

Plan De Configuración

16/02/17

1.1

|  |  |
| --- | --- |
| **Nombre.** | **Matricula** |
| **Jesús Alberto Goiz Barrales.** | **24400085** |
| **Juan Alberto Gutiérrez.** | **24400063** |
| **Brenda Robles Antonio.** | **24400073** |
| **Rene Moratilla Montes.** | **24400075** |
| **Guillermo Vivaldo Vazquez** | **24400093** |

**Control De Versiones**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre Del Archivo | Versión | Fecha de modificación | | Autor | Comentarios | |
| Csoft-PlanDeConfiguración-160217.docx | 1.0 | 16/02/17 | GVV | | | 1era Versión | |
| Csoft-PlanDeConfiguración-160217.docx | 1.1 | 23/02/17 | GVV | | | Se especificó más la sección de solicitud de cambios | |
| Csoft-PlanDeConfiguración-160217.docx | 1.2 | 01/03/17 | GVV | | | Se añadió el proceso de revisión de los documentos. | |
|  |  |  |  | | |  | |
|  |  |  |  | | |  | |
|  |  |  |  | | |  | |

# Índice

[**Índice**](#_30j0zll) **3**

[**Plan De Configuración.**](#_1fob9te) **3**

[1.- Introducción.](#_3znysh7) 3

[2.- Objetivo.](#_2et92p0) 4

[3.- Nomenclatura para el nombre de archivos.](#_tyjcwt) 4

[4.- Estándar De Codificación.](#_3dy6vkm) 4

[5.- Procedimiento para el Control De La Configuración.](#_1t3h5sf) 7

[6.- Solicitud De Cambio.](#_4d34og8) 8

[7.- Respaldos](#_2s8eyo1) 9

[8](#_2s8eyo1).- Revisión De Documentos 9

# Plan De Configuración.

## **1.- Introducción.**

En el presente documento se describen las actividades de gestión de configuración de software que deben ser llevadas a cabo durante el proceso de desarrollo del proyecto. Se definen tanto los productos que se pondrán bajo control de configuración como los procedimientos que deben ser seguidos por los integrantes del equipo de trabajo.

## **2.- Objetivo.**

Este documento describe las actividades de gestión de configuración de software que deben ser llevadas a cabo durante el proceso de desarrollo del proyecto. Aquí se definen tanto los productos que se pondrán bajo control de configuración como los procedimientos que deben ser seguidos por los integrantes del equipo de trabajo, se presentan diferentes nomenclaturas para los archivos que se añadan al repositorio del proyecto así como el estándar de codificación que se usará.

## **3.- Nomenclatura para el nombre de archivos.**

La presente nomenclatura será utilizada para nombrar cualquier entregable del proyecto ya sea documento o código.

