

Postmortem Baja Maestros

08/05/17

0.1

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

**Control De Versiones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre Del Archivo | Versión | Fecha | Autor | Comentarios |
| Csoft-PM\_Baja\_Maestros-080517.docx | 0.1 | 08/05/17 | RMM  GVV | Creación del Documento |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Índice

[**Índice**](#_kd5bfhkalq0n) **2**

[**I.- Forma PSP Project Plan Summary.**](#_jhy44tc5k06j) **3**

[**II.- Time Recording Log.**](#_1fob9te) **3**

[**III.- Defect Recording Log.**](#_3znysh7) **3**

[**IV.- Diseño.**](#_m4hw86ggwsmr) **3**

[**V.- Design CheckList.**](#_92tu2pewgguj) **3**

[**VI.- Programa Fuente.**](#_6bvnqqrwimi9) **4**

[**VII.- Code CheckList.**](#_87ns93g4124c) **4**

[**VIII.- Reporte de resultados.**](#_bo7s7l2hg3ra) **4**

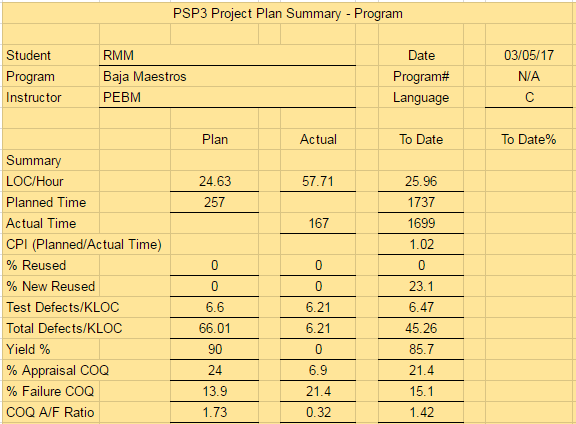
[**IX.- PIP**](#_4ocjjwajumc0) **4**

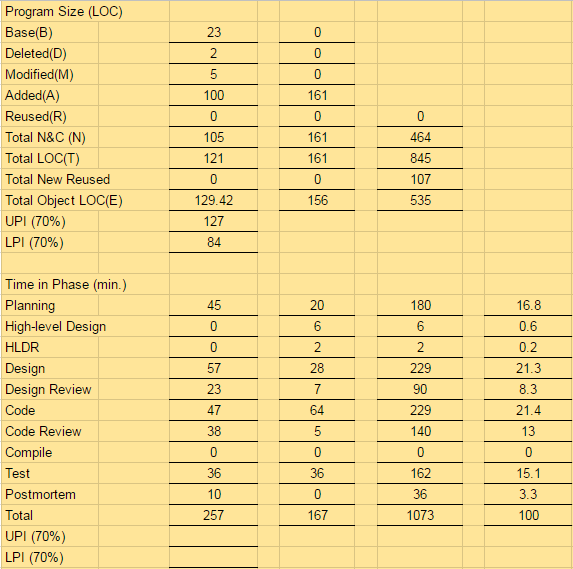
[**X.- Size Estimating Template**](#_w1ych8jaj7as) **5**

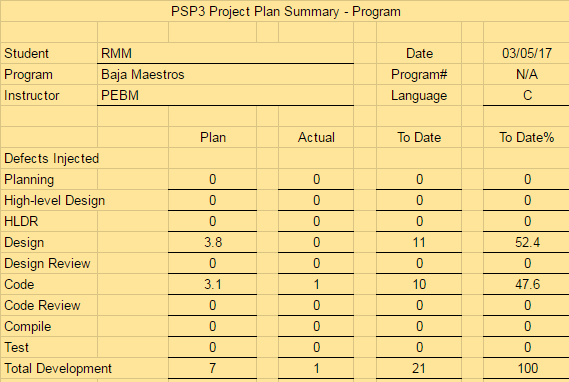
[**XI.- Task and Schedule**](#_195jonwtvlxs) **5**

