

Postmortem Alta Alumnos.

05/04/17

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\_Alta-Alumnos-050417.docx | 0.1 | 05/04/17 | JAGB | Creación del documento. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Índice

[**Índice**](#_kd5bfhkalq0n) **2**

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

[**II.- Time Recording Log.**](#_7e50s7cf9wc3) **9**

[**III.- Defect Recording Log.**](#_3znysh7) **10**

[**IV.- Diseño.**](#_m4hw86ggwsmr) **13**

[**V.- Design CheckList.**](#_92tu2pewgguj) **18**

[**VI.- Programa Fuente.**](#_6bvnqqrwimi9) **20**

[**VII.- Code CheckList.**](#_87ns93g4124c) **24**

[**VIII.- Reporte de resultados.**](#_bo7s7l2hg3ra) **27**

[**Template de pruebas**](#_1fob9te) **27**

[**IX.- PIP**](#_4ocjjwajumc0) **28**

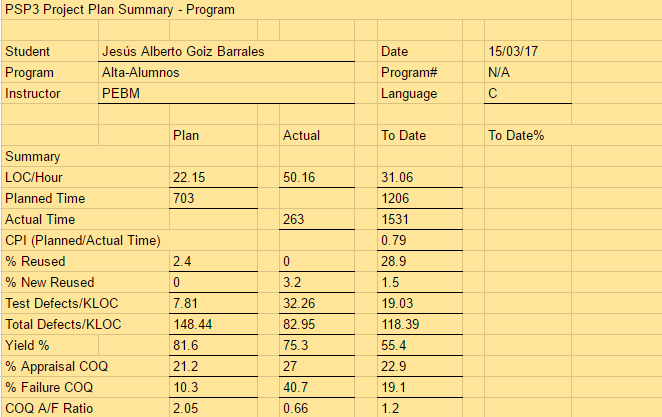
[**X.- Size Estimating Template**](#_w1ych8jaj7as) **29**

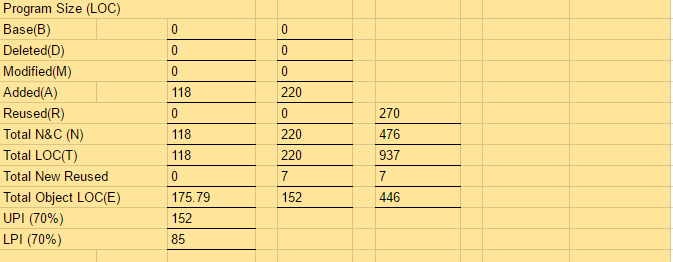
[**XI.- Task and Schedule**](#_195jonwtvlxs) **32**

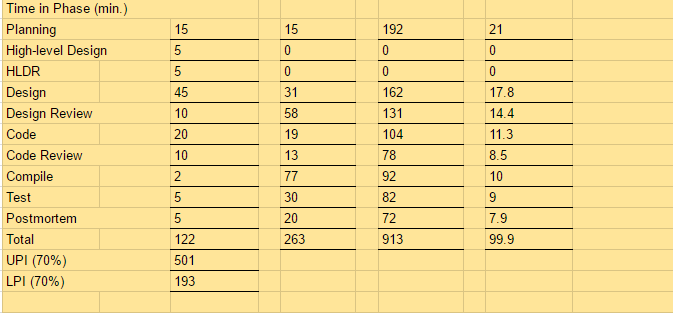
# 

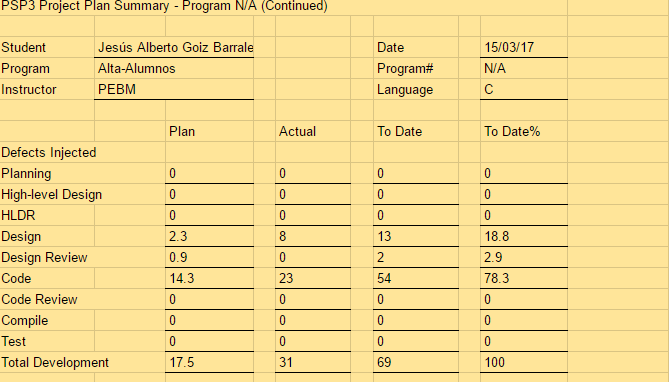
# 

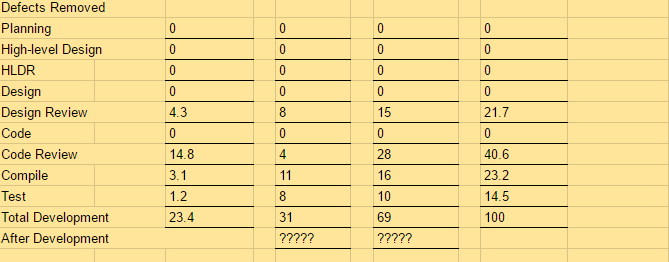
# I.- Forma PSP Project Plan Summary.

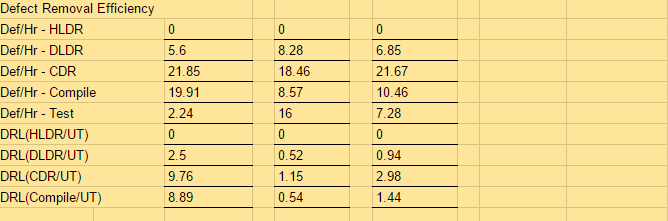












# II.- Time Recording Log.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Semana | Día | Fecha | Inicio | Fin | Delta | Fase | Actividades realizadas | Integrante (s) |
| 9 | 4 | 03/03/17 | 17:01 | 18:30 | 1:29 | Ciclo1-semana5 | Tiempo overhead, entender Insert, update. | JAGB |
| 9 | 4 | 03/03/17 | 20:00 | 22:10 | 2:10 | Ciclo1-semana5 | Tiempo overhead, entender Insert, update. | JAGB |
| 9 | 4 | 03/03/17 | 22:30 | 23:20 | 0:50 | Ciclo1-semana5 | Tiempo overhead, entender Insert, update. | JAGB |
| 9 | 6 | 05/03/17 | 0:05 | 0:12 | 0:07 | Ciclo1-semana5 | Etapa compilación-AltaAlumnos | JAGB |
| 9 | 6 | 05/03/17 | 1:05 | 1:35 | 0:30 | Ciclo1-semana5 | Etapa compilación-AltaAlumnos | JAGB |
| 9 | 6 | 05/03/17 | 16:20 | 17:00 | 0:40 | Ciclo1-semana5 | Etapa compilación-AltaAlumnos | JAGB |
| 9 | 6 | 05/03/17 | 17:15 | 17:45 | 0:30 | Ciclo1-semana5 | Etapa Pruebas-AltaAlumnos | JAGB |
| 9 | 6 | 05/03/17 | 20:40 | 21:00 | 0:20 | Ciclo1-semana5 | Etapa PM- AltaAlumnos | JAGB |

