

Tabela 2 Eventos da Região conceitual da Máquina de Estado (Continua).

<b>Nome</b>	<b>Descrição</b>
<b>ENTITY_SELECT</b>	Usuário clicou em uma entidade.
<b>ENTITY_DOUBLE_CLICK</b>	Usuário clicou duplamente em uma entidade.
<b>ENTITY_RESIZE</b>	Usuário redimensionou uma entidade.
<b>ENTITY_MOVE</b>	Usuário moveu uma entidade.
<b>ATTRIBUTE_SELECT</b>	Usuário clicou em um atributo.
<b>ATTRIBUTE_DOUBLE_CLICK</b>	Usuário clicou duplamente em um atributo.
<b>ATTRIBUTE_RESIZE</b>	Usuário redimensionou um atributo.
<b>ATTRIBUTE_MOVE</b>	Usuário moveu um atributo.
<b>RELATION_SELECT</b>	Usuário clicou em uma relação.
<b>RELATION_DOUBLE_CLICK</b>	Usuário clicou duplamente em uma relação.
<b>RELATION_RESIZE</b>	Usuário redimensionou uma relação.
<b>RELATION_MOVE</b>	Usuário moveu uma relação.
<b>CONCEPTUAL_CARDINALITY_MOVE</b>	Usuário moveu uma cardinalidade no modelo conceitual.
<b>GENERALIZATION_SELECT</b>	Usuário clicou em uma generalização.
<b>GENERALIZATION_DOUBLE_CLICK</b>	Usuário clicou duplamente em uma generalização.

Tabela 2 Eventos da Região conceitual da Máquina de Estado (Conclusão).

Nome	Descrição
<b>GENERALIZATION_RESIZE</b>	Usuário redimensionou uma generalização.
<b>GENERALIZATION_MOVE</b>	Usuário moveu uma generalização.
<b>GENERALIZATION_TYPE_MOVE</b>	Usuário moveu o rótulo de título de uma generalização.
<b>CONCEPTUAL_KEYBOARD_CLICK</b>	Usuário pressionou o teclado no modelo conceitual.
<b>CONCEPTUAL_STAGE_CLICK</b>	Usuário clicou no Stage/Canvas no modelo conceitual.

Tabela 3 Eventos da Região lógica da Máquina de Estado (Continua).

Nome	Descrição
<b>TABLE_SELECT</b>	Usuário clicou em uma Tabela.
<b>TABLE_DOUBLE_CLICK</b>	Usuário clicou duplamente em uma Tabela.
<b>TABLE_MOVE</b>	Usuário moveu uma Tabela.
<b>TABLE_RESIZE</b>	Usuário redimensionou uma Tabela.

Tabela 3 Eventos da Região lógica da Máquina de Estado (Conclusão).

Nome	Descrição
<b>LOGICAL_CARDINALITY_MOVE</b>	Usuário moveu uma cardinalidade.
<b>LOGICAL_KEYBOARD_CLICK</b>	Usuário pressionou o teclado no modelo lógico.
<b>LOGICAL_STAGE_CLICK</b>	Usuário clicou no Stage/Canvas no modelo lógico.

Assim, nos Diagrama 7 e 8 estão representadas as duas formas de alterar o estado do document/store. Naquele, é a sequência comum para alteração por meio do viewport, enquanto nesse é por meio de uma das Tabs/Paineis.

