



**UNIVERSIDADE FEDERAL DE RORAIMA  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO  
CURSO BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO  
DCC603 - BANCO DE DADOS II**

**JOÃO ROBERTO RANDEL COSTA CUNHA  
MATHEUS DE SOUZA MELO  
NATÁLIA RIBEIRO DE ALMADA**

**SISTEMA DE GERENCIAMENTO DE CURSOS**

BOA VISTA – RR

Julho de 2023

**JOÃO ROBERTO RANDEL COSTA CUNHA**  
**MATHEUS DE SOUZA MELO**  
**NATÁLIA RIBEIRO DE ALMADA**

Projeto de criação de Banco de Dados 2

Projeto da disciplina de Banco de Dados 2, a ser apresentado ao Centro de Ciência da Computação, da Universidade Federal de Roraima, como requisito parcial para obtenção de nota na disciplina obrigatória do Curso de Ciência da Computação – sob orientação do professor Msc. Acauan Cardoso Ribeiro

BOA VISTA – RR  
Julho de 2023

## **OBJETIVO**

A plataforma educacional proposta tem como objetivo oferecer um ambiente seguro e eficiente para a administração e acesso a cursos online. O sistema conta com recursos de autenticação e autorização, permitindo que usuários se cadastrem, façam login e tenham diferentes níveis de permissão.

O usuário root é criado durante a criação das tabelas do sistema e tem o poder de conceder permissões para novos usuários, incluindo o perfil de administrador. O usuário administrador possui diversas funcionalidades, como cadastrar, alterar e deletar categorias de cursos, além de cadastrar, alterar e deletar cursos individuais. O administrador também pode suspender a conta de um aluno, ter acesso à listagem dos alunos e realizar a rotina de fechamento de cursos, emitindo certificados para os alunos que tenham concluído mais de 90% da carga horária.

Os alunos têm a capacidade de se cadastrar no sistema e atualizar seu próprio perfil, incluindo informações como endereço, número de celular e foto do perfil. Além disso, os alunos podem se inscrever nos cursos disponíveis na plataforma.

No sistema, os cursos estão vinculados a uma única categoria, enquanto as categorias podem conter vários cursos vinculados a elas. Cada curso é identificado por um ID único e possui informações como nome do curso, categoria (referenciada por uma chave estrangeira), data de início, carga horária e quantidade de inscritos.

A emissão de certificados segue critérios de validação do aluno. Por exemplo, o aluno pode receber um certificado se tiver concluído um questionário com pontuação superior a 70% ou se tiver completado mais de 90% da carga horária do curso. Nessa versão do sistema, a carga horária do aluno é inserida por um perfil administrativo.

Essa plataforma educacional visa fornecer uma experiência completa e segura para alunos e administradores, possibilitando o gerenciamento eficiente de cursos, aprimoramento do aprendizado e emissão de certificados confiáveis.

## **REQUISITOS**

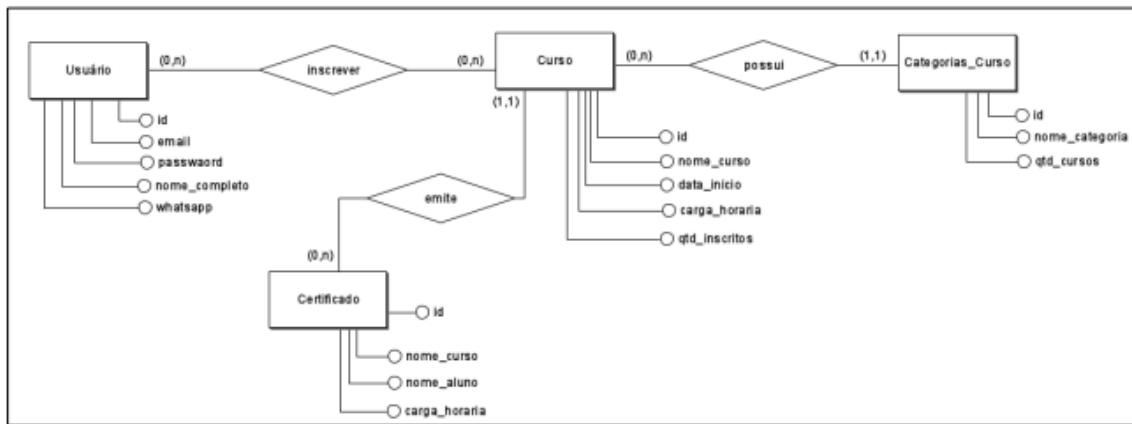
Os requisitos deste sistema, abarcam várias camadas, dentro das funções que serão capazes de realizar, podemos colocar:

