HORIZON
INOVAÇÃO E TECNOLOGIA

Edifício Office & Pool
Rua Alceu Amoroso Lima, 172, Sala 701
Caminho das Árvores, Salvador/BA
CEP: 41820-770
CNPJ: 14.497.724/0001-05

Hands-On!

Desenvolvedor Backend Pleno - API REST

Olá dev! Se chegou aqui, tá afim de desafio mesmo, ein?! Então vamos para uma parada bem legal.

Já pensou em ter sua própria empresa aérea e poder ditar os valores das passagens? Essa é a hora! Vai ser legal com a gente e já dar uns descontinhos né?!

Vamos começar a próxima startup unicórnio do Brasil, desenvolvendo a API REST do seu sistema de passagens e controle de bagagens, usando Laravel, e boas práticas de programação, padrões de projeto, principalmente aquelas descritas pelo mestre dos magos, Fowler.

Regras do Negócio:

- Cada voo tem um aeroporto de origem e uma de destino, uma numeração única, data e hora de partida.
- O voo não pode ter como origem e destino o mesmo aeroporto, e esses aeroportos não podem estar situados na mesma cidade.
- Os voos devem possuir no mínimo uma classe, contendo um tipo de classe, a quantidade de assentos, e o valor para cada assento.
- A quantidade de passagens para um voo é limitada pela quantidade de assentos disponíveis em cada classe. E não deve haver duas ou mais classes do mesmo tipo no mesmo voo.
- Os aeroportos ficam situados nas cidades, e cada um possui um nome e o próprio código aeroportuário IATA que é único.
- As cidades devem possuir nome e a unidade federativa.
- As passagens aéreas representam um assento da classe escolhida. Uma passagem contém um número de identificação único para controle interno, os dados do passageiro (nome, cpf, data de nascimento), o preço total da passagem, e a classe. Não podem ser vendidas mais passagens do que a capacidade máxima da classe.
- É possível pesquisar por passagens informando os aeroportos de origem e destino, e a data que deseja. Opcionalmente podem ser filtrados por valores. O resultado não pode mostrar voos sem assentos disponíveis. E não podem ser apresentados voos passados.
- As passagens devem ser vinculadas ao visitante que as comprou, identificando ele com o nome, cpf, data de nascimento e e-mail.

Edifício Office & Pool
Rua Alceu Amoroso Lima, 172, Sala 701
Caminho das Árvores, Salvador/BA
CEP: 41820-770
CNPJ: 14.497.724/0001-05

- Cada passagem permite no máximo uma bagagem, e se houver despache deve ser acrescida uma taxa de 10% do valor do assento ao valor da passagem. As bagagens possuem uma numeração única para controle interno.
- Durante a compra, o visitante deverá informar os dados do comprador, a quantidade de passagens, a classe escolhida, e a lista com os dados de cada passageiro e se há bagagem a ser despachada.
- Depois de compradas as passagens, é possível emitir os vouchers de cada uma contendo o número da passagem, número do voo, origem e destino, e o passageiro, além de indicar se há despacho de bagagem.
- Caso haja despacho de bagagem, é possível emitir também a etiqueta da bagagem com os números de identificação da passagem e da bagagem, e o nome do passageiro.
- O limite de tempo para emissão tanto da etiqueta quanto do voucher é de no máximo 5 horas antes do voo.

- Deve haver duas possibilidades de acesso: a gestão de voos, bagagens, passageiros e passagens só pode ser realizada por gestores autenticados; as buscas por voos, compras, consulta de passagens feita por um comprador, emissões de vouchers e etiquetas, podem ser realizadas por qualquer pessoa.

Funcionalidades:

- Autenticação de gestores
- Listar aeroportos
- Cadastrar, listar, alterar e cancelar voos
- Cadastrar e alterar preços das passagens
- Listar passageiros de um voo
- Comprar passagens
- Obter as passagens compradas pelo CPF do comprador
- Cancelar compra
- Emitir o voucher da passagem
- Emitir etiqueta de bagagem

Requisitos:

- Todas as funcionalidades devem ser implementadas sob o paradigma Rest, pfv...
- O banco de dados é da sua escolha (MySQL, PostgreSQL etc.), use o que você achar melhor
- Já seu código vai ter que ser em PHP com o framework Laravel na última versão estável. Foi mal, mas é importante...
- Arquitetura limpa, coesa e bem estruturada. O simples e bem feito fica melhor do que o enfeitado e complexo.
- Erro bem descrito protege de debugs longos.

Edifício Office & Pool
Rua Alceu Amoroso Lima, 172, Sala 701
Caminho das Árvores, Salvador/BA
CEP: 41820-770
CNPJ: 14.497.724/0001-05

- Documentar a api descrevendo passo a passo para subir em um servidor, configurações necessárias incluindo variáveis de ambiente (se houver), identificando rotas disponíveis, estruturas de dados de envio e retorno. Outros devs irão utilizar o que você construiu.

No que você deve focar:

- Disponibilizar o código publicamente no GitHub
- Corretude da sintaxe dos verbos HTTP e requisições Rest, assim a gente vai ver melhor o trampo massa que você fez
- Criações e Consultas primeiramente, deixe atualizações e exclusões de dados pro final
- Deixar a API funcional, rodando certinha e disponível em algum serviço web (Heroku, WebHost, AWS etc.) de sua preferência

Ufa... Parece muito? Que nada, você tira de letra. Dev no Brasil é outro nível.

Mas não se preocupe. Qualquer dúvida pode falar com a gente, que nós estamos prontos para te ajudar.

Vamos nessa!