

Postmortem Cambios Alumnos

07/05/17

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\_Cambios\_Alumnos-070517.docx | 0.1 | 07/05/17 | JAGC | 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**

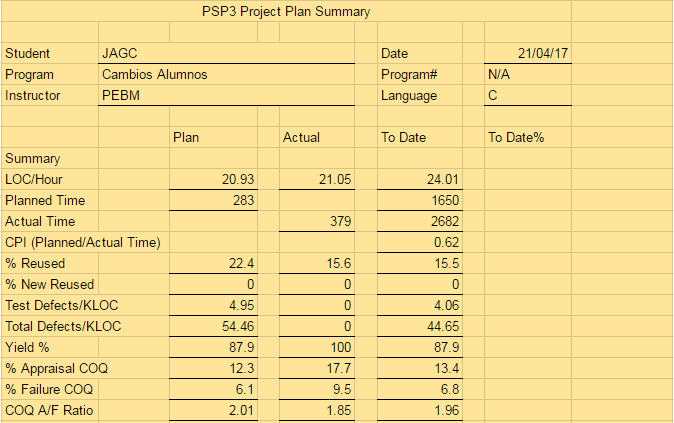
[**IX.- PIP**](#_4ocjjwajumc0) **4**

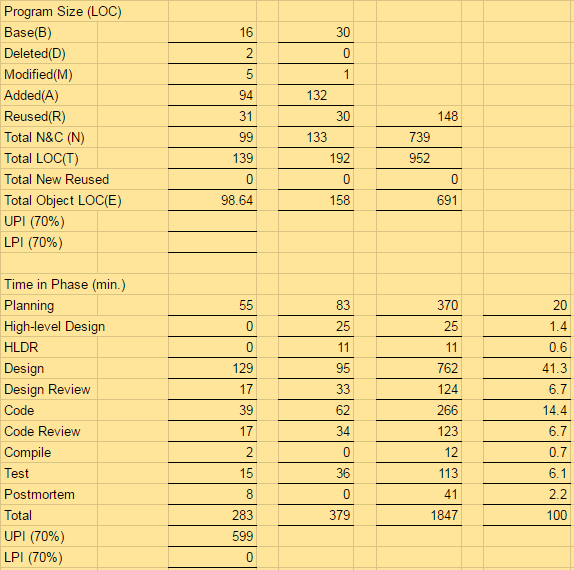
[**X.- Size Estimating Template**](#_w1ych8jaj7as) **5**

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

# I.- Forma PSP Project Plan Summary.

Se inició el proceso el 21/04/17





# 

# 

# II.- Time Recording Log.



# III.- Defect Recording Log.

N/A

# IV.- Diseño.

**State Specification Template**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Student | | JAGC | | | Date | | 01/05/17 |
| Program | | Cambios Alumnos | | | Program # | | N/A |
| Instructor | | PEBM | | | Language | | C++ |
|  | | |  | | | | |
| Cambios alumnos | | | | | | | |
| **State Name** | | | | **Description** | | | |
| Inicio | | | | Inicia el programa. | | | |
| Cambios alumnos | | | | Función principal para el cambio de alumnos. | | | |
| Query alumnos | | | | Consulta a base de datos de alumnos | | | |
| Update alumnos | | | | Función para hacer cambios en la base de datos de alumnos. | | | |
| Salir | | | | Termina el programa | | | |
|  | | | |  | | | |
| **Function/Parameter** | | | | **Description** | | | |
| printf() | | | | Imprime datos en la pantalla | | | |
| scanf() | | | | Lee datos de la pantalla | | | |
| opción | | | | opción que elegirá el usuario | | | |
| cambios\_alumnos() | | | | Función principal de cambios | | | |
| query\_alumnos() | | | | Función de consulta | | | |
| update\_alumnos() | | | | Función de cambio base de datos | | | |
| **States/Next States** | | | | **Transition Condition** | | **Action** | |
| Inicio | | | |  | |  | |
|  | Cambios alumnos | | | True | | printf()-imprime el menú  scanf()- lee opción | |
| Cambios alumnos | | | |  | |  | |
|  | Query alumnos | | | buscar un alumno | | query\_alumnos() | |
|  | Update alumnos | | | Valor de campo cambiado | | update\_alumnos() | |
|  | Salir | | | opción=5 | | Fin. | |

