

---

# **Projeto de Software**

---

**Engenharia de software I**  
**2022 – FACET/UFGD**

**Projeto: JOGO EDUCACIONAL PARA ENSINO  
DE BANCO DE DADOS**

**Equipe: Geilso Faria Rodrigues da Silva e Rodrigo  
Quinhone Veiga Durais**

# Conteúdo

<b>1</b>	<b>INTRODUÇÃO</b>	<b>3</b>
<b>2</b>	<b>MOTIVAÇÃO</b>	<b>3</b>
<b>3</b>	<b>VISÃO DA SOLUÇÃO</b>	<b>4</b>
<b>4</b>	<b>ESCOPO DA SOLUÇÃO</b>	<b>4</b>
4.1	PRINCIPAIS ENTREGÁVEIS	4
<b>5</b>	<b>LIMITES E RESTRIÇÕES DA SOLUÇÃO</b>	<b>4</b>
<b>6</b>	<b>DESCRIÇÃO DOS USUÁRIOS</b>	<b>5</b>
<b>7</b>	<b>CRONOGRAMA INICIAL</b>	<b>5</b>
7.1	PRÓXIMAS ATIVIDADES	5
<b>8</b>	<b>Identificação dos requisitos e regras do negócio</b>	<b>6</b>
8.1	ESPECIFICAÇÃO	6
8.1.1	<i>Identificação dos requisitos funcionais do sistema</i>	6
8.1.2	<i>Identificação dos requisitos não funcionais do sistema</i>	6
8.1.3	Identificação das regras de negócio do sistema	6
8.1.3	<i>Identificação dos casos de uso</i>	6
<b>9</b>	<b>ATORES</b>	<b>6</b>
<b>10</b>	<b>CASOS DE Uso</b>	<b>7</b>
	<i>&lt;Diagrama de casos de uso&gt;</i>	7
	<i>[UC01] &lt;Realizar login&gt;</i>	8
	<i>[UC02] &lt;Postar perguntas&gt;</i>	9
	<i>[UC03] &lt;Postagem de exercícios &gt;</i>	10
<b>11</b>	<b>DIAGRAMA DE CLASSES</b>	<b>11</b>
<b>12</b>	<b>DIAGRAMA DE SEQUÊNCIA</b>	<b>12</b>
<b>13</b>	<b>DIAGRAMA DE ATIVIDADES</b>	<b>13</b>
<b>14</b>	<b>PROTÓTIPO</b>	<b>15</b>
<b>15</b>	<b>REFERÊNCIAS</b>	<b>27</b>

## 1 Introdução

Este documento tem por objetivo demonstrar o projeto “JOGO EDUCACIONAL PARA ENSINO DE BANCO DE DADOS” referente ao trabalho de conclusão da disciplina de Engenharia de Software I. Nas próximas seções, serão apresentados os principais problemas que motivam a realização deste trabalho, as necessidades principais de clientes/usuários identificadas, os limites e restrições da solução e, por fim, o cronograma-macro para o projeto. O objetivo do documento é fornecer aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação da aplicação.

## 2 Motivação

A disciplina de Banco de Dados trata de muitos conceitos dentre eles descrição do minimundo, modelagem de dados, gatilhos, asserções, visões, normalização e outros, de tal maneira que estes criem algum sentido e representem corretamente algum aspecto do mundo real. E muitas vezes, os acadêmicos acabam tendo certas dificuldades para tal compreensão. Visando isso, o Jogo Educacional Laban para Ensino de Banco de Dados atuará como um forma de facilitar e ajudar os alunos que estejam com dificuldades em aprender esta respectiva disciplina. Sendo assim o jogo “Laban”, será uma boa forma de passar o tempo e aprender.

Abaixo pode-se observar a descrição do problema identificado, bem como o projeto desenvolvido impactará, os stakeholders, bem como estes são afetados e por fim a solução proposta.

<b>O problema é...</b>	Os acadêmicos muitas vezes possuem dificuldades em aprender alguns conceitos da disciplina de Banco de Dados. Dentre eles, descrição do minimundo, modelagem de dados, gatilhos, asserções, visões, normalização e outros. Visando isso, o Jogo Educacional LabBan para Ensino de Banco de Dados atuará como uma forma de facilitar e ajudar no aprendizado do aluno que esteja cursando esta disciplina.
<b>Que afeta...</b>	A equipe de desenvolvimento, que será influenciada pela construção da aplicação, os alunos da Disciplina de Banco de Dados, que poderão aprender sobre o conteúdo e sanar dúvidas, professores, que poderão propor o uso da ferramenta em suas disciplinas, bem como, qualquer outro indivíduo que utilizar do aplicativo.
<b>O impacto disto é...</b>	Acadêmicos que estão cursando a disciplina de Banco de Dados poderão testar seus conhecimentos e aprender através do uso da aplicação.
<b>A solução seria...</b>	Desenvolvimento do software de acordo com o modelo cross-platform (Android e IOS) e utilizando da linguagem React Native para construção da aplicação.

### 3 Visão da Solução

Desenvolver uma aplicação para auxiliar acadêmicos da Disciplina de Banco de Dados a fim de facilitar no desenvolvimento do indivíduo sobre o tema.

### 4 Escopo da Solução

Nessa seção, estão descritas as principais necessidades para a aplicação a ser desenvolvida.

Necessidades	Categoria
1. Ter um perfil de usuário, ou seja, um login.	essencial
2. Responder às questões sobre a disciplina de Banco de Dados.	essencial
3. Visualizar o progresso de usuário dentro do jogo.	importante
4. Responder às questões de acordo com temas específicos da disciplina de Banco de Dados	importante
5. Visualizar um ranking de usuários com maiores pontuações	desejável
6. Receber alertas de novas questões postadas na plataforma	desejável

#### 4.1 Principais entregáveis

1. Documentação do software, com os diagramas de classes, casos de uso, sequência e atividades.
2. Documento de requisitos do sistema.
3. Protótipos executáveis.
4. Aplicativo funcionando para dispositivos Android e IOS.
5. Relatório sobre o funcionamento do sistema.
6. Relatório do acompanhamento do uso do aplicativo durante o período de 1 mês.

### 5 Limites e Restrições da Solução

Por se tratar de um jogo que verifica o nível de conhecimento dos acadêmicos da disciplina de Banco de dados, uma necessidade importante seria a verificação por parte dos professores dos resultados obtidos por seus alunos, tanto para que estes possam estar cientes e sanar dúvidas de seus alunos em sala de aula, como também para eventuais bonificações. Entretanto, devido ao projeto estar na fase de desenvolvimento inicial e neste momento o foco principal do projeto é sua estruturação, manter a aplicação funcionando e os alunos testando os seus conhecimentos será um fator a ser contemplado futuramente.

Ademais, outra necessidade que não será incorporada ao sistema no momento é a integração com os demais sistemas da universidade, tanto para que certas informações não precisem ser recuperadas novamente para uso na aplicação quanto para o compartilhamento de informações dentro da comunidade acadêmica sobre o assunto. Este projeto requer maiores recursos financeiros e pessoais e neste momento tal fato não é o foco principal da construção deste aplicativo.

Por fim, outra necessidade não contemplada neste momento, é que docentes pudessem contribuir com a sugestão de questões para serem postadas na aplicação, tal ato exigiria uma

confirmação de usuários, verificação de padrões de questões, ou seja, uma verificação se o usuário é mesmo um professor e se a questão postada por estes atendem aos requisitos da aplicação. Porém, inicialmente esta função será realizada apenas pelos administradores da aplicação devido a exigência de uma maior complexidade de software, que no momento não é possível.