# I.- Forma PSP Project Plan Summary.

Se inició el proceso el 03/05/17

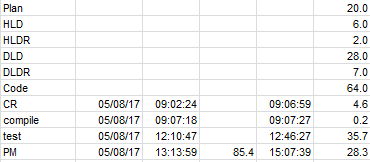






# 

# II.- Time Recording Log.



# III.- Defect Recording Log.



# IV.- Diseño.

**State Specification Template**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Student | | RMM | | | Date | | 03/05/17 |
| Program | | Baja Maestros | | | Program # | | N/A |
| Instructor | | PEBM | | | Language | | C++ |
|  | | |  | | | | |
| Cambios maestros | | | | | | | |
| **State Name** | | | | **Description** | | | |
| Inicio | | | | Inicia el programa. | | | |
| Baja maestros | | | | Función principal para la baja de maestros. | | | |
| Query maestros | | | | Consulta a base de datos de maestros | | | |
| delete\_maestros | | | | Función para hacer borrar en la base de datos de maestros. | | | |
| Salir | | | | Termina el programa | | | |
|  | | | |  | | | |
| **Function/Parameter** | | | | **Description** | | | |
| printf() | | | | Imprime datos en la pantalla | | | |
| gets() | | | | Lee datos de la pantalla | | | |
| scanf() | | | | Lee datos de la pantalla | | | |
| opción | | | | opción que elegirá el usuario | | | |
| baja\_maestros() | | | | Función principal de cambios | | | |
| query\_maestros() | | | | Función de consulta | | | |
| delete\_maestros() | | | | Función de borrado en base de datos | | | |
| **States/Next States** | | | | **Transition Condition** | | **Action** | |
| Inicio | | | |  | |  | |
|  | Baja\_maestros | | | True | | printf()-imprime el menú  scanf()- lee opción  gets()- condición de datos válidos. | |
| Cambios maestros | | | |  | |  | |
|  | Query\_maestros | | | buscar un maestro | | query\_maestros() | |
|  | Delete\_maestros | | | Borrar el registro de un maestro. | | delete\_maestros() | |
|  |  | | |  | |  | |

# Logic Specification Template

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student | RMM | | Date | 03/05/17 |
| Program | Baja Maestros | | Program # | N/A |
| Instructor | PEBM | | Language | C++ |
| Object | N/A | | Function | baja\_maestros() |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <cstdio>  #include <iostream>  #include "sqlite3.h"  Using namespace std; | | |
|  | |  | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
| 1 | Imprime la opción de insertar ID del maestro para realizar la búsqueda. |
| 2 | Se manda a llamar la función de búsqueda para imprimir los datos en pantalla del maestro. |
| 3 | Se presenta la opción de borrar maestro con un “si” o “no”  Si es si: se manda a llamar la función de delete para borrar el registro del maestro .  Si es no: Se regresa al menú principal de los maestros (alta, borrar, modificar). |
|  | Fin. |

# 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | query\_maestros() |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <cstdio>  #include <iostream>  #include "sqlite3.h"  Using namespace std; | | |
|  | |  | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
| 1 | Asignados a sql=”SELECT \* FROM MAESTROS where Id\_Al=@Id\_A; |
| 2 | Asignamos a rc el resultado de preparar la sentencia sql |
| 3 | En caso de que este mal |
| 4 | imprime el error |
| 5 | En caso de que sea correcta |
| 6 | Se busca el índice del identificador a cambiar |
| 7 | Se cambia el valor por el original |
| 8 | Se comprueba cuántos resultados existen |
| 9 | En caso de que exista uno |
| 10 | Se imprimen los valores |
| 11 | Se destruye el objeto |
| 12 | Fin. |

# 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | delete\_maestros() |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <cstdio>  #include <iostream>  #include "sqlite3.h"  Using namespace std; | | |
|  | |  | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
| 1 | Asignados a sql=”Delete MAESTROS SET @Type=@Value where Id\_Al=@Id\_A; |
| 2 | Asignamos a rc el resultado de preparar la sentencia sql |
| 3 | En caso de que este mal |
| 4 | imprime el error |
| 5 | En caso de que sea correcta |
| 6 | Se busca el índice de los identificadores a cambiar |
| 7 | Se cambia el valor por el original |
| 8 | Se comprueba cuántos resultados existen |
| 9 | En caso de que exista uno |
| 10 | Se imprimen los valores |
| 11 | Se destruye el objeto |
| 12 | Fin. |