# Logic Specification Template

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student | JAGC | | Date | 01/05/17 |
| Program | Cambios Alumnos | | Program # | N/A |
| Instructor | PEBM | | Language | C++ |
| Object | N/A | | Function | cambios\_alumnos() |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <cstdio>  #include <iostream>  #include "sqlite3.h"  Using namespace std; | | |
|  | |  | | |

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

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
| 1 | Imprime el menú cambios alumnos con todas sus opciones |
| 2 | Leer la opción del usuario |
| 3 | En caso de que sea 1 |
| 4 | Pide el id del alumno |
| 5 | En caso de que exista |
| 6 | Va a la función query\_alumnos() |
| 7 | Imprime todos los datos del alumno |
| 8 | Pide la opción del dato a cambiar |
| 9 | lee la opción |
| 10 | En caso de que sea 1 |
| 11 | Pide el valor del campo del cambio nombre |
| 12 | Manda el valor a la función update\_alumnos() |
| 13 | En caso de que sea 2 |
| 14 | Pide el valor del campo del cambio ap paterno |
| 15 | Manda el valor a la función update\_alumnos() |
| 16 | En caso de que sea 3 |
| 17 | Pide el valor del campo del cambio ap materno |
| 18 | Manda el valor a la función update\_alumnos() |
| 19 | En caso de que sea 4 |
| 20 | Pide el valor del campo del cambio fecha |
| 21 | Manda el valor a la función update\_alumnos() |
| 22 | En caso de que sea 5 |
| 23 | Pide el valor del campo del cambio direccion |
| 24 | Manda el valor a la función update\_alumnos() |
| 25 | En caso de que sea 6 |
| 26 | Pide el valor del campo del cambio teléfono |
| 27 | Manda el valor a la función update\_alumnos() |
| 28 | En caso de que sea 7 |
| 29 | Pide el valor del campo del cambio email |
| 30 | Manda el valor a la función update\_alumnos() |
| 31 | En caso de que sea 8 |
| 32 | Pide el valor del campo del cambio plan |
| 33 | Manda el valor a la función update\_alumnos() |
| 34 | En caso de que sea 9 |
| 35 | Pide el valor del campo del cambio semestre |
| 36 | Manda el valor a la función update\_alumnos() |
| 37 | En caso de que sea 10 |
| 38 | Regresa al menú cambios alumnos |
| 39 | En caso de que sea otra cosa |
| 40 | Pide de nuevo el valor |
| 41 | En caso que el id no exista |
| 42 | Imprime que es incorrecto el id |
| 43 | regresa a el menú cambios alumnos |
| 44 | En caso de que sea 2 |
| 45 | Sale del ciclo. |
| 46 | Fin. |

