

Modelo App\Models\Marca (Tabela Marca)

- **materiais(): hasMany(Material::class)**
 - **Significado:** Uma Marca pode ter muitos Material associados.
 - **Como funciona:** O Eloquent irá procurar na tabela Material por registros onde a coluna cod_marca corresponda ao cod_marca (chave primária) desta Marca.
 - **Uso:** \$marca->materiais (retorna uma coleção de objetos Material).
-

Modelo App\Models\Categoria (Tabela Categoria)

- **materiais(): hasMany(Material::class)**
 - **Significado:** Uma Categoria pode ter muitos Material associados.
 - **Como funciona:** O Eloquent irá procurar na tabela Material por registros onde a coluna cod_categoria corresponda ao cod_categoria (chave primária) desta Categoria.
 - **Uso:** \$categoria->materiais (retorna uma coleção de objetos Material).
-

Modelo App\Models\Modelo (Tabela Modelo)

- **materiais(): hasMany(Material::class)**
 - **Significado:** Um Modelo pode ter muitos Material associados.
 - **Como funciona:** O Eloquent irá procurar na tabela Material por registros onde a coluna cod_modelo corresponda ao cod_modelo (chave primária) deste Modelo.
 - **Uso:** \$modelo->materiais (retorna uma coleção de objetos Material).
-

Modelo App\Models\MaterialEstado (Tabela Material_Estado)

- **materiais(): hasMany(Material::class)**
 - **Significado:** Um MaterialEstado pode ter muitos Material associados.

- **Como funciona:** O Eloquent irá procurar na tabela Material por registos onde a coluna cod_estado corresponda ao cod_estado (chave primária) deste MaterialEstado.
- **Uso:** \$materialEstado->materiais (retorna uma coleção de objetos Material).

Modelo App\Models\Nivel (Tabela Nivel)

- **funcionarios(): hasMany(Funcionario::class)**
 - **Significado:** Um Nivel pode ter muitos Funcionario associados.
 - **Como funciona:** O Eloquent irá procurar na tabela Funcionarios por registos onde a coluna cod_nivel corresponda ao cod_nivel (chave primária) deste Nivel.
 - **Uso:** \$nivel->funcionarios (retorna uma coleção de objetos Funcionario).

Modelo App\Models\Funcao (Tabela Funcao)

- **funcionarios(): hasMany(Funcionario::class)**
 - **Significado:** Uma Funcao pode ter muitos Funcionario associados.
 - **Como funciona:** O Eloquent irá procurar na tabela Funcionarios por registos onde a coluna cod_funcao corresponda ao cod_funcao (chave primária) desta Funcao.
 - **Uso:** \$funcao->funcionarios (retorna uma coleção de objetos Funcionario).

Modelo App\Models\FuncionarioEstado (Tabela Funcionario_Estado)

- **funcionarios(): hasMany(Funcionario::class)**
 - **Significado:** Um FuncionarioEstado pode ter muitos Funcionario associados.
 - **Como funciona:** O Eloquent irá procurar na tabela Funcionarios por registos onde a coluna cod_estado corresponda ao cod_estado (chave primária) deste FuncionarioEstado.

- **Uso:** \$funcionarioEstado->funcionarios (retorna uma coleção de objetos Funcionario).
-

Modelo App\Models\TipoServico (Tabela TiposServico)

- **servicos(): hasMany(Servico::class)**
 - **Significado:** Um TipoServico pode ter muitos Servico associados.
 - **Como funciona:** O Eloquent irá procurar na tabela Servicos por registos onde a coluna cod_tipo_servico corresponda ao cod_tipo_servico (chave primária) deste TipoServico.
 - **Uso:** \$tipoServico->servicos (retorna uma coleção de objetos Servico).
-

Modelo App\Models\Localizacao (Tabela Localizacoes)

- **servicos(): hasMany(Servico::class)**
 - **Significado:** Uma Localizacao pode ter muitos Servico associados.
 - **Como funciona:** O Eloquent irá procurar na tabela Servicos por registos onde a coluna cod_local_servico corresponda ao cod_local_servico (chave primária) desta Localizacao.
 - **Uso:** \$localizacao->servicos (retorna uma coleção de objetos Servico).
-