## 6 Descrição dos Usuários

O padrão de usuários esperados para uso da aplicação são acadêmicos dos cursos de Engenharia de Computação e Sistemas de Informação da Universidade Federal da Grande Dourados, independentemente de sua idade ou gênero.

Os principais problemas enfrentados pelos acadêmicos destes cursos da UFGD que podem limitar a sua produtividade quanto ao assunto geralmente estão relacionados com a falta de tempo para um melhor entendimento sobre o assunto. Para tal, a aplicação visa que estes possam aprender de modo rápido, seguro e eficaz.

## 7 Cronograma Inicial

Esta seção apresenta um cronograma inicial destacando quais serão os principais marcos do projeto, o que conterão e quando eles ocorrerão.

Fases/Marcos do projeto	Entregáveis	Data de início prevista	Data de término prevista
Análise de requisitos	Documento de requisitos do sistema	29/07/2022	04/08/2022
Documentação do software, com os principais diagramas.	Diagramas de classes, casos de uso, sequência e atividades.	05/08/2022	15/08/2022
Proposta de sistema, com protótipos	Protótipos executáveis do sistema	16/08/2022	23/08/2022
Software finalizado	Aplicativo funcionando para dispositivos Android e IOS.	24/08/2022	23/09/2022

### 7.1 Próximas atividades

Esta subseção apresenta o cronograma das próximas atividades a serem realizadas, antes da elaboração do cronograma detalhado para todo o projeto.

Fases	Entregáveis	Data de início prevista	Data de término prevista
Relatório sobre o funcionamento do sistema.	Relatório sobre o funcionamento geral do sistema.	24/09/2022	30/09/2022
Acompanhamento pós desenvolvimento	Relatório do acompanhamento do uso do aplicativo	24/09/2022	25/10/2022

	durante o período de 1 mês.		
--	-----------------------------	--	--

## 8 Identificação dos requisitos e regras do negócio

A seguir são descritos os principais requisitos do sistema bem como os casos de uso derivados.

### 8.1.1 Identificação dos requisitos funcionais do sistema

A seguir são listados os principais requisitos funcionais do sistema.

- [RF01] - O sistema deve cadastrar usuários.
- [RF02] - O sistema deve permitir postar perguntas .
- [RF03] - O sistema deve registrar as respostas selecionadas.
- [RF04] - O sistema deve mostrar as respostas das perguntas.
- [RF05] - O sistema deve conter um ranking geral de pontuação por usuários.
- [RF06] - O sistema deve conter um nível individual de pontuação por usuários.
- [RF07] - O sistema deve fornecer exercícios e questões para serem resolvidos.
- [RF08] - O sistema deve fornecer níveis de dificuldade mediante às questões e exercícios.

### 8.1.2 Identificação dos requisitos não funcionais do sistema

A seguir são listados os principais requisitos não funcionais do sistema.

- [RNF01] - O sistema deve controlar o acesso às funcionalidades de postar perguntas sendo que estas devem ser feitas apenas por administradores.
- [RNF02] - O sistema deve estar disponível para o uso nas plataformas Android e IOS.

### 8.1.3 Identificação das regras de negócio do sistema

A seguir são listados as principais regras de negócio do sistema.

- [RNN01] - Apenas será mostrado no ranking global usuários que acertarem a partir de X questões.
- [RNN02] - O sistema deverá fornecer uma aba para propor uma solução diferente em casos no qual algum indivíduo discorda da resposta.
- [RNN03] - Uma resposta não poderá conter mais de uma seleção de alternativa.
- [RNN04] - Um usuário poderá usar o sistema sem realizar um cadastro prévio

### 8.1.3 Identificação dos casos de uso

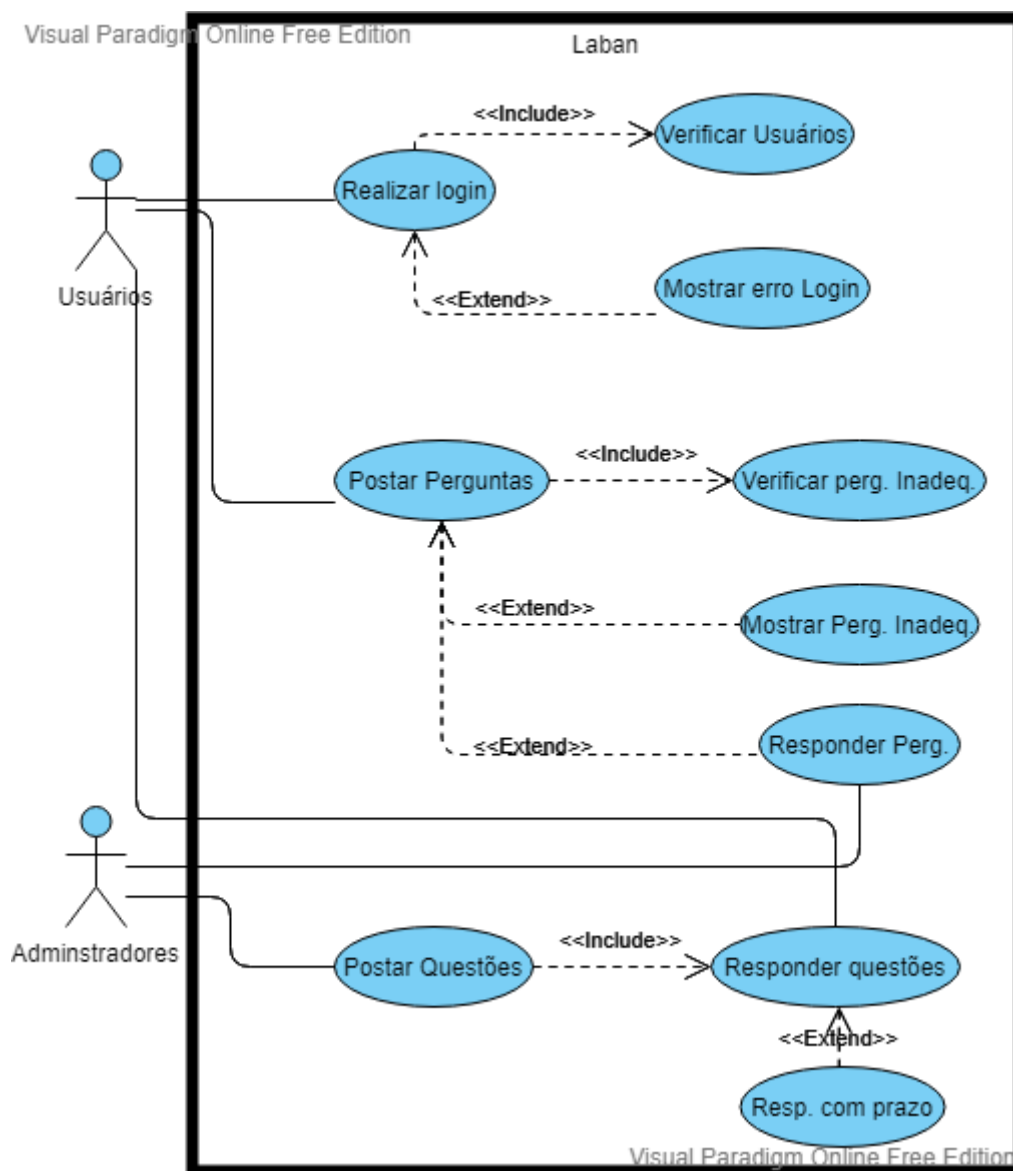
A tabela abaixo descreve brevemente cada ator da aplicação.

