

Mini Mundo

A ConstrAdmin é uma plataforma de gerenciamento de obras que auxilia na organização, visualização e coordenação eficiente de projetos de construção civil.

“Usuários” representam as pessoas que vão usar o sistema, podendo ser elas, administradores, supervisores, apontadores, empreiteiros ou um usuário comum. Esses usuários são diferenciados por roles em um controle de acesso. Eles possuem como atributos o **idUsuario** que seria seu identificador, **estoque_idEstoque** que é a chave estrangeira de estoque, **nome** do usuário, **lastName** que seria seu sobrenome, **cep** da sua residência, **cpf** do usuário, como endereço temos **país, cidade, estado** onde habita, **email** do usuário e **sexo** que diz o gênero do usuário.

- “Usuário” pertence apenas um estoque;
- “Usuário” possui pelo menos uma listas de obras;
- “Usuário” possui pelo menos um comentário;
- “Usuário” possui pelo menos um telefone;
- “Usuário” possui pelo menos uma lista de atividade;
- “Usuário” possui pelo menos uma models_has_roles;
- “Usuário” possui pelo menos uma models_has_permissions.

A tabela “telefoneUsuarios” vem da primeira forma normal em relação aos atributos multivalorados. Então, aqui são armazenados os números dos usuários. Nessa tabela, temos como atributos: **usuarios_idUsuarios** que é o identificador e a chave estrangeira da tabela e o próprio **telefone** do usuário.

- “telefoneUsuario” pertence a um usuário.

A “lista_obras” é uma tabela associativa criada a partir do relacionamento de usuários e obras. Nela temos o **obras_idObras** que é a chave estrangeira de obras e **usuarios_idUsuario** que é a chave estrangeira dos usuários. Essa tabela serve para identificar em qual obra um usuário específico está fazendo parte. Usuários podem ou não fazer parte de uma obra.

- “lista_obras” pertence a um usuário;
- “lista_obras” pertence a uma obra.

A tabela “obras” representa as obras registradas no sistema. Essa tabela possui o **idObras** que simboliza o identificador de uma obra, **nome** da obra, **statusObra** que caracteriza os estados de uma obra (podendo ela estar: **começando, em andamento ou finalizada**), **descricao** é um campo pro usuário descrever informações em relação à obra, **tamanho** representando a área da obra, o **tipo** corresponde ao tipo de obra (podendo ela ser: **residencial, comercial, industrial, infraestrutura, saneamento ou restauro**), como atributos de endereço temos: **logradouro, numResidencial, bairro, cidade e estado**, o **cep** do endereço, a **dtFinal** representando a data que a obra acabou, a **dtInicial** com a data que a obra foi iniciada, a estrutura do local onde vai ocorrer a obra (podendo ela ser:

metálica, concreto e madeira) e o **proposito** que representa o porque daquela obra está sendo feita.

- “Obra” possui pelo menos uma lista obras;
- “Obra” possui pelo menos um card_atividades;
- “Obra” possui pelo menos um arquivo;
- “Obra” possui pelo menos uma lista_materiais_necessarios.

A “lista_materiais_necessarios” é uma tabela associativa formada a partir do relacionamento entre a tabela “Obras” e “materiais_Estoque”. Ela tem como função administrar os materiais utilizados para cada obra, definindo assim a quantidade que pode ser utilizada para uma obra. Como atributos ela possui: **materiais_Estoque_idMateriais** que é seu identificador, **obras_idObras** que é a chave estrangeira de obras e quantidade que especifica a **quantidade** do material.

- “lista_materiais_necessarios” pertence a uma obra;
- “lista_materiais_necessarios” pertence a um materiaisEstoque.

Os materiais são representados no banco pela tabela chamada “materiaisEstoque”. Ele possui como atributos: **idMateriais** que é seu identificador na tabela, **estoque_idEstoque** que é a chave estrangeira da tabela de estoque, **kg** que seria o peso do material, **metros** que seria o tamanho do material caso ele tenha um que precise ser informado, **quantidade** que representa a quantidade de materiais, **dtVencimento** representando a data que o material vá vencer caso tenha alguma, **dtEntrada** que diz a data que o material foi registrado no sistema, a **dtSaida** que diz a data que o material foi retirado do sistema e o **Status_2** que diz se o material é **novo** ou **usado**, que basicamente fala se o material já foi utilizado e ainda resta alguma quantia a ser usada ou se é novo.

- “materiaisEstoque” pertence a apenas um estoque;
- “materiaisEstoque” possui uma lista de materiais necessários.

O sistema possui apenas um estoque compartilhado, ele está presente no banco de dados para deixar isso claro. Como atributos ele possui **idEstoque** que seria o identificador e o **nomeEstoque** que é como a empresa vai chamar esse estoque.

