

Formação <pra.elas> Banco BV | Turma 2

Grupo She Can do It <luiza.difante> <maira.oliveira> <michele.strehl>

REQUISITOS

1. ADMIN

- a. *Registrar cadastro de usuários.*
- b. *Classificar usuários em diferentes níveis administrativos (incluindo professores) e alunos.*
- c. *Acesso dos usuários por meio de validação de login e senha.*
- d. *Usuários administrativos poderão inserir, ler, atualizar e excluir informações conforme permissão de acesso.*
- e. *Alunos podem efetuar cadastro e realizar as atividades ali contempladas, conforme solicitação de matrícula nos cursos.*
- f. *Emitir relatório de usuários cadastrados.*
- g. *Emitir relatório de acessos e histórico de navegação.*

2. CURSOS

- a. *Registrar alunos.*
- b. *Registrar professores.*
- c. *Registrar cursos.*
- d. *Registrar matrícula de alunos.*
- e. *Registrar depoimentos dos alunos.*
- f. *Registrar andamento da trilha aprendizagem e realização das atividades.*
- g. *Registrar desempenho (notas) dos alunos e emitir certificados.*
- h. *Registrar avaliação do curso efetuada pelos alunos.*
- i. *Registrar e autorizar pagamentos.*
- j. *Emitir comprovantes de aquisição e/ou pagamento.*
- k. *Registrar e aplicar promoções.*
- l. *Registrar contato com a interface.*
- m. *Emitir relatório de oferta de cursos.*
- n. *Emitir relatório de cursos mais vendidos.*
- o. *Emitir relatório de alunos mais assíduos.*
- p. *Emitir sugestões de compra baseada em compras anteriores.*

3. POSTS BLOG

- a. *Registrar postagens.*
- b. *Registrar acessos.*
- c. *Registrar interações como curtidas e compartilhamentos.*
- d. *Registrar comentários e respostas a comentários.*
- e. *Emitir relatório de postagens por período.*
- f. *Emitir relatórios de postagens mais acessadas, curtidas, compartilhadas e comentadas.*
- g. *Emitir relatório de comentários e respostas a comentários.*

4. FORMULÁRIO DE CONTATO

- a. *Registrar envio de formulários de contato: geral e de professores.*
- b. *Registrar resposta ao contato.*
- c. *Emitir relatório de contatos.*

She Can Do It!



A. Tabela Usuário

- 1.1. ID Usuário = número
- 1.2. Nome = texto curto
- 1.3. Sobrenome = texto curto
- 1.4. CPF = dado comprimento fixo
- 1.5. Data de nascimento = data
- 1.6. Imagem de avatar = texto curto
- 1.7. Senha = texto curto
- 1.8. Confirmar senha = texto curto
- 1.9. Tipo de usuário = Dindin, Professor, Aluno
- 1.10. Data de cadastro = tempo real
- 1.11. Data de atualização = tempo real

```
create table usuario
(
  id_user int not null auto_increment,
  nome_user varchar(40) not null,
  sobrenome_user varchar(60) not null,
  cpf_user char(14) not null,
  dt_nascimento date not null,
  img_user varchar(60) null,
  senha_user varchar(20) not null,
  cf_senha_user varchar(20) not null,
  tipo_user enum('D', 'P', 'A') not null,
  dt_cadastro_user timestamp default
current_timestamp not null,
  dt_atualizacao_user timestamp default
current_timestamp null,
  constraint pk_usuario primary key(id_user)
);
```

B. Tabela Contato

- 1.1. ID Contato = número
- 1.2. ID Usuário = [Tabela Usuário \(A\)](#)
- 1.3. E-mail = texto curto
- 1.4. DDD = dado comprimento fixo
- 1.5. Celular = dado comprimento fixo

```
create table contato
(
  id_contato int not null auto_increment,
  id_user int not null,
  email_user varchar(80) not null,
  ddd_user char(4) not null,
  cel_user char(10) not null,
  constraint pk_contato primary key(id_contato),
  constraint fk_contato_user foreign key(id_user)
references usuario(id_user)
);
```

C. Tabela Professor

- 1.1. ID Professor = número
- 1.2. ID Usuário = [Tabela Usuário \(A\)](#)
- 1.3. ID Contato = [Tabela Contato \(B\)](#)
- 1.4. Mini biografia = texto longo
- 1.5. LinkedIn = texto curto
- 1.6. Status = ativo / inativo

```
create table professor
(
  id_prof int not null auto_increment,
  id_user int not null,
  id_contato int not null,
  minibio_prof varchar(300) not null,
  linkedin_prof varchar(120) null,
  status_prof enum('A', 'I') not null default 'A',
  constraint pk_professor primary key(id_prof),
  constraint fk_professor_user foreign key(id_user)
references usuario(id_user),
  constraint fk_professor_contato foreign key
(id_contato) references contato(id_contato)
);
```