- O sistema contém autenticação/autorização com login e Cadastro de Usuários.
- O usuário poderá se cadastrar na plataforma usando um email e uma senha, podendo ter acesso ao seu perfil dentro da plataforma .
- Usuário root (criado junto ao criação das tabelas) do sistema que pode dar permissão para novos usuários tipo admin.
- Um usuário root foi criado para atender as demandas dentro da plataforma, com este podendo criar cursos, deletá-los ou modificá-los.
- Usuário admin pode cadastrar/alterar/deletar novas categorias de curso.
- Usuário admin é capaz de criar, deletar e modificar os cursos da plataforma, sendo este também capaz de validar os certificados emitidos para os alunos que foram capazes de terminar cursos.
  - Tem acesso a listagem dos alunos.
  - Faz a rotina de “fechar” um curso, emitindo certificados para quem terminou mais de 90% da carga horária.
- O usuário pode se cadastrar no sistema e nos cursos
- Os alunos com seus devidos perfis criados e com acesso a plataforma , poderão se cadastrar no sistema, bem como os cursos presentes na plataforma.
- Certificados são emitidos seguindo algum critério de validação do aluno. Exemplo: - Aluno terminou de responder um questionário e acertou > 70% E/OU - Quando o aluno termina mais de 90% da carga horária do curso. 6.1 Para essa versão do sistema quem lança a carga horária do aluno é algum perfil admin do sistema.

Para atender estes requisitos, o sistema possui um banco de dados relacional feito em PostgreSQL, capaz de armazenar as informações pertinentes aos cursos, como seu nome, duração, categoria e outros aspectos. Também pertinentes aos alunos, como nome de perfil, cursos feitos e certificados obtidos.

As rotas do backend foram feitas com Node e Javascript, foi feito um CRUD com o objetivo de armazenar dados específicos e que esses dados não sejam acessados ou armazenados de forma indevida.

## MODELAGEM



O banco de dados é composto por quatro tabelas: "Usuários", "Categorias de Cursos", "Cursos" e "Inscrições".

- A tabela "Usuários" armazena informações sobre os usuários do sistema, como nome, email e senha. Cada usuário possui um ID único.
- A tabela "Categorias de Cursos" armazena as diferentes categorias de cursos disponíveis no sistema. Cada categoria possui um ID único, um nome e uma descrição.
- A tabela "Cursos" armazena informações sobre os cursos oferecidos. Cada curso possui um ID único, um nome, uma referência à categoria de curso a qual está vinculado, uma data de início e uma carga horária. A quantidade de inscritos pode ser mantida nessa tabela ou ser calculada com base nas inscrições na tabela "Inscrições".
- A tabela "Inscrições" registra as inscrições dos alunos nos cursos. Cada inscrição possui um ID único, uma referência ao curso e ao aluno que realizou a inscrição, e uma data de inscrição.

Essa estrutura de banco de dados permite que os usuários se cadastrem no sistema, se inscrevam nos cursos disponíveis, alterem seus perfis e que os usuários com permissão de administrador realizem a gestão das categorias de cursos, cursos e inscrições.

## Banco de Dados

O projeto utiliza o banco de dados PostgreSQL. O PostgreSQL é um sistema de gerenciamento de banco de dados relacional de código aberto, conhecido por sua confiabilidade, robustez e recursos avançados. Ele é amplamente utilizado em projetos de desenvolvimento de software devido à sua capacidade de lidar com grandes volumes de dados e oferecer recursos avançados, como suporte a transações ACID, integridade referencial e consultas complexas. No projeto em questão, o PostgreSQL é utilizado para armazenar e gerenciar os dados relacionados ao sistema gerenciador de cursos.



