**Desenvolvimento da Aplicação**

**Funcionalidades Para o Desenvolvimento**

* Criação de Contas
* Login de Usuários
* Criação de Eventos
* Criação de Trilhas
* Mudança de Papéis
* Submissão
* Emissão de Certificados

**Tecnologias**

* NodeJS – Plataforma de Desenvolvimento
* ExpressJS – Framework Back - End
* Sequelize – ORM para banco de Dados
* MariaDB – Banco de Dados Relacional, opção ao MySQL
* React – Biblioteca para o Front – End Web
* React-Native – Biblioteca para o Front – End Mobile
* Expo – Biblioteca para Comunicação entre Mobile e API

**Estrutura do Back-End**

-Sige\_aplicacao: Pasta main do sistema.

--- app: Pasta que contém os arquivos base da aplicação.

-------- controllers: Pasta dos Controllers

-------- models: Pasta dos Models

-------- routes: Pasta das rotas

--- config: Pasta das configurações de banco de dados e etc.

--- database: Pasta das configurações de tabelas e seeds do banco de dados.

--- public: Pasta que contém os arquivos de view.

--- package.json: Arquivo de configuração relacionado ao NodeJS

--- server.js: Aquivo de inicialização do sistema e servidor.

--- .gitignore: Arquivo de declarar para ignorar pastas ou arquivos, para submeter ao git.

--- .sequelizerc: Arquivo de configuração das pastas do Sequelize

**Geral**

Rotas:

* / - Tela de Login do sistema, Index do Sistema
* /login – Tela de login do sistema

Métodos:

* /login: Post – Enviar informação de login e senha ao sistema
* /login: Get – Redireciona para tela de Login

Campos do formulário: login e senha

Tempo da Sessão de Usuário no Sistema: 30 minutos

Caso login seja realizado, redirecionar para:

* /inscrito/home

**Cadastro**

Rotas:

* /cadastro – Formulário de cadastro de participantes

Métodos:

* /cadastro: Post – Enviar informações do formulário de cadastro ao sistema
* /cadastro: Get – Redireciona para tela de Cadastro

Modelo:

Classe geral que contém informações que todos que usam o sistema possui.

Inscrito:

* cpf - numérico
* nome - string
* email - string
* senha - string
* deficiência - booleano
* tipo\_deficiencia – string, padrão: Nenhum ou Nulo
* papel – numérico (1 - adm, 2 - revisor, 3 - participante)

Controller:

InscritoController:

* create(): Cria novo usuário
* find(): Lista todos os usuários
* findById(id): Busca um usuário pelo id
* edit(id): Edita um usuário
* delete(id): Deleta um usuário

**Papéis de Usuários**

Participante:

* pedir\_certifica()
* submeter\_resumo()
* inscrever\_evento()
* liberar\_resumo() – Caso tenha cpf cadastrado em submissão
* pedir\_correcao() – Caso tenha cpf cadastrado em submissão

Revisor:

* area\_conhecimento – String
* buscar\_submissao()
* aprovar\_submissao()
* reprovar\_submissao()

Administrador:

* liberar\_certificados()
* gerar\_anal()
* buscar\_usuarios()
* atribuir\_papel()

**Home**

Rotas

* /inscrito/home – Página principal do sistema. Somente autenticados.

**Evento**

Rotas

* /inscrito/evento – Pagina de Eventos Cadastrados no Sistema
* /inscrito/evento/new – Formulário para criar novo evento
* / inscrito/evento/edit/id – Formulário de edição de eventos.

Métodos:

* / inscrito/evento: Get – Redireciona para página de eventos
* / inscrito/evento/new: Get – Redireciona para formulário de criar novos eventos.
* / inscrito/evento/new: Post – Envia informações de evento criado ao sistema.
* / inscrito/evento/edit/id: Get – Redireciona para formulário de editar eventos.
* / inscrito/evento/edit/id: Put – Envia informações de edição do evento ao sistema.
* / inscrito/evento/id: Delete – Deleta o evento selecionado do sistema.

Modelo:

Classe referente ao evento criado pelo administrador e utilizado por todos os usuários do sistema.