D. Tabela Aluno

- 1.1. ID Aluno = número
- 1.2. ID Usuário = [Tabela Usuário \(A\)](#)
- 1.3. ID Contato = [Tabela Contato \(B\)](#)
- 1.4. Status = ativo / inativo

```
create table aluno
(
  id_aluno int not null auto_increment,
  id_user int not null,
  id_contato int not null,
  status_aluno enum('A', 'I') not null default 'A',
  constraint pk_aluno primary key (id_aluno),
  constraint fk_aluno_user foreign key (id_user)
  references usuario (id_user),
  constraint fk_aluno_contato foreign key (id_contato)
  references contato (id_contato)
);
```

E. Tabela Curso

- 1.1. ID curso = número
- 1.2. Nome curso = texto curto
- 1.3. Descrição curso = texto longo
- 1.4. ID Usuário = [Tabela Usuário \(A\)](#)
- 1.5. ID Professor = [Tabela Professor \(C\)](#)
- 1.6. Duração = texto curto (horas)
- 1.7. Curso gratuito = sim / não
- 1.8. Valor do curso = valor
- 1.9. Data de cadastro = tempo real
- 1.10. Data de atualização = tempo real
- 1.11. Imagem curso = texto curto

```
create table curso
(
  id_curso int not null auto_increment,
  nome_curso varchar(50) not null,
  descricao_curso varchar(400) not null,
  id_user int not null,
  id_prof int not null,
  duracao_curso varchar(20) not null,
  gratuidade_curso enum('S', 'N') not null default 'S',
  valor_curso decimal(6,2) not null,
  dt_cadastro timestamp default
  current_timestamp not null,
  dt_atualizacao timestamp default
  current_timestamp null,
  img_curso varchar(60) null,
  constraint pk_curso primary key (id_curso),
  constraint fk_curso_usuario foreign key (id_user)
  references usuario (id_user),
  constraint fk_curso_professor foreign key (id_prof)
  references professor (id_prof)
);
```

F. Tabela Matrícula

- 1.1. ID Matrícula = número
- 1.2. ID Usuário = [Tabela Usuário \(A\)](#)
- 1.3. ID Curso = [Tabela Curso \(E\)](#)
- 1.4. Data de início = data e hora
- 1.5. Data de término = data e hora
- 1.6. Data de cadastro = tempo real

```
create table matricula
(
  id_matricula int not null auto_increment,
  id_usuario int not null,
  id_curso int not null,
  dt_inicio datetime not null,
  dt_termino datetime not null,
  dt_cadastro timestamp default
  current_timestamp not null,
  constraint pk_matricula primary key (id_matricula),
  constraint fk_matricula_usuario foreign key (id_user)
  references usuario (id_user),
  constraint fk_matricula_curso foreign key (id_curso)
  references curso (id_curso)
);
```

G. Tabela Depoimento

- 1.1. ID Depoimento = número
- 1.2. ID Usuário = [Tabela Usuário \(A\)](#)
- 1.3. Depoimento = Texto longo
- 1.4. Data de cadastro = tempo real

```
create table depoimento
(
  id_depoimento int not null auto_increment,
  id_user int not null,
  depoimento varchar(130) not null,
  dt_cadastro timestamp default
  current_timestamp not null,
  constraint pk_depoimento primary key
  (id_depoimento),
  constraint fk_depoimento_user foreign key (id_user)
  references usuario (id_user)
);
```

H. Tabela Trilha Aprendizagem

I. Tabela Realização Atividade

J. Tabela Desempenho Aluno

K. Tabela Certificado

L. Tabela Avaliação Curso

M. Tabela Categoria Blog

- 1.1. ID Categoria = número
- 1.2. Título Categoria = Texto curto
- 1.3. Status = ativa / inativa

```
create table categoria_blog
(
  id_categoria int not null auto_increment,
  titulo_categoria varchar(30) not null,
  status_categoria enum('A', 'I') not null default 'A',
  constraint pk_categoria primary key (id_categoria)
);
```



N. Tabela Post Blog

- 1.1. ID Post = número
- 1.2. Autor Post = Tabela Usuário (A)
- 1.3. Categoria = Tabela Categoria (K)
- 1.4. Data da publicação = data e hora
- 1.5. Título post = Texto curto
- 1.6. Resumo post = Texto longo
- 1.7. Conteúdo post = Texto longo
- 1.8. Data de cadastro = tempo real
- 1.9. Data de edição = tempo real
- 1.10. Imagem post = texto curto
- 1.11. Status = ativa / inativa

```
create table post_blog
(
  id_post int not null auto_increment,
  id_autor int not null,
  id_categoria int not null,
  dt_publicacao datetime not null,
  titulo_post varchar(50) not null,
  resumo_post varchar(250) not null,
  conteudo_post varchar(5000) not null,
  dt_cadastro timestamp default
current_timestamp not null,
  dt_edicao timestamp default
current_timestamp not null,
  img_post varchar(60) null,
  status_post enum('A', 'I') not null default 'A',
  constraint pk_blog primary key(id_post),
  constraint fk_blog_autor foreign key(id_autor)
references usuario(id_user),
  constraint fk_blog_categoria foreign key
(id_categoria) references categoria_blog
(id_categoria)
);
```

DIAGRAMA ER

