

**Normas de Desarrollo**

**GRUPO**

N°3

**CURSO**

5K2

**DOCENTES**

Ing. Zohil, Julio

Ing. Liberatori, Marcelo

Ing. Jaime, Natalia

**ALUMNOS**

Allemand, Facundo leg. 58971

Herrera, Antonio leg. 57824

Pedrosa, Paula Melania leg. 58822

Rojas Amaya, M. Florencia leg. 58577

### 

### 

**UNIVERSIDAD TECNOLÓGICA NACIONAL**

**UNIVERSIDAD TECNOLÓGICA NACIONAL**

**INGENIERÍA EN SISTEMAS DE INFORMACIÓN** **PROYECTO FINAL**

INGENIERÍA EN SISTEMAS DE INFORMACIÓN PROYECTO FINAL

**PROYECTO**

**PROYECTO**



**Que Golazo!**

**Que Golazo!**

**Sistema de Gestión de Torneos de Fútbol**

Sistema de Gestión de Torneos de Fútbol

16/04/2015

31/05/2014

|  |  |  |  |
| --- | --- | --- | --- |
| HISTORIAL DE VERSIONES | | | |
| VERSION | **FECHA** | **RESPONSABLE** | **OBSERVACION** |
| 1.0 | 21/10/2014 | Paula Pedrosa | Creación del documento. |
| 1.1 | 22/10/2014 | Paula Pedrosa | Se agregó todo lo asociado a las herramientas de HW y SW que estamos utilizando |
| 1.2 | 23/10/2014 | Paula Pedrosa | Se agregó todo lo asociado a Gestión de la Configuración |
| 1.2 | 16/04/2015 | Paula Pedrosa | Cambios menores |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Tabla de contenido

[Tabla de contenido 2](#_Toc401945820)

[Introducción 3](#_Toc401945821)

[Descripción de Documento 3](#_Toc401945822)

[Desarrollo 3](#_Toc401945823)

[Definición de Software y Hardware para Desarrollo 3](#_Toc401945824)

[Reglas de Nombrado 4](#_Toc401945825)

[Gestión de Configuración 7](#_Toc401945826)

# Introducción

## Descripción de Documento

El objetivo de este documento es detallar las Normas de Desarrollo del Sistema de Gestión de Torneos de Fútbol QueGolazo. Incluye la definición y descripción de Software y Hardware que fueron utilizados para el desarrollo del proyecto, criterios, métodos y notación a aplicar. Además se menciona la gestión de Configuración, especificando estructuras, acceso a repositorios del proyecto y del sistema.

# Desarrollo

## Definición de Software y Hardware para Desarrollo

A continuación se detalla los recursos de hardware y software que están siendo utilizados para el desarrollo e implementación:

|  |  |
| --- | --- |
| **RECURSOS DE HARDWARE PARA EL DESARROLLO** | * 4 equipos con Sistema Operativo Windows 7 o superior |
| * Hosting para alojar el Proyecto |
| * Hosting IIS para alojar el Sistema y la Base de Datos |

|  |  |
| --- | --- |
| **RECURSOS DE SOFTWARE PARA EL DESARROLLO** | * IDE de Desarrollo: **Visual Studio 2012** |
| * Lenguaje de Programación: **C# (Framework .net).** |
| * Base de Datos: **SQL Server 2012 y SQL Management Studio.** |
| * Herramienta de versionado: **Tortoise SVN y Ankh SVN.** |
| * Repositorio de Código: **Google Code** |
| * Repositorio de Documentación: **Google Code** |
| * Diseño de Interfaces de Usuario:   Framework Front End: **Bootstrap**  IDE Front End: **Brackets** |
| * Soporte para la Comunicación: **Skipe, Teamviewer 9, Facebook, WhatsApp, Gmail.** |
| * Respaldo de Información: **GoogleDrive** |
| * Herramientas para Gestión de Documentos: **Word, Excel, Power Point, Project** |
| * Herramienta de Modelado: **Start UML** |
| * Herramienta para Gestión Ágil de Proyectos: **Visual Studio**   Link para acceder a esta herramienta: *https://quegolazo.visualstudio.com/DefaultCollection/ProyectoFinal/\_home/index* |

