Buenas Prácticas

Guía de desarrollo

*V 1.0.0.0*

Historial de Revisiones

| Fecha | Versión | Release | Autor | Principales Cambios | Revisión | Fecha Aprobación | Aprobación |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 17/03/2017 | 1.0 |  | Adrián Ayala | Creación del Documento |  |  |  |

**Índice**

1 Alcance 4

2 Descripción 4

2.1 Clases, estructuras e interfaces 4

2.2 Métodos 4

2.3 Propiedades 4

2.4 Variables 5

2.5 Parámetros 5

2.6 Enumeraciones 5

2.7 Eventos 5

2.8 Controladores 5

2.9 General 6

3 Diagramas 7

4 Información adicional 7

5 Definiciones, Siglas y Abreviaturas 7

# Alcance

El objetivo de este documento dejar asentado algunas reglas o tips a la hora de programar.

# Descripción

## Clases, estructuras e interfaces

1. Los nombres deben ser un sustantivo, en castellano con el formato PascalCase. Sino se puede definir una entidad como un sustantivo, revisar. Apoyarse en el dominio del sistema para el nombramiento de los mismos.
2. Utilizar nombres comunes, que no den lugar a ambigüedades.
3. Solo en las interfaces prefijarlas con I. Ej: Ipersistable
4. No utilizar abreviaciones en los nombres.

## Métodos

1. Nombrar los métodos de acuerdo a la tarea que realicen. Por lo general un método debería realizar una única tarea.
2. Nombrar los métodos con verbos en castellano con el formato de escritura PascalCase.
3. No utilizar abreviaciones en los nombres.

## Propiedades

1. Los nombres deben ser sustantivos o adjetivos, no verbos. Esto es porque el nombre de una propiedad se refiere a datos.
2. Nombrar las propiedades en castellano con el formato de escritura PascalCase.
3. No utilizar abreviaciones en los nombres.
4. No utilizar Get como parte del nombre de una propiedad.
5. Nombrar a las propiedades booleanas en forma afirmativa (SiValdio en vez de NoValido). Si agrega valor (solamente si así lo es), utilizar prefijos (Si, Tiene, Puede). Un buen ejercicio para el nombramiento de las propiedades booleanas es testearlas en el escenario de uso. Por ejemplo que como está en el framework de .Net:

If(collection.Contains(item))

If(regularExpression.Matches(text))

Es más natural que:

If(collection.IsContained(item))

If(regularExpression.Match(text))

1. Preferir la voz activa en vez de la vos pasiva, ejemplo:

If(Persona.SiValida) // major que

If(Persona .EsValida)

## Variables

1. Los nombres deben ser sustantivos o adjetivos, no verbos.
2. Nombrar las variables en castellano con el formato de escritura PascalCase para el caso de variables globales, y para el caso de variables locales utilizar el formato de escritura CamelCase.
3. No utilizar prefijos como m\_, g\_, etc.
4. No utilizar abreviaciones en los nombres.

## Parámetros

1. Utilizar para los nombres de los parámetros el formato de escritura CamelCase.
2. Más allá de la legibilidad, los parámetros son mostrados en el designer y en la documentación del framework (intellisense y class browsing). Por eso deben ser descriptivos y concisos.
3. No utilizar abreviaciones en los nombres, ni prefijo ni sufijo.

## Enumeraciones

1. Nombrar las Enumeraciones en castellano con el formato de escritura PascalCase.
2. Utilizar nombres singular para las enumeraciones.
3. No utilizar Enum como prefijo ni sufijo.
4. No utilizar abreviaciones en los nombres.

## Eventos

1. Nombras a los eventos con verbos o frases verbales (Actualizando, Guardando, etc)
2. Utilizar el concepto temporal para los nombres, Ej: Cerrando y Cerrar.
3. No utilizar prefijos Before o After para indicar el momento del evento.
4. Nombrar a los event handlers (delegados utilizados como tipos) con el sufijo EventHandler.

## Controladores

Controladores, contenidos en la carpeta llamada Controllers. La mayoría extiende de WebSecurityController, este controlador es el encargado de validar que el usuario tenga el permiso en seguridad para realizar la operación (método) que se está definiendo. Ejemplo

|  |
| --- |
| [Authorize]  public ActionResult ListadoSolicitudes()  {  **validateAction** se define en webSecurityController  try  {  this.validateAction("listarSolicitudesEnVerificacion");  …  }  …  } |

1. El nombre del controlador debe estar asociado a un conjunto de funcionalidades determinadas o bien a una clase en particular, nunca a una acción. Preferiblemente que sean en Español.
2. Son aceptados ExigibilidadController, AutorizacionController, FTPController, ReporteController.
3. No deberían ser de la forma: NuevaAutorizacionController, EditarExigibilidadController, etc.
4. Para los ABM dentro de los Controladores siempre definimos los métodos Nuevo, Editar y Eliminar.
5. Tenemos Nuevo con GET y POST, y Editar con GET y POST.
6. No olvidar utilizar el Authorize en caso de ser necesario (operaciones que requieren login de usuario)

|  |
| --- |
| [Authorize]  public ActionResult Nuevo()  {  …  }  [Authorize, HttpPost]  public ActionResult Nuevo(Actividad actividad)  {  …  }  [Authorize]  public ActionResult Editar(long idActividad)  {  …  }  [Authorize, HttpPost]  public ActionResult Editar(Actividad actividad)  {  …  }  [Authorize]  public ActionResult Eliminar(long idActividad)  {  …  } |

## General

1. Incorporar comentarios, de triple barra, en las Clases, Métodos, Propiedades y secciones de código que ayuden a la legibilidad del código y además ayuda porque son mostrados en el designer y en la documentación (intellisense y class browsing).
2. Incorporar los tags #region / #endregion para sectorizar el código. Ejemplo en las clases

|  |
| --- |
| #region // Variables Publicas //  #endregion  #region // Variables Privadas //  #endregion  #region // Propiedades Publicas //  #endregion  #region // Propiedades Privadas //  #endregion  #region // Métodos Públicos //  #endregion  #region // Métodos Privados //  #endregion |

En los controladores por ejemplo:

|  |
| --- |
| #region // Vistas Públicas //  #endregion  #region // Métodos Públicos //  #endregion  #region // Métodos Privados //  #endregion |

1. Código duplicado, todos los bloques de código idéntico o muy similar se deben generalizar y colocar en clases genéricas para ser heredadas por las demás clases que corresponda.

# Diagramas

N/A

# Información adicional

# Definiciones, Siglas y Abreviaturas

|  |  |
| --- | --- |
| **Término o Abreviatura** | **Descripción** |
| N/A | No Aplica |
|  |  |