

DIRETRIZES DE DESENVOLVIMENTO DE SOFTWARE

Nº	CATEGORIA	DESCRIÇÃO
1	Apresentação	Conceitos/Termos de negócio devem ser referenciados da mesma forma em todas as telas e relatórios. Por exemplo, não se pode exibir "Departamento responsável pelo edital" em uma tela e "Unidade Finep responsável pelo edital" em outra.
2	Apresentação	Informações em tabelas devem ter o mesmo alinhamento e a mesma formatação em todas as telas nas quais sejam exibidos, caso seja estabelecido para o sistema um padrão nesse sentido.
3	Apresentação	Se for definido um padrão para a posição de botões como "Salvar", "Cancelar", "Voltar", etc., verificar se eles estão nos locais esperados e na ordem correta.
4	Apresentação	Telas com funcionalidades e/ou objetivos similares deveriam ter interfaces com apresentação e comportamento iguais ou, no mínimo, bastante semelhantes. Por exemplo, a seleção de consultores para um edital não deveria ter interface com muitas diferenças em relação à seleção de analistas para um edital.
5	Arquitetura	A camada de apresentação não deve acessar a base de dados.
6	Arquitetura	Classes de domínio e que implementam serviços não podem fazer referências a classes da camada de apresentação nem a bibliotecas de <i>front-end</i> (como aquelas relacionadas ao JSF, por exemplo).
7	Arquitetura	O controle de status de ciclo de vida de entidades do domínio não deve ser baseado nas informações do BPMS.
8	Arquitetura	O domínio do negócio da aplicação deve ser criado e representado pelas classes, e não pelo banco de dados. Dessa forma, deve-se primeiramente criar as classes, com todos os relacionamentos que forem necessários e deixar que as tabelas do banco de dados sejam geradas automaticamente pelo Hibernate.
9	Arquitetura	Regras do Negócio não podem ser implementadas nas classes <i>Managed Bean</i> (quando se usa JSF) ou nos <i>Controllers / ViewModels</i> (quando se usa ZK).
10	Arquitetura	Transações que envolvam a instanciação de um processo no BPMS devem ser revertidas caso o processo em questão não tenha sido instanciado com sucesso. Ou seja, a funcionalidade não poderia prosseguir se falhar a instanciação de um processo que precisa ser criado para que a transação seja considerada completa.
11	Banco de Dados	As chaves primárias devem ser artificiais (<i>surrogate key</i>).
12	Banco de Dados	As tabelas devem estar pelo menos na Terceira Forma Normal.
13	Banco de Dados	Não devem ser utilizados <i>Triggers, Stored Procedures, Functions</i> , campos BLOB ou CLOB, exceto com aval do responsável pelo projeto na Finep.
14	Codificação / Java	Quando se usa JSF/Primefaces, as classes <i>Controllers</i> são, na verdade, <i>Managed Beans</i> e devem ser nomeadas como tal (isto é, terminando em "MB"). E, mais importante ainda, não devem ser compartilhados por várias funcionalidades. Idealmente, cada funcionalidade tem o seu <i>Managed Bean</i> .

www.finep.gov.br

Rio de Janeiro

Ventura Corporate Towers
Av. República do Chile, 330
Torre Oeste – Centro
10º - 12º e 15º - 17º andares
20031 170 Rio de Janeiro RJ
t. (21) 2555 0330

São Paulo

JK Financial Center
Av. Presidente Juscelino
Kubitschek, 510
9º andar Itaim Bibi
04543 000 São Paulo SP
t. (11) 3847 0300

Brasília

SCN QD. 02 Bloco "D" Torre A,
Sala 1102
Centro Empresarial Liberty Mall
70712 903 Brasília DF
t. (61) 3035 7408

SAC

t. (21) 2555 0555
sac@finep.gov.br

Ouvidoria

t. (21) 2557 2414
ouvidoria@finep.gov.br

Nº	CATEGORIA	DESCRIÇÃO
15	Codificação / Java	Nos projetos ZK, as classes <i>Controllers</i> (padrão de projeto MVC) e <i>ViewModels</i> (padrão de projeto MVVM) devem ser nomeadas de acordo, ou seja, terminando em "Controller" e "VM", respectivamente. E também não devem ser compartilhadas por diferentes <i>views</i> .
16	Codificação / Java	Erros no acesso a banco de dados devem ser capturados e tratados.
17	Codificação / Java	Listas de opções - usadas em <i>combos</i> , <i>radio buttons</i> e <i>checkboxes</i> - que tiverem um domínio pequeno (até mais ou menos 10 itens) devem ser colocadas em <i>Enums</i> , a não ser que haja um requisito explicitando a necessidade de manutenção deste cadastro. Somente neste caso essas informações deverão ser colocadas em tabelas no banco de dados. Ex: Status de Operação: ATIVA("Ativa"), PENDENTE("Pendente"), CONCLUÍDA("Concluída").
18	Codificação / Java	Não devem existir <i>warnings</i> de sintaxe no código-fonte das classes.
19	Codificação / Java	<i>Raw types</i> não devem ser usados.
20	Codificação / Java	Todas as classes que sejam itens de uma classe que implemente <code>java.util.Collection</code> devem sobrescrever os métodos <code>equals()</code> e <code>hashCode()</code> .
21	Codificação / Java	Todas as Entidades que são persistidas devem sobrescrever os métodos <code>equals()</code> e <code>hashCode()</code> .
22	Codificação / Java	Upload de arquivos deve verificar, após o arquivo ter sido subido, se o arquivo realmente existe no local esperado e se seu tamanho é correto (se não for possível comparar o tamanho no servidor com o tamanho do arquivo na máquina cliente, no mínimo verificar se o arquivo tem tamanho maior que zero).

www.finep.gov.br

Rio de Janeiro

Ventura Corporate Towers
Av. República do Chile, 330
Torre Oeste – Centro
10º - 12º e 15º - 17º andares
20031 170 Rio de Janeiro RJ
t. (21) 2555 0330

São Paulo

JK Financial Center
Av. Presidente Juscelino
Kubitschek, 510
9º andar Itaim Bibi
04543 000 São Paulo SP
t. (11) 3847 0300

Brasília

SCN QD. 02 Bloco "D" Torre A,
Sala 1102
Centro Empresarial Liberty Mall
70712 903 Brasília DF
t. (61) 3035 7408

SAC

t. (21) 2555 0555
sac@finep.gov.br

Ouidoria

t. (21) 2557 2414
ouvidoria@finep.gov.br