## Reglas de Nombrado

base de datos

|  |  |  |
| --- | --- | --- |
|  | **DESCRIPCIÓN** | **EJEMPLO** |
| **NOMBRE DE TABLAS** | UperCamelCase y los nombres siempre en plural | Equipos – Jugadores - JugadoresXEquipo |
| **ATRIBUTOS DE LAS TABLAS** | camelCase | nombre – goles - cantidadDePartidos |
| **CLAVES PRIMARIAS** | “id”+ el nombre de la tabla en singular con Mayúscula la primer letra | idJugador - idTorneo |
| **CLAVES FORÁNEAS** | Exactamente igual que la clave primaria a la que referencia |  |

Programación

|  |  |  |
| --- | --- | --- |
|  | **DESCRIPCIÓN** | **EJEMPLO** |
| **NOMBRE DE LAS CLASES** | UperCamelCase en singular | Equipo, GestorEquipo, Cancha |
| **NOMBRE DE LOS ATRIBUTOS** | camelCase, siempre teniendo en cuenta que los atributos de aquellas clases que tengan persistencia datos deben ser exactamente igual en el código C# y en el nombre en la base de datos |  |
| **ENNUMERADOS Y CONSTANTES** | mayúsculas separadas con guiones bajos | CESPED, MOSAICOS, TIPO\_DE\_CANCHA |
| **VECTORES** | vector+ “Nombre de la clase que contiene” | vectorEquipos |
| **LISTAS** | lista+ “nombre del tipo de objetos que contiene” | listaJugadores |
| **MÉTODOS** | camelCase (agregarEquipo() ), uso obligatorio de la etiqueta <sumary> <sumary/> describiendo la funcionalidad del método. | agregarEquipo() |
| **MÉTODOS GESTORES** | siempre deben estar implementados los métodos:  obtenerTodos() : Lista o Datatable  obtenerPorId(long id) : Object  registrar(Object objetoAInsertar)  modificar () |  |

CONTROLES DE FORMULARIOS

|  |  |  |
| --- | --- | --- |
|  | **DESCRIPCIÓN** | **EJEMPLO** |
| **CONTROLES DE FORMULARIOS** | Abreviación del nombre del control + nombre significativo |  |
| **TEXTBOX** | **txt**NombreDeJugador |
| **GRILLAS** | **gv**Partidos |
| **BOTONES** | **btn**AgregarEquipo |
| **DROPDOWNLIST** | **ddl**Campeonatos |
| **LABELS** | **lbl**Fecha |
| **RADIOBUTTONS** | **rb**Sexo |
| **RADIOBUTTONGROUP** | **rbg**TiposDeCampeonato |
| **CHECKBOX** | **cb**Eliminatorias |
| **PANEL** | **panel**Jugadores |
| **LITERAL** | **lit**NombreDelJugador |
| **TEXTAREA** | **txa**Descripcion |
| **LISTBOX** | **lbox**Jugadores |

Consultas a la Base de Datos:

Las sentencias **INSERT, UPDATE, DELETE y SELECT** que sonutilizadas en las consultas a la Base de Datos, las palabras reservadas dentro de la consulta deben ir en mayúscula.

Así también como las palabras reservadas que se utilizan en la sentencia SELECT, tales como **SELECT, FROM, INNER JOIN, JOIN, LEFT JOIN, RIGHT JOIN, WHERE, HAVING, ORDEY BY, GROUP BY** deben ir en mayúscula.

Las consultas en la Base de Datos, deben realizarse de la siguiente manera:

|  |
| --- |
| **SELECT** idTorneo, nombre **FROM** Torneos t **INNER JOIN** Ediciones e ON t.idTorneo = e.idTorneo **WHERE** idUsuario = 1 **GROUP BY** idTorneo, nombre **HAVING** COUNT(idEdicion) > 3 **ORDER BY** nombre |

Es decir, cada palabra reservada debajo de otra.