## Relatório de Dependências do Projeto

O projeto utiliza várias dependências para fornecer funcionalidades específicas. Abaixo está uma descrição das principais dependências utilizadas:

1. bcrypt (^5.1.0):
  - O bcrypt é uma biblioteca para criptografia de senhas.
  - É usado para proteger e armazenar senhas de forma segura no banco de dados.
2. cookie-parser (~1.4.4):
  - O cookie-parser é um middleware para processar cookies em aplicativos Express.
  - Ele analisa os cookies recebidos nas solicitações e os converte em objetos JavaScript.
3. cors (^2.8.5):
  - O cors é um middleware que fornece suporte a CORS (Cross-Origin Resource Sharing) em aplicativos Express.
  - Ele permite solicitações de origens diferentes, permitindo a comunicação entre domínios diferentes.

4. debug (~2.6.9):
  - O debug é uma biblioteca de depuração para Node.js.
  - Ele fornece uma maneira conveniente de imprimir mensagens de depuração durante o desenvolvimento.
5. express (~4.16.1):
  - O express é um framework web rápido e minimalista para Node.js.
  - Ele é usado para criar APIs e manipular solicitações e respostas HTTP.
6. http-errors (~1.6.3):
  - O http-errors é uma biblioteca para criar erros HTTP facilmente.
  - É usado para gerar erros personalizados em rotas e tratamento de erros.
7. jsonwebtoken (^9.0.0):
  - O jsonwebtoken é uma biblioteca para criar e verificar tokens JWT (JSON Web Tokens).
  - É usado para autenticação e autorização baseadas em tokens.
8. method-override (^3.0.0):
  - O method-override é um middleware para suportar verbos HTTP adicionais em aplicativos Express.
  - Ele permite que os métodos HTTP PUT e DELETE sejam usados em navegadores que não os suportam.
9. morgan (~1.9.1):
  - O morgan é um middleware de registro de solicitação HTTP para aplicativos Express.
  - Ele registra detalhes das solicitações recebidas, como método, URL, código de status, tempo de resposta, etc.
10. multer (^1.4.5-lts.1):
  - O multer é um middleware para o gerenciamento de uploads de arquivos em aplicativos Express.
  - Ele lida com o recebimento e armazenamento de arquivos enviados por meio de formulários.
11. pg (^8.10.0) e pg-hstore (^2.3.4):
  - O pg é um driver Node.js para interagir com bancos de dados PostgreSQL.
  - O pg-hstore é um pacote que permite a manipulação de dados do tipo hstore no PostgreSQL.
12. pug (2.0.0-beta11):
  - O pug (anteriormente conhecido como Jade) é um mecanismo de modelo de alto desempenho para Node.js.
  - Ele é usado para criar templates HTML de forma mais eficiente e expressiva.
13. sequelize (^6.31.0):
  - O sequelize é um ORM (Object-Relational Mapping) para Node.js.
  - Ele fornece uma interface para interagir com bancos de dados relacionais, como MySQL, PostgreSQL, SQLite, etc.
14. stylus (0.54.5):
  - O stylus é um pré-processador CSS para Node.js.

- Ele permite escrever estilos de forma mais poderosa e dinâmica, com recursos como variáveis, mixins e aninhamento.

**15. dotenv (^16.0.3):**

- O dotenv é uma biblioteca que permite carregar variáveis de ambiente a partir de um arquivo .env.
- É útil para armazenar informações sensíveis, como chaves de API, senhas de banco de dados, etc.

**16. sequelize-cli (^6.6.1):**

- O sequelize-cli é uma interface de linha de comando para a biblioteca Sequelize.
- Ele fornece comandos úteis para criar e gerenciar migrações, inicializar modelos, etc.

Essas dependências desempenham um papel fundamental no funcionamento e nas funcionalidades do sistema gerenciador de cursos. Cada uma delas fornece recursos específicos que contribuem para o desenvolvimento do projeto.