# III.- Defect Recording Log.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. De defecto | Componente | Fase inyectado | Fase removido | Tiempo | Descripcíon |  |
| 40 | Csoft-Revisión\_Diseño\_Alta-Alumnos-080317.docx | C1-semana4-Diseño | C1-semana4-Revisión de diseño | 2 | Error en variables | 8 |
| 40 | Csoft-Revisión\_Diseño\_Alta-Alumnos-080317.docx | C1-semana4-Diseño | C1-semana4-Revisión de diseño | 2 | Error en variables | 8 |
| 40 | Csoft-Revisión\_Diseño\_Alta-Alumnos-080317.docx | C1-semana4-Diseño | C1-semana4-Revisión de diseño | 2 | Error en variables | 8 |
| 40 | Csoft-Revisión\_Diseño\_Alta-Alumnos-080317.docx | C1-semana4-Diseño | C1-semana4-Revisión de diseño | 2 | Error en variables | 8 |
| 80 | Csoft-Revisión\_Diseño\_Alta-Alumnos-080317.docx | C1-semana4-Diseño | C1-semana4-Revisión de diseño | 5 | Error en la función de alumnos | 8 |
| 80 | Csoft-Revisión\_Diseño\_Alta-Alumnos-080317.docx | C1-semana4-Diseño | C1-semana4-Revisión de diseño | 4 | Error en la función de alumnos | 8 |
| 40 | Csoft-Revisión\_Diseño\_Alta-Alumnos-080317.docx | C1-semana4-Diseño | C1-semana4-Revisión de diseño | 1 | Error en declaración de variable | 8 |
| 80 | Csoft-Revisión\_Diseño\_Alta-Alumnos-080317.docx | C1-semana4-Diseño | C1-semana4-Revisión de diseño | 6 | Error en la función de alumnos | 8 |
| 80 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Codifcación | 10 | Error en la función de allta\_alumno | 9 |
| 80 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Compilación | 15 | Error en la función para imprimir txt | 9 |
| 80 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Compilación | 10 | Error en la función insert\_alumnos | 9 |
| 40 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Compilación | 2 | Error en las declaración de variables en la función insert\_alumnos | 9 |
| 80 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Compilación | 5 | Error en la función callback, error en el llamado de la función | 9 |
| 80 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Compilación | 10 | Error en el apartado de la función de sql | 9 |
| 80 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Compilación | 5 | Error en el apartado de la función de sql | 9 |
| 80 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Compilación | 10 | Error en el apartado de la función de sql | 9 |
| 80 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Compilación | 3 | Error en la función de insert\_alumnos en "sqlite3\_bind\_text" | 9 |
| 80 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Compilación | 15 | Error en la función print\_archivo | 9 |
| 20 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Compilación | 2 | Error de sintaxis línea de código 156 | 9 |
| 50 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Pruebas | 10 | Error en el formato de impresión de menú | 9 |
| 50 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Pruebas | 5 | Error al momento de imprimir el .txt de planes |  |
| 50 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Pruebas | 3 | Error de falta de acentos en las impresiones |  |
| 50 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Pruebas | 3 | Error de falta de acentos en las impresiones |  |
| 50 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Pruebas | 3 | Error de falta de acentos en las impresiones |  |
| 50 | Csoft-Codificación\_Alta-Alumnos-160317 | c1-semana5-Codifcación | c1-semana5-Pruebas | 3 | Error de falta de acentos en las impresiones |  |

# IV.- Diseño.

# 

**Table C70 State Specification Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Jesus Alberto Goiz Barrales | Date | 23/06/2016 |
| Program | Menú Alumnos | Program # | Alta  Alumnos |
| Instructor | PEBM | Language | C |
| Object |  | Routine |  |

|  |  |
| --- | --- |
| **State name** | **Description** |
| Inicio | Inicia el programa. |
| Int Main | Se utiliza para llamar a nuestra función de menú- alumnos junto con toda su respectiva acción. |
| Alta\_Alumnos | Los datos necesarios para agregar un nuevo alumno. |
| Insert\_Alumnos | Función para almacenar datos del alumno. |
| Query\_Alumnos | Muestra los planes que se encuentran en la base de datos. |
| Baja\_Alumnos | Función para buscar y eliminar un alumno |
| Modificación\_Alumnos | Función para buscar y modificar los campos de un alumno. |
| Fuction/Parameter | Descripción de la Función |
| printf() | Imprimir los datos en pantalla. |
| gets() | Leer datos |
| opción | Indica el tipo de acciones que elige el usuario. |
| char nombres\_alumnos[40]; char apellido\_materno[30] char apellido\_paterno[40]; int fecha\_alumnos; char direccion\_alumnos[60]; char correo\_alumnos[30];  int telefono\_alumnos;  char IdAlumnos[15];  Char opcionBaja:  Char opcionModifcación:  Char opcionAlta. | Variables globales. |

# 

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **States/Next States** | | | | **Transition Condition** | | **Action** | |
| Inicio | | | |  | |  | |
|  | Maín | | | Ninguna | | Se muestra el menú en pantalla. | |
| Alta\_Alumnos | | | |  | |  | |
|  |  | | | Ninguna. | | Se solicitan todos los datos del alumno.  char nombres\_alumnos[40]; char apellido\_materno[30] char apellido\_paterno[40]; int fecha\_alumnos; char direccion\_alumnos[60]; char correo\_alumnos[30];  int telefono\_alumnos; | |
|  | Baja\_Alumno | | | Se necesita haber dado de alta algún Alumno | | Con la variable IdAlumno se buscará en la base de datos para poder eliminarlo. | |
|  |  | | |  | |  | |
|  | Modificación\_Alumno | | | Se necesita tener alumnos en la base de datos. | | Se podrá modificar los valores de las variables del alumno:  char nombres\_alumnos[40]; char apellido\_materno[30] char apellido\_paterno[40]; int fecha\_alumnos; char direccion\_alumnos[60]; char correo\_alumnos[30];  int telefono\_alumnos; | |

**Table C66 Operational Scenario Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Jesús Alberto Goiz Barrales | Date | 06/03/2017 |
| Program | Menú Alumnos | Program # | Alta Alumnos |
| Instructor | PEBM | Language | C |