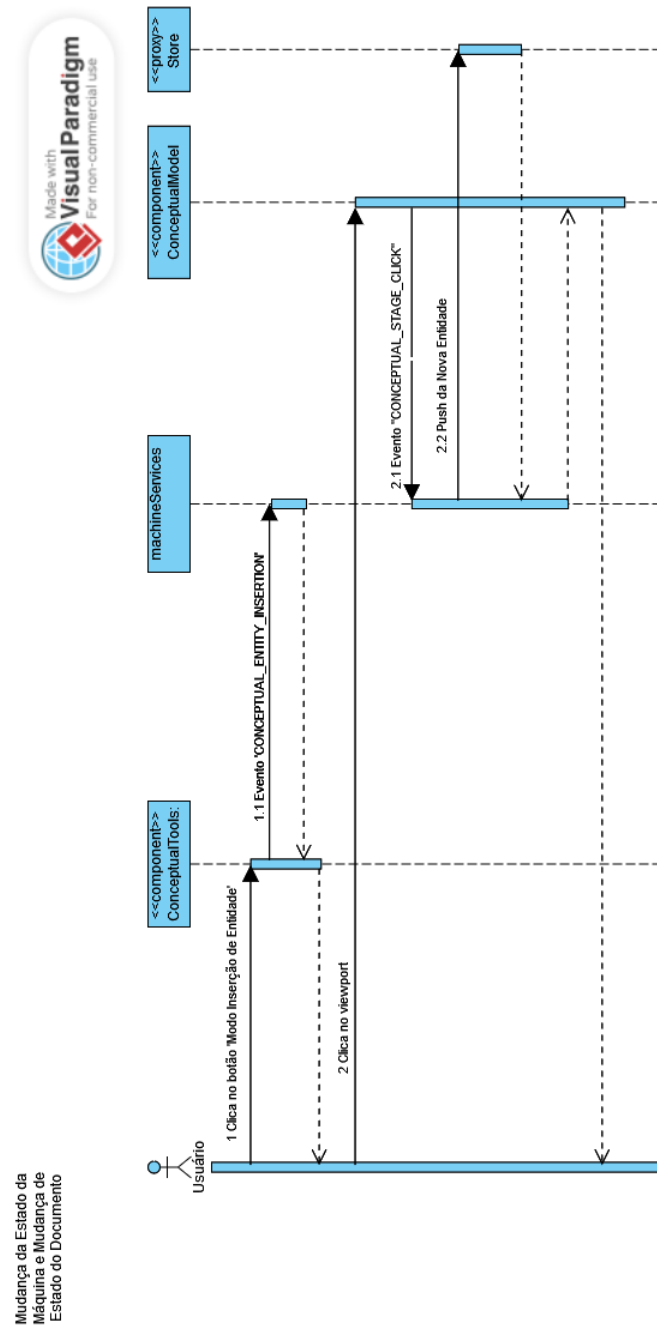


Diagrama 7 Diagrama de Sequência da Interação do usuário a partir do Viewport.