# Functional Specification Template

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Student** | | | RMM | | **Date** | 03/05/17 |
| **Program** | | | Baja Maestros | | **Program #** | N/A |
| **Instructor** | | | PEBM | | **Language** | C++ |
|  | | | | | | |
| **Class Name** | | void baja\_maestros() | | | | |
| **Parent Class** | | N/A | | | | |
|  | | | | | | |
| **Attributes** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  | char opcion | | | Corresponde a la opción que el usuario eligió. | | |
|  | char valor [40] | | | Variable para el valor del nombre. | | |
|  | sqlite3 \*db | | | Base de datos | | |
|  | | | | | | |
| **Items** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  | query\_maestros() | | | Consulta de la base de datos | | |
|  | delete\_maestros() | | | Cambio directo a la base de datos | | |
|  | printf() | | | Para imprimir directamente en la interfaz de usuario. | | |
|  | cin() | | | Para leer datos de la interfaz de usuario. | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | | void query\_maestros() | | | | |
| **Parent Class** | | N/A | | | | |
|  | | | | | | |
| **Attributes** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  | sqlite3 \*db | | | Base de datos | | |
|  | char\* id | | | Id del maestro | | |
|  | char \*sql | | | Para la sentencia sql | | |
|  | char \*err | | | Para el error de la consulta sql | | |
|  | int rc | | | Valor de consulta sql | | |
|  | int step | | | Número de resultados | | |
|  | sqlite3\_stmt \*res | | | Resultado de consulta sql | | |
|  | | | | | | |
| **Items** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  | sqlite3\_prepare\_v2() | | | Prepara la sentencia sql para la consulta | | |
|  | SQLITE\_OK | | | Para saber si es correcta la sentencia sql | | |
|  | sqlite3\_bind\_parameter\_index() | | | Busca dónde está el identificador del parámetro a cambiar. | | |
|  | sqlite3\_bind\_text() | | | Inyecta un valor en la sentencia sql | | |
|  | sqlite3\_step() | | | Resultados de la consulta | | |
|  | SQLITE\_ROW | | | Para saber si hay datos | | |
|  | sqlite3\_column\_text() | | | Localiza el valor de la consulta en memoria | | |
|  | sqlite3\_finalize() | | | Finaliza el objeto creado. | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | | void delete\_maestro() | | | | |
| **Parent Class** | | N/A | | | | |
|  | | | | | | |
| **Attributes** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  | sqlite3 \*db | | | Base de datos | | |
|  | char\* id | | | Id del maestro | | |
|  | char \*sql | | | Para la sentencia sql | | |
|  | char \*err | | | Para el error de la consulta sql | | |
|  | int rc | | | Valor de consulta sql | | |
|  | int step | | | Número de resultados | | |
|  | sqlite3\_stmt \*res | | | Resultado de consulta sql | | |
|  | char\* valor | | | Valor a cambiar | | |
|  | char\* type | | | Tipo de valor a cambiar | | |
|  | | | | | | |
| **Items** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  | sqlite3\_prepare\_v2() | | | Prepara la sentencia sql para la consulta | | |
|  | SQLITE\_OK | | | Para saber si es correcta la sentencia sql | | |
|  | sqlite3\_bind\_parameter\_index() | | | Busca dónde está el identificador del parámetro a cambiar. | | |
|  | sqlite3\_bind\_text() | | | Inyecta un valor en la sentencia sql | | |
|  | sqlite3\_step() | | | Resultados de la consulta | | |
|  | SQLITE\_ROW | | | Para saber si hay datos | | |
|  | sqlite3\_column\_text() | | | Localiza el valor de la consulta en memoria | | |
|  | sqlite3\_finalize() | | | Finaliza el objeto creado. | | |