Ator	Descrição
Usuários	Se trata dos acadêmicos que estão cursando a disciplina de Banco de Dados dos cursos de Engenharia de Computação e Sistemas de Informação.
Administradores	Se trata dos administradores da aplicação que terão permissões especiais sobre o software.

## 9 Casos de Uso

Essa seção apresenta todos os requisitos funcionais da aplicação, especificados como casos de uso.

### Diagrama de casos de uso



### [UC01] Realizar Login

<b>Prioridade:</b>	X	Essencial		Importante		Desejável
<b>Ator(es):</b>	Usuários					

**Descrição:** Os usuários após realizarem o cadastro na aplicação entrarão com seus respectivos cadastro de usuário e senha, após isso os seus dados serão verificados.

**Pré-condições:**

**<Download>**

O indivíduo deve baixar o aplicativo.

**Pós-condições:**

**<Tutorial>**

Após o login o usuário poderá usar todas as funcionalidades do sistema.

**Fluxo principal**

1. O usuário entra no aplicativo;
2. O sistema solicita do usuário seus cadastro e sua senha;
3. O usuário digita as informações.
4. O sistema verifica as informações do usuário;
5. O sistema mostra a tela inicial do aplicativo, com todas as funções.

**Fluxos alternativos**

**[FA 001]**

1. O usuário opta por não logar no aplicativo;
2. O sistema emite uma mensagem confirmando se este que mesmo sair da aplicação;
3. O usuário confirma se o usuário não pretende continuar o login.
4. O sistema finaliza sua operação

**Fluxos de erro**

**[FE 001]**

1. O usuário digita dados inválidos;
2. O sistema emite um alerta informando que os dados não conferem;
3. O sistema solicita que o usuário digite novamente os dados;
4. O usuário digita os dados corretos;
5. O sistema verifica se as informações de login estão corretas;
6. O sistema mostra a tela inicial do aplicativo, com todas as funções.

**[UC02] Postar perguntas**

<b>Prioridade:</b>		Essencial	X	Importante		Desejável
--------------------	--	-----------	---	------------	--	-----------



<b>Ator(es):</b>	Usuários e administradores
------------------	----------------------------

**Descrição:** O sistema permite aos usuários do aplicativo que postem perguntas caso haja alguma dúvida sobre o assunto ou determinado exercício. Tanto os administradores quanto outros usuários poderão responder às perguntas.

**Pré-condições:**

<Login>

Para que haja a postagem de perguntas é necessário que o usuário esteja logado.

**Pós-condições:**

<Visibilidade das perguntas>

Após a postagem das perguntas a mesma ficará disponível para que outros usuários vejam e possam responder, bem como, os administradores.

**Fluxo principal**

1. O usuário loga no sistema;
2. O usuário entra na aba de postagem de perguntas / dúvidas;
3. O usuário escreve um pergunta e envia a solicitação para postagem;
4. O sistema verifica se o conteúdo da pergunta é adequado.
5. O sistema posta a pergunta na aba de dúvidas
6. A pergunta ficará visível para todos que entrarem na aba de postagem de perguntas /dúvidas;
7. Administradores respondem as perguntas;
8. O usuário visualiza a resposta de sua dúvida.

**Fluxos alternativos**

**[FA 001]**

1. O usuário loga no sistema;
2. O usuário entra na aba de postagem de perguntas / dúvidas;
3. O usuário escreve uma pergunta inadequada e envia a solicitação para postagem;
4. O sistema identifica como inadequada a pergunta e não a publica;
5. O sistema emite uma mensagem ao usuário informando que o conteúdo da pergunta é inadequado.
6. O sistema emite uma mensagem informando que o usuário responsável ficará um determinado período de tempo sem poder postar perguntas.

**Fluxos de erro**

**[FE 001]**

1. O usuário loga no sistema;
2. O usuário entra na aba de postagem de perguntas / dúvidas;
3. O usuário escreve um pergunta e envia a solicitação para postagem;
4. O sistema identifica um erro interno e não posta a pergunta;
5. O sistema informará ao usuário que a pergunta não foi postada;
6. O sistema aconselhará o usuário a postar a pergunta novamente mais tarde.

**[UC03] Postagem de exercícios**

<b>Prioridade:</b>	<b>X</b>	Essencial		Importante		Desejável
<b>Ator(es):</b>	Administradores					

**Descrição:** O sistema irá disponibilizar periodicamente exercícios para que os usuários do aplicativo respondam as questões de assuntos específicos da disciplina de Banco de Dados.

**Pré-condições:**

<Login>

O deverá estar logado no sistema.

**Pós-condições:**

<Visualizar respostas>

Após responder um exercício o usuário terá acesso ao gabarito das questões.

**Fluxo principal**

- 1.Os administradores postam uma questão
- 2.O sistema identifica a categoria da pergunta e atualiza o cadastro em seu banco de questões.
3. O usuário entra na área de teste
4. O usuário inicia um novo teste que poderá conter a nova questão.
5. O usuário resolve a(s) questão(ões);
6. O sistema irá registrar a resposta enviada;
7. O sistema mostra ao usuário sua pontuação;
8. O sistema exibe o gabarito das questões.

**Fluxos alternativos**

**[FA 001]**

1. Os administradores postam uma questão com prazo para ser resolvida;
2. O sistema notifica os usuários que uma questão especial foi postada;
3. O usuário abre a notificação;
4. O usuário loga no aplicativo;
5. O usuário resolve a questão, se resolvida dentro do prazo, acarretará no ganho de pontos extras;
6. O sistema irá registrar a resposta enviada
7. Posteriormente, o sistema enviará uma lista dos usuários que conseguiram resolver a respectiva questão dentro do prazo.
8. O sistema atribui pontos aos usuários de acordo com a velocidade em que a questão for respondida. Quanto mais rápido, mais pontos;
9. O sistema mostra ao usuário sua pontuação;
10. O sistema exibe o gabarito das questões.

**Fluxos de erro**

**[FE 001]**

1. O sistema identifica divergências no registro de pontuação dos exercícios;
3. O sistema identifica a questão e contacta os administradores.
4. Os administradores corrigem a divergência e enviam os dados ao sistema.
5. O sistema notifica o usuário e mostra a sua nova pontuação obtida na questão.

## 10. Diagramas de Classes

Esta seção provê um maior detalhamento dos componentes da aplicação, apresentando suas classes através de diagramas.

