DOCUMENTO DE VISÃO

Destrava

Adrielson Justino

HISTÓRICO DE VERSÕES			
VERSÃO	**DATA**	**DESCRIÇÃO**	**RESPONSÁVEL**
1.0	25/11/2022	Elaboração do Documento de visão	Adrielson
2.0	02/02/2023	Inclusão do modelo conceitual e lógico do banco	Adrielson

Documento de Visão de Projeto

1. Introdução

Objetivo: O propósito deste documento é coletar, analisar e definir as necessidades de alto-nível e características do projeto de software, focando nas potencialidades requeridas pelos afetados e usuários-alvo, e como estes requisitos serão abordados no projeto de software.

A visão do projeto documenta o ambiente geral de processos a ser desenvolvido para o sistema durante o projeto, fornecendo a todos os envolvidos uma descrição compreensível deste e de suas macro-funcionalidades.

O Documento de Visão de Projeto documenta apenas as necessidades e funcionalidades do sistema que estão sendo atendidas no projeto de software.

2. Descrição do Projeto

O projeto “destrava” tem como propósito facilitar o encontro entre professores empregados ou não que buscam por uma renda a mais, e alunos que precisam destravar em alguma disciplina, reforço escolar, aprender um novo idioma, treinamento de técnicas ou tecnologias e muito mais.

3. Papel das Partes Interessadas

3.1 - Cliente

Descrição	Responsável pelos requisitos funcionais e não funcionais do sistema. Responsável pelos testes do sistema para homologação do projeto.
Papel no desenvolvimento	• Atuar como facilitador e especificador dos requisitos sistêmicos perante a equipe de desenvolvimento. • Garantir que as regras de negócio sejam suportadas pela base legal. • Validar e aprovar os requisitos. • Fornecer os parâmetros operacionais do sistema. • Acompanhar o desenvolvimento do sistema. • Decidir sobre a realização de reuniões com os colaboradores. • Participar da homologação das decisões relacionadas aos sistemas. • Participar da Homologação do produto final.
Insumos ao projeto de software	• Requisitos Funcionais. • Requisitos Não-Funcionais. • Casos de testes para homologação do sistema. • Consultas diversas para validação do sistema.

	● Homologação das aplicações.
Representante	A definir

3.2 - Equipe de Desenvolvimento destrava	
Descrição	Responsável pela identificação dos requisitos do software e pelo desenvolvimento dos modelos estáticos e dinâmicos do projeto.
Papel no desenvolvimento	● Identificar e descrever as necessidades do usuário, especificando as funcionalidades do software que irão atendê-las. ● Levantar os requisitos funcionais e não funcionais do projeto. ● Definir o que irá interagir com o sistema. ● Identificar dentro da visão lógica do sistema, a melhor forma de acomodar as necessidades do usuário, e o impacto da solução adotada sobre os requisitos do sistema. ● Em suma, responsável pela geração de um produto ● que atenda aos requisitos que foram identificados junto ao usuário
Insumos ao projeto de software	● Requisitos Funcionais. ● Requisitos Não-Funcionais. ● Casos de testes para homologação do sistema. ● Consultas diversas para validação do sistema. ● Homologação das aplicações.
Representante	Adrielson Justino

3.3 - Papel dos Atores	
Descrição	Administradores
Papel	Manutenção e suporte do sistema.
Insumos ao sistema	Moderar conteúdos (conteúdos inadequados, denúncias reportadas)
Representante	Adrielson Justino

Descrição	Usuário (Aluno)
Papel	Consulta.
Insumos ao sistema	Buscar professores, contactar professores.
Representante	Público Geral

Descrição	Usuário (Professor)
Papel	Criador de anúncios.
Insumos ao sistema	Cadastrar anúncios, confirmar solicitação.
Representante	Público Geral

4. Necessidades e Funcionalidades	
Necessidade 1	
Cadastro de Usuários	
Id Func.	**Descrição das Funcionalidades/atores envolvidos**
F1.1	Cadastrar Usuário
	Usuário, Administradores
F1.2	Deletar Cadastro
	Usuário, Administradores
F1.1	Consultar Cadastro
	Administradores

Necessidade 2	
Login/Logout na plataforma	
Id Func.	**Descrição das Funcionalidades/atores envolvidos**
F2.1	Efetuar login
	Administradores, Usuários
F2.2	Efetuar logout
	Administradores, Usuários
Necessidade 3	
Gerenciar Perfil	
Id Func.	**Descrição das Funcionalidades/atores envolvidos**
F3.1	Editar dados do perfil
	Usuários
F3.2	Consultar dados do perfil

	Administradores, Usuários

Necessidade 4	
Anúncios de pacotes	
Id Func.	**Descrição das Funcionalidades/atores envolvidos**
F4.1	Criar Anúncio
	Usuários
F4.2	Editar Anúncio
	Usuários, Administradores
F4.3	Excluir Anúncio
	Usuários, Administradores
F4.6	Consultar Anúncio
	Usuários, Administradores

