



TESTES

DEFINIÇÃO:

- **TESTE CAIXA PRETA:** Você **não olha o que tem dentro** do programa, só vê **o que entra e o que sai**. Exemplo: Você aperta um botão na TV (entrada) e vê se ela liga (saída). Você **não abre a TV** para ver os fios e como ela funciona.
- **TESTE CAIXA BRANCA:** Você **olha dentro do código** para testar cada caminho possível. Exemplo: Você desmonta o carrinho de controle remoto e vê **como os fios funcionam**. Você **testa cada parte**: o motor, a roda, o controle...
- **TESTE REGRESSÃO:** Serve para garantir que **nada quebrou** depois de uma mudança no código. Você conserta a buzina do seu carrinho, mas agora as luzes pararam de funcionar. O teste de regressão garante que **consertar uma coisa não estrague outra**.

VISÃO GERAL:

Este documento apresenta um resumo técnico dos testes realizados nas nas funções de coleta e validação de dados do sistema de gerenciamento de imigrantes, desenvolvido em Portugol WebStudio tais funções envolvem **fluxos interativos**, com laços e condicionais, destinados a garantir que apenas dados válidos e completos sejam registrados antes que o sistema avance, replicando testes de caixa-preta para validar entrada de dados, estruturas de repetição e confirmações de usuário. Os testes também são aplicados no banco de dados do sistema de gerenciamento de imigrantes abrangendo 13 tabelas e suas respectivas validações de integridade.

Tabela Principal

- **imigrante:** Dados pessoais básicos dos imigrantes (nome, nacionalidade, data de nascimento, tipo sanguíneo).

Tabelas Relacionadas

- **documento:** Documentos dos imigrantes (passaporte, RNE, CPF, etc.)
- **escolaridade:** Histórico educacional
- **formacao_profissional:** Cursos e formações técnicas
- **emprego:** Histórico profissional
- **local_de_apoio:** Instituições de apoio disponíveis
- **imigrante_local:** Registro de acesso aos locais de apoio

- **contato_emergencia:** Contatos de emergência
- **moradia:** Histórico de moradia/alojamento
- **hospital:** Cadastro de hospitais
- **saude:** Perfil médico dos imigrantes
- **doenca:** Catálogo de doenças
- **imigrante_doenca:** Relação imigrante-doença
- **vacina:** Catálogo de vacinas
- **imigrante_vacina:** Registro de vacinação

TESTES NAS FUNÇÕES

Cada função foi analisada considerando: objetivo, fluxo principal, validações, laços e confirmação.

1. Função preencher_escolaridade()

Objetivo:

Coletar dados complementares de escolaridade e garantir sua confirmação pelo usuário antes de avançar.

Fluxo e Validações Principais:

- Loop “faça...enquanto” que repete a coleta até confirmação.
- Coleta: nível, país, instituição, ano.
- Exibição de resumo para confirmação.
- Se inválido, repete.

2. Função preencherFormacoes()

Objetivo:

Permitir cadastro de múltiplas formações profissionais com flexibilidade na quantidade.

Fluxo e Validações Principais:

- Solicita número de formações.
 - Se > 0 : inicia loop para coletar cada formação e armazena como se fosse um array.
 - Se $= 0$: informa ausência e segue.
- Exibição da lista cadastrada para confirmação visual.

3. Função preencherEmpregos()

Objetivo:

Cadastrar experiências profissionais apenas para usuários ≥ 18 anos.

Fluxo e Validações Principais:

- Verifica idade:
 - Se < 18 : exibe mensagem e segue sem coletar empregos.
 - Se ≥ 18 : pergunta se já trabalhou.
 - Se sim: solicita número de empregos e coleta nomes em loop, armazena em array.
 - Se não: segue sem cadastro.
- Exibição da lista de empregos cadastrados para confirmação.

Estrutura clara para cada função

- Identificação da função e objetivo.
- Descrição do fluxo, incluindo loops e condições.
- Pontos de entrada, saída e confirmação de dados.

Validações e Controle de Fluxo

- Uso de laços controlados por confirmação (escolaridade) ou número informado (formações/empregos).
- Condicionais baseadas em idade ou valor informado.

TESTES REALIZADOS NO BANCO DE DADOS

1. Testes de Integridade Referencial (Foreign Keys)

Objetivo: Validar que as relações entre tabelas funcionem corretamente

Resultados:

- Todas as tentativas de inserção com FKs inválidas foram rejeitadas
- Tentativas de exclusão de registros com dependentes foram bloqueadas
- Inserções com FKs válidas foram aceitas

Tabelas testadas: documento, escolaridade, formacao_profissional, emprego, imigrante_local, contato_emergencia, moradia, saude, imigrante_doenca, imigrante_vacina

2. Testes de Restrições NOT NULL

Objetivo: Verificar se campos obrigatórios são validados

Resultados:

- Inserções com valores NULL em campos obrigatórios foram rejeitadas
- Inserções com todos os campos obrigatórios preenchidos foram aceitas
- Strings vazias ("") foram aceitas em alguns campos que deveriam ser obrigatórios

Observação: Identificada necessidade de validação adicional para strings vazias em campos críticos

3. Testes de Valores Padrão

Objetivo: Validar se valores padrão são aplicados automaticamente

Resultados:

- Campo `data_cadastro` na tabela `imigrante` recebe automaticamente `CURRENT_TIMESTAMP`
- Campos opcionais aceitam valores NULL quando não informados

4. Testes de Restrições UNIQUE

Objetivo: Verificar se duplicatas são impedidas em campos únicos

Resultados:

- Coluna email na tabela imigrante rejeita duplicatas
- Constraint composta (tipo_documento, numero_documento) na tabela documento funciona corretamente

5. Testes de Restrições CHECK

Objetivo: Validar regras de negócio implementadas via constraints

Resultados:

- Validação de anos de conclusão (entre 1900 e ano atual)
- Validação de datas de emprego (data_termino >= data_inicio)
- Validação de formato de email via REGEXP
- Validação de formato de horário de funcionamento

6. Testes de Tipos de Dados

Objetivo: Verificar se os tipos de dados são respeitados

Resultados:

- Campos DATE, INT, VARCHAR respeitam seus limites
- Campos TEXT armazenam conteúdo extenso
- Campos BOOLEAN funcionam corretamente
- Tentativa de inserir string em campo INT foi rejeitada (comportamento esperado)

7. Testes de Alterações Estruturais

Objetivo: Verificar se mudanças na estrutura não quebram funcionalidades existentes

Resultados:

- Adição de novas colunas não afeta inserções existentes
- Adição de constraints não impacta dados válidos
- Alterações estruturais mantêm integridade referencial