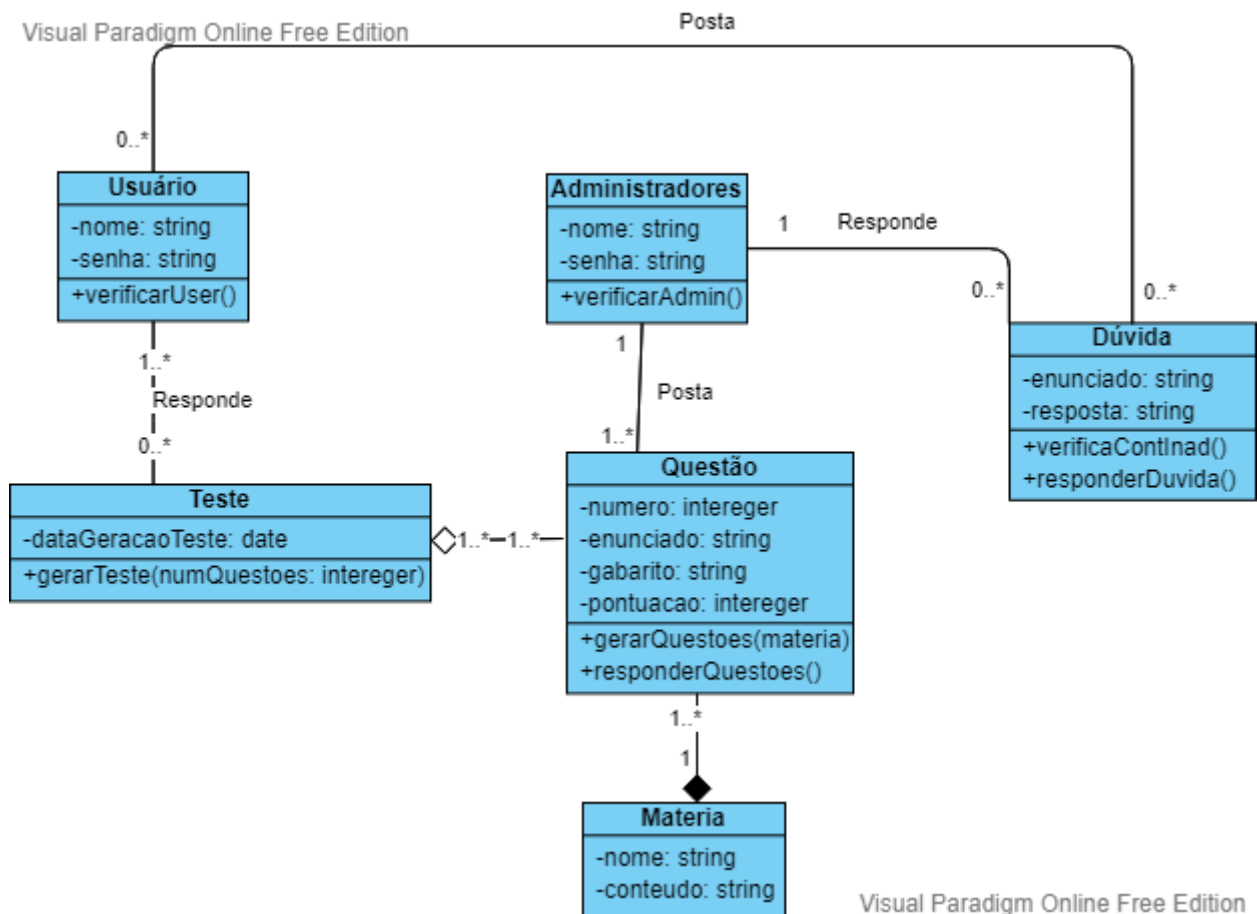


Figura 2. Diagrama de Classes do Sistema

No seguinte diagrama temos as respectivas classes com seus respectivos atributos, métodos, cardinalidade e relacionamentos.

**Usuário:** Classe referente ao usuários da aplicação com o relacionamento (associação) entre as classes *teste* e *Dúvida*.

- Atributos: Atributos privados, *nome* do tipo string, que serve como identificador do usuário e *senha* do tipo string, que serve como uma chave para uso da aplicação.
- Método: Método público *verificarUser()*, que verifica se os dados de usuário e senha conferem
- Cardinalidade: Um ou muitos usuários podem responder a nenhum ou muitos testes e nenhum ou muitos usuários postam nenhuma ou muitas dúvidas.

**Administradores:** Classe referente aos administradores do aplicativo, com uma associação entre as classes *Questão* e *Dúvida*.

- Atributos: Atributos privados, *nome* do tipo string, que serve como identificador do administrador e *senha* do tipo string, que serve como uma chave para uso da aplicação.
- Método: Método público *verificarAdmin()*, que verifica se os dados de usuário e senha do administrador conferem.
- Cardinalidade: Um administrador posta uma ou muitas questões. Um administrador responde a uma ou mais dúvidas.

**Teste:** Classe referente a cada teste gerado por matérias específicas e com certo conjunto de questões.

- Atributo: Atributo privado *dataGerecaoTeste* do tipo date, que serve como identificador de testes gerados conforme cada dia.
- Método: Método público *gerarTeste(numQuestoes: integer)*, usado para gerar um teste de acordo com o número de questões, sendo que uma questão pode existir fora de um teste (agregação).
- Cardinalidade: Um teste pode conter uma ou muitas questões e um teste pode ser respondido por um ou muitos usuários.

**Questão:** Classe referente a cada questão, que só poderá existir se existir alguma matéria sobre o tema (composição).

- Atributos: Atributos privados *numero* do tipo integer, que serve como identificador de cada questão, *enunciado* do tipo string, que contém o enunciado de cada questão, *gabarito* do tipo string, que contém as respostas das questões e *pontuacao* do tipo integer, que contém o número total de pontos que cada questão representa.
- Métodos: Métodos públicos *gerarQuestoes(materia)* de acordo com uma matéria específica e o método *responderQuestao()* referente ao ato de responder uma questão.
- Cardinalidade: Uma ou muitas questões são referentes apenas a uma matéria.

**Matéria:** Classe referente a cada assunto da disciplina de Banco de dados.

- Atributos privados, *nome* do tipo string, que serve como identificador de cada conteúdo e *conteudo* do tipo string, que contém toda a informação sobre o tema.
- Cardinalidade: Uma matéria pode conter uma ou muitas questões.

**Dúvida:** Classe referente às dúvidas dos usuários e respostas dos administradores.

- Atributos: Atributos privados, *enunciado* do tipo string, referente ao conteúdo da dúvida e *resposta* do tipo string, referente a resposta obtida da pergunta.
- Métodos: Método público *verificarContInad()* para verificar se uma dúvida não possui conteúdo inadequado e o método *responderDuvida()* referente a ação de responder uma dúvida.
- Cardinalidade: Nenhuma ou muitas dúvidas são respondidas por uma ou muitas pessoas

## 10 Diagrama de Sequência

Esta seção descreve as interações que acontecem entre os atores e/ou objetos do sistema. Detalhando como eles colaboram.