Modelo App\Models\Cliente (Tabela Clientes)

- **servicos(): hasMany(Servico::class)**
 - **Significado:** Um Cliente pode ter muitos Servico associados.
 - **Como funciona:** O Eloquent irá procurar na tabela Servicos por registos onde a coluna cod_cliente corresponda ao cod_cliente (chave primária) deste Cliente.
 - **Uso:** \$cliente->servicos (retorna uma coleção de objetos Servico).
-

Modelo App\Models\Material (Tabela Material)

- **categoria(): belongsTo(Categoria::class)**

- **Significado:** Um Material pertence a uma Categoria.
- **Como funciona:** O Eloquent irá procurar na tabela Categoria por um registo onde a chave primária (cod_categoria) corresponda ao valor na coluna cod_categoria deste Material.
- **Uso:** \$material->categoria (retorna um objeto Categoria).
- **marca(): belongsTo(Marca::class)**
 - **Significado:** Um Material pertence a uma Marca.
 - **Como funciona:** Procura na tabela Marca onde a PK (cod_marca) corresponda a material.cod_marca.
 - **Uso:** \$material->marca (retorna um objeto Marca).
- **modelo(): belongsTo(Modelo::class)**
 - **Significado:** Um Material pertence a um Modelo.
 - **Como funciona:** Procura na tabela Modelo onde a PK (cod_modelo) corresponda a material.cod_modelo.
 - **Uso:** \$material->modelo (retorna um objeto Modelo).
- **estado(): belongsTo(MaterialEstado::class)**
 - **Significado:** Um Material pertence a um MaterialEstado.
 - **Como funciona:** Procura na tabela Material_Estado onde a PK (cod_estado) corresponda a material.cod_estado.
 - **Uso:** \$material->estado (retorna um objeto MaterialEstado).
- **avarias(): hasMany(Avaria::class)**
 - **Significado:** Um Material pode ter muitas Avaria.
 - **Como funciona:** Procura na tabela Avarias onde a coluna cod_material corresponda à PK (cod_material) deste Material.
 - **Uso:** \$material->avarias (retorna uma coleção de objetos Avaria).
- **perdas(): hasMany(Perda::class)**
 - **Significado:** Um Material pode ter muitas Perda.
 - **Como funciona:** Procura na tabela Perdas onde a coluna cod_material corresponda à PK (cod_material) deste Material.
 - **Uso:** \$material->perdas (retorna uma coleção de objetos Perda).

- **servicos(): belongsToMany(Servico::class)->using(ServicoEquipamento::class)**
 - **Significado:** Um Material pode estar associado a muitos Servicos (relação muitos-para-muitos).
 - **Como funciona:** Usa a tabela pivot Servico_Equipamento. Procura na Servico_Equipamento por registos onde cod_material corresponda à PK deste Material, depois usa o cod_servico desses registos para encontrar os Servicos correspondentes. O ->using() permite aceder a colunas extras na tabela pivot (data_levantamento, data_devolucao) através do modelo ServicoEquipamento na relação (\$material->servicos[0]->pivot->data_levantamento).
 - **Uso:** \$material->servicos (retorna uma coleção de objetos Servico).
-

Modelo App\Models\Funcionarios (Tabela Funcionarios)

- **nivel(): belongsTo(Nivel::class)**
 - **Significado:** Um Funcionario pertence a um Nivel.
 - **Como funciona:** Procura na tabela Nivel onde a PK (cod_nivel) corresponda a funcionarios.cod_nivel.
 - **Uso:** \$funcionario->nivel (retorna um objeto Nivel).
- **funcao(): belongsTo(Funcao::class)**
 - **Significado:** Um Funcionario pertence a uma Funcao.
 - **Como funciona:** Procura na tabela Funcao onde a PK (cod_funcao) corresponda a funcionarios.cod_funcao.
 - **Uso:** \$funcionario->funcao (retorna um objeto Funcao).
- **estado(): belongsTo(FuncionarioEstado::class)**
 - **Significado:** Um Funcionario pertence a um FuncionarioEstado.
 - **Como funciona:** Procura na tabela Funcionario_Estado onde a PK (cod_estado) corresponda a funcionarios.cod_estado.
 - **Uso:** \$funcionario->estado (retorna um objeto FuncionarioEstado).
- **user(): hasOne(User::class)**
 - **Significado:** Um Funcionario **pode ter** um User associado (para login).

