**UNIVERSIDAD TECNOLÓGICA DE LA RIVIERA MAYA**

Organismo PúblicoDescentralizado Del Gobierno Del Estado De Quintana Roo

**P R E S E N T A:**

*ESCOBAR HERNÁNDEZ LUIS ALFREDO*

*Cuatrimestre: 5to.   Aula: D-105  Grupo: “TI-51”*

**C A R R E R A:**

*Tecnologías de la información y comunicación*

**M  A  T  E  R  I  A**

*Calidad de software*

**D  O  C  E  N  T E:**

*Ing. Cristian Bernal*

*Playa del Carmen, Quintana Roo a 9 de marzo del 2016*

Contenido

[Información General 3](#_Toc445273655)

[Diagramas 3](#_Toc445273656)

[Gantt 4](#_Toc445273657)

[Estimaciones de costo 5](#_Toc445273658)

[Bitácora de Tiempo 8](#_Toc445273659)

[Tecnologías equipadas al software 9](#_Toc445273660)

[C# 9](#_Toc445273661)

[Sybase 9](#_Toc445273662)

[SQL SERVER 9](#_Toc445273663)

[Estadar de Codificación 10](#_Toc445273664)

[Introducción 10](#_Toc445273665)

[Nombre de los Archivos 11](#_Toc445273666)

[Variables 11](#_Toc445273667)

[Constantes 11](#_Toc445273668)

[Estructuras de Control 12](#_Toc445273669)

[Clases 13](#_Toc445273670)

[Métodos 13](#_Toc445273671)

[Atributos 14](#_Toc445273672)

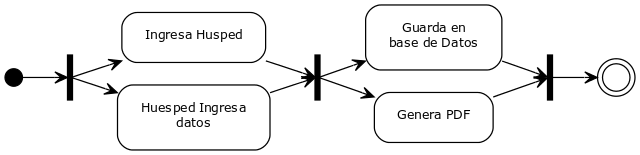
# Información General

*En el marco del curso de “Calidad en el desarrollo de software” se pidió en desarrollo de un software, cuyo propósito fuese hacer registro de los clientes que hacen uso de la salas de spa, dentro de algún hotel.*

*Las principales líneas generales sobre la empresa renova spa está entre sus colaboradores, mucho empeño y muchas ganas de trabajar tanto en cualquier orden jerárquico de estos mismos.*