# Operational Scenario Template

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | RMM | **Date** | 03/05/17 |
| **Program** | Bajas Maestros | **Program #** | N/A |
| **Instructor** | PEBM | **Language** | C++ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario Number** | **1** | **User Objective** | Dar de baja un maestro. | |
| **Scenario Objective** | | Borrar un maestro, opción Si | | |
| **Source** | **Step** | **Action** | | **Comments** |
| Usuario | 1 | Abrir el programa. | |  |
| Programa | 2 | Se muestra el menú de Maestros en la opción de dar de baja. | | Menú de maestros de alta, baja y modificar. |
| Usuario | 3 | Selecciona la opción de Baja. | | La opción 2 es para bajas. |
| Programa | 4 | Pide el id del maestro | | Este id es el del maestro que se desea cambiar los datos. |
| Usuario | 5 | Introduce el id del maestro | | Para hacer el query del maestro |
| Programa | 6 | Leer opción del usuario | | Verificar que sea valida |
| Programa | 7 | Muestra los datos del maestro | | Imprime en pantalla los datos del maestro |
| Programa | 8 | Se le pregunta al usuario si desea eliminar los datos que se muestran en pantalla del maestro. | | Al seleccionar solo son dos opciones “Si” y “No” |
| Usuario | 8 | Introduce la opción de Si- | |  |
| Programa | 9 | Muestra un mensaje de “ Datos eliminados con éxito” y regrese al menú de maestros. | |  |
| Usuario | 10 | Visualiza el mensaje de confirmación y regresa al menú principal de maestros. | |  |
| Programa | 11 | Fin de la función Baja maestros | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario Number** | **1** | **User Objective** | Dar de baja un maestro. | |
| **Scenario Objective** | | Borrar un maestro, opción No | | |
| **Source** | **Step** | **Action** | | **Comments** |
| Usuario | 1 | Abrir el programa. | |  |
| Programa | 2 | Se muestra el menú de Maestros en la opción de dar de baja. | | Menú de maestros de alta, baja y modificar. |
| Usuario | 3 | Selecciona la opción de Baja. | | La opción 2 es para bajas. |
| Programa | 4 | Pide el id del maestro | | Este id es el del maestro que se desea cambiar los datos. |
| Usuario | 5 | Introduce el id del maestro | | Para hacer el query del maestro |
| Programa | 6 | Leer opción del usuario | | Verificar que sea valida |
| Programa | 7 | Muestra los datos del maestro | | Imprime en pantalla los datos del maestro |
| Programa | 8 | Se le pregunta al usuario si desea eliminar los datos que se muestran en pantalla del maestro. | | Al seleccionar solo son dos opciones “Si” y “No” |
| Usuario | 8 | Introduce la opción de “No” | |  |
| Programa | 9 | Muestra un mensaje de alerta “ Los datos no serán eliminados” y regresa al menú principal del maestro. | |  |
| Usuario | 10 | Visualiza el mensaje de alerta del programa y este mismo lo lleva al menú principal de los maestros. | |  |
| Programa | 11 | Fin de la función Baja maestro | |  |

# V.- Design CheckList.

# Revisión diseño

**PSP2 Design Review Checklist**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | RMM | Date | 03/05/17 |
| Program | Baja Maestros | Program # | N/A |
| Instructor | PEBM | Language | C++ |

|  |  |
| --- | --- |
| Purpose | To guide you in conducting an effective design review |
| General | - Review the entire program for each checklist category; do not attempt to review for more than one category at a time!  - As you complete each review step, check off that item in the box at the right.  - Complete the checklist for one program or program unit before reviewing the next. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Complete | Verify that the design covers all of the applicable requirements.  - All specified outputs are produced.  - All needed inputs are furnished.  - All required includes are stated. | MB  X | B    X  X | R | M |
| External Limits | Where the design assumes or relies upon external limits, determine if behavior is correct at nominal values, at limits, and beyond limits. |  |  |  |  |
| Logic | - Verify that program sequencing is proper.  Stacks, lists, and so on are in the proper order.  Recursion unwinds properly.  - Verify that all loops are properly initiated, incremented, and terminated.  - Examine each conditional statement and verify all cases. | X        X | X |  |  |
| Internal Limits | Where the design assumes or relies upon internal limits, determine if behavior is correct at nominal values, at limits, and beyond limits. |  |  |  |  |
| Special Cases | - Check all special cases.  - Ensure proper operation with empty, full, minimum, maximum, negative, and zero values for all variables.  - Protect against out-of-limits, overflow, and underflow conditions.  - Ensure “impossible” conditions are absolutely impossible.  - Handle all possible incorrect or error conditions. | X  X    X  X  X |  |  |  |
| Functional Use | - Verify that all functions, procedures, or methods are fully understood and properly used.  - Verify that all externally referenced abstractions are precisely defined. | X    X |  |  |  |
| System Considerations | - Verify that the program does not cause system limits to be exceeded.  - Verify that all security-sensitive data are from trusted sources.  - Verify that all safety conditions conform to the safety specifications. | X    X    X |  |  |  |
| Names | Verify that  - all special names are clear, defined, and authenticated  - the scopes of all variables and parameters are self-evident or defined  - all named items are used within their declared scopes | X  X  X |  |  |  |
| Standards | Ensure that the design conforms to all applicable design standards. | X |  |  |  |