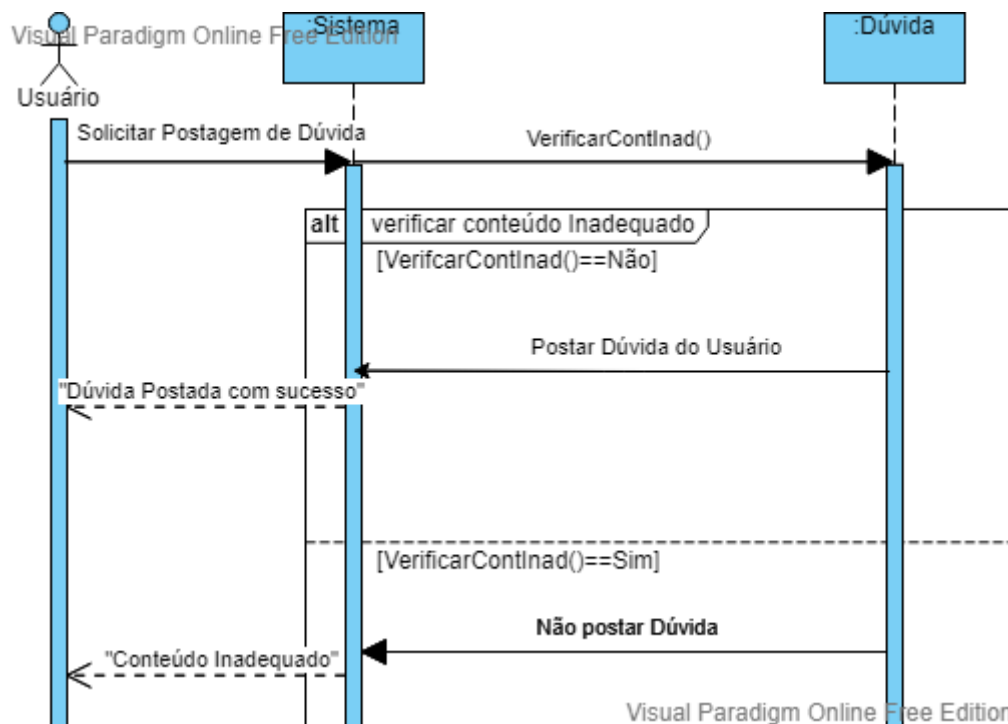


Figura 3. Diagrama de Sequência para postagem de dúvidas

O diagrama apresentado na figura 3, descreve o caso de uso da postagem de dúvidas por parte dos usuários, onde o usuário após escrita a sua dúvida solicita ao sistema para esta seja postada, posteriormente o sistema faz a chamada ao método *verificarContInad()* que verifica se o conteúdo da mensagem não é inadequado. Após isso, se o conteúdo da mensagem for adequado o sistema posta a dúvida do usuário e emite uma mensagem informando que sua dúvida foi postada, senão, se o conteúdo da dúvida for inadequado o sistema não posta e emite uma mensagem informando que o conteúdo é inadequado.

No cenário principal: O usuário escreve sua dúvida, faz a solicitação para postagem, o sistema verifica o conteúdo e se este for adequado posta a dúvida.

Um possível caminho alternativo: Se o usuário escrever uma questão com conteúdo inadequado o sistema verifica, não realiza a postagem e notifica o usuário sobre tal ação.

## 11 Diagrama de Atividades

O objetivo desta seção é apresentar o fluxo de execução dos processos que compõem a aplicação.

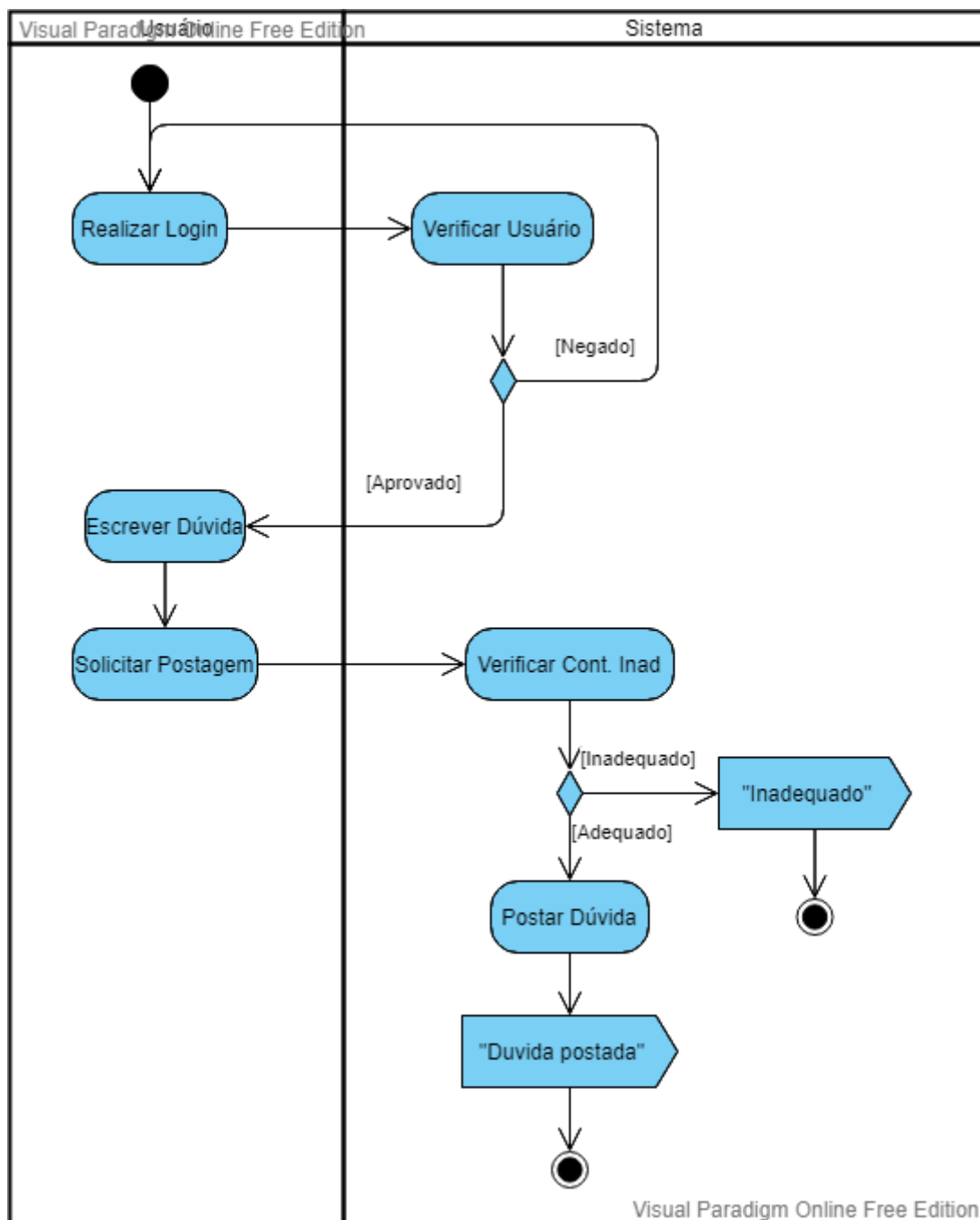
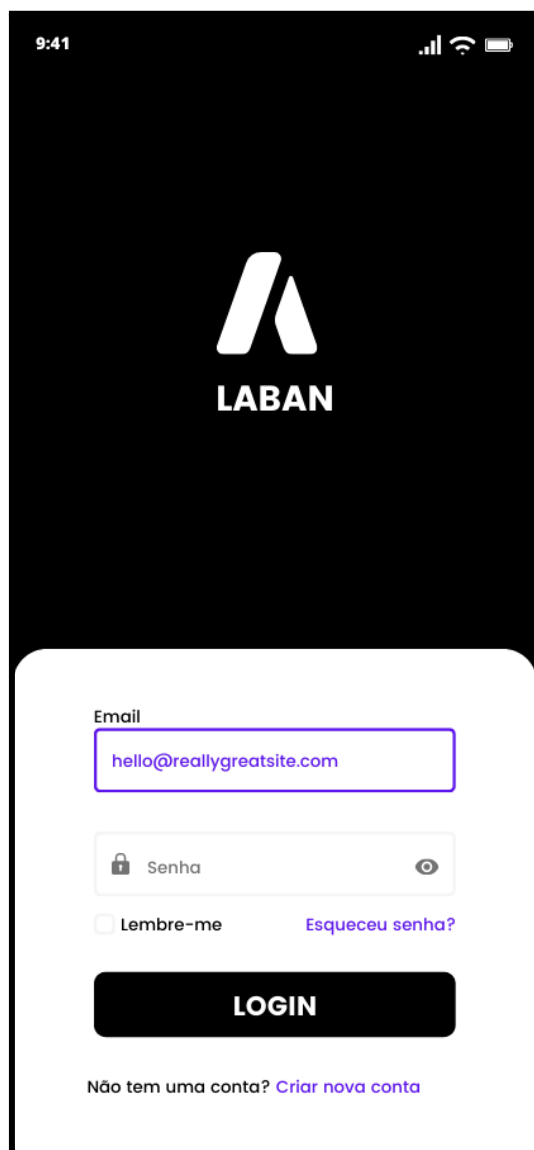


Figura 4. Diagrama de atividades do processo de postar uma dúvida.