- “Estoque” possui pelo menos um usuário;
- “Estoque” possui materiais_Estoque.

A tabela “comentários” surge do relacionamento associativo entre usuários e atividades. Nela, além das chaves estrangeiras como **atividade_idAtividade** e **usuarios_idUsuario**, tem o seu identificador **idComentarios** e o atributo **comentario** onde um usuário com cargo específico de supervisor pode fazer algum comentário relacionado a uma atividade da obra.

- “comentários” pertence a um usuário;
- “comentários” pertence a uma atividade;

A tabela atividade representa as atividades realizadas em uma obra. Quanto aos atributos ela possui: **idAtividades** como seu identificador, **card_Atividades_idCard** é a chave

estrangeira da tabela card_Atividades, **nome** caracterizando o nome da atividade, **etiqueta** seria uma imagem para por na capa do card, **anexo** sendo uma imagem, vídeo ou documento que pode ser anexada dentro de um card, **descricao** da atividade sendo realizada, **dtFinal** a data que a atividade foi realizada, **dtInicial** sendo a data que a atividade foi iniciada e **status** podendo ele estar: **começando, andamento ou finalizado**.

- “atividade” possui pelo menos um comentário;
- “atividade” pertence a um card_Atividades;
- “atividade” possui pelo menos uma lista_atividades

A tabela “card_Atividades” representa os cartões contendo o **idCard** que é seu identificador, **titulo** do card e a chave estrangeira **obras_idObras**. Nele são colocadas as atividades realizadas.

- “card_Atividades” pertence a uma obra;
- “card_Atividades” possui pelo menos uma atividade;

“lista_atividades” é uma tabela criada também por um relacionamento associativo entre usuários e atividades. Nela são listadas as atividades designadas a um usuário específico, por isso ela possui como atributos: **usuarios_idUsuario** e **atividade_idAtividade** que são ambos chaves estrangeiras e ids da tabela.

- “lista_atividades” pertence a um usuário;
- “lista_atividades” pertence a uma atividade;

A tabela “permissions” vem de uma biblioteca do laravel, ela tem as permissões que um usuário pode realizar dentro do sistema. Quanto aos atributos temos: **id** como seu identificador, **name** como nome e o **guard_name** serve para que o laravel faça a autenticação e verificação.

- “permissions” possui pelo menos um model_has_permissions;
- “permissions” possui pelo menos um role_has_permissions.

Em “roles” se encontra as funções de um usuário no sistema. Essa tabela também vem da biblioteca do laravel. Seus atributos são: **id** representando seu identificador, **guard_name** onde o laravel faz a autenticação e verificação, **name** como nome, podendo ele ser: **administrador**: geralmente se referindo a empresa; **supervisor**: as pessoas de confiança da empresa que vão como o nome já diz, supervisionar o processo da obra e possuir um grande acesso a diversas funcionalidades do sistema; **apontador**: tendo cargo de administrar o estoque; **engenheiro**: são os usuários que tomam conta das questões relacionadas às atividades; **cliente**: a pessoa que solicitou a obra e **comum** são usuários que têm acesso assim como os clientes mas não realizam nada diretamente no sistema.

- “roles” possui pelo menos uma roles_has_permissions;
- “roles” possui pelo menos uma models_has_roles.

A “roles_has_permissions” surge de um relacionamento associativo entre as tabelas da biblioteca do laravel “roles” e “permissions”. Nela, as permissões são atribuídas para cada

função. Seus atributos são: **roles_id**, **permissions_id** ambos sendo identificadores e chaves estrangeiras.

- “roles_has_permissions” pertence a uma role;
- “roles_has_permissions” pertence a uma permission.

A tabela “model_has_roles” também é fruto de um relacionamento associativo, dessa vez, entre roles e usuários. Como o nome indica, ela dá uma função (role) para um usuário. Como atributos temos: **roles_id**, **usuarios_id** **Usuario** como chaves estrangeiras e identificadores da tabela e **model_type** serve para indicar qual tabela faz relação com as roles.

- “model_has_roles” pertence a um role;
- “model_has_roles” pertence a um usuário;

A tabela “model_has_permissions” também é de um relacionamento associativo, dessa vez, entre permissions e usuários. Ela serve para ligar permissões a, nesse caso, usuários. Como atributos temos: **roles_id**, **usuarios_id** **Usuario** como chaves estrangeiras e identificadores da tabela e **model_type** serve para indicar qual tabela faz relação com as permissions.

- Uma “model_has_permissions” pertence a uma permission;
- Uma “model_has_permissions” pertence a um usuário;