## TELAS FRONTEND

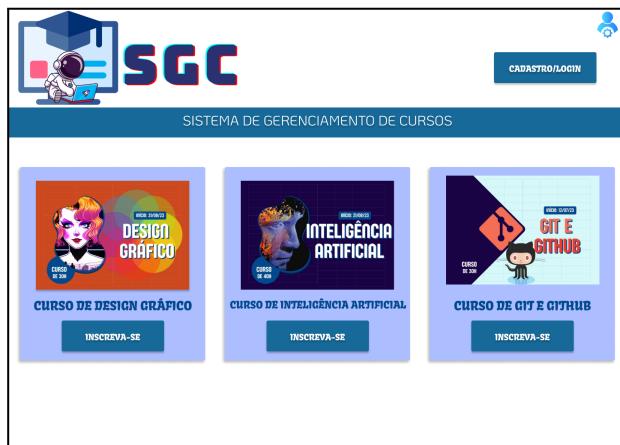


Figura 1: Página Inicial

Figura 2: Cadastro/Login



Figura 3: Página de descrição de Curso

## RELATÓRIO SOBRE A IMPLEMENTAÇÃO

O presente relatório descreve as funcionalidades e características do Sistema Gerenciador de Cursos, bem como as limitações encontradas durante o desenvolvimento. O sistema utiliza a linguagem JavaScript com conexão com o frontend e é integrado ao banco de dados PostgreSQL.

**Autenticação/autorização:** O sistema possui recursos de autenticação e autorização, permitindo o login e cadastro de usuários. No entanto, durante o desenvolvimento, foram encontradas dificuldades relacionadas à correção de importações e conexão com o frontend, o que impactou o pleno funcionamento dessas funcionalidades.

**Usuário root:** O sistema conta com um usuário root, criado durante a criação das tabelas, que têm permissão para conceder privilégios de administrador a novos usuários. Apesar de não ter sido implementado o seeder para facilitar essa atribuição, o sistema permite a realização dessa tarefa.

**Controle de permissões:** A exclusão de novas categorias de curso e novos cursos é restrita ao usuário root. Os administradores têm permissão para cadastrar e alterar cursos, mas não podem realizar exclusões. No entanto, a exclusão do aluno não foi implementada no sistema.

**Acesso à listagem de alunos:** Tanto o usuário root quanto os administradores têm acesso à listagem de alunos, permitindo visualizar informações relevantes sobre os estudantes.

**Funcionalidade de cursos:** O sistema apresenta uma funcionalidade de cursos, permitindo que os usuários marquem o curso como concluído utilizando um checkmark, de forma semelhante ao AVA/Moodle. No entanto, o seeder necessário para fechar e emitir certificados não foi implementado.

Cadastro de usuários e atualização de perfil: Os usuários podem se cadastrar no sistema e atualizar seus próprios perfis, incluindo informações como endereço, celular e foto do perfil. Além disso, eles podem se inscrever nos cursos disponíveis no sistema.

Relacionamento entre cursos e categorias: A partir da implementação do "associate", os cursos só podem estar vinculados a uma categoria, enquanto as categorias podem estar vinculadas a mais de um curso, seguindo a relação de 1 para n.

Limitações e funcionalidades não implementadas: Durante o desenvolvimento, algumas funcionalidades não puderam ser completamente implementadas. Essas incluem o lançamento da carga horária integralizada do aluno pelo administrador e a implementação do controle de certificados utilizando blockchain.

Hospedagem do sistema: O sistema foi publicado na plataforma de serviços em nuvem da AWS (Amazon Web Services), proporcionando acesso e disponibilidade aos usuários.

Este relatório apresentou uma visão geral do Sistema Gerenciador de Cursos, destacando suas funcionalidades, limitações e ações não implementadas. O sistema demonstra potencial para auxiliar no gerenciamento de cursos, embora alguns ajustes e aprimoramentos sejam necessários para sua plena utilização.

## **Acesso ao Projeto**

O projeto pode ser acessado através do link do repositório no GitHub:  
<https://github.com/nataliaalmada/BancoDeDados2-2023.2-Grupo1.git>

Além disso, o projeto está disponível na AWS, através do seguinte endereço IP:  
<https://52.67.162.154/>

Esses links fornecem acesso ao projeto e permitem explorar o código fonte, colaborar com o desenvolvimento e visualizar a aplicação em funcionamento na plataforma AWS.