# 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | query\_alumnos() |
| **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 ALUMNOS 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 | update\_alumnos() |
| **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=”UPDATE ALUMNOS 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** | | | JAGC | | **Date** | 01/05/17 |
| **Program** | | | Cambios Alumnos | | **Program #** | N/A |
| **Instructor** | | | PEBM | | **Language** | C++ |
|  | | | | | | |
| **Class Name** | | void cambios\_alumnos() | | | | |
| **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\_alumnos() | | | Consulta de la base de datos | | |
|  | update\_alumnos() | | | 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. | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | void update\_alumnos() | | | | |
| **Parent Class** | | N/A | | | | |
|  | | | | | | |
| **Attributes** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  | sqlite3 \*db | | | Base de datos | | |
|  | char\* id | | | Id del alumno | | |
|  | 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** | JAGC | **Date** | 01/05/17 |
| **Program** | Cambios Alumnos | **Program #** | N/A |
| **Instructor** | PEBM | **Language** | C++ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario Number** | **1** | **User Objective** | Cambiar un dato del alumno | |
| **Scenario Objective** | | Pasos para hacer cambios en los datos del alumno | | |
| **Source** | **Step** | **Action** | | **Comments** |
| Usuario | 1 | Abrir el programa. | |  |
| Programa | 2 | Se muestra el menú cambios alumnos para elegir entre buscar a un alumno o regresar al menú alumnos | |  |
| Usuario | 3 | Introduce la opción buscar alumno | | La opción 1 para buscar a un alumno por el id. |
| Programa | 4 | Pide el id del alumno | | Este id es el del alumno que se desea cambiar los datos. |
| Usuario | 5 | Introduce el id del alumno | | Para hacer el query del alumno |
| Programa | 6 | Leer opción del usuario | | Verificar que sea valida |
| Programa | 7 | Muestra los datos del alumno | | Imprime en pantalla los datos del alumno |
| Programa | 8 | Pide el número de opción que desea modificar | | Cada dato tiene un número para la selección |
| Usuario | 8 | Introduce la opción del campo a modificar | | Las opciones solo corresponden a los campos de alumno |
| Programa | 9 | Pide el valor a modificar del alumno | | Solamente va a leer el valor después un salto de línea. |
| Usuario | 10 | Introduce el valor del cambio | |  |
| Programa | 11 | Manda a hacer el update | | Modifica en la base de datos. |
| Programa | 12 | Imprime los datos del alumno | | Se muestra una opción para salir o buscar a otro alumno. |
| Usuario | 13 | Introduce la opción de salir | | Fin. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario Number** | **2** | **User Objective** | Salir sin hacer cambio 1 | |
| **Scenario Objective** | | Salir sin buscar a un alumno y no hacer cambios. | | |
| **Source** | **Step** | **Action** | | **Comments** |
| Usuario | 1 | Abrir el programa. | |  |
| Programa | 2 | Se muestra el menú cambios alumnos para elegir entre buscar a un alumno o regresar al menú alumnos | |  |
| Usuario | 3 | Introduce la opción regresar al menú alumnos | | La opción 2 para regresar. |
| Programa | 4 | Termina el programa | | Fin. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario Number** | **3** | **User Objective** | Salir sin hacer cambio 2 | |
| **Scenario Objective** | | salir sin hacer cambios del alumno | | |
| **Source** | **Step** | **Action** | | **Comments** |
| Usuario | 1 | Abrir el programa. | |  |
| Programa | 2 | Se muestra el menú cambios alumnos para elegir entre buscar a un alumno o regresar al menú principal | |  |
| Usuario | 3 | Introduce la opción buscar alumno | | La opción 1 para buscar a un alumno por el id. |
| Programa | 4 | Pide el id del alumno | | Este id es el del alumno que se desea cambiar los datos. |
| Usuario | 5 | Introduce el id del alumno | | Para hacer el query del alumno |
| Programa | 6 | Leer opción del usuario | | Verificar que sea valida |
| Programa | 7 | Muestra los datos del alumno | | Imprime en pantalla los datos del alumno |
| Programa | 8 | Pide el número de opción que desea modificar | | Cada dato tiene un número para la selección |
| Usuario | 8 | Introduce la opción regresar al menú cambios alumno | | La última opción. |
| Programa | 9 | Regresa al menú de cambios | | Paso 2 |
| Usuario | 10 | Introduce opción de salir | | La número 2 |
| Programa | 11 | Fin programa | | Fin. |

# V.- Design CheckList.

# Revisión diseño

**PSP2 Design Review Checklist**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | JAGC | Date | 01/05/17 |
| Program | Creacion Alumnos | 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  X  X | B | 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: Juan Alberto Gutierrez Canto \*/

/\*Componente: Cambios Alumnos \*/

/\*Fecha: 02/05/17 \*/

/\*Descripción: Codificación que muestra los cambios para alumnos\*/

/\*Versión: 0.1 \*/

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

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

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

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

/\* void update\_alumnos(sqlite3 \*db,char\* type,char\* value,char\* id\_alumno) \*/