Generales:

En los ABMC (Alta – Baja – Modificación - Consulta), los botones de las interfaces deben decir:   
Alta: **“Registrar”**

Modificación: **“Modificar”**

Baja: **“Eliminar”**

Consulta: **“Consultar”**

## Gestión de Configuración

Como mencionamos anteriormente en las herramientas que están siendo utilizadas para el desarrollo del proyecto, utilizamos:

* **TORTOISE SVN:** es una herramienta de control de versiones
* **GOOGLE CODE:** es el repositorio que se utiliza para almacenar todas las versiones del proyecto y líneas de base del proyecto.

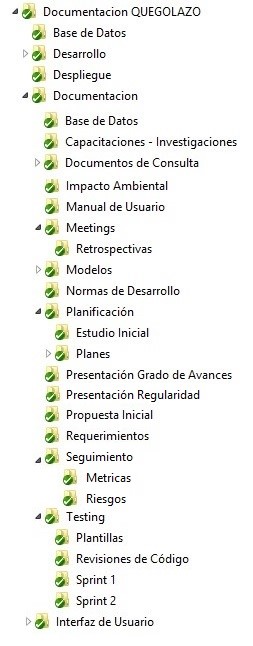
Resulta significativo aclarar que se manejan dos repositorios. Manipulamos un repositorio para el código que estamos trabajando y otro repositorio para la documentación.

Links de Repositorios:

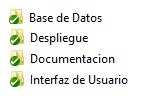
**URL Repositorio Documentación:** *https://quegolazo.googlecode.com/svn/trunk/*

**URL Repositorio Código:** *https://quegolazo-code.googlecode.com/svn/trunk/*

Estructura de Repositorio de Documentación



Como podemos ver en la imagen, el repositorio de Documentación está estructurado por cuatro carpetas principales. Estas son:



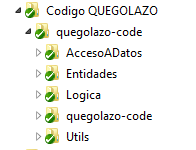
|  |  |
| --- | --- |
| **NOMBRE DE CARPETA** | **EXPLICACIÓN** |
| **BASE DE DATOS** | Esta carpeta contiene los Script de la Base de Datos de QueGolazo. Contiene 3 Script: Script de Datos, Script de Esquema y Script de Datos y Esquema. |
| **DESPLIEGUE** | Esta carpeta contendrá todo lo asociada al Despliegue del producto. |
| **DOCUMENTACIÓN** | Esta carpeta contiene todos los documentos asociados al Proyecto y al Producto que estamos desarrollando. |
| **INTERFAZ DE USUARIO** | Esta carpeta contendrá todo lo asociado a interfaz de usuario. |

Dentro de la Carpeta Documentación, se encuentran las siguientes carpetas:



