



INSTITUTO FEDERAL DE
EDUCAÇÃO CIÊNCIA E TECNOLOGIA
Baiano



Unit Test

METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS – FASE DO DESENVOLVIMENTO –

COORDENAÇÃO DE SISTEMAS (CODES)

JUNHO/2011

SUMÁRIO

1.	Controle de Versão	1
2.	Funcionalidade	1
3.	Tipo	1
4.	Teste Unitário	1
4.1.	Parâmetros no Domínio	1
4.2.	Parâmetros Fora do Domínio	1
4.3.	Tipos de Valores	1
4.4.	Data	1
4.5.	Números	2
4.6.	E-mail	2
4.7.	Senha	2
4.8.	Campos Especiais CPF, CGC, CEP, CNS e etc.	2
4.9.	Valores para Teste	2
4.10.	Sintaxes da Linguagem.....	2
4.11.	Erro na Linguagem.....	3
4.12.	Alteração de Valores	3
4.13.	Interpretação de Valores	3
4.14.	Cálculos Mesclados	3
4.15.	Formatações de região ou não	3
4.16.	Medidas	4
4.17.	Navegação	4

1. Controle de Versão

Data	Versão	Descrição	Autor
[dd/mm/aaaa]	[1.0, 1.1, ...]	[Inserir ação no documento que gerou a nova versão]	[Nome do Responsável pela Atualização da Versão]

2. Funcionalidade

[Informe qual a funcionalidade que será testada.]

3. Tipo

[Defina se o teste será Manual ou Automatizado. No caso de ser escolhida a segunda opção, informar qual será o aplicativo e suas respectivas parametrizações.]

4. Teste Unitário

4.1. Parâmetros no Domínio

[Verificar e informar aqui se todas as entradas e saídas de dados comuns ao sistema apresentam o resultado esperado. Caso contrário deve-se relatar as que apresentaram não conformidade.]

4.2. Parâmetros Fora do Domínio

[Verificar e informar aqui se todas as entradas e saídas de dados não comuns ao sistema apresentam o resultado esperado. Caso contrário deve-se relatar as que apresentaram não conformidade.]

4.3. Tipos de Valores

[Verificar e informar aqui se todos os formatos de valores que possuem regras regionais estão apropriados.]

4.4. Data

[Verificar e informar aqui se todas as datas estão no formato exigido para a região; testar se os domínios estão coerentes; em caso de cálculo de feriados devem apresentar todos eles corretamente conforme as regiões definidas pelo o usuário.]

4.5. Números

[Verificar e informar aqui se todos os números estão no formato exigido para a região; se os valores negativos e sinais estão formatados de forma padronizada; testar se os domínios estão coerentes; tratar valores com casas decimais definidas]

4.6. E-mail

[Verificar e informar aqui se todo e-mail possui um @ e um ponto, sendo que o @ nunca esta no começo ou no fim e o ponto sem esta entre o @ e o fim sendo que ele não pode estar junto com o @. Um e-mail dever ter no mínimo 5 caracteres contando o @ e o ponto.]

4.7. Senha

[Verificar e informar aqui se as senhas atendem as normas da empresa; não deve ser exibida durante a digitação somente com caractere representativo; deve ser confirmada a digitação; não deve permitir ser copiada ou colada.]

4.8. Campos Especiais CPF, CGC, CEP, CNS e etc.

[Verificar e informar aqui se existe a formatação na entrada dos dados e na armazenagem das informações; atender as regras de calculo existentes; não podem ser dados viciados; observar campos dependentes DDI-DDD-TEL e similares validando o preenchimento de ambos em validação única; validar a formatação; testar todas as regras de preenchimento; força dados inválidos ou repetir o preenchimento.]

4.9. Valores para Teste

[Verificar e informar aqui se existem tipos de dados válidos que se tornam inválidos conforme a linguagem usada.]

4.10. Sintaxes da Linguagem

[Verificar e informar aqui se a linguagem tem problemas ao tratar valores do tipo objeto, string e outros. Tais linguagens têm tratamentos especiais para estes valores para assim evitar erros ou em alguns casos não tem tratamento, pois o erro é na codificação.]

4.11. Erro na Linguagem

[Verificar e informar aqui se há erro na linguagem, por exemplo, no caso de sistemas feitos na própria base de dados que não permitem pesquisas usando termos como *else*, *while*, *for* por serem palavras reservadas da linguagem. Atualmente a maior parte dos sistemas não apresenta erros com termos reservados.]

4.12. Alteração de Valores

[Verificar e informar aqui se as linguagens ao receberem determinados valores alteram seus dados internos com tratamentos automáticos no geral elas possuem travas para estes tratamentos. Exemplo de valores: 0x00, 00FF, 1.1, 1,1, 1^2 e etc.]

4.13. Interpretação de Valores

[Verificar e informar aqui se linguagens interpretam diretamente valores sem tratar passando eles para outra aplicação em uso. Este erro ocorre geralmente quando outra linguagem é usada como intermediário. Exemplo em HTML(ASP,JSP,CGI,PHP e outras) quando passamos uma estrutura HTML(TAGs) para a página e esta estrutura é apresentada novamente porém no corpo do html podemos ter dados incorporados que mudam o resultado apresentado.]

4.14. Cálculos Mesclados

[Verificar e informar aqui se valores que são permitidos: (-,(), {}, [], ', ^, x, *, /,) que acabam por modificar os dados tornando-os inválidos; ponto flutuante ou estouros; erros de ponto flutuante dificilmente são identificados durante a especificação de um sistema ou acabam por ocorrer com a maturidade da empresa e seus dados; planejar bem o crescimento dos dados; não ter uma área de armazenamento maior que a área de entrada de dados.]

4.15. Formatações de região ou não

[Verificar e informar aqui se o sistema está usando uma região única caso não se trate todos os valores com os quais ele pode trabalhar. O não tratamento da

região pode tanto mudar o valor de um pagamento como alterar a data de uma cobrança.]

4.16. Medidas

[Verificar e informar aqui se as medidas estão iguais no sistema inteiro ou estão tratadas para isso.]

4.17. Navegação

[Verificar e informar aqui se a navegação está coerente, utilizando tabulação e se os requisitos de acessibilidade e usabilidade estão sendo atendidos.