



Sistemas de Informação - 2025
Projeto Integrador I
Plano de testes - Fase 2
30/4/2025

Eduardo Fagundes da Silva - RA 25008024
Eliseu Pereira Gili - RA 25009281
Guilherme Heron Da Silva - RA 25014708
Igor de Toledo Furtado - RA 25008239
Lucas Athanasio Bueno de Andrade - RA 25002731

1. Objetivo

Testar e validar:

- Entrada de dados do usuário
- Classificação de consumo a partir das entradas do usuário
- Inserção e consulta em banco de dados
- Exibição dos resultados

2. Escopo

Funcionalidades testadas:

- Coleta e validação dos dados de consumo
- Cálculo/classificação da sustentabilidade
- Inserção dos dados e das classificações no banco de dados
- Recuperação correta das informações armazenadas
- Consistência entre o que é inserido pelo usuário e o que é recuperado pelo BD

3. Tipos de Teste

- Teste funcional (validar lógica de classificação)
- Teste de integração (testar fluxo completo)
- Teste de banco de dados (validar INSERT, SELECT)
- Teste de limite (valores extremos)
- Teste de entrada inválida

4. Estratégia de Teste

- Execução manual
- Utilização de scripts SQL para validações diretas no banco

5. Ambiente de Teste

- Ambiente local
- Ferramentas:
 - Python 3
 - MySQL
 - VSCode

6. Casos de Teste

6.1. Testes Funcionais

Cenário	Entradas	Saídas
Entrada válida: opção 1 para uso de transportes	1	1
Entrada válida: opção 6 para uso de transportes	6	6
Entrada inválida: número fora do intervalo para uso de transportes	0, depois 2	2
Entrada inválida: número acima do intervalo para uso de transportes	10, depois 4	4
Entrada inválida: texto para uso de transportes	"abc", depois 5	5
Entrada inválida: valor em branco para uso de transportes	"" , depois 3	3
Entrada válida: valor do meio do intervalo para uso de transportes	3	3
Entrada inválida seguida de várias válidas para uso de transportes	a, 9, -1, 1	1
Energia como texto	"dez"	Mensagem de erro, pede novamente
Lixo não reciclável acima do permitido	-3	Mensagem de erro, pede novamente
Porcentagem de reciclável maior que 100	150%	Mensagem de erro, pede novamente
Transporte inválido	8	Mensagem de erro, pede novamente
Opção inválida para repetir	"x"	Mensagem de erro, pede novamente
Usuário decide repetir	Responde "S"	Executa main() novamente
Usuário finaliza	Responde "N"	Mensagem "Finalizando programa..." e encerra
Usuário pressiona Ctrl+C	Interrupção no meio de qualquer input	Mensagem "Programa encerrado..." e encerramento limpo
Consumo de água muito baixo	100	Alta sustentabilidade
Consumo de água no limite	150	Alta sustentabilidade

inferior		
Consumo médio	175	Média sustentabilidade
Consumo no limite superior	200	Média sustentabilidade
Consumo elevado	250	Baixa sustentabilidade
Baixo consumo	3	Alta sustentabilidade
Igual ao limite inferior	5	Alta sustentabilidade
Consumo intermediário	7	Média sustentabilidade
Igual ao limite superior	10	Média sustentabilidade
Acima do limite	15	Baixa sustentabilidade
Lixo não reciclável muito baixo	10	Baixa sustentabilidade
No limite inferior	20	Baixa sustentabilidade
Intermediário	35	Média sustentabilidade
No limite superior	50	Média sustentabilidade
Muito lixo	60	Alta sustentabilidade
Transporte público	1	Alta sustentabilidade
Bicicleta	2	Alta sustentabilidade
Caminhada	3	Alta sustentabilidade
Carro elétrico	5	Média sustentabilidade
Carona compartilhada	6	Média sustentabilidade
Carro a combustível fóssil	4	Baixa sustentabilidade
Valor inválido (ex: 7)	7	Baixa sustentabilidade
Valores decimais inseridos com vírgula	10,5	Sistema deve salvar 10.5

6.2. Testes de Integração

Entrada	Resultado Esperado	Observação
Dados completos corretos (água, energia, resíduos, transporte)	Cálculo correto + Gravação no banco	Classificações correta conforme definição (Alta, média e baixa sustentabilidade)
Dados extremos (água = 400 litros)	Baixa Sustentabilidade + Gravação no banco	Testar limites superiores

Dados extremos (água = 0 litros)	Alta Sustentabilidade + Gravação no banco	Testar limites inferiores
Valores inválidos (ex.: transporte número 10)	Mensagem de erro	Não deve tentar gravar no BD

6.3. Testes de Banco de Dados

Ação	Resultado Esperado	Observação
Inserir dados de consumo e classificação	Dados gravados corretamente	Conferir via SELECT**
Recuperar dados inseridos	Dados iguais aos informados	Validar integridade
Tentativa de inserção com campo nulo obrigatório	Erro controlado	Exemplo: consumo de água vazio, pede para o usuário preencher e tentar novamente - os campos são obrigatórios já no python
Inserir e consultar múltiplos registros	Listagem correta	Sem sobreposição de registros

** scripts SQL para conferência:

- SELECT * FROM monitoring ORDER BY Date DESC LIMIT 1
- SELECT * FROM monitoring_classifications ORDER BY Date DESC LIMIT 1

7. Riscos

- Falha no salvamento dos dados
- Dados corrompidos ou inconsistentes com os inputs do usuário
- Entradas inválidas não bloqueadas, todos os inputs devem ser validados pelas funções do Python
- Quebra na comunicação entre aplicação e banco - erro 500 retornado deve ser tratado

8. Critérios de aceite

- 100% dos dados inseridos devem ser recuperados com os mesmos valores - garantia de integridade

- 100% dos cálculos devem bater com as regras de classificação (alta, média e baixa sustentabilidade)
- Erros de entrada devem ser tratados antes de tentar gravar no banco (o responsável é o Python)
- Em caso de erro no banco, o sistema deve avisar o usuário sem travar (possibilitando que tente novamente ou que reinicie a aplicação)