/\* void cambios\_alumnos(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 alumno en la base de datos\*\*\*/

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

{

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 Alumnos WHERE ID\_Al LIKE @Id\_A ;"; /\* Selecciona a los alumnas 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 Los Valores del Alumno\n"); /\* Se aprueba la sentencia sql\*/

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

sqlite3\_bind\_text(res, idx, id\_alumno,45,SQLITE\_STATIC); /\* Se cambia el identificador por un valor determinado\*/

}

int step = sqlite3\_step(res);

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

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

printf("\t\t 2)Apellido paterno: %s\n", sqlite3\_column\_text(res, 2));

printf("\t\t 3)Apellido materno: %s\n", sqlite3\_column\_text(res, 3));

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

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

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

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

printf("\t\t 8)Plan: %s\n", sqlite3\_column\_text(res, 8));

printf("\t\t 9)Semestre:\%s\n", sqlite3\_column\_text(res, 9));

}

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

}

/\*\*\*Esta función se encarga de actualizar al alumno en la base de datos\*\*\*/

void update\_alumnos(sqlite3 \*db,int type,char\* value,char\* id\_alumno)

{

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

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

int rc;

sqlite3\_stmt \*res;

switch(type){

case 1:{

sql="UPDATE Alumnos SET Nombre = @Value where ID\_Al LIKE @Id\_A;"; /\* Selecciona a los alumnas con un id especifico sentencia sql\*/

break;

}

case 2:{

sql="UPDATE Alumnos SET ApPaterno = @Value where ID\_Al LIKE @Id\_A;"; /\* Selecciona a los alumnas con un id especifico sentencia sql\*/

break;

}

case 3:{

sql="UPDATE Alumnos SET ApMaterno = @Value where ID\_Al LIKE @Id\_A;"; /\* Selecciona a los alumnas con un id especifico sentencia sql\*/

break;

}

case 4:{

sql="UPDATE Alumnos SET FNacimiento = @Value where ID\_Al LIKE @Id\_A;"; /\* Selecciona a los alumnas con un id especifico sentencia sql\*/

break;

}

case 5:{

sql="UPDATE Alumnos SET Direccion = @Value where ID\_Al LIKE @Id\_A;"; /\* Selecciona a los alumnas con un id especifico sentencia sql\*/

break;

}

case 6:{

sql="UPDATE Alumnos SET Telefono = @Value where ID\_Al LIKE @Id\_A;"; /\* Selecciona a los alumnas con un id especifico sentencia sql\*/

break;

}

case 7:{

sql="UPDATE Alumnos SET Email = @Value where ID\_Al LIKE @Id\_A;"; /\* Selecciona a los alumnas con un id especifico sentencia sql\*/

break;

}

case 8:{

sql="UPDATE Alumnos SET ID\_Plan = @Value where ID\_Al LIKE @Id\_A;"; /\* Selecciona a los alumnas con un id especifico sentencia sql\*/

break;

}

case 9:{

sql="UPDATE Alumnos SET Semestre = @Value where ID\_Al LIKE @Id\_A;"; /\* Selecciona a los alumnas con un id especifico sentencia sql\*/

break;

}

}

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 Actualizando!!!\n"); /\* Se aprueba la sentencia sql\*/

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

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

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

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

}

int step = sqlite3\_step(res);

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

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

}

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

}

/\*\*\*Esta función es la funcion principal del cambio de alumnos\*\*\*/

void cambios\_alumnos(sqlite3 \*db){

char opcion;

char valor[40];

char id[40];

do{

system("cls");

printf("\t\to----------------------------------------------------------o\n"); /\* Menú de cambios alumnos\*/

printf("\t\t| ================ Cambios Alumnos =================== |\n");

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

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

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

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

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

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

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

scanf("%c",&opcion);

switch(opcion){

case '1':{

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

fflush( stdin );

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

system("cls");