Construct operational scenarios to cover the normal and abnormal program uses, including user errors.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario** Number:1 | | | Opción alta de alumnos. Alta\_Alumnos. | |
| **Scenario Objective:** Se ilustra el funcionamiento del programa en general | | | | |
| **Source** | **Step** | **Action** | | **Comments** |
| User | 1 | Se ejecuta el .exe como administrador | |  |
| System | 2 | Se visualiza el menú de  Alta\_Alumnos con los datos que se consideran en este menú. | | * Nombres. * Apellido paterno. * .Apellido materno. * Fecha de nacimiento. * ID * Direccion. * Correo electrónico. * Teléfono. * Semestre del alumno.   . |
| User | 3 | El usuario ingresa todos los datos del alumno debido a que es necesario llenar todos los campo | | Cada dato tendrá sus propias restricciones |
| System | 4 | Cuando se termine de capturar los datos se mencionara si se muestra la lista de planes actuales que tiene la base de datos . | |  |
| User | 5 | Se necesita imprimir los planes actuales que se tiene en la base de datos | |  |
| System | 6 | Solicita la asignación de un plan al alumno de los que se encuentran en la lista. | |  |
| User | 7 | Ingresa los datos del plan seleccionado | |  |
| System | 7 | Se imprimen los datos del alumno y se muestra un mensaje de que si los campos son correctos. | |  |
| User | 8 | Los campos son correctos : Se mandan a guardar en la base de datos.  Los campos son incorrectos: Se va al menu de alta de alumnos. | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario** Number:2 | | | Opción baja de alumnos. Baja\_Alumnos. | |
| **Scenario Objective:** Se ilustra el funcionamiento del programa en general | | | | |
| **Source** | **Step** | **Action** | | **Comments** |
| User | 1 | Se ejecuta el .exe como administrador | |  |
| System | 2 | Se visualiza el menú de  Alta\_Alumnos con los datos que se consideran en este menú, considerando seleccionar la baja de un Alumnos, y se manda a llamar la función Baja Alumnos. | | Falta contemplar los datos de Kardex y de plan de alumnos. |
| User | 3 | El usuario ingresa el ID del alumno a dar de baja | | Este sera un dato int. |
| System | 4 | Buscará el ID en la base de datos para mostrar la información del alumno a borrar, se mostrará una confirmación con un Si (s) o un No (n) para eliminar permanentemente al alumno y se dara un mensaje de confirmación de esta actividad de borrado. | |  |
| User | 5 | Fin de la actividad. | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario** Number:3 | | | Opción Modificación de alumnos. Modificación\_Alumnos. | |
| **Scenario Objective:** Se ilustra el funcionamiento del programa en general | | | | |
| **Source** | **Step** | **Action** | | **Comments** |
| User | 1 | Se ejecuta el .exe como administrador | |  |
| System | 2 | Se visualiza el menú de  Alta\_Alumnos con los datos que se consideran en este menú, considerando seleccionar la modificación de un alumnos, y se manda a llamar la función ;Modifcacón\_Alumno. | | Falta contemplar los datos de Kardex y de plan de alumnos. |
| User | 3 | El usuario ingresa el ID del alumno a dar de baja | | Este sera un dato int. |
| System | 4 | Buscará el ID en la base de datos para mostrar la información del alumno a modificar, se mostrará los campos que se podrán editar del alumnos y así ir haciendo los cambios necesarios.  Al finalizar se imprimirán los datos en pantalla y se mostrará un mensaje de confirmación con un Si (s) o un No (n) para guardar los datos introducidos. | |  |
| User | 5 | Fin de la actividad. | |  |

**Table C72 Logic Specification Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Jesús Alberto Goiz Barrales | Date | 08/03/2017 |
| Program | Menu Alumnos | Program # | Menu Alumnos |
| Instructor | PEBM | Language | C |
| Object | Alta\_Alumnos | Function | Alta\_Alumnos |

INCLUDES:

#include <iostream>

#include <cstdio>

#include <cstdlib>

#include "sqlite3.h"

char: nombres\_alumnos[40];

char: apellido\_materno[30]:

char: apellido\_paterno[40];

int: fecha\_alumnos;

char: dirección\_alumnos[60];

char: correo\_alumnos[30];

int teléfono;

TYPE DEFINITIONS: Function void

|  |  |
| --- | --- |
| **Declaration:** |  |
| **Reference:** |  |
|  |  |
| **Logic reference numbers** | **Program logic, in pseudocode** |
|  |  |
|  | printf("o---------------------------------------------------------o\n") |
|  | printf("| ================ Alta de Alumnos ==================== |\n") |
|  | printfi("o---------------------------------------------------------o\n") |
|  |  |
|  | printf(" o---------------------------------------------------------o\n")  printf("| Para dar de alto un alumno necesitas | \n\n")  printf(" | 1)Nombres 2)Apellido Paterno | \n \n")  printf(" | 3)Apellido Materno 4)Fecha de nacimiento. | \n\n")  printf("| 5)Dirección 6)Correo Electrónico. | \n\n")  printf("| 7)Teléfono | \n\n")  printf("| Presione una tecla para agregar los datos: | | \n\n")  printf("o---------------------------------------------------------o \n\n\n") |
|  | Pause |
|  | cls |
|  | printf("1) Introduce los nombres del alumno\n");  Pause  printf("2) Introduce el apellido paterno del alumno\n");  Pause  printf("3) Introduce el apellido materno del alumno\n");  Pause  printf("4) Introduce la fecha de nacimiento del alumno\n");  Pause  printf("5) Introduce la dirección del alumno\n");  Pause  printf("Introduce el correo electrónico del alumno\n");  Pause  printf("Introduce el teléfono del alumno\n");  Pause |
|  |  |
|  | printf("Los datos del alumno son:")  printf("1) Nombres")  Pause  printf("2 Apellido paterno del alumno\n")  Pause  printf("3) Apellido materno del alumno\n");  Pause  printf("4) Fecha de nacimiento del alumno\n");  Pause  printf("5) Dirección\n");  Pause  printf("6) Correo electrónico del alumno\n")  Pause  printf("7) Teléfono del alumno\n") |
|  | printf(“Los datos son correctos s/n)  Si los datos son correctos se almacenan los datos.  Si no:  Se vuelve a solicitar los datos del alumno. |
|  | Fin. |

**Table C72 Logic Specification Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Jesús Alberto Goiz Barrales | Date | 08/03/2017 |
| Program | Menú Alumnos | Program # | Menú Alumnos |
| Instructor | PEBM | Language | C |
| Object | Baja\_Alumnos | Function | Baja\_Alumnos |

INCLUDES:

#include <cstdlib>

#include <iostream>

#include <stdlib>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

Int idAlumno;

TYPE DEFINITIONS: Function void

|  |  |
| --- | --- |
| **Declaration:** |  |
| **Reference:** |  |
|  |  |
| **Logic reference numbers** | **Program logic, in pseudocode** |
|  |  |
|  | printf("o---------------------------------------------------------o\n") |
|  | printf("| ================ Alta de Alumnos ==================== |\n") |
|  | printfi("o---------------------------------------------------------o\n") |
|  |  |
|  | printf(" o---------------------------------------------------------o\n")  printf("| Para dar de eliminar un alumno necesitas | \n\n")  printf(" | 1)ID | \n \n")  printf("| Presione una tecla para agregar los datos: | | \n\n")  printf("o---------------------------------------------------------o \n\n\n") |
|  | Pause |
|  | cls |
|  | Se introduce una matricula. |
|  | Se genera una búsqueda |
|  | Se imprime los datos del alumno encontrado |
|  | Se mostrará el mensaje de “Deseas eliminar a este alumno (s/n)  Si= Los datos se eliminarán.  No= Los datos se conservan. |
|  | Fin. |

**Table C72 Logic Specification Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Jesús Alberto Goiz Barrales | Date | 08/03/2017 |
| Program | Menú Alumnos | Program # | Menú Alumnos |
| Instructor | PEBM | Language | C |
| Object | Modificación\_Alumnos | Function | Modificación\_Alumnos |

INCLUDES:

#include <cstdlib>

#include <iostream>

#include <stdlib>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

Int idAlumno;

TYPE DEFINITIONS: Function void

|  |  |
| --- | --- |
| **Declaration:** |  |
| **Reference:** |  |
|  |  |
| **Logic reference numbers** | **Program logic, in pseudocode** |
|  |  |
|  | printf("o---------------------------------------------------------o\n") |
|  | printf("| ================ Modificación de Alumnos ==================== |\n") |
|  | printfi("o---------------------------------------------------------o\n") |
|  |  |
|  | printf(" o---------------------------------------------------------o\n")  printf("| Para dar de modificar un alumno necesitas | \n\n")  printf(" | 1)ID | \n \n")  printf("| Presione una tecla para agregar los datos: | | \n\n")  printf("o---------------------------------------------------------o \n\n\n") |
|  | Pause |
|  | cls |
|  | Se introduce una matricula. |
|  | Se genera una búsqueda |
|  | Se imprime los datos del alumno encontrado y se muestra las opciones a editar conforme al menú de Alta\_Alumnos |
|  | Se editaran los campos y se imprimirán los nuevos en pantalla para mostrar los cambios realizar.  Se tendrá un mensaje de confirmación para guardar los datos en la base de datos.  “Deseas guardar los cambios”.  Si= Los datos se guardarán.  No= Los datos no se actualizan. |
|  | Fin. |

**Table C68 Functional Specification Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Jesús Alberto Goiz Barrales. | Date | 06/03/2017 |
| Program | Alta Alumnos | Program # | Alta-Alumnos |
| Instructor | PEBM. | Language | C |

|  |  |  |  |
| --- | --- | --- | --- |
| Object/Class Name  Main | | Parent Classes  Ninguna | Attributes  Variables globales que no se declaran en el main. |
|  | Method Declaration | Method External Specification | |
|  | Void Alta\_Alumnos . | Se tendrá el menú de alumnos de los cuáles se solicitarán los datos siguientes datos:   1. Nombres. 2. Apellido paterno. 3. .Apellido materno. 4. Fecha de nacimiento. 5. ID 6. Direccion. 7. Correo electrónico. 8. Teléfono. 9. Semestre del alumno.   Una vez mostrado los datos que se solicitará, de dará click para continuar y a continuación se hará una actualización de pantalla y se pedirán los datos mencionados en el orden que se especificó al principio.  Después se imprimirán los datos introducidos en pantalla y se dará la opción de continuar o de editar estos datos nuevamente. Si se da la opción de continuar: El alumno se guardará en el programa y se almacenará en la base de datos.  Si se da la opción de editar:  Se pedirán nuevamente los datos e iniciara todo el procedimiento. | |
|  | Baja\_Alumnos | Se necesita el ID de un alumno para poder realizar una búsqueda, una vez localizado se mostrará información y se mandará un mensaje de confirmación para eliminarlo. | |
|  | Modificación\_Alumnos | Se necesita el ID de un alumno para poder localizarlo, una vez localizado se mostrará los cambios que tiene el Alta\_Alumnos y se podrán modificar estos mismo.  Al finalizar se tendrá un mensaje de confirmación y la impresión de los datos para corroborar que sean correctos. | |
|  | Int Main. | Se mandará a llamar la función  Alta\_Alumnos()  Baja\_Alumnos()  Modificación\_Alumnos() | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |

# V.- Design CheckList.

**PSP2 Design Review Checklist**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Jesus Alberto Goiz Barrales | Date | 08/03/2017 |
| Program | Alta-Alumno | Program # | Alta-Alumno |
| 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. |  | Revisado, Se Hicieron correcciones ( JAGB). |
| External Limits | Where the design assumes or relies upon external limits, determine if behavior is correct at nominal values, at limits, and beyond limits. |  | Revisado |
| 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. |  | Revisado |
| Internal Limits | Where the design assumes or relies upon internal limits, determine if behavior is correct at nominal values, at limits, and beyond limits. |  | Revisado |
| Special Cases | * Check all special cases. * Ensure proper operation with empty, full, minimum, maximum, negative, and cero 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. |  |  |
| Functional Use | * Verify that all functions, procedures, or methods are fully understood and properly used. * Verify that all externally referenced abstractions are precisely defined. |  |  |
| 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. |  |  |
| 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 |  |  |
| Standards | Ensure that the design conforms to all applicable design standards. |  |  |

# VI.- Programa Fuente.

/\*=========================================================\*/  
/\*Autor: Jesús Alberto Goiz Barrales \*/  
/\*Componente: AltaAlumnos \*/  
/\*Fecha:16/03/17 \*/  
/\*Descripción:Función para dar de alto un alumno \*/  
/\*Versión: 0.3 \*/  
/\*=========================================================\*/  
  