**“Csoft-[Nombre\_Del\_Archivo]-[Fecha]”**

## **4.- Estándar De Codificación.**

|  |  |
| --- | --- |
| Propósito | Guiar el desarrollo de los programas que se crean utilizando el lenguaje C |
| Encabezados Del Programa | Todos los programas deberán contar con un encabezado que lleve los datos del autor del programa, número y fecha de creación del programa, descripción de lo que hace dicho programa junto con su versión. |
| Formato Del Encabezado | /\*=========================================================\*/  /\*Autor: Guillermo Vivaldo Vazquez \*/  /\*Componente: Menú \*/  /\*Fecha: 30/Enero/2016 \*/  /\*Descripción: Codificación que muestra el menú de opciones\*/  /\*Versión: 1.0 \*/  /\*=========================================================\*/ |
| Listado De Contenido | Se proporcionará una lista de las funciones que contiene el programa. |
| Ejemplo De Listado De Contenido | /\*==========================================================\*/  /\* El programa contiene lo siguiente: \*/  /\* función\_a() \*/  /\* función\_b() \*/  /\* función\_c() \*/  /\* int main() \*/  /\*==========================================================\*/ |
| Instrucciones De Reuso | * Describe cómo se usa el programa. Proporciona un formato de declaración, tipos y valores de parámetros y límites de parámetros. * Proporciona advertencias de valores ilegales, condiciones de sobreflujo u otras condiciones que podrían resultar potencialmente en una operación inadecuada. |
| Ejemplos De Reuso | /\*==========================================================\*/  /\* Instrucciones De Reuso \*/  /\* Función: int suma(int a, int b) \*/  /\* Propósito: Esta función calcula la suma de dos números \*/  /\* enteros y regresa el resultado de dicha suma. \*/  /\* =========================================================\*/ |
| Identificadores | Describe cómo se usa el programa. Proporciona un formato de declaración, tipos y valores de parámetros y límites de parámetros. Proporciona advertencias de valores ilegales, condiciones de sobreflujo u otras condiciones que podrían resultar potencialmente en una operación inadecuada. |
| Ejemplo De Identificador | Nombres De Variables: Se declararan usando como máximo 7 letras del nombre de lo que represente esa variable, por ejemplo: si representa un valor cualquiera se le nombrará “valor” en caso de que haya más variables que representen valores se le agregará al nombre de la variable un guión bajo acompañado de un número como se muestra a continuación.  int valor; /\*esto es correcto\*/  int valor\_1; /\*esto es correcto\*/  int v /\*esto es incorrecto\*/  Tipos de variables válidas para ciclos: letras simples podrán ser usadas como variables contadoras en los ciclos, nombres grandes no podrán ser usados en los ciclos.  int i; /\*esto es correcto\*/  int j; /\*esto es correcto\*/  int k\_ciclo1; /\*esto es incorrecto\*/  Nombres De Funciones: deberán ser escritas en letras minúsculas y se les nombrará de acuerdo a la operación o tarea que realice la función, en caso de que la función tenga un nombre compuesto se separará con un guión bajo, los nombres de funciones no se acentuarán.  void suma(); /\*Esto es correcto\*/  void bienv(); /\*Esto es incorrecto\*/  int desviacion\_estandar(); /\*Esto es correcto\*/  int varianzanumeros(); /\*Esto es incorrecto\*/  Parámetros de funciones: al igual que las funciones se escribirán usando las 3 primeras letras de lo que represente en caso de que haya más parámetros en la función se escribirá ese nombre separado por un numero para poder diferenciarlas, ejemplo:  int valor(int num); /\*Esto es correcto\*/  int suma(int num\_1, int num\_2); /\*Esto es correcto\*/  int suma(int número, int numero1); /\*Esto es incorrecto\*/ |
| Comentarios | * Se documentará el código para que el lector pueda comprender mejor lo que realiza. * Los comentarios deben explicar tanto el propósito como el comportamiento del código. * Comente la declaración de las variables para indicar su propósito. |
| Buen Comentario | int sum /\*Esta variable almacenará la sumatoria de diez números y se usará después para calcula la media\*/  if(raiz==NULL) /\*Verifica que la lista no tenga elementos\*/ |
| Mal Comentario | int sum /\*Variable que almacena un número entero\*/  if(raiz==NULL) /\*Compara que raíz sea igual a NULL\*/ |
| Secciones Importantes | Las secciones importantes del programa deben de tener por un bloque de comentarios que describan lo que hace. |
| Ejemplo | /\*\*\*Esta función recibe como parámetro un numero entero y calcula recursivamente el factorial de dicho número \*\*\*/  int factorial(int a) |
| Espacios En Blanco | * Los programas se escribirán con el suficiente espacio posible de tal forma que sean fáciles de leer. * Se separará cada función o instrucción del programa con al menos un espacio. |
| Sangría | * Se colocarán sangrías en cada nivel de bloque con respecto al anterior. * Llaves que abren y cierran deben estar en una sola línea y alineadas una con la otra. |
| Ejemplo De Sangrías | void lectura\_datos()  {  for(i=0;i<n\_val;i++)  {  printf("\nIngresa el valor n=%d\n",i+1);  scanf("%f", &num);  data1 = data1 + num;  agregar\_siguiente(num);  }  } |
| Mayúsculas | * Todos los defines deberán estar en mayúsculas. * Todos los otros identificadores y palabras reservadas van en minúsculas. * Los mensajes que son salidas al usuario (impresiones en pantalla) pueden tener mezcla de mayúsculas y minúsculas de tal manera que hagan más limpia la presentación al usuario. |
| Ejemplo De Mayúsculas | #define TAM  for(i=0;i<TAM;i++)  printf(“Hola Mundo”); |

## **5.- Procedimiento para el Control De La Configuración.**

Las actividades para el control de la configuración son la solicitud, evaluación, aprobación e implementación de cambios a los elementos de la línea base.

Los cambios se realizarán para la corrección y obtener una mejora cuando algo lo requiera.

Se entiende por cambio al sistema, las modificaciones que afecten a la línea base del sistema, como pueden ser:

* Cambios en los Requerimientos.
* Cambios en el Diseño.
* Cambios en la Arquitectura.
* Cambios en las herramientas de desarrollo.
* Cambios en la documentación del proyecto. (agregar nuevos documentos o modificar la estructura de los existentes)

## **6.- Solicitud De Cambio.**

Cada vez que se requiera hacer un cambio en algún documento se deberá dar aviso al líder del equipo para que apruebe dicho cambio, a la vez deberá registrar en un documento de control de cambios lo siguiente: el nombre del documento, el cambio que realizó, la fecha que realizó el cambio, quien realizó el cambio. En el documento original deberá cambiar el número de versión y registrarla en el template de control de cambios.

El líder del equipo y el encargado de la documentación del proyecto deberán estar enterados y así mismo dar el visto bueno para poder realizar el cambio que se requiera, así mismo cuando la persona que haya modificado algún documento o entregable, deberá dar aviso al encargado de soporte y calidad para su revisión correspondiente y liberación.

Es importante añadir al control de versiones la modificación hecha, así como una versión nueva, en este caso se añadirá un decimal a la versión nueva por ejemplo

Versión 1.0 (Versión Original)

Versión 1.1 (Versión Modificada)

El documento que se desea modificar se adjunta en el correo de petición de cambio, enviado a el responsable de soporte y al líder de equipo.

## 

## **7.- Respaldos**

Con el fin de asegurar la integridad de la información que se maneja en todos los documentos del repositorio cada integrante deberá ser responsable de generar una copia del documento que generó para estar prevenido ante cualquier problema o error de borrado que se pueda tener. El formato de respaldos será de la siguiente forma.

**“Csoft-[Nombre Del Archivo]\_[Fecha]”**

## **8.- Revisión De Documentos**

Una vez que algún documento se encuentre ya dentro del repositorio, el encargado de soporte deberá corroborar que el documento se encuentre en la carpeta correcta y que cumpla con el formato especificado para cualquier documento del proyecto, de lo contrario deberá notificar al dueño del documento para que realice las modificaciones necesarias para que esta persona registre a la vez sus defectos en el defectlog para así poder contribuir a la mejora. Una vez realizada esta revisión el encargado de calidad deberá realizar las revisiones pertinentes relacionadas a la ortografía y sintaxis de los documentos y de cualquier otro entregable de lo contrario se procederá a notificar al propietario del documento para que realice las correcciones necesarias y registre los defectos que tuvo.