# VI.- Programa Fuente.

/\*=========================================================\*/

/\*Autor: René Moratilla Montes \*/

/\*Componente: Baja Maestros \*/

/\*Fecha: 04/05/17 \*/

/\*Descripción: Codificación que muestra las bajas para maestros\*/

/\*Versión: 0.1 \*/

/\*=========================================================\*/

/\*==========================================================\*/

/\* El programa contiene lo siguiente: \*/

/\* void query\_maestros(sqlite3 \*db,char\* id\_alumno) \*/

/\* void delate\_maestros(sqlite3 \*db,char\* type,char\* value,char\* id\_alumno) \*/

/\* void borrar\_maestros(sqlite3 \*db) \*/

/\* int main() \*/

/\*==========================================================\*/

#include <cstdio>

#include <cstdlib>

#include <iostream>

#include "sqlite3.h"

using namespace std;

/\*\*\*Esta función se encarga de buscar los valores del maestro en la base de datos\*\*\*/

void query\_maestros(sqlite3 \*db,char\* id\_maestro)

{

char \*sql; /\*Para la sentencia sql \*/

char \*err=0; /\*Para el mensaje de error de la sentencia sql \*/

int rc;

sqlite3\_stmt \*res;

sql="SELECT \* FROM MAESTROS where Id\_Ma LIKE @Id\_M;"; /\* Selecciona a los maestros con un id especifico sentencia sql\*/

rc=sqlite3\_prepare\_v2(db,sql,-1,&res,0); /\* Verifica si es valida la instruccion sql que se solicita\*/

if(rc!=SQLITE\_OK){

printf(" QUERY-SQL error %s\n",err); /\* Imprime el error de la base de datos\*/

}

else{

printf(" Valores del maestro\n"); /\* Se aprueba la sentencia sql\*/

int idx = sqlite3\_bind\_parameter\_index(res, "@Id\_M"); /\* Se busca en que lugar esta el identificador a cambiar\*/

sqlite3\_bind\_text(res, idx, id\_maestro,40,SQLITE\_STATIC); /\* Se cambia el identificador por un valor determinado\*/

}

int step = sqlite3\_step(res);

if (step == SQLITE\_ROW) { /\* indica el resultado de la instruccion sql, valores del maestro\*/

printf(" Nombre:\t%s %s %s\n", sqlite3\_column\_text(res, 1),sqlite3\_column\_text(res, 2),sqlite3\_column\_text(res, 3));

printf(" Fecha:\t%s\n", sqlite3\_column\_text(res, 4));

printf(" Direccion:\t%s\n", sqlite3\_column\_text(res, 5));

printf(" Telefono:\t%s\n", sqlite3\_column\_text(res, 6));

printf(" Email:\t\t%s\n", sqlite3\_column\_text(res, 7));

printf(" RFC:\t\t%s\n", sqlite3\_column\_text(res, 0));

}

sqlite3\_finalize(res); /\* Finaliza el objeto creado\*/

sql="SELECT ID\_Materia,Status FROM IMPARTE where ID\_Maestro LIKE @Id\_M;";/\* Instruccion sql para la seleccion de materias en tabla imparte\*/

rc=sqlite3\_prepare\_v2(db,sql,-1,&res,0); /\* Verifica si es correcta la instruccion sql\*/

if(rc!=SQLITE\_OK){

printf("SQL error %s\n",err); /\* Imprime el error de la consulta\*/

}

else{

printf(" Materias del maestro\n"); /\* Cambia los valores para el identificador en especifico\*/

int idx = sqlite3\_bind\_parameter\_index(res, "@Id\_M");

sqlite3\_bind\_text(res, idx, id\_maestro,40,SQLITE\_STATIC);

}

step = sqlite3\_step(res);

while(step == SQLITE\_ROW){ /\* imprime los valores de la consulta de diferentes registros\*/

printf(" %s\t%s\n", sqlite3\_column\_text(res, 0),sqlite3\_column\_text(res, 1));

step=sqlite3\_step(res);

}

sqlite3\_finalize(res); /\* Libera el espacio del objeto\*/

}//45

/\*\*\*Esta función se encarga de borrar al maestro en la base de datos\*\*\*/

void delate\_maestros(sqlite3 \*db,char\* id\_maestro)