A figura 4 ilustra o processo de postagem de uma dúvida. Assim que o usuário loga no aplicativo é verificada se suas informações de Login conferem se estas estiverem incorretas o usuário é levado novamente a tela de login, caso esteja correta, no aplicativo, na aba de dúvidas o usuário escreve uma dúvida e envia uma solicitação para postagem, logo após o conteúdo é verificado se não possui algum tipo de mensagem inadequada, se possuir uma mensagem é emitida e o processo é finalizado, se o conteúdo for adequado o sistema realiza o envio e informa ao usuário que sua pergunta foi submetida e posteriormente encerra o seu processo.

## 12 Protótipo

O objetivo desta seção é apresentar as principais telas do sistema. As telas que representam as funcionalidades mais importantes. As próximas versões do documento incluirão as demais funcionalidades que compõem a aplicação.



Tela de login do aplicativo

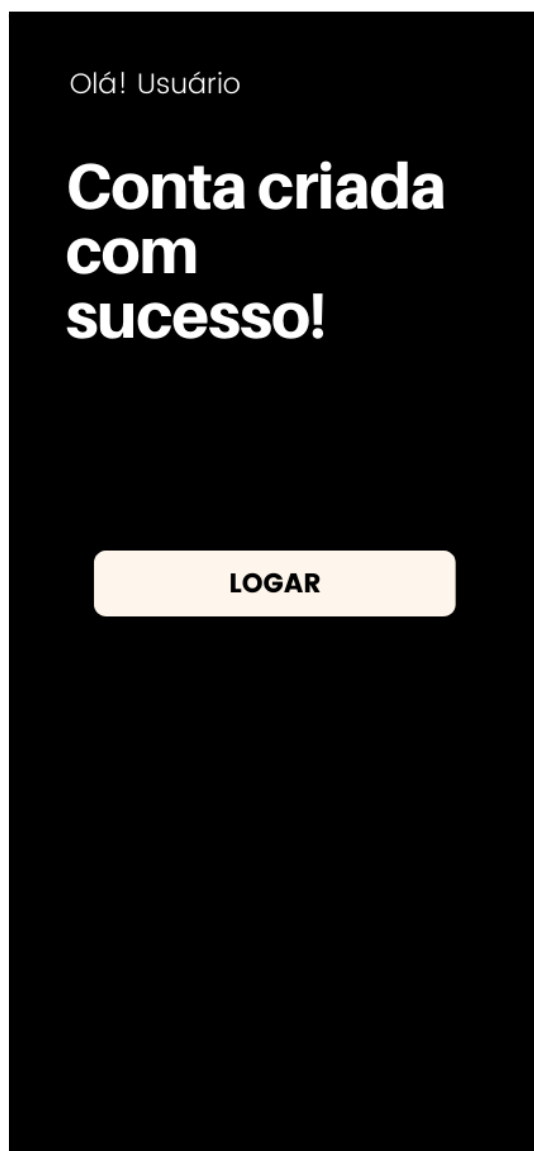
Esta será a primeira tela exibida aos usuários ao abrirem o aplicativo. Nesta, se o usuário já tiver realizado o seu cadastro ele entrará com seu respectivo usuário e senha. Ao contrário, se este não tiver uma conta será necessário que o mesmo clique em *Criar nova conta*.

A mobile application screen for creating a new account. The screen has a black header bar with a white back arrow on the left. Below the header, the title "Criar nova conta" is displayed in bold black text. The form consists of four input fields, each with a label above it: "NOME" (containing "Jiara Martins"), "EMAIL" (containing "hello@reallygreatsite.com"), "SENHA" (containing "\*\*\*\*\*"), and "DATA DE ANIVERSÁRIO" (containing "Selecionar"). At the bottom of the form is a black button with the white text "Enviar".

Tela de criação de conta

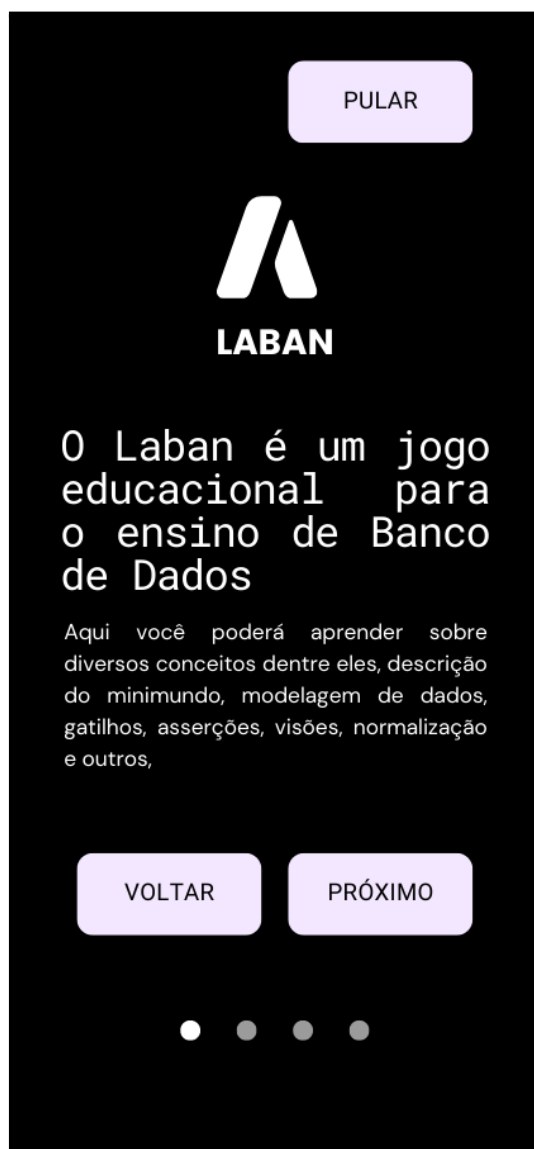
Nesta tela o usuário entrará com seu nome, e-mail (que servirá como identificador), sua senha e sua data de aniversário.





