Contenido

[Introducción 2](#_Toc439289408)

[Objetivo 2](#_Toc439289409)

[Alcance 2](#_Toc439289410)

[Diagrama del repositorio 3](#_Toc439289411)

[Criterios de nombrado 3](#_Toc439289412)

[Criterios de aceptación para conformación de línea base 3](#_Toc439289413)

[Herramientas de repositorios 3](#_Toc439289414)

[Herramientas e infraestructura 4](#_Toc439289415)

[Recursos 4](#_Toc439289416)

[Identificación de la configuración 6](#_Toc439289417)

[Estructura de directorios 6](#_Toc439289418)

[Convenciones sobre uso de mayúsculas y minúsculas 7](#_Toc439289419)

[*Definiciones* 7](#_Toc439289420)

[*Nuestras convenciones* 8](#_Toc439289421)

[Convención sobre nombrado 9](#_Toc439289422)

[*Clases* 9](#_Toc439289423)

[*Métodos* 9](#_Toc439289424)

[*Variables* 9](#_Toc439289425)

[*Comentarios* 10](#_Toc439289426)

# Introducción

En el siguiente documento se define la estructura de los repositorios la cual brinda organización, y permite la gestión de los archivos, la definición de líneas base que sirven de punto de control y permiten el regreso en caso de desperfectos, y también los criterios de aceptación de las mismas por parte del plantel docente.

# Objetivo

El objetivo del plan de configuración es definir y mantener la integridad, y trazabilidad de los documentos, entregables, o todo artefacto realizado por el equipo a lo largo del proyecto. Es decir poder proporcionar una visión general de la organización, de las actividades y tareas en general llevadas a cabo por el equipo de proyecto.

# Alcance

La gestión de configuración contempla las siguientes actividades:

• Identificación de ítems de configuración

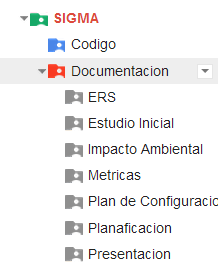
• Mantenimiento de descripciones de los ítems de configuración.

• Establecimiento y administración del repositorio.

• Mantenimiento de la historia de los ítems.

• Control de los cambios.

# Diagrama del repositorio



# Criterios de nombrado

Se define a continuación y de forma genérica como se muestra:

Nombre del documento\_v[n°]

# Criterios de aceptación para conformación de línea base

Las líneas base se corresponderán con la finalización de cada uno de los módulos del sistema los cuales deberán haber sido codificados, comentados y testeados en su línea crítica con pruebas de estrés.

# Herramientas de repositorios

Se manejara una dualidad de repositorios tanto en Google drive como en TortoiseSVN, en el primero se actualizaría por línea base al finalizarla, mientras que en TortoiseSVN se irá actualizando por módulos, librerías y funcionalidades que se incorporen.

# Herramientas e infraestructura

Para el manejo del versionado en el cliente se utilizan:

* SVN Tortoise para el versionado del código fuente de la aplicación
* Github como plataforma de repositorio y administración del codigo fuente.
* GoogleDrive para repositorio de la documentación.

Tortoise SVN y Github son instalados en los clientes permiten:

* Registrar todos los cambios efectuados sobre los archivos de un proyecto.
* Recuperar versiones anteriores del código de un proyecto.
* Conocer qué cambios se han efectuado sobre un archivo determinado, quién los ha realizado y cuándo.
* Obtener información pertinente a los distintos cambios realizados, pudiendo visualizar el log de información.
* Brindar información en forma de métricas sobre el desarrollo del código fuente, diferenciando por usuario desarrollador.

Por el lado del servidor, se cuenta con tres cuenta en Github para el manejo del código fuente.

Link: https://github.com/maxitarno/SiGMA

# Recursos

En esta sección se describirán los recursos a utilizar para poder generar los ítems de configuración.

Para los ítems de configuración correspondientes a la gestión del proyecto:

* Microsoft Office 2007.
* Enterprise Architect.
* SQL Server 2008.
* Microsoft Project 2007
* Microsoft Excel 2007

Para los ítems de configuración correspondientes a la implementación:

* Visual Studio 2010
* SQL Server 2008

# Identificación de la configuración

Será ítem de configuración todo aquel archivo, documento, etc., que pueda sufrir cambio a lo largo del tiempo. Como así también todo aquello que esté especificado en la WBS.

## Estructura de directorios

Con respecto a la documentación de proyecto

// INSERTAR IMAGEN DE REPOSITORIO DE – PRODUCTO - PROYECTO //

# Convenciones sobre uso de mayúsculas y minúsculas

## Definiciones

* **EstiloPascal**

La primera letra del identificador y la primera letra de las siguientes palabras concatenadas están en mayúsculas. El estilo de mayúsculas y minúsculas Pascal se puede utilizar en identificadores de tres o más caracteres.

Ejemplo:  
*ImageSprite*

* **Estilo camelCase**

La primera letra del identificador está en minúscula y la primera letra de las siguientes palabras concatenadas en mayúscula

Ejemplo:  
*imageSprite*

* **Estilo Mayúsculas (ALL\_CAPS)**

Todas las letras del identificador se encuentran en mayúsculas

Ejemplo:  
*IO*

* **Estilo minúsculas (small\_caps)**

Todas las letras del identificador se encuentran en minúsculas

Ejemplo  
*system*

## Nuestras convenciones

|  |  |
| --- | --- |
| Tipo | Código C# |
| Clase | Estilo Pascal |
| Método | camelCase |
| Namespace / Package | Estilo Pascal |
| Propiedades | Estilo Pascal |
| Parámetro | camelCase |
| Variable local | camelCase |
| Interface | Estilo Pascal |

**Abreviaturas de variable:**

* txt = TextBox
* lbl = Label
* rfv = RequiredFieldValidator
* cv = CustomValidator
* rev = RegularExpressionValidator
* btn = Button
* cov = CompareValidator
* chk = CheckBox
* rb = RadioButton
* ddl = DropDownList
* lst = Listbox
* pnl = panel
* fu = FileUpload
* img = Image
* ibtn = image button
* hf = hidden field

## Convención sobre nombrado

### Clases

* El nombre de las clases que representan entidades deberá comenzar con una letra “E” (e mayúscula de entidad) seguido del nombre del objeto en Pascal Case.
* El nombre de las clases que representan acceso a datos deberá comenzar con el texto “LogicaBD” seguido del nombre del objeto en Pascal Case.
* El nombre de la clase debe estar en singular, salvo que la clase represente multiplicidad de cosas.
* Las Nombres de las clases deberían ser Sustantivos: ejemplo *carro, hombre, tienda, país, empleado, proveedor.*

### Métodos

* Los nombres de los métodos deberían ser un verbo, dado que describe una acción; ejemplo *buscar (), modificar (), cargar (), registrar()*
* Los Métodos dentro de las clases siempre deben declarar su visibilidad tales como *privadas, protegidas, públicas, etc.*

### Variables

* Limitar el uso de variables que sean de un solo carácter, Los nombres comunes para las variables temporales son i, j, k, m, y n para los números enteros; c, d, y e para los caracteres.
* Nombres de variables sólo pueden contener caracteres alfabéticos

### Comentarios

Será considerado una buena práctica la inclusión comentarios en métodos y atributos utilizando las herramientas del lenguaje para la visualización de métodos, siempre que sea el desarrollador lo considere necesario.

La inserción de los comentarios se realizará sin seguir un estándar, utilizando las formas de comentar código brindado por el lenguaje. El desarrollador deberá indicar en un pequeño texto el comportamiento del método.