Necessidade 5	
Buscar anúncios	
Id Func.	**Descrição das Funcionalidades/atores envolvidos**
F5.1	Pesquisar anúncios
	Administradores, Usuários

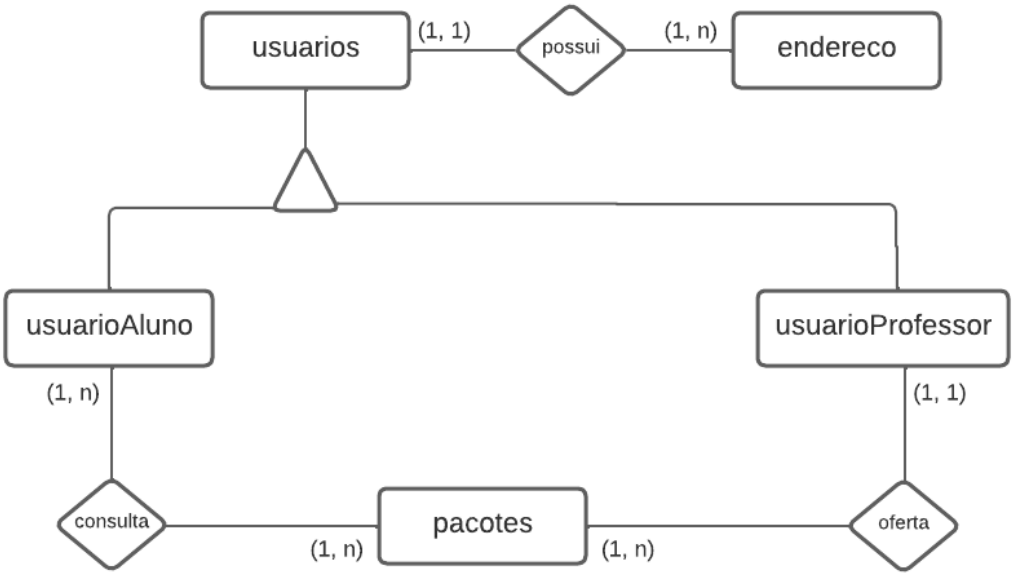
Necessidade 6	
Contactar professor	
Id Func.	**Descrição das Funcionalidades/atores envolvidos**
F6.1	Mandar solicitação para professor

	Usuários

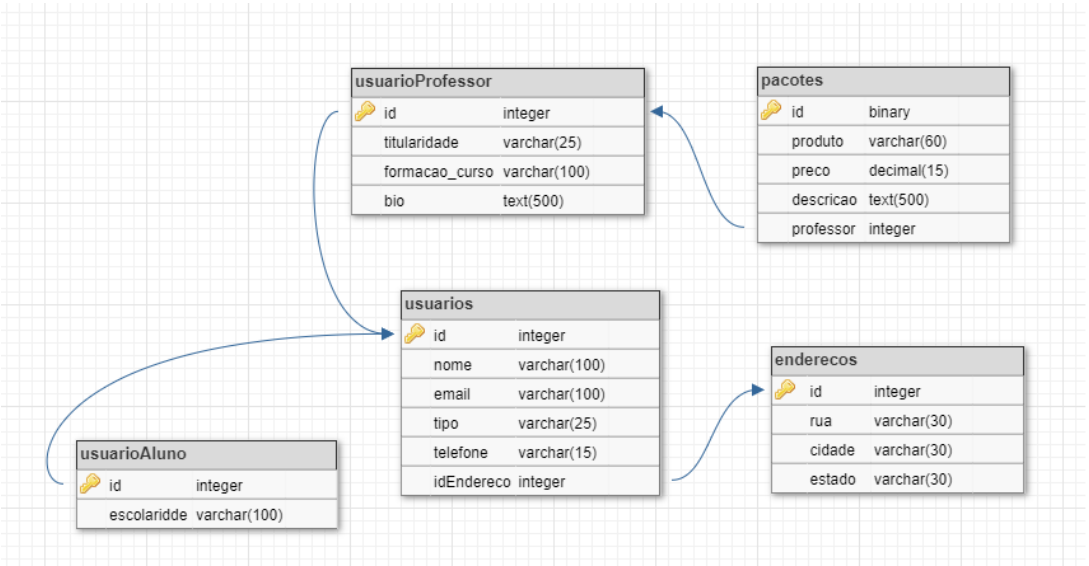
Necessidade 7	
Confirmar solicitação do aluno	
Id Func.	**Descrição das Funcionalidades/atores envolvidos**
F7.1	Confirmar solicitação do aluno
	Administradores, Usuários

6. Premissas e Restrições	
Premissas	**Restrições**
• Não se Aplica	• O sistema não deve ser modificado por seus usuários;