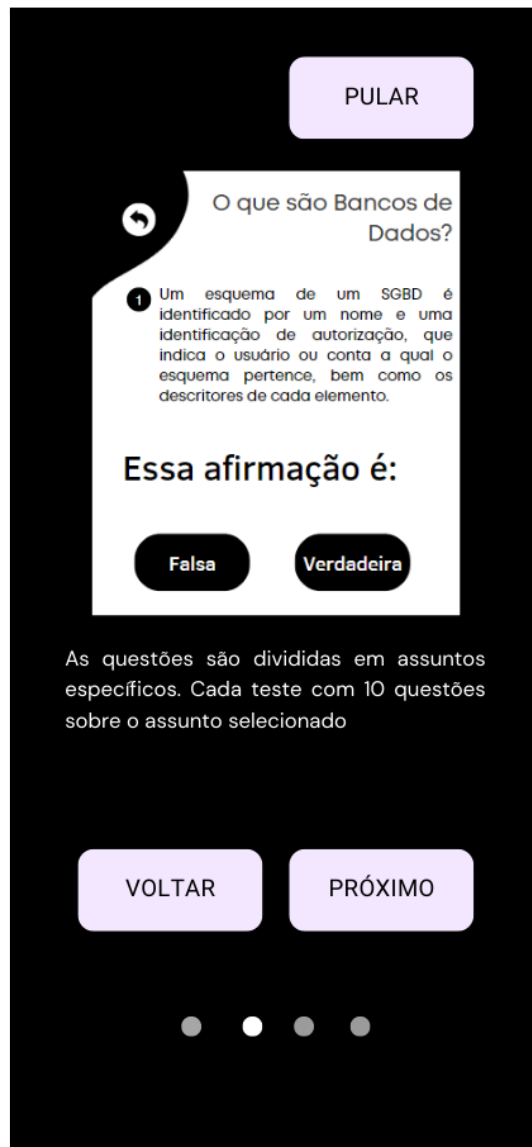
*Tela referente a criação com sucesso de login*

Esta tela será exibida após os usuários enviarem sua informação de cadastro. Após esta o usuário será redirecionado novamente para a tela de login, na qual este agora poderá entrar com seu novo cadastro.



*Tela exibida pela primeira vez após o login com um tutorial*

Após o usuário entrar pela primeira vez no aplicativo será exibido um tutorial informando sobre a aplicação de modo a facilitar o entendimento sobre o uso do aplicativo. Durante todo o tutorial o usuário poderá pular a exibição.



*Segunda tela de tutorial*

Nesta tela já será mostrado detalhes sobre a aplicação, como por exemplo os testes que ocorrem nesta e suas dinâmicas. Novamente o usuário poderá pular o tutorial exibido.



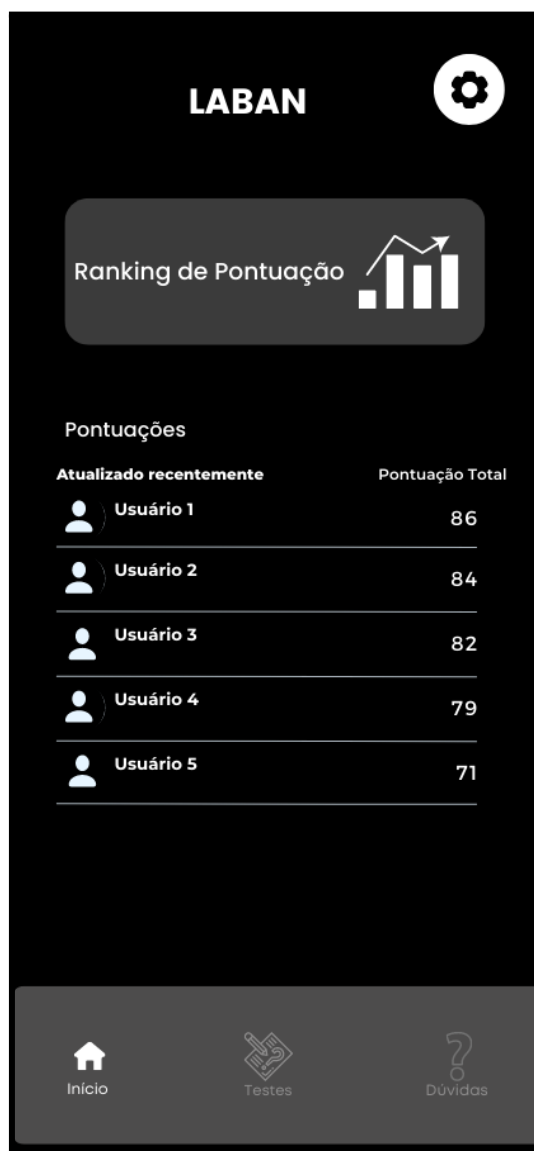
*Terceira tela do tutorial*

Nesta tela será mostrado os detalhes de pontuação obtidos pelos jogadores da aplicação, bem como uma descrição sucinta. Novamente o tutorial poderá ser pulado nesta etapa.



*Tela final do tutorial*

Após todas as tela de tutorial o usuário ao clicar no botão *ir* será redirecionado ao menu da aplicação. E a opção de pular a exibição não será mais exibida, pois esta é a última tela do tutorial.



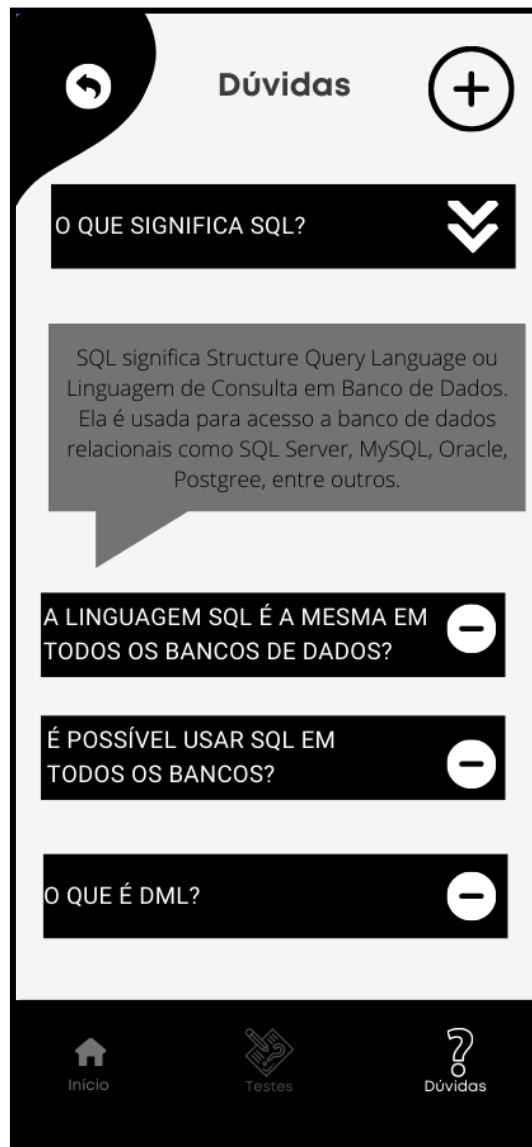
*Tela de menu de uso da aplicação*

Nesta tela inicial será mostrado o ranking de usuários da aplicação. Bem como os ícones para o menu teste e dúvidas.



