**Casos de Testes do Cadastro de Clientes**

|  |  |  |
| --- | --- | --- |
| Entrada | **Classes de Equivalência Válidas** | **Classes de Equivalência Inválidas** |
| Nome | Quantidade de caracteres menores que 100. | Quantidade de caracteres maior que 100. |
| Telefone | Dígitos maiores que zero | Caracteres alfabéticos ou dígitos negativos |
| Rua | Quantidade de caracteres menores que 100. | Quantidade de caracteres maior que 100. |
| Número | Dígitos maiores que zero | Caracteres alfabéticos ou dígitos negativos |
| Cidade/Estado | Quantidade de caracteres menores que 100. | Quantidade de caracteres maior que 100. |

**Casos de Testes**

|  |  |
| --- | --- |
| Nome do Cliente & Rua & Cidade/Estado | Saída |
| “Pedro” | Programa seguirá para próxima instrução. |
| 1234 | Inválido. |
| @Pedro | Inválido. |

|  |  |
| --- | --- |
| Telefone | Saída |
| 12345678901 | Programa seguirá para próxima instrução. |
| 1, 12, 123, 1234, 12345, 123456, 1234567, 12345678, 123456789, 1234567890, 123456789012, [...] | Inválido. |
| “abc” | Inválido. |
| -12345678901, @12345678901, [...] | Inválido. |

|  |  |
| --- | --- |
| Número | Saída |
| 123 | Programa seguirá para próxima instrução. |
| 123A | Programa seguirá para próxima instrução. |
| -123, @1231, [...] | Inválido. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Teste: Cadastro de Clientes Nome** | | | | | | |
| **Entradas** | **Classes válidas** | **Resultado Esperado** | | **Classes Inválidas** | **Resultado Esperado** | |
| Caracteres | Numero de caracteres maior que 1 e menor que 100 | Prosseguimento das instruções | | Número de caracteres menor que 1 e maior que 100 | Solicitar uma entrada válida | |
| **Relatório de execução de testes** | | | | | | |
| **Entradas** | | | **Resultado** | | | **Aprovado?** |
| Pedro Lucas | | | Solicita a entrada do telefone do cliente | | | SIM |
| - | | | Solicitado novamente o nome de usuário | | | SIM |
| **Barnaby Marmaduke Aloysius Benjy Cobweb Dartagnan Egbert Felix Gaspar Humbert Ignatius Jayden Kasper Leroy Maximilian Neddy Obiajulu Pepin Quilliam Rosencrantz Sexton Teddy Upwood Vivatma Wayland Xylon Yardley Zachary Usansky**. | | | Solicitado novamente o nome de usuário | | | SIM |
| @Pedro Lucas | | |  | | | NÃO |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Teste: Cadastro de Clientes Telefone** | | | | | | |
| **Entradas** | **Classes válidas** | **Resultado Esperado** | | **Classes Inválidas** | **Resultado Esperado** | |
| Números inteiros | Números inteiros maiores que 0 e menores ou igual a 9 | Prosseguimento das instruções | | Número de dígitos | Solicitar uma entrada válida | |
| **Relatório de execução de testes** | | | | | | |
| **Entradas** | | | **Resultado** | | | **Aprovado?** |
| 31999999999 | | | Solicita a entrada da Rua do cliente | | | SIM |
| 31999 | | | Solicitado novamente o telefone de usuário | | | SIM |
| **“Abc”** | | | Solicitado novamente o Telefone de usuário | | | SIM |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Teste: Cadastro de Clientes Endereço: Rua** | | | | | | |
| **Entradas** | **Classes válidas** | **Resultado Esperado** | | **Classes Inválidas** | **Resultado Esperado** | |
| Caracteres | Número de caracteres maior que 1 e menor que 100 | Prosseguimento das instruções | | Número de caracteres menor que 1 e maior que 100 | Solicitar uma entrada válida | |
| **Relatório de execução de testes** | | | | | | |
| **Entradas** | | | **Resultado** | | | **Aprovado?** |
| Bernardo Vasconcelos | | | Solicita o número da rua. | | | SIM |
| 123 | | | Solicitado novamente a Rua | | | SIM |
| @Bernardo Vasconcelos | | | Solicitado novamente a Rua | | | SIM |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Teste: Cadastro de Clientes Endereço: Numero** | | | | | | |
| **Entradas** | **Classes válidas** | **Resultado Esperado** | | **Classes Inválidas** | **Resultado Esperado** | |
| Números inteiros | Números inteiros maiores que 0 e menores ou igual a 9 | Prosseguimento das instruções | | Número de dígitos menores que zero | Solicitar uma entrada válida | |
| **Relatório de execução de testes** | | | | | | |
| **Entradas** | | | **Resultado** | | | **Aprovado?** |
| 123 | | | Solicita a entrada da cidade/estado do cliente | | | SIM |
| “ABC” | | | Solicitado novamente o número | | | SIM |
| @123 | | | Solicitado novamente o Numero | | | SIM |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Teste: Cadastro de Clientes Endereço: Cidade/Estado** | | | | | | |
| **Entradas** | **Classes válidas** | **Resultado Esperado** | | **Classes Inválidas** | **Resultado Esperado** | |
| Caracteres | Número de caracteres maior que 1 e menor que 100 | Prosseguimento das instruções | | Número de caracteres menor que 1 e maior que 100 | Solicitar uma entrada válida | |
| **Relatório de execução de testes** | | | | | | |
| **Entradas** | | | **Resultado** | | | **Aprovado?** |
| Belo Horizonte MG | | | Cadastro concluído | | | SIM |
| 123 | | | Solicitado novamente a cidade/estado | | | SIM |
| @Belo horizonte MG | | | Solicitado novamente a cidade/estado | | | NAO |

Funções

void cadastrarCliente();

Insere o cadastro do cliente, em um arquivo de texto no formato binário utilizando Structs como base de escrita e estrutura.

int gerarCodigo(int tarefa);

Gera um código automático em função da contagem de Structs criadas no arquivo e retorna o mesmo para a função de chamada.

void pesquisarCadastros();

Busca um determinado nome ou cadastro armazenado na struct, lendo como base o tamanho da própria struct e, caso o item pesquisado corresponda ao item armazenado em uma das structs, imprime na tela o cadastro.

void listarCadastros();

Imprime todos os cadastros armazenados no arquivo.

void cabecalho(int tarefa);

“Limpa” o console e imprime um cabeçalho da aplicação.

int validacaoString(int string[100]);

Valida caso a string recebida possui algum caractere diferente das definições e caso haja, retorna uma mensagem de erro e um 0.