{

char \*sql; /\*Para la sentencia sql \*/

char \*err=0; /\*Para el mensaje de error de la sentencia sql \*/

int rc;

sqlite3\_stmt \*res;

sql="DELETE from Maestros where Id\_Ma LIKE @Id\_M;"; /\* Selecciona a los maestros con un id especifico sentencia sql\*/

rc=sqlite3\_prepare\_v2(db,sql,-1,&res,0); /\* Verifica si es valida la instruccion sql que se solicita\*/

if(rc!=SQLITE\_OK){

printf("\t\t QUERY-SQL error %s\n",err); /\* Imprime el error de la base de datos\*/

}

else{

printf("\t\t Borrado!!!\n"); /\* Se aprueba la sentencia sql\*/

int idx = sqlite3\_bind\_parameter\_index(res, "@Id\_M"); /\* Se busca en que lugar esta el identificador a cambiar id\*/

sqlite3\_bind\_text(res, idx, id\_maestro,40,SQLITE\_STATIC); /\* Se cambia el identificador por un valor determinado id\*/

}

int step = sqlite3\_step(res);

if (step == SQLITE\_ROW) { /\* indica el resultado de la instrucción sql, valores del maestro\*/

printf("\t\t Borrado!!!\n"); /\* Se aprueba la sentencia sql ya echa\*/

}

sqlite3\_finalize(res); /\* Finaliza el objeto creado\*/

}//20

void delate\_imparte(sqlite3 \*db,char\* id\_maestro)

{

char \*sql; /\*Para la sentencia sql \*/

char \*err=0; /\*Para el mensaje de error de la sentencia sql \*/

int rc;

sqlite3\_stmt \*res;

sql="DELETE from Imparte where ID\_Maestro LIKE @Id\_M;";

rc=sqlite3\_prepare\_v2(db,sql,-1,&res,0); /\* Verifica si es valida la instruccion sql que se solicita\*/

if(rc!=SQLITE\_OK){

printf("\t\t QUERY-SQL error %s\n",err); /\* Imprime el error de la base de datos\*/

}

else{

printf("\t\t Borrado!!!\n"); /\* Se aprueba la sentencia sql\*/

int idx = sqlite3\_bind\_parameter\_index(res, "@Id\_M"); /\* Se busca en que lugar esta el identificador a cambiar id\*/

sqlite3\_bind\_text(res, idx, id\_maestro,40,SQLITE\_STATIC); /\* Se cambia el identificador por un valor determinado id\*/

}

int step = sqlite3\_step(res);

if (step == SQLITE\_ROW) { /\* indica el resultado de la instrucción sql, valores del maestro\*/

printf("\t\t Borrado!!!\n"); /\* Se aprueba la sentencia sql ya echa\*/

}

sqlite3\_finalize(res); /\* Finaliza el objeto creado\*/

}//20

/\*\*\*Esta función es la funcion principal de borrado maestros\*\*\*/

void borrar\_maestros(sqlite3 \*db){

char opcion;

char valor[40];

char id[40];

do{

system("cls");

printf("\t\to----------------------------------------------------------o\n"); /\* Menú de borrar maestro\*/

printf("\t\t| ================ Borrar Maestros =================== |\n");

printf("\t\to----------------------------------------------------------o\n");

printf("\t\to----------------------------------------------------------o\n");

printf("\t\t| ¿Que desea hacer? |\n");

printf("\t\t| 1)Buscar Maestro |\n");

printf("\t\t| 2)Regresar Men%c Maestros |\n",163);

printf("\t\t| Ingresa la opci%cn que decea: |\n",162);

printf("\t\to----------------------------------------------------------o\n\t\t");

cin>>opcion;

switch(opcion){

case '1':{

do{

printf("\t\t Ingresa el ID.-"); /\* Busca a un maestro para hacer borrar\*/

scanf("%s",id);

system("cls");

printf("\t\to----------------------------------------------------------o\n");

query\_maestros(db,id); /\* Imprime datos del maestro\*/

printf("\t\t| 0)Salir |\n");

printf("\t\t| Desea borrar a este maestro (Y/N)): |\n");

printf("\t\to----------------------------------------------------------o\n\t\t");

cin>>opcion;

