

## Teste Técnico VB6/ ASP Clássico

O **Grupo Viceri** desenvolve softwares personalizados para empresas há mais de 25 anos e está dividida em três unidades de negócios:

- **Viceri Specialist:** Desenvolvimento de software sob medida
- **Viceri Solution:** Plataformas de produtividade Produtos e Serviços de TI
- **Viceri Ventures:** Incubadora e aceleradora de startups

## Cenário

Antes de começar, leia atentamente as orientações abaixo:

### Requisitos Obrigatórios:

- A aplicação deverá ser desenvolvida em VB6 ou ASP Clássico;
- Banco de dados SQL Server;
- O aplicativo deverá ser disponibilizado através de alguma plataforma na nuvem (GitHub, Google drive, One Drive, etc...), o projeto deve estar compilado e pronto para ser executado, os scripts de banco deverão conter além da criação dos objetos, a criação de banco de dados e usuários;
- Será preciso gerar um manual simples com as instruções de como instalar a aplicação, quais arquivos de configuração deverão ser alterados, etc...

### Requisitos Opcionais (não obrigatório, será um diferencial):

#### Caso a linguagem escolhida for VB6

- Criar uma página em ASP para listar os concorrentes de um cliente.

#### Caso a linguagem escolhida for ASP

- Criar e utilizar um serviço de componente COM+ na sua aplicação.
- Utilizar um ou mais framework de front end ex: JQuery, Handlebarsjs, etc...

### Cenário proposto:

Desenvolver um sistema para cadastro de corretores e clientes, e consulta da carteira de clientes.

1. Deverão ser desenvolvidas **3** páginas: **Consulta de clientes, Cadastro de Corretor e cadastro de Clientes;**
  - 1.1. **Consulta de clientes** (Deverá refletir a página abaixo, layout não obrigatório, deve possuir as funcionalidades descritas).

## Teste Técnico VB6/ ASP Clássico

Filtro						
Codigo Corretor:	<input type="text"/>	Ativo?	<input checked="" type="checkbox"/>			
Nome Corretor:	<input type="text"/>	Estado:	<input checked="" type="checkbox"/>			
Nome Cliente:	<input type="text"/>	Cidade:	<input checked="" type="checkbox"/>			
CPF Cliente:	<input type="text"/>					
<input type="button" value="Cadastrar Corretor"/>		<input type="button" value="Cadastrar Cliente"/>		<input type="button" value="Pesquisar"/>		

  

Nome Cliente	CPF	Ativo?	Nome Corretor	Cod. Corretor	UF	Cidade	
							X
							X
							X
							X
							X
							X
							X
							X
							X
							X
							X
							X
							X
							X
							X

- Nenhum filtro é obrigatório, ao pesquisar sem filtros deverá trazer todos os registros de clientes;
- Ao clicar no botão "Cadastrar Corretor" deverá direcionar para a Página "Cadastro de Corretor";
- Ao clicar no botão "Cadastrar Cliente" deverá direcionar para a Página "Cadastro de Cliente";
- Ao clicar no botão pesquisar deverão ser retornados na grid os registros de clientes que atenderem aos filtros escolhidos;
- O ícone "X" em vermelho para as linhas do grid deverá ter a ação de deletar a linha de registro do cliente no banco de dados e atualizar a grid novamente;

### 1.2. Cadastro de Corretor: cadastro simples com os campos abaixo:

- Código do Corretor;
- Nome do Corretor;
- CPF do Corretor (Usar máscara "XXX.XXX.XXX-XX");

### 1.3. Cadastro de Cliente: cadastro simples com os campos abaixo:

- Corretor (lista de corretores pré cadastrados no passo 1,2, usar um "Combobox");
- Nome;
- CPF do Corretor (Usar máscara "XXX.XXX.XXX-XX");
- Endereço;
- UF ("Combobox" de estados pré cadastrados no banco de dados);
- Cidade ("Combobox" de cidades pré cadastrados no banco de dados, deve refletir a UF selecionada no campo anterior, ao trocar de UF atualizar com a lista de Cidades da mesma);

**Teste Técnico**  
**VB6/ ASP Clássico**

**Tabelas previstas Banco de Dados (Pode ser estruturado da maneira que achar melhor, o modelo abaixo é um exemplo, pode ser seguido ou não).**

- \* Campos Obrigatórios;**
- \* Não devem ser permitidos cadastros de cliente e corretor com o mesmo CPF;**
- \* Todo cliente deve ter um corretor responsável;**

Corretor		Cliente X Corretor		Cliente	
Código	String *	IdCorretor	Int(FK)	Id	Int (PK)
Nome	String *	IdCliente	Int(FK)	Nome	String *
CPF	String *			CPF	String *
				Endereço	String *
				UF	Int(FK)*
				Cidade	Int(FK)*
				Ativo	Boolean

UF		Cidade	
Id	Int(PK)	Id	Int (PK)
Nome	String *	Nome	String *
		IdUF	Int(FK)