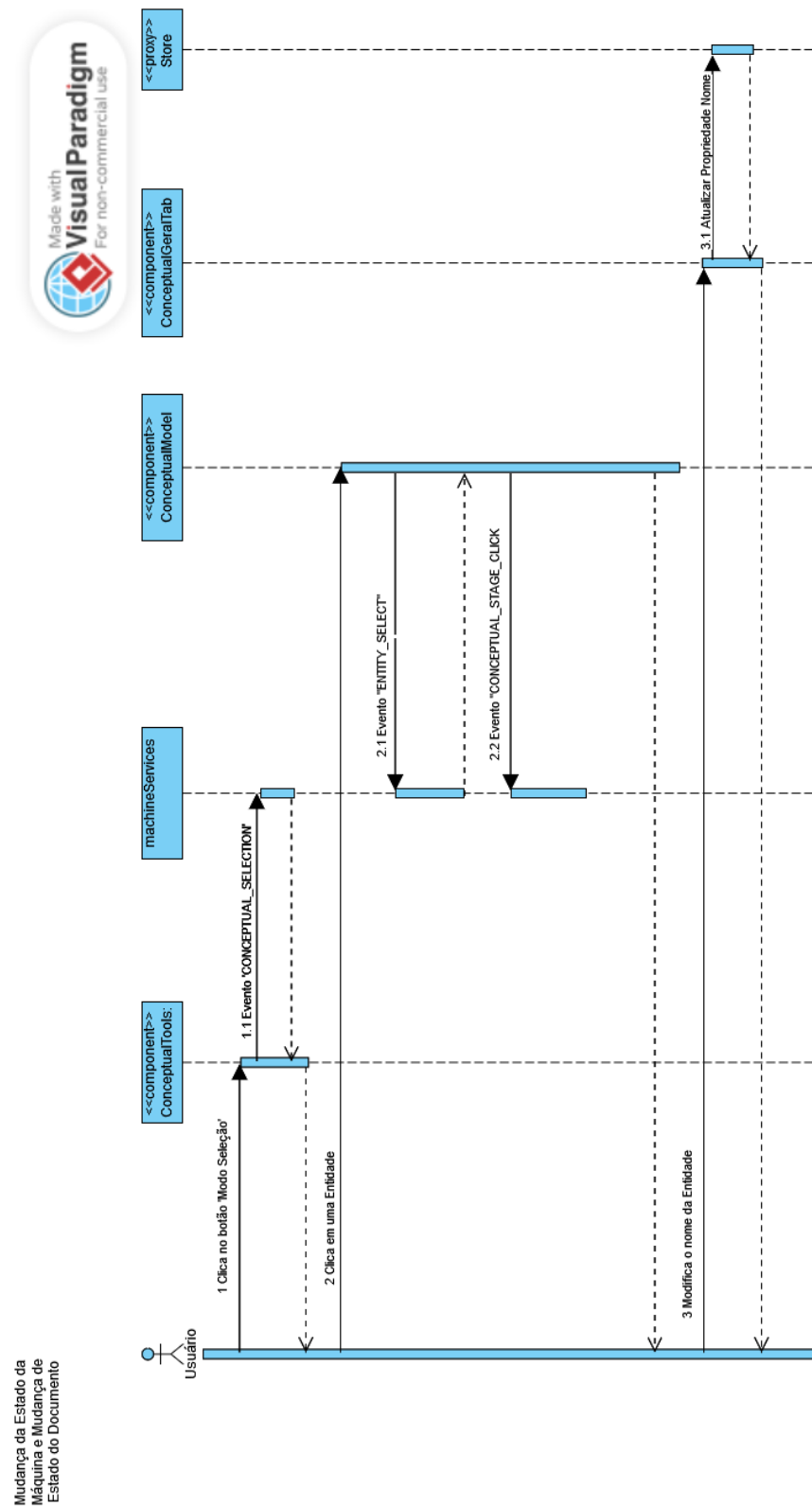


Diagrama 8 Diagrama de Sequência da Interação do usuário a partir de uma das Tabs.

## 5.2 Laboratório

No Diagrama 9 está representado a sequência de eventos para o usuário conseguir executar SQL Scripts.

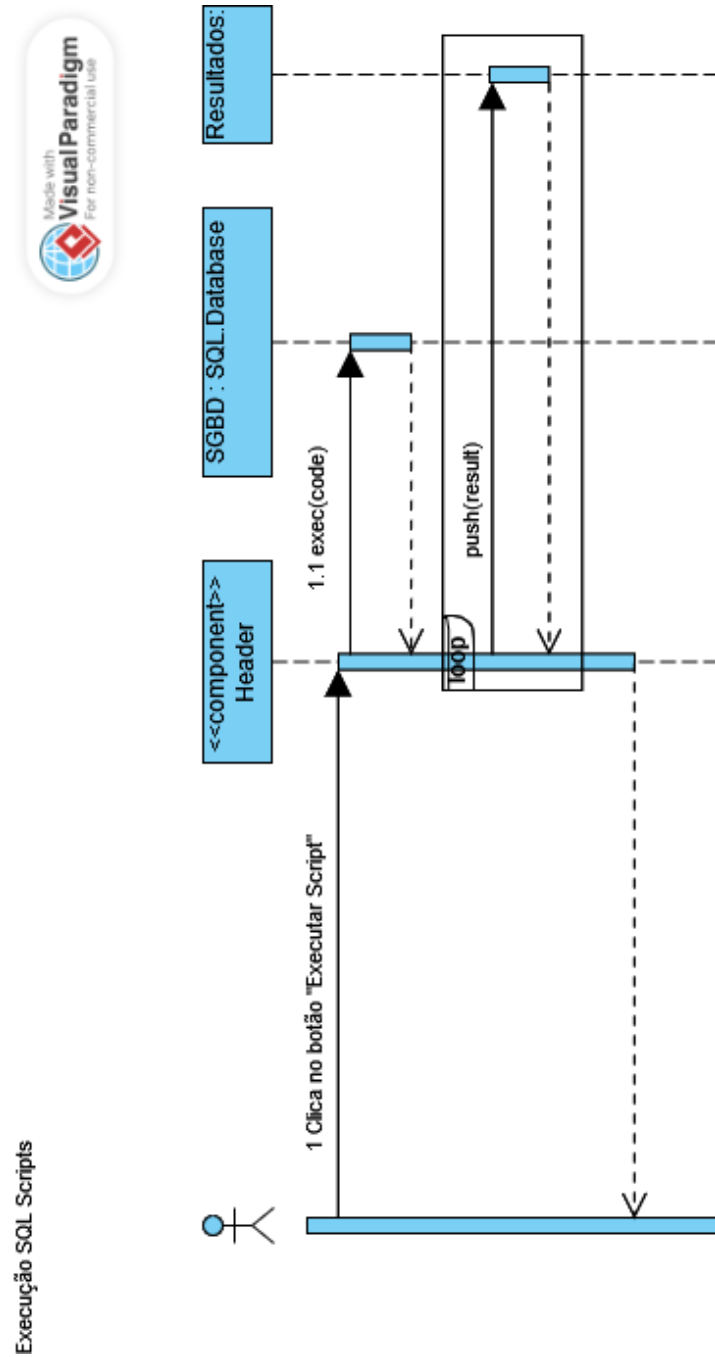


Diagrama 9 Diagrama de Sequência da execução de SQL Scripts.