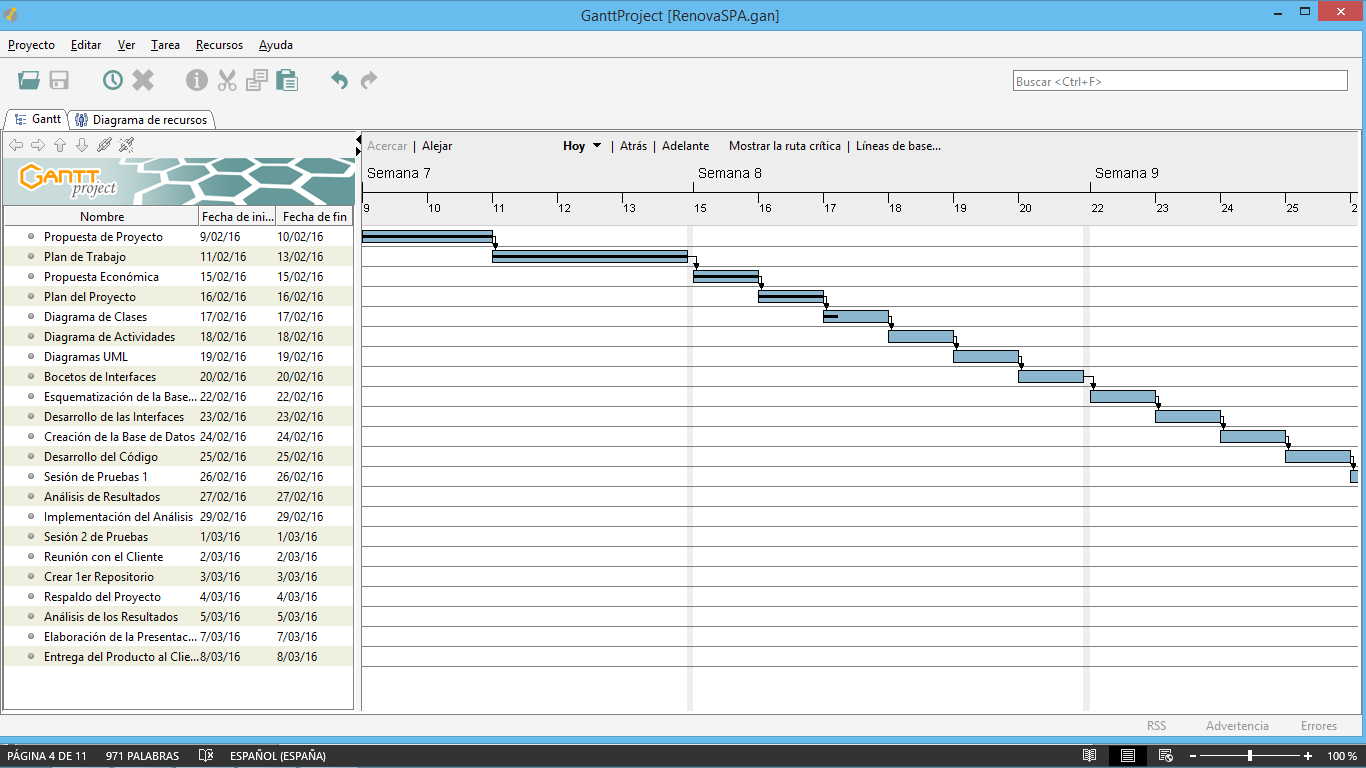
# Diagramas

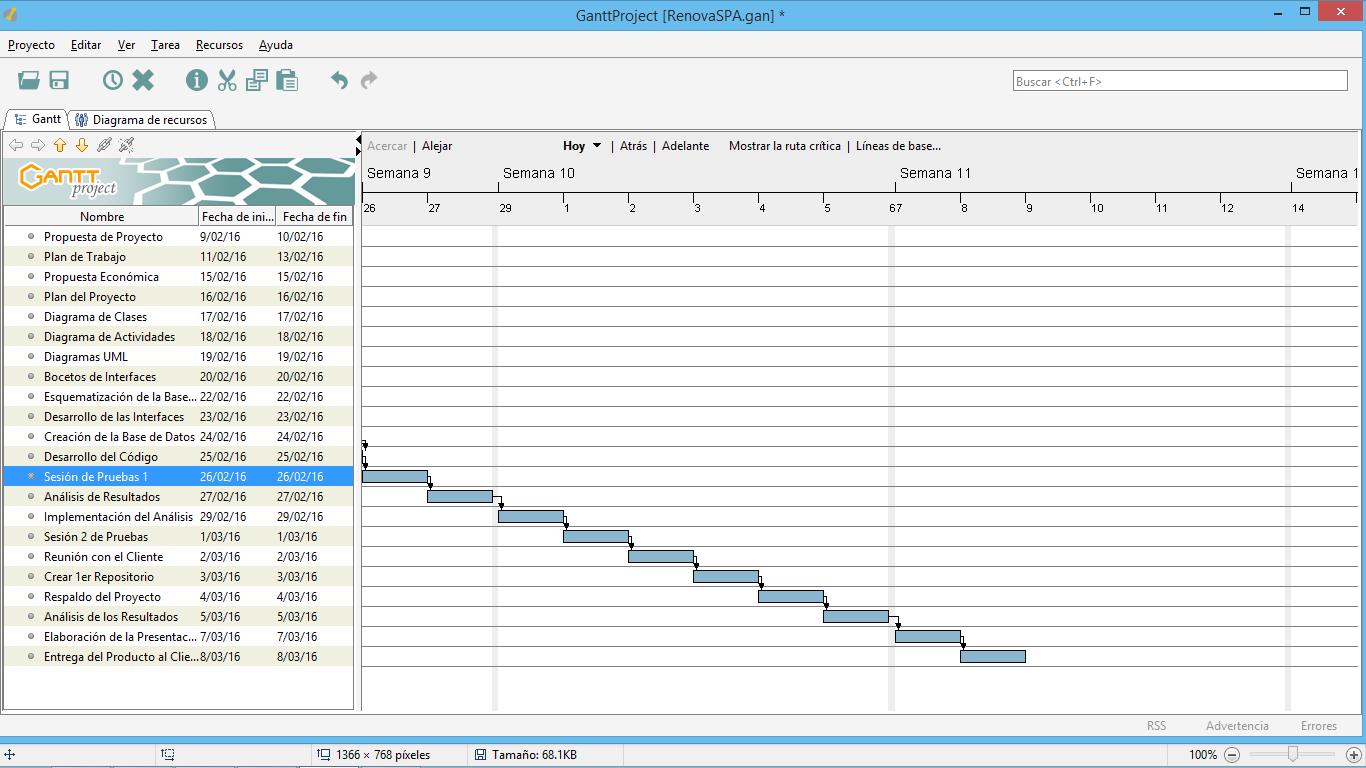




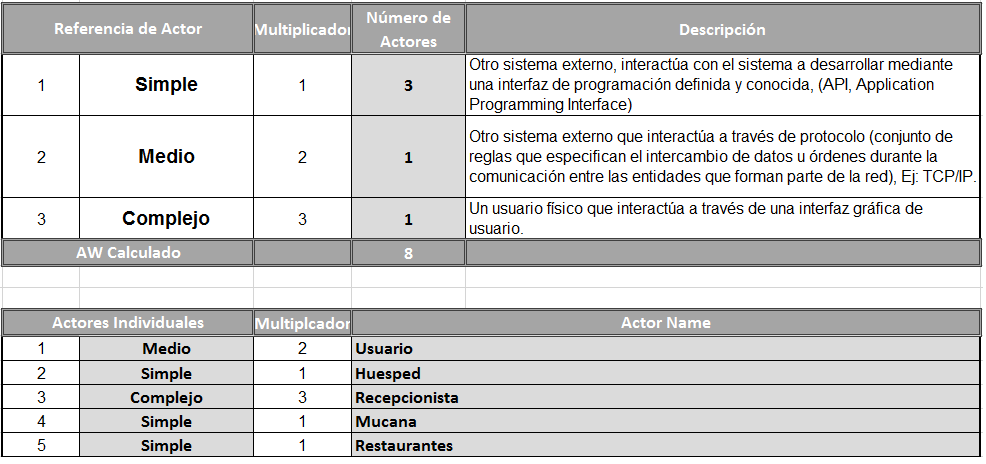


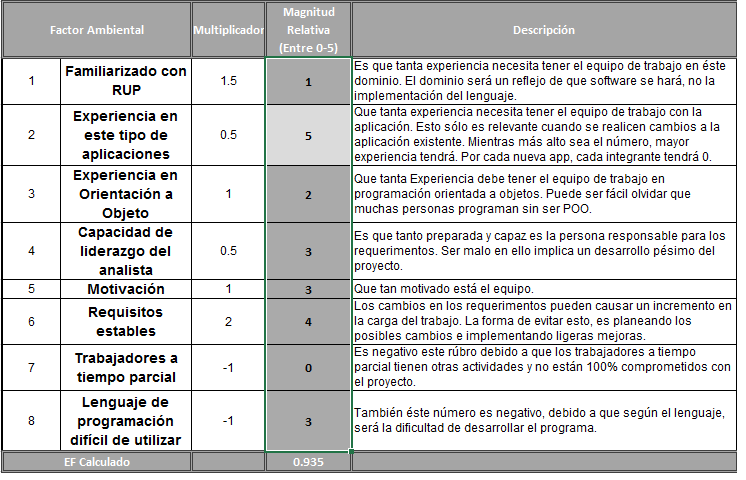
# Gantt



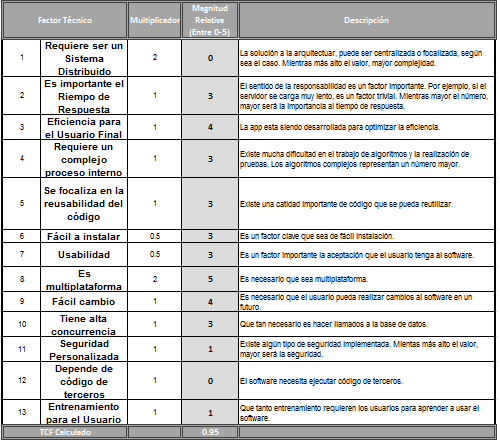


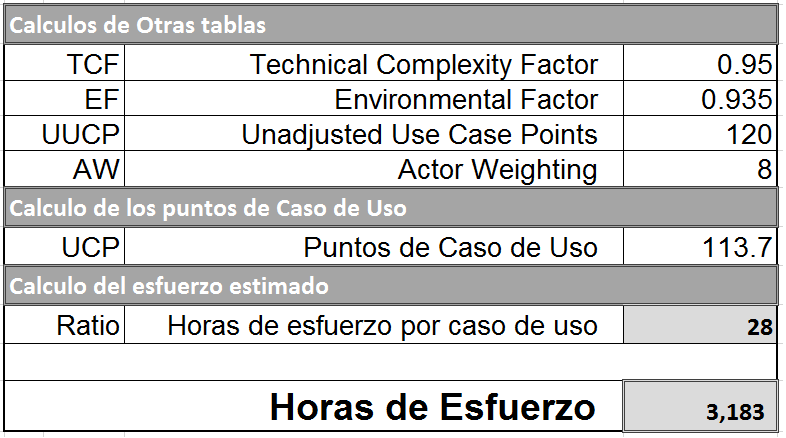
# Estimaciones de costo



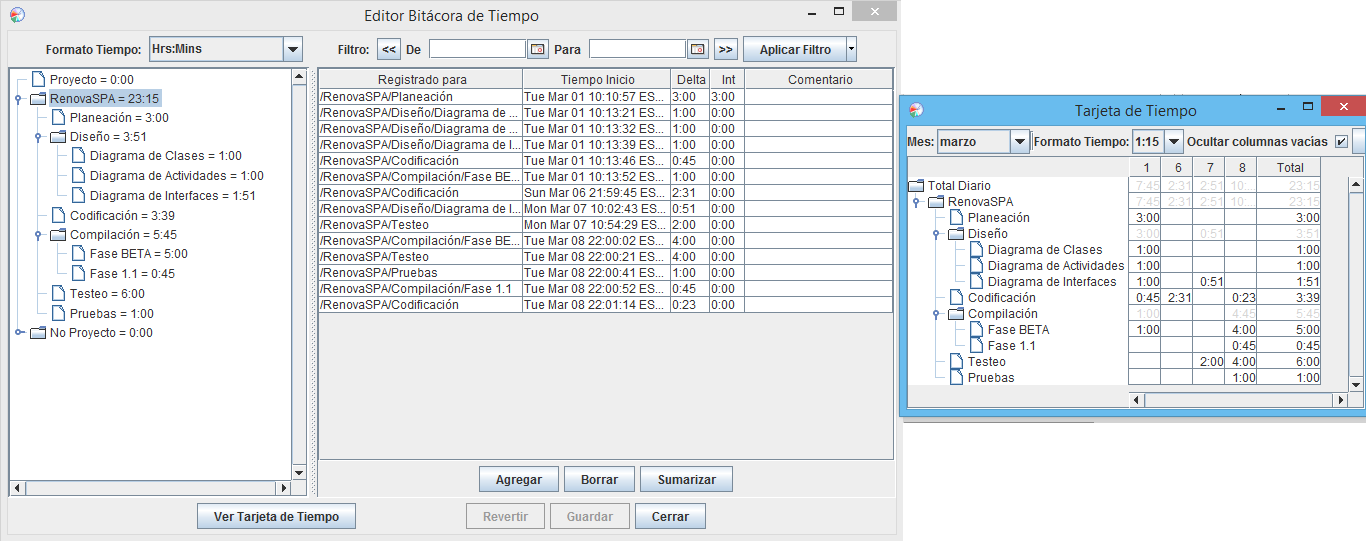








# Bitácora de Tiempo



# Tecnologías equipadas al software

## C#

Es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [orientado a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos) desarrollado y estandarizado por [Microsoft](https://es.wikipedia.org/wiki/Microsoft) como parte de su plataforma [.NET](https://es.wikipedia.org/wiki/Microsoft_.NET), que después fue aprobado como un estándar por la [ECMA](https://es.wikipedia.org/wiki/Ecma_International) (ECMA-334) e [ISO](https://es.wikipedia.org/wiki/Organizaci%C3%B3n_Internacional_de_Normalizaci%C3%B3n)(ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la [infraestructura de lenguaje común](https://es.wikipedia.org/wiki/Infraestructura_de_lenguaje_com%C3%BAn).

Su sintaxis básica deriva de [C](https://es.wikipedia.org/wiki/C_(lenguaje_de_programaci%C3%B3n))/[C++](https://es.wikipedia.org/wiki/C%2B%2B) y utiliza el modelo de objetos de la plataforma .NET, similar al de [Java](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)), aunque incluye mejoras derivadas de otros lenguajes.

## Sybase

Sybase Inc. ([NYSE](https://es.wikipedia.org/wiki/Bolsa_de_Nueva_York): [SY](http://www.nyse.com/quote/XNYS:SY)) fue una compañía dedicada al desarrollo de [tecnología de la información](https://es.wikipedia.org/wiki/Tecnolog%C3%ADa_de_la_informaci%C3%B3n). Fue fundada en [Berkeley](https://es.wikipedia.org/wiki/Berkeley_(California)),[California](https://es.wikipedia.org/wiki/California) ([EE.UU.](https://es.wikipedia.org/wiki/Estados_Unidos)), en [1984](https://es.wikipedia.org/wiki/1984), y operó como empresa independiente hasta que [SAP](https://es.wikipedia.org/wiki/SAP_AG) anunció su plan de adquirirla en mayo de 2010. En julio de ese mismo año, se completa la operación por valor de 4.567 millones de euros.

# 

## SQL SERVER

Es un sistema de manejo de bases de datos del [modelo relacional](https://es.wikipedia.org/wiki/Base_de_datos_relacional), desarrollado por la empresa [Microsoft](https://es.wikipedia.org/wiki/Microsoft).

El lenguaje de desarrollo utilizado (por línea de comandos o mediante la interfaz gráfica de Management Studio) es [Transact-SQL](https://es.wikipedia.org/wiki/Transact-SQL) (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos ([DML](https://es.wikipedia.org/wiki/Lenguaje_de_manipulaci%C3%B3n_de_datos)), crear tablas y definir relaciones entre ellas ([DDL](https://es.wikipedia.org/wiki/Lenguaje_de_definici%C3%B3n_de_datos)).

Dentro de los competidores más destacados de SQL Server están: [Oracle](https://es.wikipedia.org/wiki/Oracle), [MariaDB](https://es.wikipedia.org/wiki/MariaDB), [MySQL](https://es.wikipedia.org/wiki/MySQL), [PostgreSQL](https://es.wikipedia.org/wiki/PostgreSQL). SQL Server solo está disponible para [sistemas operativos](https://es.wikipedia.org/wiki/Sistema_operativo) [Windows](https://es.wikipedia.org/wiki/Windows) de Microsoft.