//printf("\t\t Ingresa la respuesta.- "); /\*Opción de borrado\*/

//scanf("%[^\n]",valor);

switch(opcion){

case 'Y':{

delate\_maestros(db,id);

delate\_imparte(db,id);

break;

}

case 'N':{

printf("\t\t Regresando a men%c\n",163);

break;

}

case '0':{

printf("\t\t Regresando a men%c\n",163);

break;

}

default:{

printf("\t\t Valor incorrecto!!! intenta otro \n");

break;

}

}

}while(opcion!='0');

break;

}

case '2':{

printf("\t\t Regresando a men%c\n",163);

break;

}

default:{

printf("\t\t Valor incorrecto!!! intenta otro \n");

break;

}

}

}while(opcion !='2');

}//61

int main()

{

sqlite3 \*db; /\* Base de datos en sqlite3\*/

int res;

res=sqlite3\_open("test.db", &db); /\* Se abre la db\*/

borrar\_maestros(db); /\*A borrar maestros\*/

return 0;

}

# VII.- Code CheckList.

# Revisión codificación

|  |  |  |  |
| --- | --- | --- | --- |
| Student | RMM | Date | 08/05/17 |
| Program | Bajas Maestros | Program # | N/A |
| Instructor | PEBM | Language | C++ |

|  |  |
| --- | --- |
| Purpose | To guide you in conducting an effective code review |
| General | - Review the entire program for each checklist category; do not attempt to review for more than one category at a time!  - As you complete each review step, check off that item in the box at the right.  - Complete the checklist for one program or program unit before reviewing the next. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Complete | Verify that the code covers all of the design. | MB | B | R | M |
| Includes | Verify that the includes are complete. | X |  |  |  |
| Initialization | Check variable and parameter initialization.  - at program initiation  - at start of every loop  - at class/function/procedure entry | X  X  X |  |  |  |
| Calls | Check function call formats.  - pointers  - parameters  - use of ‘&’ | X  X  X |  |  |  |
| Names | Check name spelling and use.  - Is it consistent?  - Is it within the declared scope?  - Do all structures and classes use ‘.’ reference? | X  X  X |  |  |  |
| Strings | Check that all strings are  - identified by pointers  - terminated by NULL | X  X |  |  |  |
| Pointers | Check that all  - pointers are initialized NULL  - pointers are deleted only after new  - new pointers are always deleted after use | X  X  X |  |  |  |
| Output Format | Check the output format.  - Line stepping is proper.  - Spacing is proper. | X  X |  |  |  |
| () Pairs | Ensure that () are proper and matched. |  |  |  |  |
| Logic Operators | - Verify the proper use of ==, =, ||, and so on.  - Check every logic function for (). | X  X |  |  |  |
| Line-by-line check | Check every line of code for  - instruction syntax  - proper punctuation | X  X |  |  |  |
| Standards | Ensure that the code conforms to the coding standards. | X |  |  |  |
| File Open and Close | Verify that all files are  - properly declared  - opened  - closed | X  X  X |  |  |  |

