Roteiro para Seminário Acadêmico: Desenvolvimento do Projeto "Universo Down"

1. Introdução

- **Objetivo do Projeto**: Desenvolver uma aplicação web para o projeto "Universo Down" que integra um backend robusto utilizando Node.js e um frontend dinâmico com Vue.js.
- **Visão Geral**: Explicar a arquitetura geral da aplicação, incluindo a comunicação entre o backend e o frontend, e o uso do banco de dados MySQL.

2. Desenvolvimento do Backend

2.1. Configuração do Ambiente

- **Instalação do Node.js e npm**: Explicar como instalar o Node.js e o npm.
- **Inicialização do Projeto Node.js**:

```
npm init -y
```

- **Instalação das Dependências Necessárias**:

```
npm install express mysql sequelize dotenv
```

- **Estrutura de Pastas do Projeto**:
  - `src/`
    - `config/`
    - `controllers/`
    - `models/`
    - `routes/`
    - `app.js`

2.2. Configuração do Banco de Dados

- **Configuração do Sequelize**: Criação do arquivo `config/database.js` para configurar o Sequelize.
- **Modelagem das Tabelas no Sequelize**:
  - Exemplificar com um modelo `User`.

```
const { DataTypes } = require('sequelize');
const sequelize = require('../config/database');

const User = sequelize.define('User', {
  name: {
    type: DataTypes.STRING,
    allowNull: false
  },
  email: {
```

```
    type: DataTypes.STRING,
    unique: true,
    allowNull: false
  }
});

module.exports = User;
```

2.3. Criação das Rotas e Controladores

- **Definição de Rotas**: Explicar como criar rotas RESTful.
- **Implementação dos Controladores**: Mostrar exemplos de controladores para criar, ler, atualizar e deletar dados.

```
const express = require('express');
const UserController = require('../controllers/UserController');
const router = express.Router();

router.post('/users', UserController.createUser);
router.get('/users', UserController.getAllUsers);

module.exports = router;
```

2.4. Inicialização do Servidor

- **Configuração do Servidor Express**: Criação do arquivo `app.js` para iniciar o servidor.

```
const express = require('express');
const userRoutes = require('./routes/userRoutes');
const app = express();

app.use(express.json());
app.use('/api', userRoutes);

app.listen(3000, () => {
  console.log('Server is running on port 3000');
});
```

3. Desenvolvimento do Frontend

3.1. Configuração do Ambiente

- **Instalação do Vue CLI**:

```
npm install -g @vue/cli
```

- **Criação do Projeto Vue**:

```
vue create universo-down-frontend
```

3.2. Estrutura do Projeto Vue

- **Estrutura de Pastas**:
  - `src/`
    - `assets/`
    - `components/`
    - `router/`
    - `views/`
    - `App.vue`
    - `main.js`

3.3. Configuração do Vue Router

- **Criação de Rotas**: Configuração das rotas em `src/router/router.js`.

```
import { createRouter, createWebHistory } from 'vue-router';
import HomeView from '../views/Home.vue';

const routes = [
  {
    path: '/',
    name: 'home',
    component: Home
  }
];

const router = createRouter({
  history: createWebHistory(process.env.BASE_URL),
  routes
});

export default router;
```

3.4. Desenvolvimento de Componentes

- **Criação de Componentes**: Exemplificar com componentes `Navbar.vue` e `Footer.vue`.
- **Integração dos Componentes**: Mostrar como integrar componentes no `App.vue`.

3.5. Estilização com Bootstrap-Vue

- **Instalação do Bootstrap-Vue**:

```
npm install bootstrap bootstrap-vue-3
```

- **Configuração do Bootstrap-Vue em `main.js`**:

```
import { createApp } from 'vue';
import App from './App.vue';
import router from './router';
import BootstrapVue3 from 'bootstrap-vue-3';

import 'bootstrap/dist/css/bootstrap.css';
import 'bootstrap-vue-3/dist/bootstrap-vue-3.css';

const app = createApp(App);
app.use(router);
app.use(BootstrapVue3);
app.mount('#app');
```

3.6. Comunicação com o Backend

- **Configuração do Axios**:

```
npm install axios
```

- **Exemplo de Requisição Axios**:

```
import axios from 'axios';

export default {
  name: 'HomeView',
  data() {
    return {
      users: []
    };
  },
  created() {
    axios.get('http://localhost:3000/api/users')
      .then(response => {
        this.users = response.data;
      });
  }
};
```

4. Conclusão

- **Demonstração da Aplicação**: Mostrar a aplicação em funcionamento.
- **Discussão sobre Desafios e Soluções**: Abordar problemas encontrados e como foram resolvidos.
- **Próximos Passos**: Planos futuros para melhorias e expansões do projeto.

5. Perguntas e Respostas

- **Abrir para Perguntas**: Dar espaço para que os participantes façam perguntas e obtenham esclarecimentos sobre o projeto.