

Prova Prática para Desenvolvedor Odoo

Instruções:

- 1) Esta prova tem o objetivo de avaliar habilidades práticas em desenvolvimento Python e Odoo, incluindo algoritmos, modelagem de sistemas orientados a objetos, e conhecimentos específicos nos módulos de contabilidade e contas a pagar do Odoo.
- 2) Para a parte prática em Odoo, você pode assumir que os modelos básicos e estrutura já estão configurados.
- 3) Crie um repositório público no github
- 4) Envie o link do repositório para o e-mail rh@mutualizo.com com o título da mensagem "Prova Desenvolvedor Odoo - Seu Nome"

Questão 1: Sequência de Fibonacci em Python

Implemente uma função em Python chamada `fibonacci(n)` que retorna o n-ésimo termo da sequência de Fibonacci.

Exemplo de entrada/saída:

```
Python
fibonacci(6) # Deve retornar 8 (os primeiros termos da
sequência são: 0, 1, 1, 2, 3, 5, 8)
```

Questão 2: Modelagem e Arquitetura de Sistemas Web com API REST

Você está trabalhando em um sistema de e-commerce. Modele classes em Python que representem os seguintes elementos: `Produto`, `Carrinho`, e `Pedido`.

- a) `Produto`: Deve ter atributos como `nome`, `preço` e `estoque`.
- b) `Carrinho`: Deve ser capaz de adicionar e remover produtos, calcular o total e finalizar a compra.
- c) `Pedido`: Deve conter informações sobre os produtos comprados e o total do pedido.

Implemente também uma API REST para fornecer acesso a esses recursos.

Questão 3: Módulo de Contabilidade do Odoo

Suponha que você esteja desenvolvendo um sistema de contabilidade para uma empresa com operações internacionais. A empresa lida com várias moedas e precisa de uma funcionalidade para converter automaticamente transações em diferentes moedas para a

moeda base da empresa.

Tarefa:

- 1) Crie um novo modelo chamado ``account.foreign.exchange.rate`` que permita o registro de taxas de câmbio entre diferentes moedas. O modelo deve conter os seguintes campos:
 - a) ``date`` (Date): Data da taxa de câmbio.
 - b) ``currency_from_id`` (Many2one): Moeda de origem.
 - c) ``currency_to_id`` (Many2one): Moeda de destino.
 - d) ``exchange_rate`` (Float): Taxa de câmbio.
- 2) Modifique o modelo ``account.move`` para incluir um campo ``currency_id`` que represente a moeda em que a transação está sendo registrada. Certifique-se de que este campo esteja relacionado com o modelo ``res.currency``.
- 3) Implemente uma função no Odoo que, ao criar um lançamento contábil, verifique se a moeda da transação é diferente da moeda base da empresa. Em caso afirmativo, a função deve converter o valor da transação para a moeda base usando a taxa de câmbio correspondente.

Observações:

- Considere que os modelos ``res.currency`` e ``account.move`` já estão definidos no sistema.

Questão 4. Integração com Sistemas Externos no Módulo de Contas a Receber do Odoo

Suponha que você esteja trabalhando em um projeto de implementação do módulo de contas a receber do Odoo para uma empresa que precisa integrar dados de faturas com seu sistema de gestão financeira externo. O sistema externo utiliza um serviço RESTful para receber e registrar informações de faturas.

Tarefa:

1. Crie um módulo personalizado no Odoo chamado ``account_invoice_integration`` que permita a integração de faturas com o sistema financeiro externo. Este módulo deve incluir um modelo chamado ``account.invoice.integration`` com os seguintes campos:
 - a. ``invoice_id`` (Many2one): Relacionamento com a fatura no Odoo.
 - b. ``external_system_id`` (Char): ID da fatura no sistema externo.
 - c. ``status`` (Selection): Status da integração (pendente, sucesso, erro).
 - d. ``response_message`` (Text): Mensagem de resposta do sistema externo.
2. Implemente uma função no módulo personalizado que, quando uma fatura é validada e confirmada no Odoo, envie automaticamente os dados da fatura para o sistema externo via API REST. A função deve atualizar o registro no modelo

`account.invoice.integration` com o status da integração e a mensagem de resposta do sistema externo.

3. Simule a integração de uma fatura de exemplo no sistema Odoo. Crie uma fatura no Odoo e confirme-a. Certifique-se de que os dados da fatura sejam enviados corretamente ao sistema externo e que o registro no modelo `account.invoice.integration` seja atualizado com o status e a mensagem de resposta.

Observações:

- Considere que o sistema externo já possui uma API RESTful configurada para receber informações de faturas e retornar mensagens de resposta.