/\*==========================================================\*/  
/\* El programa contiene lo siguiente: \*/  
/\*static int callback (void \*data,int argc, char \*\*argv,char \*\*az\_col\_name)  
/\*void query\_alumnos(sqlite3 \*db,char\* id\_alumno) \*/  
/\*void insert\_alumnos(sqlite3 \*db,char\* id\_alumno, char\* nombre, char\* ap\_paterno, char\* ap\_materno, char\* fecha\_nac, char\* direccion, char\* telefono, char\* email,char\* semestre,char\* id\_plan)\*/  
/\*static int print\_archivo(void \*data,int argc, char \*\*argv,char \*az\_col\_name)\*/  
/\*void actualizar\_planes(sqlite3 \*db)\*/  
/\*void alta\_alumnos(sqlite3 \*db)\*/  
/\*int main()\*/  
/\*==========================================================\*/  
/\*Funcion que tendra el menu de alumnos para poder ingresar  
los datos de este mismo de una manera ordenada y  
almacenarlos en la base de datos con el respctivo Id de  
cada plan que se tenga previamente registrado \*/  
  
#include <iostream>  
#include <cstdio>  
#include <cstdlib>  
#include "sqlite3.h" /\*Interaccion base de datos\*/  
  
using namespace std;  
FILE \*pfile;  
  
/\*\*\*Esta función se encarga de imprimir los datos encontrados en la base de datos\*\*\*/  
static int callback (void \*data,int argc, char \*\*argv,char \*\*az\_col\_name)  
{  
 int i;  
 for(i=0;i<argc;i++){  
 fprintf(pfile,"%s\t",argv[i]); /\* Imprime en el archivo txt los planes que estan disponibles en la bd\*/  
 }  
 fprintf(pfile,"\n");  
 return 0;  
}  
  
/\*\*\*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=@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,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 instrucción sql, valores del maestro\*/  
  
 printf("\t\t Nombre: %s %s %s\n", sqlite3\_column\_text(res, 1),sqlite3\_column\_text(res, 2),sqlite3\_column\_text(res, 3));  
 printf("\t\t Fecha: %s\n", sqlite3\_column\_text(res, 4));  
 printf("\t\t Direccion: %s\n", sqlite3\_column\_text(res, 5));  
 printf("\t\t Telefono: %s\n", sqlite3\_column\_text(res, 6));  
 printf("\t\t Email: %s\n", sqlite3\_column\_text(res, 7));  
 printf("\t\t ID Alumno: %s\n", sqlite3\_column\_text(res, 0));  
 printf("\t\t Plan: %s\n", sqlite3\_column\_text(res, 8));  
 printf("\t\t Semestre:\%s\n", sqlite3\_column\_text(res, 9));  
  
 }  
 sqlite3\_finalize(res); /\* Finaliza el objeto creado\*/  
  
}  
  
  
void insert\_alumnos(sqlite3 \*db,char\* id\_alumno, char\* nombre, char\* ap\_paterno, char\* ap\_materno, char\* fecha\_nac, char\* direccion, char\* telefono, char\* email,char\* semestre,char\* id\_plan){  
 char \*sql;  
 char \*err=0;  
 int rc;  
 sqlite3\_stmt \*res; /\* Inserta todos los valores del alumno sentncia sql\*/  
 sql="INSERT INTO ALUMNOS (Id\_Al, Nombre, ApPaterno, ApMaterno, FNacimiento, Direccion, Telefono, Email,ID\_Plan,Semestre) VALUES (@Id\_A,@nom,@ap\_p,@ap\_m,@fec,@dir,@tel,@ema,@plan,@sem);";  
 rc=sqlite3\_prepare\_v2(db,sql,-1,&res,0); /\* Prepara la sentencia sql\*/  
 if(rc!=SQLITE\_OK){  
 printf("SQL error %s\n",err); /\* Imprime el mensaje de error\*/  
 }  
 else{  
 printf("\t\t Valores del alumno guardados correctamente.\n");/\* Cambia los identificador por los valores\*/  
 int idm = sqlite3\_bind\_parameter\_index(res, "@Id\_A"); /\* Busca el lugar del identificador\*/  
 int nom = sqlite3\_bind\_parameter\_index(res, "@nom");  
 int ap\_p = sqlite3\_bind\_parameter\_index(res, "@ap\_p");  
 int ap\_m = sqlite3\_bind\_parameter\_index(res, "@ap\_m");  
 int fec = sqlite3\_bind\_parameter\_index(res, "@fec");  
 int dir = sqlite3\_bind\_parameter\_index(res, "@dir");  
 int tel = sqlite3\_bind\_parameter\_index(res, "@tel");  
 int ema = sqlite3\_bind\_parameter\_index(res, "@ema");  
 int plan= sqlite3\_bind\_parameter\_index(res, "@plan");  
 int sem = sqlite3\_bind\_parameter\_index(res, "@sem");  
  
 sqlite3\_bind\_text(res, idm, id\_alumno,40,SQLITE\_STATIC); /\* cambia el valor por elidenitficador \*/  
 sqlite3\_bind\_text(res, nom, nombre,40,SQLITE\_STATIC);  
 sqlite3\_bind\_text(res, ap\_p, ap\_paterno,40,SQLITE\_STATIC);  
 sqlite3\_bind\_text(res, ap\_m, ap\_materno,40,SQLITE\_STATIC);  
 sqlite3\_bind\_text(res, fec, fecha\_nac,40,SQLITE\_STATIC);  
 sqlite3\_bind\_text(res, dir, direccion,40,SQLITE\_STATIC);  
 sqlite3\_bind\_text(res, tel, telefono,40,SQLITE\_STATIC);  
 sqlite3\_bind\_text(res, ema, email,40,SQLITE\_STATIC);  
 sqlite3\_bind\_text(res, plan, id\_plan,40,SQLITE\_STATIC);  
 sqlite3\_bind\_text(res, sem, semestre,40,SQLITE\_STATIC);  
  
 }  
 int step = sqlite3\_step(res);/\* \*/  
 if (step == SQLITE\_ROW) {  
 printf("\t\t Datos correctos %s\n", sqlite3\_column\_text(res, 0)); /\* Imprime el resultado de la consulta\*/  
 }  
}  
  
static int print\_archivo(void \*data,int argc, char \*\*argv,char \*az\_col\_name)  
{  
 int i;  
 for(i=0;i<argc;i++){  
 fprintf(pfile,"%s\t",argv[i]); /\* Imprime en el archivo txt con los planes disponibles en la bd\*/  
 }  
 fprintf(pfile,"\n");  
 return 0;  
}  
  