|  |  |
| --- | --- |
| **NOMBRE DE CARPETA** | **EXPLICACIÓN** |
| **BASE DE DATOS** | Esta carpeta contiene todos los Documentos asociados a Base de Datos, es decir los Diagramas de Entidad - Relación |
| **CAPACITACIONES - INVESTIGACIONES** | Esta carpeta contiene todos los Documentos asociados a las Investigaciones y Capacitaciones que hemos realizado y que consideramos significativas para el desarrollo de nuestro producto. |
| **DOCUMENTOS DE CONSULTA** | Esta carpeta contiene todos los Documentos de Consulta, tales como los resultado de investigaciones, capacitaciones, etc. |
| **IMPACTO AMBIENTAL** | Esta carpeta contiene todos los Documentos asociados al análisis de Impacto Ambiental de nuestro Proyecto. |
| **MANUAL DE USUARIO** | Esta carpeta contiene el Manual de Usuario de nuestro Sistema. |
| **MEETINGS** | Esta carpeta contiene todas las reuniones documentadas. Incluye las Retrospectivas. |
| **MODELOS** | Esta carpeta contiene todos los modelos que fueron realizados. |
| **NORMAS DE DESARROLLO** | Esta carpeta contiene el documento de Normas de Desarrollo. |
| **PLANIFICACIÓN** | Esta carpeta contiene el resultado del Estudio Inicial junto con la WBS y la Calendarización y los Planes realizados, tales como el Plan de Testing, Plan de Riesgos, etc. |
| **PRESENTACIÓN DE GRADO DE AVANCES** | Esta carpeta contiene todo lo asociado a la Presentación de Grado de Avance. |
| **PROPUESTA INICIAL** | Esta carpeta contiene todo lo asociado al análisis de la Propuesta Inicial |
| **REQUERIMIENTOS** | Esta carpeta contiene todo lo asociado a la Especificación de Requerimientos de Software (ERS) y Product Backlog. |
| **SEGUIMIENTO** | Esta carpeta contiene toda la documentación asociada al Seguimiento del proyecto. Dentro de ella encontramos dos carpetas: Métricas y Riegos. En la carpeta de Métrica se encuentra todo lo referido a las métricas medidas durante el proyecto y en la carpeta de Riesgos, lo asociado al seguimiento de riesgos durante el proyecto. |
| **TESTING** | Esta carpeta contiene toda la documentación asociada a Testing, como lo son los Casos de Prueba, resultado de Testing, resultados de revisiones de código, etc. |

Estructura de Repositorio de CÓDIGO



Como podemos ver en la imagen el repositorio de Código está estructurado por cinco carpetas principales. Cada una de estas carpetas representa un proyecto dentro de nuestro código de aplicación web. Cada uno de estos proyectos agrupa un conjunto de clases que se comportan de manera similar.

|  |  |
| --- | --- |
| **NOMBRE DE CARPETA** | **EXPLICACIÓN** |
| **ACCESOADATOS** | Esta carpeta contiene todas las clases asociada al acceso a datos, es decir todas aquellas clases que contemplen métodos de acceso y consulta a la base de datos.  Ejemplo: DAOTorneo: contiene todos los métodos de acceso a datos asociado a la entidad Torneo, es decir, registrarTorneo(), actualizarTorneo(). |
| **ENTIDADES** | Esta carpeta contiene todas las clases que representan Entidades de nuestro proyecto  Ejemplo: Torneo, Equipo, etc. |
| **LÓGICA** | Esta carpeta contiene todos las clases que están asociadas a la lógica de negocio  Ejemplo: GestorTorneo, GestorEquipo |
| **QUEGOLAZO-CODE** | Es el Proyecto Web, agrupa las clases de presentación.  Ejemplo: Páginas web, y archivos de configuración web. |
| **UTILS** | Esta carpeta contiene las clases que dan soporte a los gestores de la capa lógica. |

Versionado

En cuanto al manejo de las versiones, cuando hacemos un cambio en cualquier documento, se maneja de la siguiente manera:

Versión 1.0 -> Creación del Documento

Versión 1.1 -> Cuando se agregaron cambios significativos

Si los cambios que se realizaron fueron cambios menores o estéticos sigue conservando la versión anterior.

Todos los documentos deben contar con una tabla al comienzo del mismo para gestionar el historial de versiones. En la misma se debe indicar Versión, Fecha, Responsable del Cambio y Observación.

A continuación mostramos la tabla de historia de versiones:

|  |  |  |  |
| --- | --- | --- | --- |
| HISTORIAL DE VERSIONES | | | |
| VERSION | **FECHA** | **RESPONSABLE** | **OBSERVACION** |
|  |  |  |  |
|  |  |  |  |

En cuanto a los cambios en el código, cada vez que se realiza un commit, se debe comentar el cambio realizado y ese comentario debe ser representativo de la funcionalidad agregada o modificada, de manera que cualquier persona pueda entender rápidamente lo que se realizó. Es de fundamental importancia, que cuando se realice el commit todo el código debe compilar y además, no se deben encontrar ningún tipo de error que impidiera la ejecución de la funcionalidad o afectara el trabajo anteriormente realizado.