## 6 Visão de Processos

No Diagrama 10 está representado a sincronização entre dois documentos que colaboram na mesma sessão por meio de um servidor de broadcast. Toda modificação do documento notifica o servidor de broadcast, que se comunica com o dispositivo do usuário por meio de *websocket*. Esse servidor, então, propaga essa modificação aos outros nós.

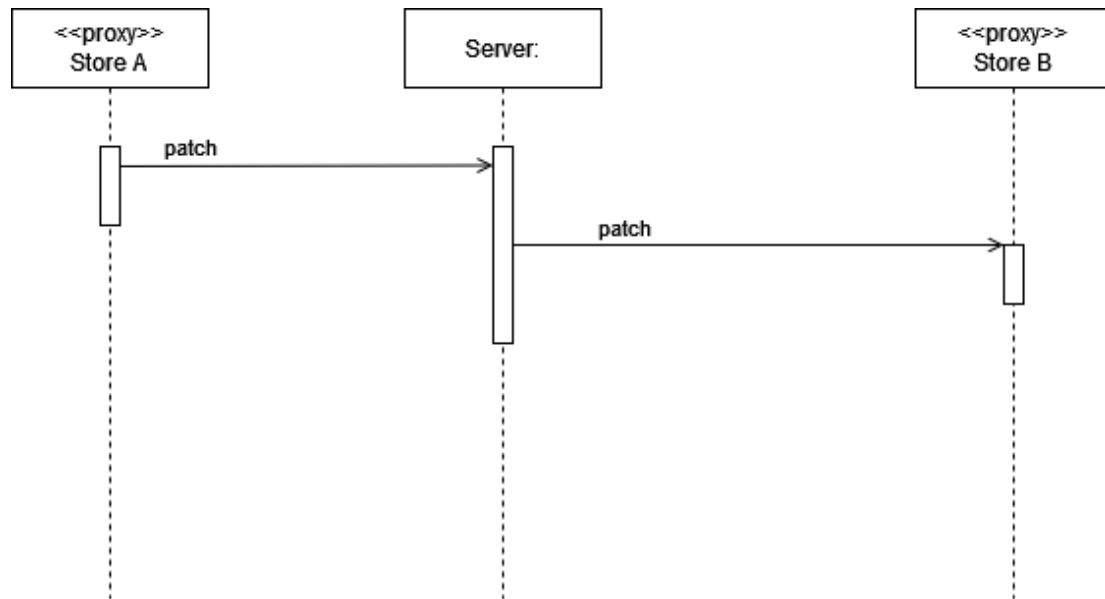


Diagrama 10 Diagrama de Processos

## 7 Visão de Implementação

No Diagrama 11 está a hierarquia de pacotes da implementação.