Evento:

* nome - string
* url - string
* instituição - string
* máximo\_inscricoes - numérico
* sigla - string
* virtual - booleano
* descrição - string
* submissão - booleano
* inicio\_evento - data
* fim\_evento - data
* inicio\_submissao - data
* fim\_submissao - data
* inicio\_revisao - data
* fim\_revisao - data
* pago - booleano
* valor\_inscricao - float
* máximo\_ativ\_inscrito – numérico
* dias\_evento - numérico
* trilhas – Classe Trilha
* Inscritos – associação Classe Lista Inscritos

Controller:

EventoController:

* create(): Cria novo evento
* find(): Lista todos os eventos
* findById(id): Busca um evento especifico
* edit(id): Edita um evento especifico
* delete(id): Deleta um evento especifico

**Trilhas**

Rotas:

* /evento/trilha/new – Pagina de Cadastrar Trilha
* /evento/trilha – Pagina de Listar Trilha

Metodos:

* /evento/trilha/new: Post – Cadastrar Trilha no Sistema
* /evento/trilha: Get – Recebe Lista de Trilhas Para o Evento Selecionado

Modelos:

Trilha:

* nome: string

Controller:

TrilhaController:

* create(): Cria nova Trilha
* find(): Lista todas as Trilhas de um Evento
* findById(id): Busca uma trilha especifica
* edit(id): Edita uma trilha especifica
* delete(id): Deleta uma trilha especifica

Submissão

Rotas

/evento/submissao/new – Criar Nova Submissao

/inscrito/submissão – Lista de Submissoes

Metodos:

/evento/submissao/new: Post – Cadastra nova submissão no sistema

/inscrito/submissao: Get – Recebe lista de submissões do inscrito

Modelo:

* titulo - string
* texto - text
* palavra\_chave – lista de string
* autores – lista de string
* cpf\_orientador - numérico
* mensagem\_apoio - text
* ods - string
* status\_aprovacao – numérico (0 reprovado, 1 aprovado, 2 analise)
* área\_conhecimento - string
* feedback – text (Visualizado apenas por revisor)

Lista Padrão ODS(Número deve estar presente):

* 1 – Erradicação da Fome
* 2 – Fome Zero e Agricultura Responsável
* 3 – Saúde e Bem-Estar
* 4 – Educação de Qualidade
* 5 – Igualdade de Gênero
* 6 – Água Potável e Saneamento
* 7 – Energia Acessível e Limpa
* 8 – Trabalho Decente e Crescimento Econômico
* 9 – Indústria, Inovação e Infraestrutura
* 10 – Redução das Desigualdades
* 11 – Cidades e Comunidades Sustentáveis
* 12 – Consumo e Produção Responsáveis
* 13 – Ação Contra a Mudança Global do Clima
* 14 – Vida na Água
* 15 – Vida Terrestre
* 16 – Paz, Justiça, e Instituições Eficazes
* 17 – Parcerias e Meios de Implementação

Lista Padrão Área do Conhecimento:

* Ciências Exatas e da Terra
* Ciências Biológicas
* Engenharias
* Ciências da Saúde
* Ciências Agrarias
* Ciências Sociais Aplicadas
* Ciências Humanas
* Linguística, Letras e Artes
* Outros

Na submissão vem Perguntas Extras Proveniente da Trilha.

Controller:

* create(): Cria nova Submissão
* find(): Lista todas as Submissões
* findById(id): Busca uma Submissão especifica
* edit(id): Edita uma Submissão especifica
* delete(id): Deleta uma Submissão especifica

**Inscrever em Evento**

Formulário:

* Lista de Eventos Ativos
* Instituição do Participante
* Categoria: Discente, Docente/Pesquisador, Pesquisador, Outros.
* Tipo de Participante: Bolsista, Não Bolsista, Ouvinte.
* Número do Celular

**Certificado**

Campos - Apresentação:

* Nome do Participante
* Tipo de Apresentação
* Título da Submissão
* Nome dos Autores
* Código do Certificado

Campos – Revisor:

* Nome do Revisor
* Código do Certificado