

Aluno: Bruno Santos Ferreira

Projeto de Software Web



Domínio da Informação e Descrição geral do Produto:

Site de certificado de Cursos Preparatórios para Concursos O site tem como objetivo oferecer certificados de cursos preparatórios, qualidade em diversas áreas para concurseiros. Nosso foco é fornecer uma plataforma online acessível e conveniente para os candidatos que precisam de certificados para realizar a inscrição em concursos públicos. Nosso público-alvo principal são os concurseiros que buscam se preparar para concursos públicos em diversas áreas, como administrativa, jurídica, saúde, educação, entre outras. Além disso, também visamos profissionais que desejam aprimorar suas habilidades e conhecimentos através de cursos preparatórios específicos.

Os processos de negócio envolvem desde a seleção e desenvolvimento dos cursos preparatórios, a criação e disponibilização dos materiais de estudo, até a matrícula dos alunos e a emissão dos diplomas após a conclusão dos cursos.

O site deve operar em conformidade com todas as regulamentações e normas pertinentes ao fornecimento de cursos preparatórios e emissão de diplomas, garantindo a legalidade e validade dos diplomas emitidos aos alunos

O software deve permitir aos alunos explorar e se inscrever em uma variedade de cursos preparatórios disponíveis, acessar materiais de estudo, acompanhar seu progresso de aprendizagem, realizar as provas e, ao final, obter e imprimir seus diplomas de forma rápida e fácil.

Requisito Primário: Gerenciamento de Cursos

O requisito primário do site é permitir o gerenciamento eficaz dos cursos preparatórios oferecidos aos concurseiros. Isso inclui a capacidade de adicionar, atualizar e excluir cursos, garantindo que os alunos tenham acesso às informações mais recentes sobre os cursos disponíveis.

Principais Funcionalidades: Adicionar curso atualizar curso excluir curso visualização de detalhes do curso.

Caso de Uso: Gerenciamento de Cursos

Ator Principal: Sistema

Fluxo Principal:

Adicionar Curso: O sistema permite que novos cursos sejam adicionados automaticamente à plataforma, sem intervenção do administrador. Os cursos são adicionados com base em um sistema de integração com fornecedores de conteúdo educacional ou por meio de um sistema de entrada de dados automatizado.

Atualizar Curso: O sistema permite que os cursos existentes sejam atualizados automaticamente com informações atualizadas, como horários das aulas, materiais do curso e custo. Essas atualizações são realizadas com base em feeds de dados fornecidos pelos fornecedores de conteúdo educacional ou por meio de integrações automatizadas.

Emitir Certificado: Após a conclusão bem-sucedida de um curso por parte de um aluno, o sistema gera automaticamente um certificado para o aluno. O sistema verifica se o aluno atende aos requisitos de conclusão do curso, como assistir a todas as aulas e completar os

exercícios necessários. Uma vez verificado, o sistema gera um certificado personalizado para o aluno, contendo as informações do curso e do aluno. O certificado é automaticamente disponibilizado para download pelo aluno em sua conta no site.

Fluxos Alternativos: Se houver problemas técnicos durante o processo de adição, atualização ou emissão de certificados, o sistema deve registrar o erro e alertar os administradores para investigar e resolver o problema.

Fluxo de Exceção: Se ocorrer algum erro durante a emissão do certificado, como dados incompletos ou inconsistentes, o sistema deve registrar o erro e notificar o aluno sobre a situação, fornecendo orientações sobre como resolver o problema.

DER - Modelo Lógico Simplificado de primeiro momento

Entidades:

Curso: Atributos: ID (identificador único), Título, Descrição, Instrutor, Preço, Duração.

Aluno: Atributos: ID (identificador único), Nome, E-mail, Senha.

Certificado: Atributos: ID (identificador único), Data de Emissão.

Matrícula: Atributos: ID (identificador único), Data de Matrícula.

Relacionamentos: Relacionamento 1 para Muitos (1:N) - Curso para Matrícula: Um curso pode ter várias matrículas de alunos. Uma matrícula está associada a apenas um curso.
Cardinalidade: Um curso pode ter muitas matrículas, mas uma matrícula está vinculada a apenas um curso.

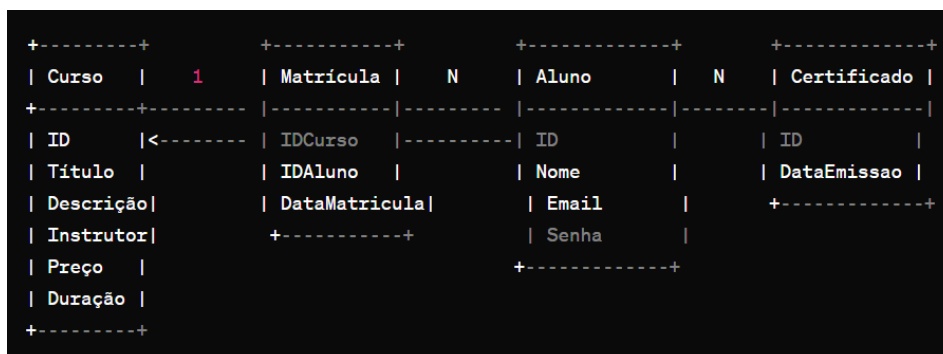
Relacionamento 1 para Muitos (1:N) - Aluno para Matrícula: Um aluno pode fazer várias matrículas em cursos diferentes. Uma matrícula está associada a apenas um aluno.
Cardinalidade: Um aluno pode ter muitas matrículas, mas uma matrícula está vinculada a apenas um aluno.

