**Prática 2**

1. Configurar o VSCode para acessar o GitHub.

a) Abra o aplicativo VSCode. Certifique-se de estar na pasta onde deseja clonar o repositório (ex:. Documents).

b) Clone o seu repositório backend do GitHub, utilizando o terminal do VSCode. Caso o repositório já exista no seu computador, atualize-o com o repositório remoto.

c) Abra a pasta do repositório clonado no VSCode.

2. Criar um projeto NodeJS.

a) Crie uma pasta chamada praticas, se não existir, e dentro dela crie uma pasta chamada pratica02.

b) No terminal digite o comando **cd praticas/pratica02** para alterar a pasta.

c) Inicie um novo projeto NodeJS usando **npm init -y** para criar um arquivo package.json.

d) Instale o pacote readline-sync para facilitar a interação do usuário usando **npm install readline-sync**.

e) Abra o arquivo package.json e acrescente na seção scripts o comando **"dev": "node index.js"**.

3. Criar um modelo Contato.

a) Crie um arquivo modelo.js dentro da pasta pratica02.

b) Abra o arquivo modelo.js e declare uma classe Contato.

c) Implemente um construtor para a classe Contato que deve inicializar as propriedades nome, email e telefone.

d) Exporte a classe Contato para ser utilizada por outros módulos.

4. Criar um controlador para operações CRUD.

a) Crie um arquivo controlador.js dentro da pasta pratica02.

b) Abra o arquivo controlador.js e importe a classe Contato na constante Contato.

c) Declare uma constante contatos inicializando com um array vazio.

d) Declare a função adicionarContato() que deve receber os parâmetros nome, email e telefone.

e) Crie uma instância de Contato na constante novoContato.

f) Adicione novoContato no array contatos.

g) Declare a função listarContatos() sem parâmetros.

h) Faça a função listarContatos() retornar o array contatos.

i) Declare a função buscarContato() que deve receber o parâmetro nome.

j) Faça a função buscarContato() localizar um contato no array contatos pelo nome.

k) Declare a função atualizarContato() que deve receber os parâmetros nome, email e telefone.

l) Faça a função atualizarContato() cham-ar a função buscarContato(). Se encontrar o contato, atualize as propriedades email e telefone com os parâmetros da função.

m) Declare a função removerContato() que deve receber o parâmetro nome.

n) Faça a função removerContato() localizar a posição de um contato no array contatos pelo nome. Se encontrar, recorte um elemento do array contatos a partir da posição do contato.

o) Exporte todas as funções para serem utilizadas por outros módulos.

5. Criar um index para chamar o controlador.

a) Crie um arquivo index.js dentro da pasta pratica02.

b) Abra o arquivo index.js e importe o pacote readline-sync na constante readline.

c) Importe o controlador na constante contatoControlador.

d) Declare uma função menu() sem parâmetro. Faça a função imprimir as opções: 1. Adicionar contato; 2 - Listar contatos; 3 - Buscar contato; 4 - Atualizar contato; 5 - Remover contato; 6 - Sair.

e) Declare a função escolherOpcao() que deve receber o parâmetro opcao.

f) Faça a função escolherOpcao() testar o parâmetro opcao com os valores do menu.

g) Caso a opção seja 1, chame a função readline.question() para ler o nome, email e telefone do contato. Em seguida, chame a função adicionarContato() com os valores lidos.

h) Caso a opção seja 2, chame a função listarContatos() e imprima o nome, email e telefone de cada contato.

i) Caso a opção seja 3, chame a função readline.question() para ler o nome do contato. Em seguida, chame a função buscarContato() com o valor lido. Se encontrar o contato, imprima o nome, email e telefone. Senão imprima a mensagem de contato não encontrado.

j) Caso a opção seja 4, chame a função readline.question() para ler o nome, email e telefone do contato. Em seguida, chame a função atualizarContato() com os valores lidos.

k) Caso a opção seja 5, chame a função readline.question() para ler o nome do contato. Em seguida, chame a função removerContato() com o valor lido.

l) Caso a opção seja 6, chame a função process.exit() para encerrar a execução.

m) Declare a função main() e crie um laço infinito.

n) Dentro do laço, chame a função readline.question() para ler a opção do menu. Em seguida, chame a função escolherOpcao() com o valor lido.

o) Faça a chamada a função main().

p) Execute o projeto através do comando **npm run dev**.

6. Subir as modificações no GitHub.

a) Adicione todas as modificações realizadas no projeto no stage do git.

b) Faça um commit com as modificações adicionadas.

c) Caso não esteja configurado os parâmetros user.name e user.email, configure-os através do git.

d) Envie o commit para o branch principal do seu repositório no GitHub.