/\*\*\*Esta función se encarga de actualizar el doucmento Planes\_Disponibles.txt\*\*/  
  
void actualizar\_planes(sqlite3 \*db){  
 char \*sql;  
 char \*err=0;  
 int rc;  
 const char\*data;  
 pfile=fopen("Planes\_Disponibles.txt","w");  
 fprintf(pfile,"----- PLANES DISPONIBLES PARA ALUMNOS -------\nID\tNombre\t\tNumeroSemestres\tFecha\n");  
 sql="SELECT \* from PLANES";  
 rc=sqlite3\_exec(db,sql,callback,(void\*)data,&err);  
 if(rc!=SQLITE\_OK){  
 printf("SQL error %s\n",err);  
 }  
 else{  
 printf("\t\t Nueva lista disponible\n");  
 }  
 fclose(pfile);  
}  
  
/\*\*\*\*Esta función realiza un insert en la tabla Alumnos en la base de datos\*\*\*\*/  
  
void alta\_alumnos(sqlite3 \*db)  
{  
 char nombre[40];  
 char ap\_materno[40];  
 char ap\_paterno[40];  
 char f\_nacimiento[40];  
 char direccion[40];  
 char telefono[40];  
 char correo[40];  
 char id\_alumno[40];  
 char semes\_alumno[40];  
  
 char id\_plan[40];  
 int status=2;  
 bool correcto=false;  
while(status != 1){  
 system("cls");  
 printf("\t\to----------------------------------------------------------o\n");  
 printf("\t\t| ================ Alta de Alumnos =================== |\n");  
 printf("\t\to----------------------------------------------------------o\n");  
 printf("\t\to----------------------------------------------------------o\n");  
 printf("\t\t| Para dar de alto un alumno necesitas |\n");  
 printf("\t\t| 1)Nombres 2)Apellido Paterno |\n");  
 printf("\t\t| 3)Apellido Materno 4)Fecha de nacimiento. |\n");  
 printf("\t\t| 5)Direcci%cn 6)Correo Electr%cnico. |\n",162,162);  
 printf("\t\t| 7)Tel%cfono 8)ID-Curp |\n",130);  
 printf("\t\t| 9)Semestre |\n");  
 printf("\t\t| Presiona una tecla para agregar los datos: |\n");  
 printf("\t\to----------------------------------------------------------o\n");  
 fflush( stdin );  
 printf("\t\t 1) ");  
 scanf("\t\t%[^\n]",nombre);  
 fflush( stdin );  
 printf("\t\t 2) ");  
 scanf("%s",ap\_paterno);  
 fflush( stdin );  
 printf("\t\t 3) ");  
 fflush( stdin );  
 scanf("%s",ap\_materno);  
 correcto=true;  
  
 do{  
 printf("\t\t 4) ");  
 scanf("%s",&f\_nacimiento);  
  
 if(f\_nacimiento[6]>='0' && f\_nacimiento[6]<='9' && f\_nacimiento[7]>='0' && f\_nacimiento[7]<='9'){ /\*año validacion numeros\*/  
 if(f\_nacimiento[3]>='0' && f\_nacimiento[3]<'2' && f\_nacimiento[4]>='0' && f\_nacimiento[4]<='9'){ /\*mes validacion numeros\*/  
 if(f\_nacimiento[3]=='1'){  
 if(!(f\_nacimiento[4]>='0' && f\_nacimiento[4]<='2')){ /\*mes validacion 12\*/  
 printf("\t Mes Incorrecto\n");  
 }  
 }  
  
 if(f\_nacimiento[0]>='0' && f\_nacimiento[0]<'4' && f\_nacimiento[1]>='0' && f\_nacimiento[1]<='9')  
 { /\*dia validacion numero\*/  
 if(f\_nacimiento[0]=='3'){  
 if(!(f\_nacimiento[1]>='0' && f\_nacimiento[1]<='1')){ /\*dia validacion 31\*/  
 printf("\t D%ca Incorrecto\n",161); /\*dia incorrecto\*/  
 continue;  
 }  
 }  
 correcto=false; /\*validado correctamente\*/  
 }  
 else{  
 printf("\t D%ca Incorrecto\n",161); /\*dia incorrecto\*/  
 }  
 }  
 else{  
 printf("\t Mes Incorrecto\n"); /\*mes incorrecto numero \*/  
 }  
 }  
 else{  
 printf("\t A%co Incorrecto\n",164); /\*año incorrecto numero\*/  
 }  
 }while(correcto);  
  
  
 printf("\t\t 5) ");  
 fflush( stdin );  
 scanf("%[^\n]",direccion);  
  
 do{ /\* validacion del correo\*/  
 printf("\t\t 6) ");  
 fflush( stdin );  
 scanf("%s",&correo);  
 for(int i=0;i<40;i++)  
 {  
 if(correo[i]== '@')  
 { /\* si tiene @ es correcto\*/  
 correcto=false;  
 }  
 }  
 if(correcto){  
 printf("\t Correo Incorrecto\n"); /\* no tiene @ es incorrecto\*/  
 }  
 }while(correcto);  
  
 printf("\t\t 7) ");  
 fflush( stdin );  
 scanf("%[0-9]",&telefono);  
 correcto=true;  
  
 printf("\t\t 8) ");  
 fflush( stdin );  
 scanf("%s",&id\_alumno);  
  
 printf("\t\t 9) ");  
 fflush( stdin );  
 scanf("%s",&semes\_alumno);  
  
 system("cls");  
  
 printf("\t\to---------------------------------------------------------o\n");  
 printf("\t\t| ================ Alta de Alumnos ====================|\n");  
 printf("\t\to---------------------------------------------------------o\n");  
 printf("\t\to---------------------------------------------------------o\n");  
 printf("\t\t| | \n");  
 printf("\t\t| Son correctos los datos? 1)si 2)no: | \n");  
 printf("\t\to---------------------------------------------------------o\n");  
 printf("\t\t Nombre: %s %s %s\n",&nombre,&ap\_paterno,&ap\_materno);  
 printf("\t\t Nacimiento: %s\n",&f\_nacimiento);  
 printf("\t\t Direcci%cn: %s\n",162,&direccion);  
 printf("\t\t Tel%cfono: %s\n",130,&telefono);  
 printf("\t\t Email: %s\n",&correo);  
 printf("\t\t CURP: %s\n",&id\_alumno);  
 printf("\t\t Semestre %s\n",&semes\_alumno);  
 printf("\n\t\t Opci%cn .-",162);  
 cin>>status;  
 }  
  
 system("cls");  
  