Relacionamento Muitos para Muitos (N:N) - Aluno para Certificado: Um aluno pode ter vários certificados de conclusão de cursos. Um certificado está associado a apenas um

aluno. Cardinalidade: Um aluno pode ter muitos certificados, e um certificado está vinculado a apenas um aluno.

Relacionamento Muitos para Muitos (N:N) - Curso para Certificado: Um curso pode ter vários certificados de conclusão emitidos. Um certificado está associado a apenas um curso. Cardinalidade: Um curso pode ter muitos certificados, e um certificado está vinculado a apenas um curso.

Diagrama de Entidade-Relacionamento (DER):



database mysql

Para criar o banco de dados no MySQL para o meu webapp de diplomas de cursos preparatórios, vamos seguir os requisitos e o modelo lógico que você forneceu. Vou criar as tabelas principais e suas relações.

Estrutura do Banco de Dados

Tabela **Curso**

Armazena informações sobre os cursos oferecidos.

- **ID** (identificador único)
- **Título**
- **Descricao**



- Instrutor
- Preço
- Duracao

Tabela *Aluno*

Armazena informações dos alunos.

- ID (identificador único)
- Nome
- Email
- Senha

Tabela *Certificado*

Armazena informações sobre os certificados emitidos.

- ID (identificador único)
- Data_Emissao
- Curso_ID (chave estrangeira para *Curso*)
- Aluno_ID (chave estrangeira para *Aluno*)

Tabela *Matricula*

Armazena informações sobre as matrículas dos alunos nos cursos.

- ID (identificador único)
- Data_Matricula
- Curso_ID (chave estrangeira para *Curso*)
- Aluno_ID (chave estrangeira para *Aluno*)

Código SQL

Aqui está o código SQL para criar as tabelas e definir as relações no MySQL:

```
CREATE DATABASE IF NOT EXISTS CursosPreparatorios;
USE CursosPreparatorios;

-- Tabela Curso
CREATE TABLE Curso (
    ID INT AUTO_INCREMENT PRIMARY KEY,
```



```
        Titulo VARCHAR(255) NOT NULL,  
        Descricao TEXT,  
        Instrutor VARCHAR(255),  
        Preco DECIMAL(10, 2),  
        Duracao INT  
    );  
  
-- Tabela Aluno  
CREATE TABLE Aluno (  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(255) NOT NULL,  
    Email VARCHAR(255) NOT NULL UNIQUE,  
    Senha VARCHAR(255) NOT NULL  
);  
  
-- Tabela Certificado  
CREATE TABLE Certificado (  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    Data_Emissao DATE NOT NULL,  
    Curso_ID INT,  
    Aluno_ID INT,  
    FOREIGN KEY (Curso_ID) REFERENCES Curso(ID),  
    FOREIGN KEY (Aluno_ID) REFERENCES Aluno(ID)  
);  
  
-- Tabela Matricula  
CREATE TABLE Matricula (  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    Data_Matricula DATE NOT NULL,  
    Curso_ID INT,  
    Aluno_ID INT,  
    FOREIGN KEY (Curso_ID) REFERENCES Curso(ID),  
    FOREIGN KEY (Aluno_ID) REFERENCES Aluno(ID)  
);
```

Explicação do Código



Criação do Banco de Dados:

```
CREATE DATABASE IF NOT EXISTS CursosPreparatorios;  
USE CursosPreparatorios;
```

1.
 - Cria um banco de dados chamado **CursosPreparatorios** se ele não existir e define-o como o banco de dados em uso.

Tabela **Curso**:

```
CREATE TABLE Curso (  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    Titulo VARCHAR(255) NOT NULL,  
    Descricao TEXT,  
    Instrutor VARCHAR(255),  
    Preco DECIMAL(10, 2),  
    Duracao INT  
);
```

2.
 - Cria a tabela **Curso** com as colunas **ID**, **Titulo**, **Descricao**, **Instrutor**, **Preco** e **Duracao**.

Tabela **Aluno**:

```
CREATE TABLE Aluno (  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(255) NOT NULL,  
    Email VARCHAR(255) NOT NULL UNIQUE,  
    Senha VARCHAR(255) NOT NULL  
);
```

3.
 - Cria a tabela **Aluno** com as colunas **ID**, **Nome**, **Email** e **Senha**. O campo **Email** é único.

Tabela **Certificado**:

```
CREATE TABLE Certificado (  
    ID INT AUTO_INCREMENT PRIMARY KEY,
```

```
Data_Emissao DATE NOT NULL,  
Curso_ID INT,  
Aluno_ID INT,  
FOREIGN KEY (Curso_ID) REFERENCES Curso(ID),  
FOREIGN KEY (Aluno_ID) REFERENCES Aluno(ID)  
);
```

- 4.
- Cria a tabela **Certificado** com as colunas **ID**, **Data_Emissao**, **Curso_ID** e **Aluno_ID**. As colunas **Curso_ID** e **Aluno_ID** são chaves estrangeiras que referenciam as tabelas **Curso** e **Aluno**, respectivamente.

Tabela **Matricula:**

```
CREATE TABLE Matricula (  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    Data_Matricula DATE NOT NULL,  
    Curso_ID INT,  
    Aluno_ID INT,  
    FOREIGN KEY (Curso_ID) REFERENCES Curso(ID),  
    FOREIGN KEY (Aluno_ID) REFERENCES Aluno(ID)  
);
```

- 5.
- Cria a tabela **Matricula** com as colunas **ID**, **Data_Matricula**, **Curso_ID** e **Aluno_ID**. As colunas **Curso_ID** e **Aluno_ID** são chaves estrangeiras que referenciam as tabelas **Curso** e **Aluno**, respectivamente.