- **Como funciona:** Procura na tabela users por um registo onde a coluna cod_funcionario corresponda à PK (cod_funcionario) deste Funcionario.
 - **Uso:** \$funcionario->user (retorna um objeto User ou null).
 - **servicos(): belongsToMany(Servico::class)->using(ServicoFuncionario::class)**
 - **Significado:** Um Funcionario pode estar associado a muitos Servicos (relação muitos-para-muitos).
 - **Como funciona:** Usa a tabela pivot Servico_Funcionario. Procura na Servico_Funcionario por registos onde cod_funcionario corresponda à PK deste Funcionario, depois usa o cod_servico desses registos para encontrar os Servicos correspondentes. O ->using() permite aceder a colunas extras na tabela pivot (data_alocacao_inicio, etc.) através do modelo ServicoFuncionario.
 - **Uso:** \$funcionario->servicos (retorna uma coleção de objetos Servico).
-

Modelo App\Models\Servicos (Tabela Servicos)

- **cliente(): belongsTo(Cliente::class)**
 - **Significado:** Um Servico pertence a um Cliente.
 - **Como funciona:** Procura na tabela Clientes onde a PK (cod_cliente) corresponda a servicos.cod_cliente.
 - **Uso:** \$servico->cliente (retorna um objeto Cliente).
- **tipoServico(): belongsTo(TipoServico::class)**
 - **Significado:** Um Servico pertence a um TipoServico.
 - **Como funciona:** Procura na tabela TiposServico onde a PK (cod_tipo_servico) corresponda a servicos.cod_tipo_servico.
 - **Uso:** \$servico->tipoServico (retorna um objeto TipoServico).
- **localizacao(): belongsTo(Localizacao::class)**
 - **Significado:** Um Servico pertence a uma Localizacao. (Esta relação é nullable).

- **Como funciona:** Procura na tabela Localizacoes onde a PK (cod_local_servico) corresponda a servicos.cod_local_servico.
 - **Uso:** \$servico->localizacao (retorna um objeto Localizacao ou null).
- **avarias(): hasMany(Avaria::class)**
 - **Significado:** Um Servico pode ter muitas Avaria.
 - **Como funciona:** Procura na tabela Avarias onde a coluna cod_servico corresponda à PK (cod_servico) deste Servico.
 - **Uso:** \$servico->avarias (retorna uma coleção de objetos Avaria).
- **perdas(): hasMany(Perda::class)**
 - **Significado:** Um Servico pode ter muitas Perda.
 - **Como funciona:** Procura na tabela Perdas onde a coluna cod_servico corresponda à PK (cod_servico) deste Servico.
 - **Uso:** \$servico->perdas (retorna uma coleção de objetos Perda).
- **detalhesCasamento(): hasOne(ServicoDetalhesCasamento::class)**
 - **Significado:** Um Servico pode ter um registo de detalhes de casamento.
 - **Como funciona:** Procura na tabela Servico_Detalhes_Casamento onde a PK (cod_servico) corresponda à PK (cod_servico) deste Servico.
 - **Uso:** \$servico->detalhesCasamento (retorna um objeto ServicoDetalhesCasamento ou null). *(Similar para Batizado, ComunhaoParticular, ComunhaoGeral, EvCorporativo)*
- **funcionarios(): belongsToMany(Funcionario::class)->using(ServicoFuncionario::class)**
 - **Significado:** Um Servico pode ter muitos Funcionarios associados (relação muitos-para-muitos).
 - **Como funciona:** Usa a tabela pivot Servico_Funcionario. Acessa os dados extras da pivot via modelo ServicoFuncionario.
 - **Uso:** \$servico->funcionarios (retorna uma coleção de objetos Funcionario).
- **materiais(): belongsToMany(Material::class)->using(ServicoEquipamento::class)**