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

query\_alumnos(db,id); /\* Imprime datos del alumno\*/

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

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

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

fflush( stdin );

scanf("%c",&opcion);

if(opcion !='0'){

printf("\t\t Ingresa el nuevo valor.- "); /\* Nuevo valor a actualizar en la base de datos\*/

fflush( stdin );

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

}

switch(opcion){

case '1':{

update\_alumnos(db,1,valor,id); /\* Actualiza nombre\*/

break;

}

case '2':{

update\_alumnos(db,2,valor,id); /\* Actualiza ap paterno\*/

break;

}

case '3':{

update\_alumnos(db,3,valor,id); /\* Actualiza ap materno\*/

break;

}

case '4':{

update\_alumnos(db,4,valor,id); /\* Actualiza fecha nacimiento\*/

break;

}

case '5':{

update\_alumnos(db,5,valor,id); /\* Actualiza direccion\*/

break;

}

case '6':{

update\_alumnos(db,6,valor,id); /\* Actualiza telefono\*/

break;

}

case '7':{

update\_alumnos(db,7,valor,id); /\* Actualiza email\*/

break;

}

case '8':{

update\_alumnos(db,8,valor,id); /\* Actualiza id\_plan\*/

break;

}

case '9':{

update\_alumnos(db,9,valor,id); /\* Actualiza semestre\*/

break;

}

case '0':{

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

break;

}

default:{

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

break;

}

}

printf("\t\t Ingresa 1 para continuar\n");

fflush( stdin );

scanf("%c",&opcion);

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');

}

int main()

{

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

int res;

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

cambios\_alumnos(db); /\*A cambios alumnos\*/

return 0;

}

# VII.- Code CheckList.

# Revisión codificación

|  |  |  |  |
| --- | --- | --- | --- |
| Student | JAGC | Date | 02/05/17 |
| Program | Cambios Alumnos | 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. |  |  |  |  |
| File Open and Close | Verify that all files are  - properly declared  - opened  - closed | X  X  X |  |  |  |