 printf("\t\to---------------------------------------------------------o\n");  
 printf("\t\t| ================ Alta de Alumnos =================== |\n");  
 printf("\t\to---------------------------------------------------------o\n");  
 printf("\t\to---------------------------------------------------------o\n");  
 printf("\t\t| Para dar de alta al alumno es |\n");  
 printf("\t\t| es necesario conocer el ID del plan . |\n");  
 printf("\t\t| La lista est%c en la carpeta del programa. |\n",160);  
 printf("\t\t| Desea actualizar la lista? 1)si 2)no: |\n");  
 printf("\t\to---------------------------------------------------------o\n");  
 printf("\n\t\t Opci%cn .-",162);  
 cin>>status;  
 if(status == 1){  
 actualizar\_planes(db); /\*Se actualiza el txt con las materias de la db\*/  
 }  
 int cant\_planes=1;  
 while(cant\_planes--){  
 printf("\n\t\t Ingrese el No. del plan ID .- "); /\*id del plan\*/  
 scanf("%s",&id\_plan);  
 printf("\t\t Es correcto el plan asingnado 1)si 2)no\n\t\t "); /\*definicion si es correcto el id del plan o no\*/ cin>>status;  
 }  
  
 insert\_alumnos(db,id\_alumno,nombre,ap\_paterno,ap\_materno,f\_nacimiento,direccion,telefono,correo,semes\_alumno,id\_plan);/\* Mandamos a guardar datos a la bd\*/  
 system("cls");  
 printf("\t\to---------------------------------------------------------o\n");  
 printf("\t\t| ================ Alta de Alumnos ====================|\n");  
 printf("\t\to---------------------------------------------------------o\n");  
 printf("\t\to---------------------------------------------------------o\n");  
 printf("\t\t| Los datos del alumno son: |\n\n");  
 printf("\t\to---------------------------------------------------------o\n");  
 query\_alumnos(db,id\_alumno); /\* selecciona los datos del Alumno y los imprime de la db\*/  
 printf("\t\t Ingresa 1 para continuar\n");  
 printf("\n\t\t Opci%cn .-",162);  
 scanf("%d",&status);  
 system("cls");  
  
}  
  
  
int main()  
{  
 sqlite3 \*db;  
 int res;  
  
 res=sqlite3\_open("test8.db",&db);  
  
 alta\_alumnos(db);  
  
 return 0;  
}

# VII.- Code CheckList.

**Table C58 C++ Code Review Checklist**

PROGRAM NAME: Alta-Alumnos

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose | To guide you in conducting an effective code review |  |  |
| General | • As you complete each review step, check that item in the box to the right.  • Complete the checklist for one program unit before you start to review the next. |  |  |
| Complete | Verify that the code covers all the design. |  |  |
| Includes | Verify that includes are complete. |  |  |
| Initialization | Check variable and parameter initialization:  • At program initiation  • At start of every loop  • At function/procedure entry |  |  |
| Calls | Check function call formats:  • Pointers  • Parameters  • Use of ‘&’ |  |  |
| Names | Check name spelling and use:  • Is it consistent?  • Is it within declared scope?  • Do all structures and classes use '.' reference? |  |  |
| Strings | Check that all strings are  • identified by pointers  • terminated in NULL. |  |  |
| Pointers | Check that pointers are  • initialized NULL  • only deleted after new, and  • new pointers are always deleted after use. |  |  |
| Output Format | Check the output format:  • Line stepping is proper.  • Spacing is proper. |  | Se hicieron correcciones |
| {} Pairs | Ensure that the {} are proper and matched. |  | Se hicieron correcciones |
| Logic Operators | • Verify the proper use of ==, =, //, and so on.  • Check every logic function for proper (). |  | Se hicieron correcciones |
| Line-by-line Check | Check every LOC for  • instruction syntax and  • proper punctuation. |  | Se hicieron correcciones |
| Standards | Ensure that the code conforms to the coding standards. |  | Sin corrección |
| File Open and Close | Verify that all files are  • properly declared,  • opened, and  • closed. |  |  |

# VIII.- Reporte de resultados.

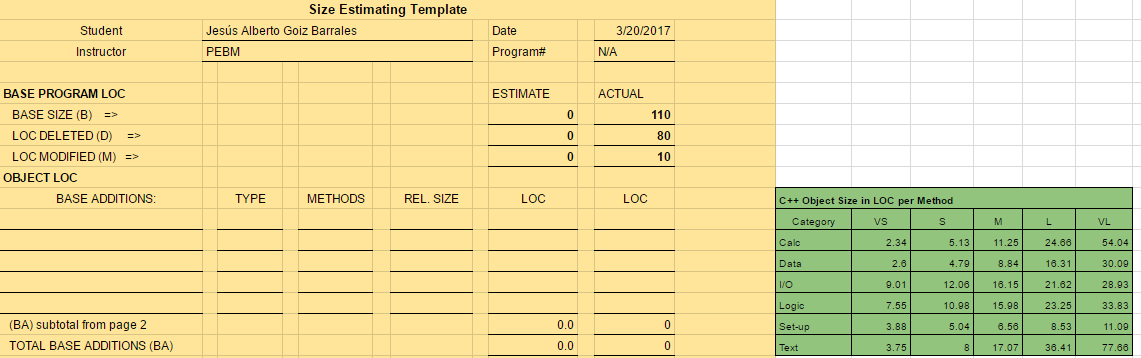
# Template de pruebas

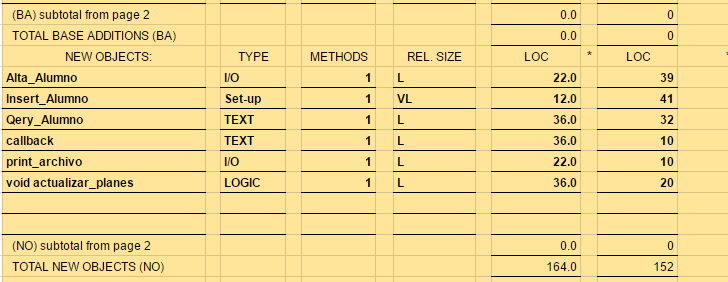
|  |  |
| --- | --- |
| Nombre de la Prueba | Prueba de Alta de Alumnos |
| Caso de Uso | 2.0 |
| Objetivo de la Prueba | Verificar que la carga de datos a la base de datos se ha realizado correctamente. |
| Descripción de la Prueba | Se ingresará a sistema como Administrador y se seleccionará la opción de “Alta Alumnos” |
| Test | 1. Direccionarse a la opción de Alta alumnos. 2. Ingresar Datos del alumno. 3. Se le asigna un plan. 4. Se confirma alta de alumno. 5. Se puede seguir agregando alumnos (regresar al paso 1). |
| Resultados Esperados | Mensaje de Éxito:  Datos Enviados: (Se mostrará los datos almacenados en variables locales)   1. Nombres 2. Apellido paterno 3. Apellido materno 4. Fecha de nacimiento 5. Dirección 6. Correo electrónico 7. Teléfono 8. Curp. 9. ID Plan   Datos Almacenados: (Se hará consulta a DB)   1. Nombres 2. Apellido paterno 3. Apellido materno 4. Fecha de nacimiento 5. Dirección 6. Correo electrónico 7. Teléfono 8. Curp. 9. ID Plan   Mensaje de Error:  Datos Enviados: (Se mostrará los datos almacenados en variables locales)   1. Nombres 2. Apellido paterno 3. Apellido materno 4. Fecha de nacimiento 5. Dirección 6. Correo electrónico 7. Teléfono 8. Curp. 9. ID Plan   “Error de consulta el Alumno no se ha dado de alta” (Debido a que no se cargó correctamente el alumno)  En caso de tener un campo vacío:  “Error ‘Campo Vacío’ Se encuentra vacío Favor de ingresar un valor” |
| Capturas de pantalla |  |
| Estado caso de prueba | Posibles estados:   * Ejecutado * Exitoso\*\*\*\* * Fallido * Frenado * Pendiente de ejecución * En construcción |
| Errores obtenidos | Se corrigió problemas con la tabulación.  Se validó más de dos nombre y dirección con espacios.  Se corrigió la mala impresión de los datos.  Se corrigió la visualización del txt de Planes. |
| Responsable de prueba | JAGB |
| Comentarios |  |

