**Padrão de Nomenclatura**

# Sumário

Sumário 2

I PARTE 3

Introdução 4

Definições Gerais 4

Pacote 4

Interfaces 4

Classes 4

Enumeraçoes e Lists 5

Métodos 5

Parâmetros de Métodos 5

Variáveis 5

Exceções 5

Documentação de Métodos e Classes 6

Comentário de Código 6

Definições Específicas 7

Classes de Acesso a Dados (DAO) 7

Classes de Negócio (Services) 7

Classes de Valores (Value Objects) 7

# PARTE

***Padrões de Nomenclatura***

Classes

Interfaces

Enumerações

Pacotes, etc.

# Introdução

O objetivo deste documento é apresentar o padrão de nomenclatura dos artefatos de código, escritos na linguagem Java.

# Definições Gerais

## Pacote

Todos os pacotes do projeto deverão estar sob a hierarquia formada conforme domínio e nome do sistema. Por exemplo.

com.br.managertranschool

Para os pacotes relacionados a arquitetura do sistemas os mesmos estarão sob o seguinte pacote.

com.br.managertranschool.architecture

Para os pacotes relacionados a camada de negócio do sistemas os mesmos estarão sob o seguinte pacote.

com.br.managertranschool.business

Para os pacotes relacionados a camada de activity do sistemas os mesmos estarão sob o seguinte pacote.

com.br.managertranschool.view.activity

## Interfaces

Todas interfaces deverão iniciar com a letra I em maiúscula. Por exemplo:

IUsuarioService

## Classes

As classes adotarão o padrão de nomenclatura CamelCase, com exceção das siglas que permanecerão todas em maiúsculo. Por exemplo:

MainActivity e UsuarioDAO

## Enumeraçoes e Lists

As Enumerações e Lists deverão conter o sufixo “List”. Por exemplo:

ClienteStatusList e UsuarioTipoList

## Métodos

Os métodos devem ter a primeira letra minúscula e demais seguindo o padrão CamelCase. Conforme exemplos abaixo:

findByFilter

## Parâmetros de Métodos

Os parâmetros NÃO deverão ser prefixados pela letra “p”. As demais letras seguirão o padrão CamelCase conforme exemplo abaixo.

codCliente

## Variáveis

As variáveis deverão atender a um determinado:

Seja variável estática ou de instancia e primeira letra deverá ser minúscula e demais letras seguindo o padrão CamelCase, por exemplo:

nomeCliente

## Exceções

Todas as exceções do sistema deverá conter o sufixo “Exception” conforme exemplo abaixo.

SystemException

IntegrationException

Observação: Variáveis do tipo exceção em blocos “try-catch” deverão ser de nome “e”.

## Documentação de Métodos e Classes

Todas as classes e métodos deverão conter documentações com textos claros em linguagem de negócio. Tais documentações deverão atender as especificações JavaDoc. Documentação do autor é composta por nome completo e e-mail.

## Comentário de Código

Os comentários de código deverão existir conforme objetivo do bloco de operações. Conforme padrão abaixo

//Obter dados do cliente.

...findCliente();

//Alterar dados do cliente

...atualizarCliente();

# Definições Específicas

## Classes de Acesso a Dados (DAO)

Todas implementações dos DAOs conterão o sufixo “DAO”.

## Classes de Negócio (Services)

Todas implementações dos Services conterão o sufixo “Service”.

## Classes de Valores (Value Objects)

As classes de valores representam os dados.

Tais classes podem representar:

* Entidades de banco de dados. Seu sufixo é “VO”.
* Dados de entrada utilizados como filtro. Seu sufixo é “FilterVO”.