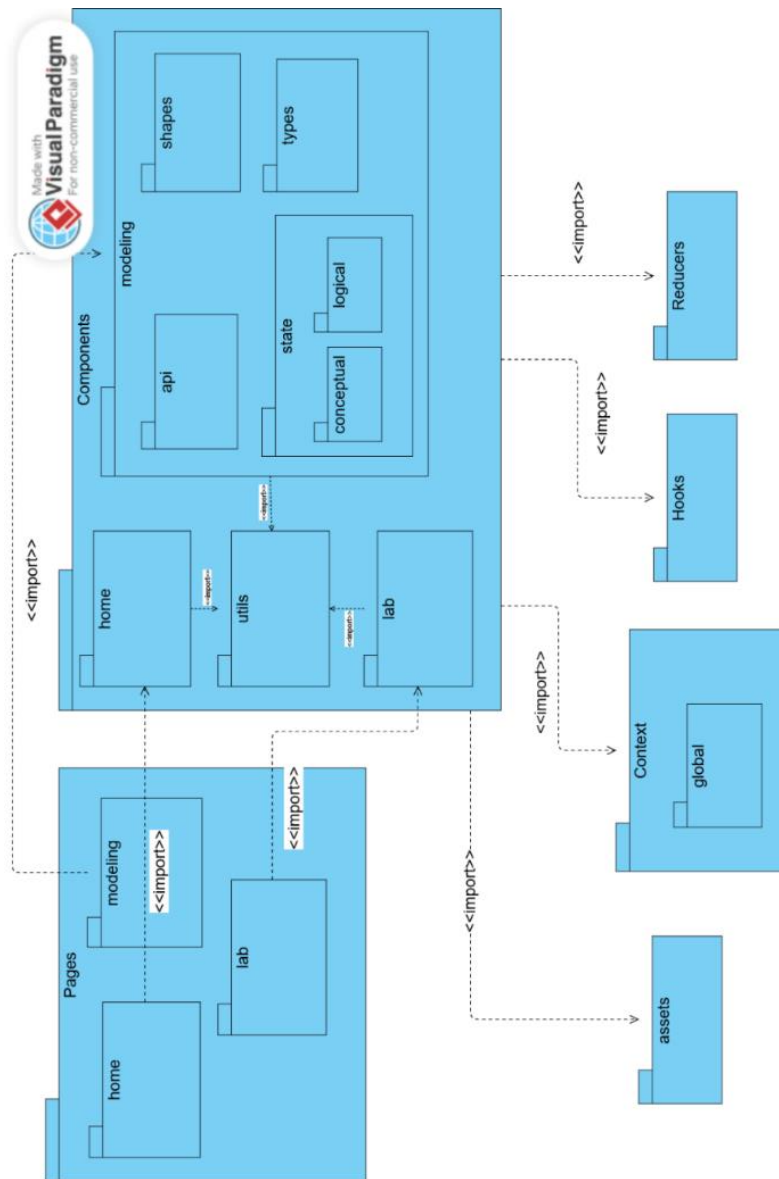


Diagrama 11 Diagrama de Pacotes.

## 8 Visão de Implantação

No Diagrama 12 está representada a topologia dos dispositivos que participam do sistema.

O sistema consiste de um servidor para prover os artefatos estáticos html/css/js (Application Server), que se comunica com o dispositivo do usuário por meio do protocolo *http*, e um servidor de broadcast para distribuir modificações em documentos de uma mesma sessão, que se comunica com o dispositivo do usuário por meio do protocolo *websocket*.

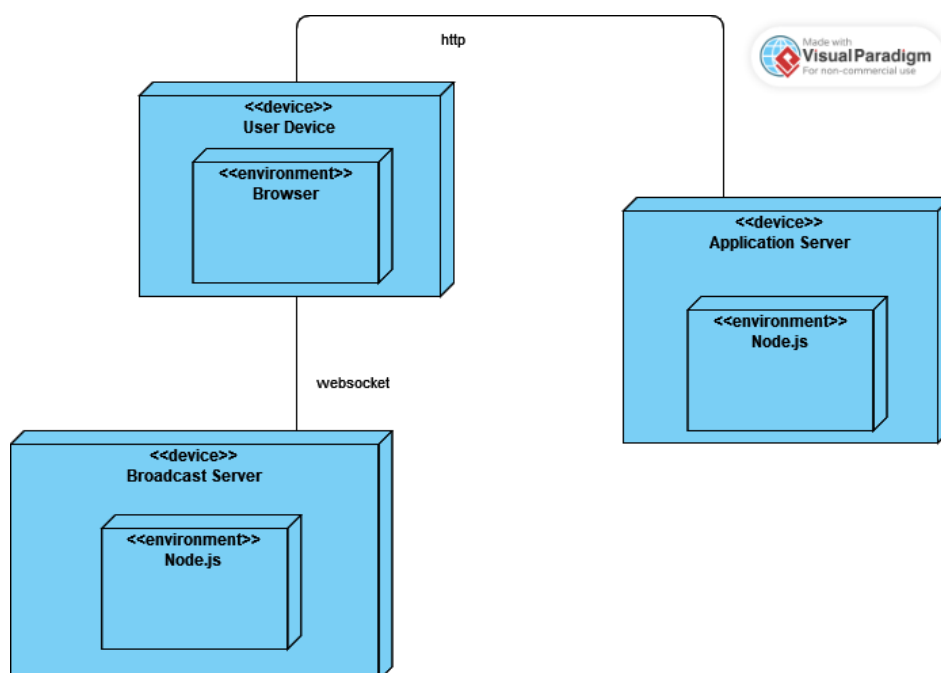


Diagrama 12. Diagrama de Implantação.

## 9 Testes

A aplicação não foi satisfatoriamente testada.

## **10 Problemas conhecidos**

- As transformações podem ter comportamento não-especificado;
- As conexões consistem de apenas um segmento;
- Os atalhos de teclado não foram implementados;
- A seleção do texto do visualizador de PDF não está funcionando corretamente;

## REFERÊNCIAS

HU, Chenglie. **An Introduction to Software Design** - Concepts, Principles, Methodologies, and Techniques. Springer, 2023.

## **ANEXOS**

## ANEXO I – Diagrama de Classes



Diagrama 13 Diagrama de Classes