

# **Pedidos Online – DUSA**

## **Estándar de Documentación Técnica**

### **Versión 1.2**

#### **Historia de revisiones**

| Fecha      | Versión | Descripción                    | Autor              |
|------------|---------|--------------------------------|--------------------|
| 28/08/2013 | 1.0     | Creación del Documento         | Ignacio Gil        |
| 30/08/2013 | 1.1     | Modificaciones varias          | Luciana Guartechea |
| 31/08/2013 | 1.2     | Revisión final Responsable SQA | Verónica Gamarra   |
|            |         |                                |                    |

# Índice

|           |                      |   |
|-----------|----------------------|---|
| <b>1.</b> | <b>Diagramas</b>     | 3 |
| 1.1.      | Herramientas         | 3 |
| 1.2.      | Convenciones         | 3 |
| 1.2.1     | Clases               | 4 |
| 1.2.2     | Atributos            | 4 |
| 1.2.3     | Operaciones          | 4 |
| 1.2.4     | Parámetros           | 4 |
| 1.2.5     | Asociaciones         | 4 |
| 1.2.6     | DataTypes            | 4 |
| <b>2.</b> | <b>Subsistemas</b>   | 4 |
| <b>3.</b> | <b>Componentes</b>   | 5 |
| <b>4.</b> | <b>Interfaces</b>    | 5 |
| <b>5.</b> | <b>Generalidades</b> | 5 |

## **1. Diagramas**

### **1.1. Herramientas**

Para la construcción de artefactos UML se utilizarán las herramientas Cacao y Enterprise Architect dependiendo de la necesidad de desarrollar de forma concurrente estos artefactos se optará por una de las dos herramientas, siendo que Cacao soporta edición concurrente y el cual permite la integración con Google Drive. En cuanto a la construcción de diagramas o esquemas que no apliquen a conceptos UML se utilizará parte de la suite de Microsoft Office como ser: Visio, Word, Excel. Cada diagrama será representado de forma tal que su visión sea clara para el lector.

### **1.2. Convenciones**

Se definen las siguientes convenciones con respecto a los distintos estilos en el nombramiento de elementos:

#### **Pascal case**

La primera letra del identificador y la primera letra de cada palabra siguiente se ponen en mayúscula.

Ejemplo: ObtenerDato

#### **Camel case**

La primera letra del identificador se pone en minúscula y la primera letra de cada palabra siguiente se pone en mayúscula.

Ejemplo: obtenerDato

#### **Upper case**

Todas las letras en el identificador se ponen en mayúscula. Usar esta convención sólo para identificadores que consisten de dos o menos letras.

Ejemplo: UI

Además se debe considerar:

- No usar abreviaciones o contracciones en los nombres de identificadores.
- No usar acrónimos que no sean globalmente aceptados. Solo usar acrónimos conocidos, por ejemplo UI para Interfaz de Usuario.

### **1.2.1 Clases**

El nombre debe estar compuesto por caracteres alfabéticos y debe cumplir con el formato Pascal case, a su vez se deben usar Sustantivos para denotarlas. Además las abreviaciones se deben utilizar con moderación.

### **1.2.2 Atributos**

El nombre debe estar compuesto por caracteres alfanuméricos y debe cumplir con el formato Camel case.

### **1.2.3 Operaciones**

Los nombres de las operaciones deben estar compuestos por caracteres alfanuméricos, debiéndose usar verbos en infinitivo que representen dicha operación, y deben cumplir con el formato Camel case.  
Ejemplos: obtenerPedido(), invocar().

### **1.2.4 Parámetros**

El nombre debe estar compuesto por caracteres alfanuméricos y debe cumplir con el formato Camel case.

### **1.2.5 Asociaciones**

El nombre es opcional; en el caso de existir debe estar compuesto por caracteres alfanuméricos y debe cumplir con el formato Camel case.

### **1.2.6 DataTypes**

Con respecto al elemento DataType, se seguirá el mismo criterio que en una clase. Dicho elemento será distinguido mediante el estereotipo <<DataType>>.

## **2. Subsistemas**

El estilo utilizado para los subsistemas es Pascal Case. Para cada subsistema se puede adjuntar una nota en forma de comentario para detallar un poco más su propósito. En la representación se utilizará el estereotipo provisto <<subsystem>>.

### **3. Componentes**

Para los nombres de las componentes se va a utilizar el mismo criterio que sigue la nomenclatura de los subsistemas. Si la funcionalidad o propósito de una componente no es demasiado intuitiva se adjuntará un comentario explicando más en detalle. En la representación se utilizará el estereotipo <<component>>.

### **4. Interfaces**

El nombre debe estar compuesto por caracteres alfabéticos y debe cumplir con el formato Pascal case. Se debe usar sustantivos o adjetivos que describan su comportamiento y comenzarán con la letra "I" (en mayúscula) al comienzo del nombre. Esto permitirá interpretar que es una interface del subsistema que indica el resto del nombre. Los componentes serán distinguidos por el estereotipo <<interfaz>>.

### **5. Generalidades**

Para todos los casos, los nombres tienen que ser elegidos cuidadosamente, ya que una buena elección genera una mejor comunicación y entendimiento del problema, en lo posible usar nombres mnemotécnicos para poder tener una mejor interpretación de lo que se quiere representar.