

Vendinha

Cenário

Uma vendinha precisa informatizar o controle de contas de seus clientes (dívidas penduradas) para facilitar a busca e o cadastro desses dados que antes eram feitas por papel. O cliente chega na loja, faz a compra e pede para o atendente pendurar para que seja acertado no final do mês.

Pensando nisso, é necessário criar um sistema simples de cadastro para que o dono da venda consiga controlar as dívidas de seus clientes.

Requisitos

- O sistema deve permitir o cadastro, recuperação, atualização e exclusão de clientes
 - Os dados necessários para se criar a conta são:
 - Nome completo *
 - CPF *: deve ser um número válido de CPF
 - Data de Nascimento *
 - E-mail: e-mail para contato
- *: campos obrigatórios
- Para cada cliente deve ser possível cadastrar dívidas. As dívidas possuem:
 - Valor: valor da dívida
 - Situação: se está paga ou não
 - Data da criação
 - Data de Pagamento
 - Descrição: descrição do que foi comprado
 - Deve haver uma tela que liste os clientes:
 - Os clientes devem ser ordenados pelo que mais deve para o que menos deve
 - Mostrar campo idade deve ser calculada com base na data de nascimento
 - A lista de clientes deve ser carregada de 10 em 10 (a forma que isso deverá aparecer na tela fica a sua escolha)
 - Deve haver um campo de busca por nome, onde o atendente digita um texto e os clientes cujo nome contém aquele texto devem ser mostrados.
 - Deve haver no final da listagem uma soma do total de dívidas dos clientes
 - Deve haver uma tela que liste as dívidas:
 - Deve ser possível marcar uma dívida como paga
 - Deve aparecer a soma da dívidas de um cliente
 - A soma das dívidas abertas do cliente não devem ultrapassar R\$ 200,00.

O projeto

Você fará uma API que possibilite o CRUD de clientes com dados no formato JSON, além de desenvolver uma interface para que o atendente possa fazer essas operações.

Requisitos

- A API deve ser desenvolvida preferencialmente em Asp.NET Core com a linguagem C#, quaisquer bibliotecas ou frameworks estão permitidos.
- A API deve aceitar e devolver dados no formato JSON
- A API deve possibilitar a persistência dos dados em banco de dados PostgreSQL. Utilizando NHibernate
- A interface deve ser desenvolvida com HTML, CSS e Javascript (ReactJS está permitido).
- A interface deve possuir uma listagem que possibilite o cadastro, alteração e exclusão do cliente
- O cadastro deve ser validado e caso haja algum erro de validação (um campo obrigatório não foi preenchido, por exemplo), você deve mostrar o erro na tela para que o atendente possa consertar.
- A interface deve possuir uma listagem de dívidas para cada cliente e possibilitar informar o pagamento de uma ou mais dívidas.
- O projeto deve estar salvo num repositório git **público** da sua escolha (BitBucket, GitHub, GitLab etc)
- O projeto deve ter um arquivo SQL responsável pela criação do schema do banco de dados.
- README.md explicando o motivo do uso das bibliotecas escolhidas (caso se aplique)
- README.md deve conter instruções de como executar a aplicação

Avaliação

Serão avaliados nesse projeto o seu domínio nas tecnologias .NET e Javascript, além da forma que você levou para resolver o problema, a organização do código e a estrutura dos projetos. A avaliação será com base naquilo que for entregue, então caso não tenha algum domínio em um conhecimento específico, atente-se a entregar o que você domina com a melhor qualidade possível.

Pontos Essenciais

- Linguagem C#
- Linguagem Javascript
- Programação orientada a objetos
- Construção de API
- Apresentação de informação
- Qualidade de código