# VIII.- Reporte de resultados.

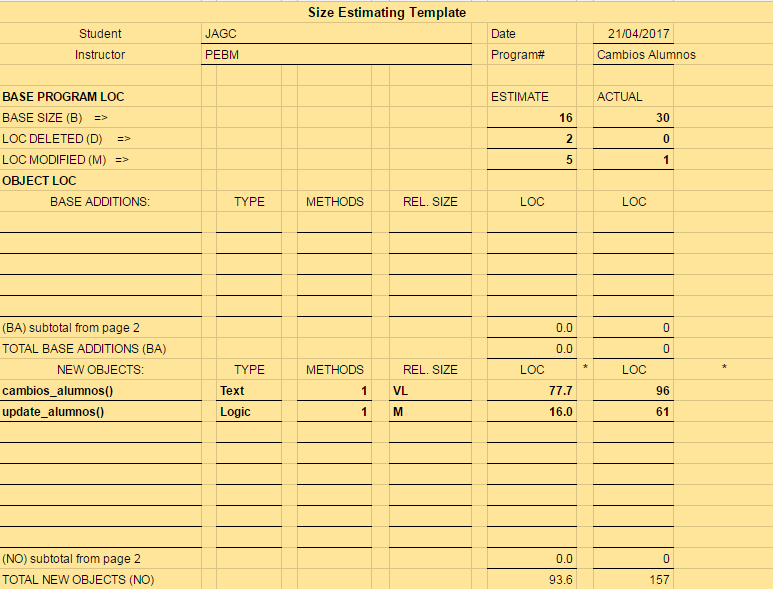
# Template de pruebas

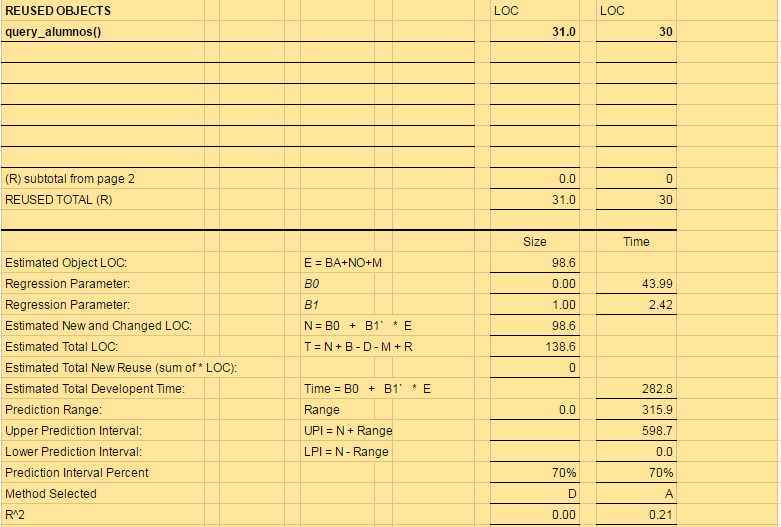
|  |  |
| --- | --- |
| Nombre de la Prueba | Prueba de cambio de alumno |
| Caso de Uso | 2.2 |
| Objetivo de la Prueba | Verificar en DB que la modificación realizada sea correcta. |
| Descripción de la Prueba | Se ingresa a Sistema como usuario Administrador y se realiza por medio de la opción de Cambio Alumno. |
| Test | 1.- Direccionarse a la opción Cambio Alumno.  2.- Ingresar ID alumno.  3.- Seleccionar Campo a modificar.  4.- Ingresar datos nuevos.  5.- Confirmar  6.- Se repite el paso 3 hasta seleccionar la opción salir. |
| Resultados Esperados | Mensaje de Éxito.  Datos Alumno (Se hará consulta a DB)   1. Nombres 2. Apellido paterno 3. Apellido materno 4. Fecha de nacimiento 5. Dirección 6. Correo electrónico 7. Teléfono   Mensaje de Error:  “No se ha modificado el valor ‘Campo a Cambiar’” |
| Capturas De Pantalla |  |
| Estado caso de prueba | Se realizan los cambios correctamente, se probó con el alumno con ID VIVG250694 |
| Errores obtenidos | Los querys estaban mal diseñados y no imprimían los valores almacenados en la BD, los update se optimizaron porque no se actualizaban los campos indicados. |
| Responsable de prueba | JAGC |
| Comentarios | Sin comentarios |

# IX.- PIP

|  |  |  |
| --- | --- | --- |
| Menú Reportes Planes | | |
| No. PIP | Descripción del Problema | Descripción de la propuesta. |
| 1 | Se llevó demasiado tiempo en el diseño | Planear un tiempo mayor en esta tarea |
|  |  |  |
| Nombre Módulo | | |
| No. PIP | Descripción del Problema | Descripción de la propuesta. |

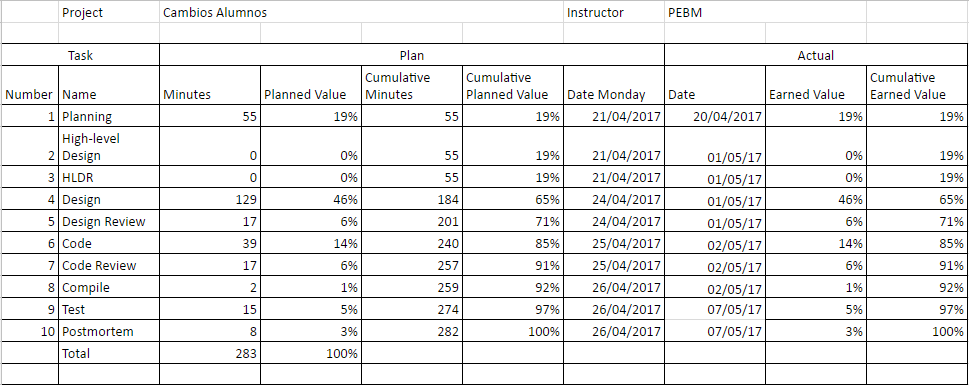
# X.- Size Estimating Template





**XI.- Task and Schedule**

Task



Schedule