*Menu de testes do aplicativo*

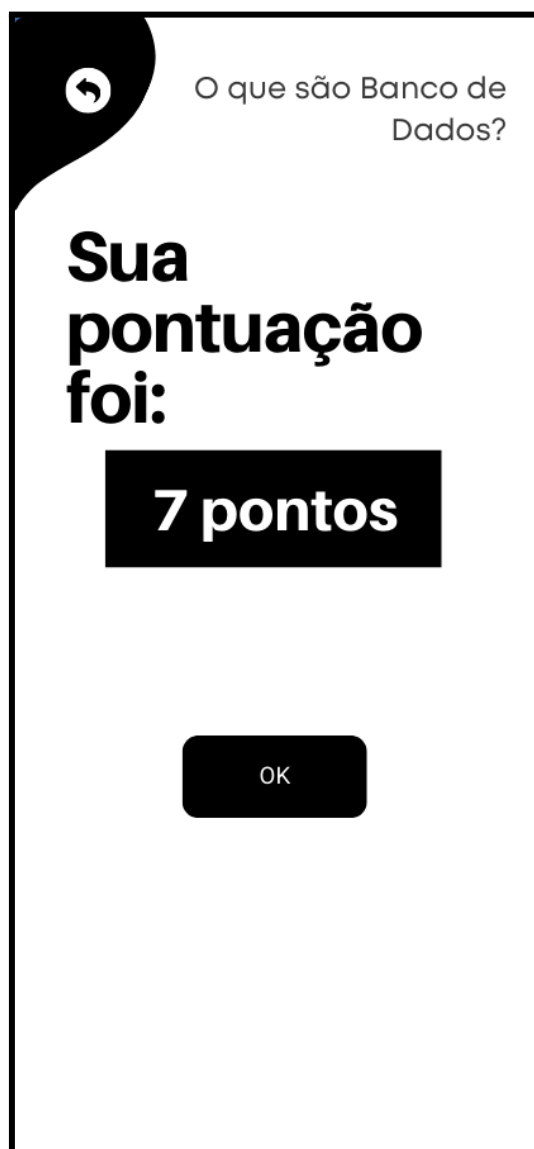
Neste menu os usuários poderão responder aos testes de acordo com uma matéria específica. Após feitos, o ícone de círculo da aplicação ficará com um sinal de visto indicando que aquele conjunto de testes já foi realizado.



*Tela referente a aba de dúvidas*

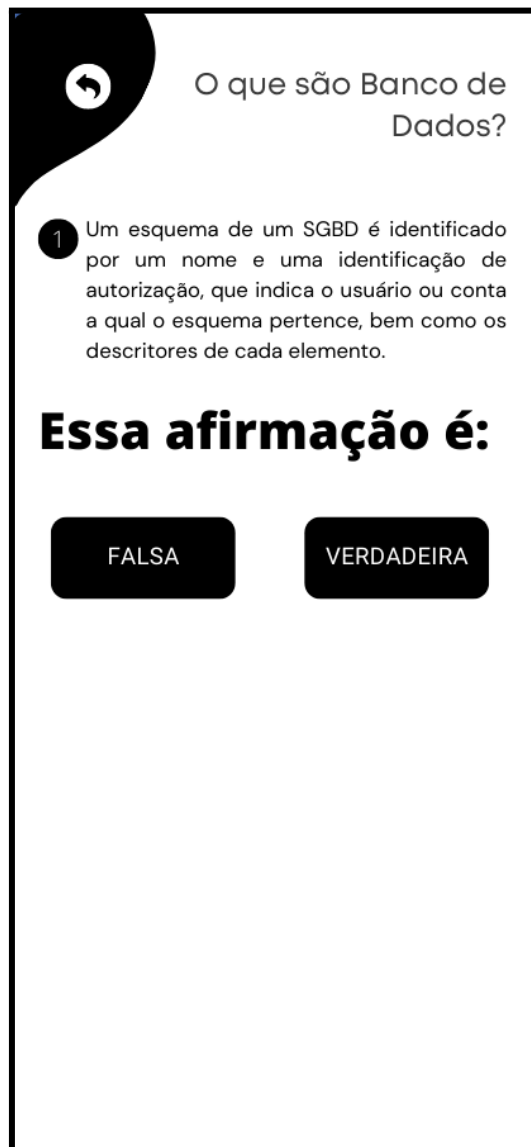
Nesta aba os usuários além de poderem visualizar as dúvidas respondidas que outros usuários obtiveram estes poderão realizar a postagem de uma nova dúvida.





*Tela de pontuação*

Esta tela de pontuação será exibida ao final de cada teste informando a pontuação total obtida por usuários em cada teste.

A simulated test question screen with a black header bar on the left containing a white circular icon with a right-pointing arrow. The title "O que são Banco de Dados?" is in the top right. A question number "1" in a black circle is followed by the text: "Um esquema de um SGBD é identificado por um nome e uma identificação de autorização, que indica o usuário ou conta a qual o esquema pertence, bem como os descritores de cada elemento." Below this is the bold text "Essa afirmação é:". At the bottom are two black buttons with white text: "FALSA" and "VERDADEIRA".

O que são Banco de Dados?

1 Um esquema de um SGBD é identificado por um nome e uma identificação de autorização, que indica o usuário ou conta a qual o esquema pertence, bem como os descritores de cada elemento.

**Essa afirmação é:**

FALSA VERDADEIRA

*Tela simulando uma questão de um teste*

Cada teste contará com questões aleatórias definidas com base no Banco de questões da aplicação. No caso do exemplo é uma pergunta do tópico *O que são Banco de dados*.

Comando SELECT

**Bancos**

Codigo	Nome
001	Banco do Brasil
033	Santander
237	Bradesco
341	Itaú

Tabela 1: Bancos Comerciais.

**Pessoas**

CPF	Nome
86277635697	José da Silva
88208811874	Manoel da Silva
66516764743	Maria dos Santos

Tabela 2: Pessoas.

**Conta Corrente**

Banco	Pessoa	Numero
033	86277635697	98876788
237	86277635697	96645727
341	66516764743	9102947
001	88208811874	8120938

Tabela 3: Contas Correntes.

**Marque a alternativa que possui o comando SQL utilizado para que sejam selecionadas as informações (nome do correntista e o número de conta corrente) dos correntistas do Banco do Brasil.**

**A** `SELECT Nome, Numero FROM Pessoas, Conta_Corrente WHERE Pessoa=CPF AND Banco IN (SELECT Codigo AS Banco FROM Bancos WHERE Nome='Banco do Brasil');`

**B** `SELECT Nome, Numero FROM Pessoas, Conta_Corrente, Bancos WHERE Pessoa=CPF AND Banco IN (SELECT Codigo AS Banco FROM Bancos WHERE Nome='Banco do Brasil');`

**C** `SELECT Nome, Numero FROM Pessoas, Conta_Corrente WHERE Pessoa=CPF AND Banco='Banco do Brasil';`

**D** `SELECT Nome, Numero FROM Pessoas, Conta_Corrente, Bancos WHERE Pessoa=CPF AND Nome='Banco do Brasil';`

*Tela simulando uma questão de um teste*

Exemplo de questão aleatória definida com base no Banco de questões da aplicação. No caso do exemplo é uma pergunta do tópico *Comando SELECT*.

## 13 Referências

JÚNIOR, Edwar Saliba. Diagrama de Caso de Uso. 2020. Disponível em: <[http://esj.eti.br/IFTM/Disciplinas/Grau03/APOO/APOO\\_Unidade\\_04\\_DiagramaDeCasoDeUso.pdf](http://esj.eti.br/IFTM/Disciplinas/Grau03/APOO/APOO_Unidade_04_DiagramaDeCasoDeUso.pdf)>

How to make a UML Sequence Diagram. Disponível em <[https://www.youtube.com/watch?v=pCK6prSq8aw&list=LL&index=1&t=346s&ab\\_channel=LucidSoftware](https://www.youtube.com/watch?v=pCK6prSq8aw&list=LL&index=1&t=346s&ab_channel=LucidSoftware)>

Tutorial de Diagrama de Classes UML. Disponível em: <[https://www.youtube.com/watch?v=rDidOn6KN9k&list=LL&index=2&ab\\_channel=LucidSoftwarePortugu%C3%AAs](https://www.youtube.com/watch?v=rDidOn6KN9k&list=LL&index=2&ab_channel=LucidSoftwarePortugu%C3%AAs)>

Marco Tulio Valente. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade, Editora: Independente, 395 páginas, 2020. Disponível em:<<https://engsoftmoderna.info/cap4.html>>