7. Banco de Dados
Modelo Conceitual



Modelo Lógico



Modelo Físico

```sql
CREATE TABLE enderecos (
id INTEGER,
rua VARCHAR(30),
cidade VARCHAR(30),
bairro VARCHAR(30),
estado VARCHAR(30),
PRIMARY KEY(id)
);

CREATE TABLE usuarios (
  id INTEGER AUTO_INCREMENT,
  nome VARCHAR(100) NOT NULL,
  email VARCHAR(100) NOT NULL,
  senha VARCHAR(100) NOT NULL,
  tipo ENUM('professor', 'aluno') NOT NULL,
  telefone VARCHAR(15),
  idEndereco INTEGER,
  PRIMARY KEY(id),
  FOREIGN KEY (idEndereco) REFERENCES enderecos(id)
);

CREATE TABLE usuariosprofessor (
  id INTEGER,
  titularidade ENUM('Tecnico', 'Graduado', 'Mestre', 'Doutor'),
  formacao_curso VARCHAR(100),
  bio LONGTEXT,
  PRIMARY KEY(id),
  FOREIGN KEY (id) REFERENCES usuarios(id)
);

CREATE TABLE usuariosaluno (
  id INTEGER,
  escolaridade ENUM('Educação infantil', 'Fundamental', 'Médio', 'Superior', 'Mestrado', 'Doutorado'),
  PRIMARY KEY(id),
  FOREIGN KEY (id) REFERENCES usuarios(id)
);

CREATE TABLE pacotes (
idPacote INTEGER AUTO_INCREMENT,
produto VARCHAR(60) NOT NULL,
preco DECIMAL(10, 2),
descricao LONGTEXT,
professor integer,
PRIMARY KEY(idPacote),
FOREIGN KEY (professor) REFERENCES usuariosprofessor(id)
);
```

Triggers

```sql
-- gatilho pra adicionar id à tabela dos professores para deixar vinculado à tabela usuario caso um novo professor seja cadastrado :

DELIMITER $$
CREATE TRIGGER tr_usuariosprofessor_insert
AFTER INSERT ON usuarios
FOR EACH ROW
BEGIN
  IF NEW.tipo = 'professor' THEN
    INSERT INTO usuariosprofessor (id)
    VALUES (NEW.id);
  END IF;
END$$
DELIMITER ;



-- gatilho pra adicionar id à tabela dos alunos para deixar vinculado à tabela usuario caso um novo aluno seja cadastro :

DELIMITER $$
CREATE TRIGGER tr_usuariosaluno_insert
AFTER INSERT ON usuarios
FOR EACH ROW
BEGIN
  IF NEW.tipo = 'aluno' THEN
    INSERT INTO usuariosaluno (id)
    VALUES (NEW.id);
  END IF;
END$$
DELIMITER ;




-- gatilho pra adicionar id à tabela dos enderecos para deixar vinculado à tabela usuario caso um novo usuario seja cadastro :

DELIMITER $$
CREATE TRIGGER tr_endereco_insert
AFTER INSERT ON usuarios
FOR EACH ROW
BEGIN
    INSERT INTO enderecos (id)
    VALUES (NEW.id);
END$$
DELIMITER ;
```

Consultas, inserções, updates

CONSULTAS

Página de login

```
// Consulta a tabela "professores"
$sql = "SELECT * FROM usuarios WHERE email='$email' AND senha='$senha' AND tipo='professor'";
```

```
// Consulta a tabela "alunos"
$sql = "SELECT * FROM usuarios WHERE email='$email' AND senha='$senha' AND tipo='aluno'";
```

Dados perfil professor

```
// exibir as informações de contato do usuario na pagina professor
$sql = "SELECT usuarios.*, enderecos.*, usuariosprofessor.*
FROM usuarios
JOIN enderecos ON usuarios.id = enderecos.id
JOIN usuariosprofessor ON usuarios.id = usuariosprofessor.id
WHERE usuarios.id ='$id_usuario'";
```

Dados perfil aluno

```
// exibir as informações de contato do usuario na pagina aluno
$id_usuario = $_SESSION['idUsuario'];
$sql = "SELECT * FROM  usuarios  WHERE usuarios.id ='$id_usuario'";
```

Resultados de busca de pacotes

```
//recupera dados da tabela de usuarios e pacotes. Filtro de busca de aulas.
$sql = "SELECT pacotes.*, usuarios.nome as nome_professor, usuarios.telefone
FROM pacotes
JOIN usuarios ON pacotes.professor = usuarios.id
WHERE produto LIKE '%$produto%'";
```

INSERÇÕES

Cadastro de usuarios

```
$query = "INSERT INTO usuarios (nome, email, senha, tipo)
VALUES ('$nome', '$email', '$senha','$tipo')";
```

Cadastro de Pacotes

```
$sql = "INSERT INTO pacotes (produto, descricao, preco, professor)
VALUES ('$titulo', '$descricao', '$valor', '" . $_SESSION['idUsuario'] . "')";
```

UPDATES

```php
if ($nome != null) {
    $stmt = $conn->prepare("UPDATE usuarios SET nome = ? WHERE id = ?");
    $stmt->bind_param("si", $nome, $id_usuario);
    $stmt->execute();
}
if ($telefone != null) {
    $stmt = $conn->prepare("UPDATE usuarios SET telefone = ? WHERE id = ?");
    $stmt->bind_param("si", $telefone, $id_usuario);
    $stmt->execute();
```