- **Significado:** Um Servico pode ter muitos Material (Equipamentos) associados (relação muitos-para-muitos).
 - **Como funciona:** Usa a tabela pivot Servico_Equipamento. Acessa os dados extras da pivot via modelo ServicoEquipamento.
 - **Uso:** \$servico->materiais (retorna uma coleção de objetos Material).
-

Modelo App\Models\Avaria (Tabela Avarias)

- **material(): belongsTo(Material::class)**
 - **Significado:** Uma Avaria pertence a um Material.
 - **Como funciona:** Procura na tabela Material onde a PK (cod_material) corresponda a avarias.cod_material.
 - **Uso:** \$avaria->material (retorna um objeto Material).
 - **servico(): belongsTo(Servico::class)**
 - **Significado:** Uma Avaria pertence a um Servico. (Relação nullable).
 - **Como funciona:** Procura na tabela Servicos onde a PK (cod_servico) corresponda a avarias.cod_servico.
 - **Uso:** \$avaria->servico (retorna um objeto Servico ou null).
-

Modelo App\Models\Perda (Tabela Perdas)

- **material(): belongsTo(Material::class)**
 - **Significado:** Uma Perda pertence a um Material.
 - **Como funciona:** Procura na tabela Material onde a PK (cod_material) corresponda a perdas.cod_material.
 - **Uso:** \$perda->material (retorna um objeto Material).
 - **servico(): belongsTo(Servico::class)**
 - **Significado:** Uma Perda pertence a um Servico. (Relação nullable).
 - **Como funciona:** Procura na tabela Servicos onde a PK (cod_servico) corresponda a perdas.cod_servico.
 - **Uso:** \$perda->servico (retorna um objeto Servico ou null).
-

Modelos de Detalhes de Serviço (ServicoDetalhesCasamento, ServicoDetalhesBatizado, etc.)

- **servico(): belongsTo(Servico::class)**
 - **Significado:** Cada registo de detalhe (Casamento, Batizado, etc.) pertence a um Servico.
 - **Como funciona:** Procura na tabela Servicos onde a PK (cod_servico) corresponda à PK/FK (cod_servico) deste registo de detalhe.
 - **Uso:** \$detalheCasamento->servico (retorna um objeto Servico).
-

Modelo App\Models\ServicoFuncionario (Tabela Pivot Servico_Funcionario)

- **servico(): belongsTo(Servico::class)**
 - **Significado:** Um registo nesta tabela pivot pertence a um Servico.
 - **Como funciona:** Procura na tabela Servicos onde a PK (cod_servico) corresponda a Servico_Funcionario.cod_servico.
 - **Uso:** \$servicoFuncionarioPivot->servico (retorna um objeto Servico).
 - **funcionario(): belongsTo(Funcionario::class)**
 - **Significado:** Um registo nesta tabela pivot pertence a um Funcionario.
 - **Como funciona:** Procura na tabela Funcionarios onde a PK (cod_funcionario) corresponda a Servico_Funcionario.cod_funcionario.
 - **Uso:** \$servicoFuncionarioPivot->funcionario (retorna um objeto Funcionario).
-

Modelo App\Models\ServicoEquipamento (Tabela Pivot Servico_Equipamento)

- **servico(): belongsTo(Servico::class)**
 - **Significado:** Um registo nesta tabela pivot pertence a um Servico.
 - **Como funciona:** Procura na tabela Servicos onde a PK (cod_servico) corresponda a Servico_Equipamento.cod_servico.

- **Uso:** \$servicoEquipamentoPivot->servico (retorna um objeto Servico).
 - **material(): belongsTo(Material::class)**
 - **Significado:** Um registo nesta tabela pivot pertence a um Material.
 - **Como funciona:** Procura na tabela Material onde a PK (cod_material) corresponda a Servico_Equipamento.cod_material.
 - **Uso:** \$servicoEquipamentoPivot->material (retorna um objeto Material).
-

Modelo App\Models\User (Tabela users)

- **funcionario(): belongsTo(Funcionario::class)**
 - **Significado:** Um User pertence a um Funcionario.
 - **Como funciona:** Procura na tabela Funcionarios onde a PK (cod_funcionario) corresponda a users.cod_funcionario.
 - **Uso:** \$user->funcionario (retorna um objeto Funcionario ou null).