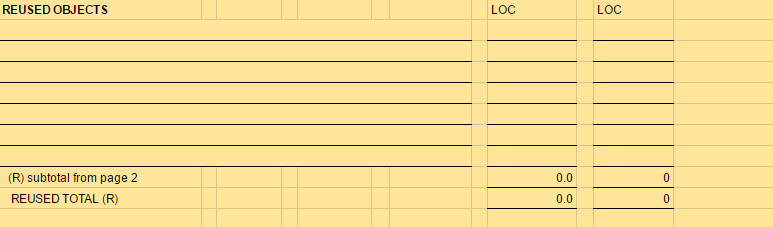
# IX.- PIP

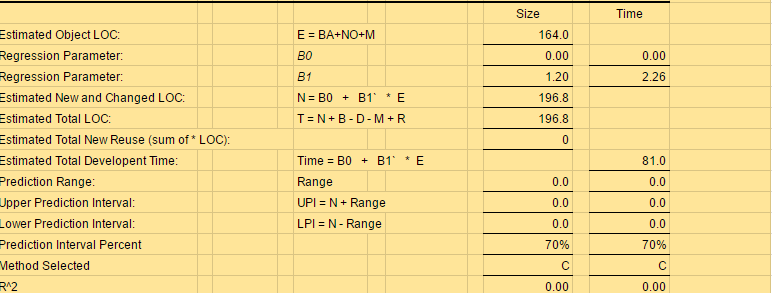
|  |  |  |
| --- | --- | --- |
| Menú-Alumnos | | |
| No. PIP | Descripción del Problema | Descripción de la propuesta. |
| 1 | Mala organición de ideas | Tener mejor panorama al momento de hacer la planeación de nuestro PSP |
| 2 | Confusión de nombres de módulos | Tener una mejor organización en cuestión de análisis del problema. |
| 3 | Verificar el diseño para facilitar la programación y entendimiento del programa | Tomar en cuenta y analizar de manera concreta cada función que se vaya a programar para no dejar pasar errores y se descubran hasta compilación o pruebas |
| 4 | No dejar pausado el módulo debido a que se pierde información | Al momento de hacer el recuento de errores como no se terminó a la par, se dejó pausado el progreso y al momento de hacer el recuento de errores fue difícil de encontrar errores pasados. |

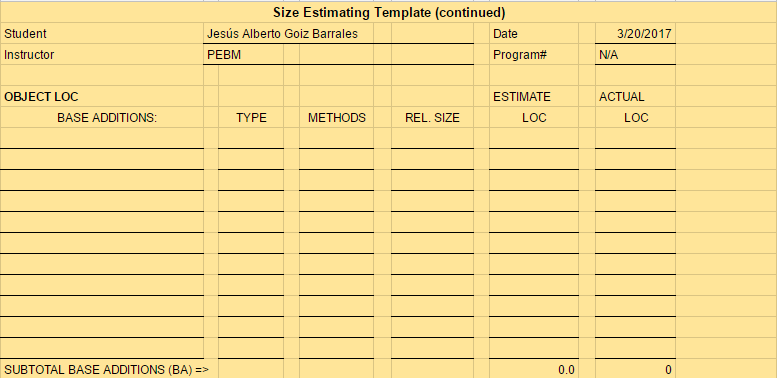
# X.- Size Estimating Template

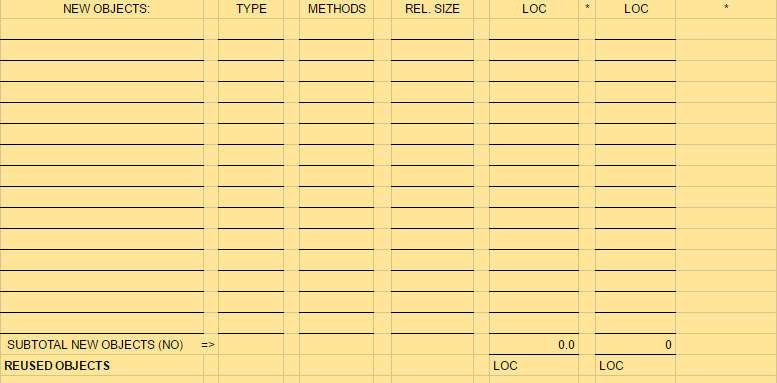


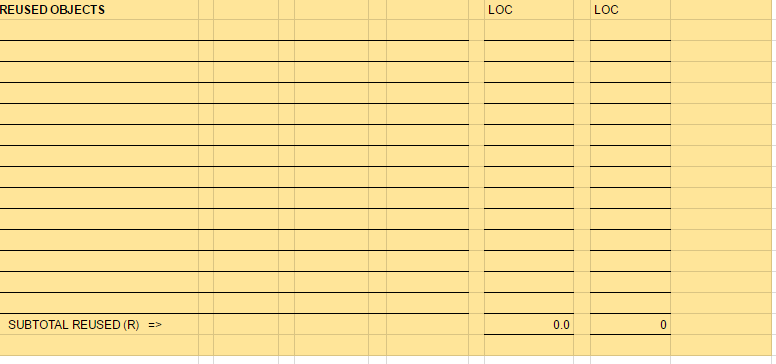












# XI.- Task and Schedule