# VIII.- Reporte de resultados.

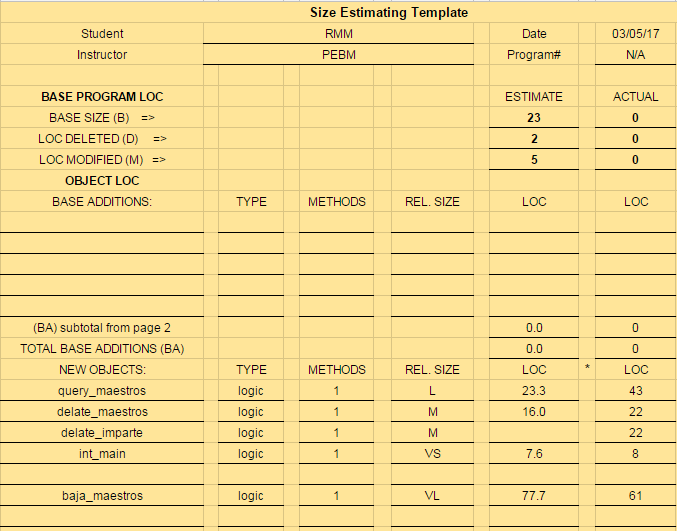
# Template de pruebas

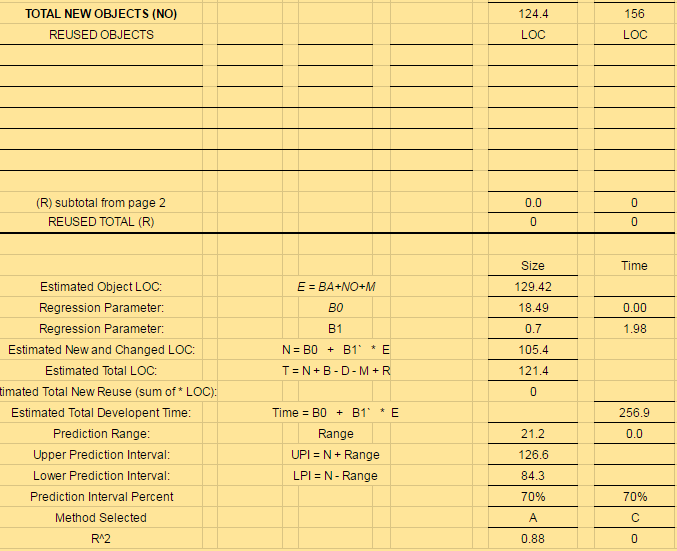
|  |  |
| --- | --- |
| Nombre de la Prueba | Prueba de Baja de maestros |
| Caso de Uso | 3.1 |
| Objetivo de la Prueba | Verificar la correcta eliminación de los datos de un maestro en la DB. |
| Descripción de la Prueba | Se ingresa como usuario Administrador y se realiza por medio de la opción de “Baja de Maestro”. |
| Test | 1.-Direccionarse a la opción Baja Maestro.  2.- Ingresar ID de Maestro.  3.-Confirmación de eliminación. |
| Resultados Esperados | Mensaje de Éxito:  Datos Maestro (Se hará consulta a DB)   1. Nombres 2. Apellido paterno 3. Apellido materno 4. Fecha de nacimiento 5. Dirección 6. Correo electrónico 7. Teléfono   “Ha sido borrado ‘Maestro´ de Sistema” (Si no se encuentra en Consulta a DB)  Mensaje de Error:  “No fue borrado de sistema” (Se encuentra en consulta a DB). |
| Capturas de pantalla |  |
| Estado caso de prueba | Se realizan las bajas correctamente para todos los registros relacionados a un maestro. (registro de prueba, maestro con ID VIVAG). |
| Errores obtenidos | No se podían dar de baja los registros del maestro en la tabla de imparte. |
| Responsable de prueba | RMM y GVV |
| Comentarios | Sin comentarios |

# IX.- PIP

|  |  |  |
| --- | --- | --- |
| Menú Reportes Planes | | |
| No. PIP | Descripción del Problema | Descripción de la propuesta. |
| 1 | Se detectaron problemas en un borrado en la fase de pruebas | Realizar una mejor revisión y corrida manual antes de realizar la compilación y las pruebas |
|  |  |  |
| Nombre Módulo | | |
| No. PIP | Descripción del Problema | Descripción de la propuesta. |

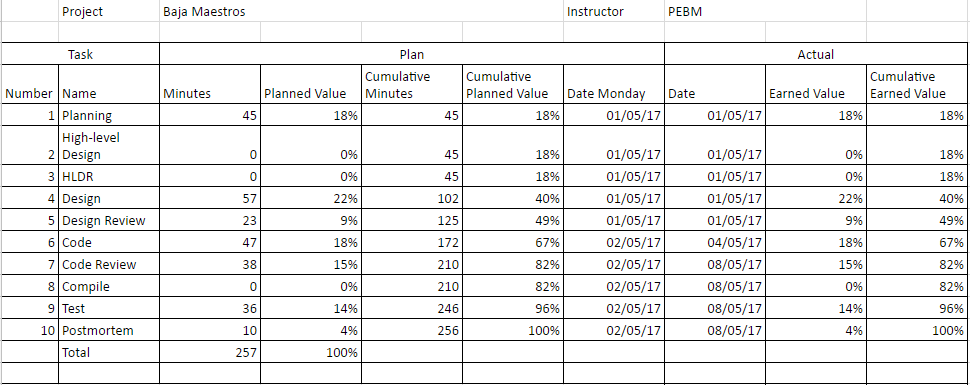
# X.- Size Estimating Template





**XI.- Task and Schedule**

Task



Schedule

