

Desafio Ruby on Rails

O desafio consiste em implementar uma aplicação Web utilizando o framework Ruby on Rails (versão 4.x ou 5.x), e um banco de dados relacional SQLite, MySQL ou Postgres, a partir de uma modelagem de dados inicial desnormalizada, que deve ser normalizada para a implementação da solução.

A Aplicação

Você vai criar uma aplicação de cadastro de pedidos de compra, a partir de uma modelagem inicial, com as seguintes funcionalidades:

- CRUD de clientes.
- CRUD de produtos.
- CRUD de pedidos de compra, com status (Em Aberto, Pago ou Cancelado).
- Cada CRUD:
 - deve ser filtrável e ordenável por qualquer campo, e possuir paginação de 20 itens.
 - deve possuir formulários para criação e atualização de seus itens.
 - deve permitir a deleção de qualquer item de sua lista.
- Barra de navegação entre os CRUDs.
- Links para os outros CRUDs nas listagens (Ex: link para o detalhe do cliente da compra na lista de pedidos de compra)
- API Rest JSON para todos os CRUDS listados acima.

- Multilíngue (Português e Inglês).

Modelo de dados

A modelagem inicial para a implementação solução é a seguinte:

Pedido			
	Column Name	Data Type	Allow Nulls
	NumeroPedido	int	<input type="checkbox"/>
	NomeCliente	varchar(100)	<input type="checkbox"/>
	CPF	char(11)	<input type="checkbox"/>
	Email	nchar(10)	<input checked="" type="checkbox"/>
	DtPedido	smalldatetime	<input type="checkbox"/>
	CodBarras	varchar(20)	<input type="checkbox"/>
	NomeProduto	varchar(100)	<input checked="" type="checkbox"/>
	ValorUnitario	money	<input type="checkbox"/>
	Quantidade	int	<input type="checkbox"/>
			<input type="checkbox"/>

Você deve alterar esta modelagem para que a mesma cumpra com as três primeiras formas normais.

Além disso, a implementação deste modelo em um banco de dados relacional deve ser realizada levando em consideração os seguintes requisitos:

- O banco de dados deve ser criado utilizando Migrations do framework Ruby on Rails.
- Implementação das validações necessárias, e relacionamentos nas classes de Model do framework Ruby on Rails.

Requisitos da aplicação

Para que a solução do desafio seja aceita, ela deve passar pelos seguintes requisitos:

- O modelo de dados deve cumprir com as três primeiras formas normais e ser ACID.
- O front-end deve ser responsivo, ou seja, a interface deve se adaptar a diferentes tamanhos de tela.
- A aplicação deve ser implementada utilizando todas as features principais do framework Ruby on Rails (ActionController e Resourceful Routes, ActiveRecord, Views ERB , Asset Pipeline, ActiveMigrations, I18n, etc.).

Entrega do desafio

O código da solução do desafio pode ser entregue através de um link para um repositório no [Github](#), [Bitbucket](#), ou em formato zip por e-mail.

OBS: É recomendável subir a solução em um repositório do [Github](#) ou do [Bitbucket](#) para facilitar a correção e solução de dúvidas.

Dicas

- Você pode usar qualquer gem existente no mercado para implementar funcionalidades como filtros, paginação, ordenação e navegação.

- Utilize frameworks CSS como Bootstrap ou Materialize para implementar a responsividade do Front-end.
- Crie uma camada de regra de negócios entre o Controller e o Model para elaborar uma estrutura de aplicação mais limpa e escalável.
- Leia os [Ruby on Rails guides](#).

Bônus

- Implementar autenticação de usuário na aplicação, criando um novo modelo para usuários.
- Implementação de testes unitários, utilizando preferencialmente RSpec.
- Permitir que o usuário mude o número de itens por página.
- Permitir deleção em massa de itens nos CRUDs.
- Implementar aplicação de desconto em alguns pedidos de compra.
- Implementar a camada de Front-End em aplicação 100% javascript, utilizando as tecnologias Node.js, Webpack e Babel. Esta camada deve se comunicar com o Back-End Rails através de sua API Rest JSON.
- Implementar testes unitários para a camada de Front-End (javascript).