Puede ser configurado para utilizar varias instancias en el mismo servidor físico, la primera instalación lleva generalmente el nombre del servidor, y las siguientes - nombres específicos (con un guion invertido entre el nombre del servidor y el nombre de la instalación).

# Estadar de Codificación

## Introducción

La Especificación del lenguaje C# no define un estándar de codificación. Sin embargo, Microsoft utiliza las instrucciones de este tema para desarrollar ejemplos y documentación.

Las convenciones de codificación tienen los objetivos siguientes:

• Crean una apariencia coherente en el código, para que los lectores puedan centrarse en el contenido, no en el diseño.

• Permiten a los lectores comprender el código más rápidamente al hacer suposiciones basadas en la experiencia anterior.

• Facilitan la copia, el cambio y el mantenimiento del código.

• Muestran los procedimientos recomendados de C#.

## Nombre de los Archivos

* *No utilizar infraguiones para nombrar los archivos de código fuente.*
* *Nombrar el archivo de código fuente igual a la clase principal que implementa.*

El archivo de código fuente de una clase, debe tener el mismo nombre que la clase. Adicionalmente no debe incluirse más de una clase principal por archivo.

## Variables

En breves ejemplos que no incluyen directivas using, utilice calificaciones de espacio de nombres. Si sabe que un espacio de nombres se importa en un proyecto de forma predeterminada, no es necesario completar los nombres de ese espacio de nombres. Los nombres completos pueden partirse después de un punto (.) si son demasiado largos para una sola línea, como se muestra en el ejemplo siguiente:

var currentPerformanceCounterCategory = new System.Diagnostics.

PerformanceCounterCategory();

## Constantes

Los nombres de las constantes se deben escribir en mayúsculas separadas por guiones bajos. Los nombres de constantes pueden contener dígitos, pero no como el primer carácter.

final static int MAX\_PARTICIPANTS = 10;

## Estructuras de Control

* **No cambiar una variable del loop dentro de un bloque for.**

La actualización de una variable loop dentro de un bloque loop es generalmente considerada confuso, y aún más cuando el valor de la variable se modifica en más de un lugar.

* **Todas las primitivas de control de flujo** *(if, else, while, for, do, foreach, switch)* **deben codificarse con bloque, aun cuando esté vacío.**
* **Todas las sentencias switch deben poseer una etiqueta default en el último case.**

Es recomendable un comentario como “no action” donde sea explícita esta intención. Si la etiqueta default es la última, es más sencillo de ubicar.

* **Evitar el uso de sentencias return condicionales o múltiples.**

La norma para un flujo simple es: “una entrada, una salida”. En algunos casos tales como validación de precondiciones, puede ser una buena práctica forzar la salida si una de las precondiciones no se cumple.

* **No realizar comparaciones explícitas true o false.**

Usualmente es considerada una mala práctica comparar una expresión booleana contra expresiones true o false.

Ejemplo:

while [condition == false] // Mala práctica

while [condition != true] // Mala práctica

while (((condition == true) == true) == true) // Mala práctica

while (booleanCondition) // Práctica recomendada

## Clases

Utiliza sustantivo o frases nominales para nombrar una clase. Así es consistente con .NET y fácil de recordar.

|  |  |
| --- | --- |
|  | **public class** Developer  {  }  **public class** ItemLocation  {  }  **public class** WorkItem  {  } |

## Métodos

Habrá que usar PascalCasing para los métodos:

|  |  |
| --- | --- |
|  | **public class** DeveloperAction  {  **public void** CodeSomething()      {          //...      }  **public void** CommentSomething()      {          //...      }  } |

## Atributos

* **No utilizar atributos public o protected.**

Evitando el uso de atributos expuestos directamente al programador, las clases pueden ser actualizadas con más facilidad pues el atributo no debe ser transformado en propiedad, manteniendo así la compatibilidad binaria.

La recomendación es que siempre se implementen los métodos Get y Set en lugar de declarar los atributos como públicos. El siguiente ejemplo muestra el uso correcto de atributos privados y propiedades con métodos Get y Set.

Ejemplo:

public struct Point

{

private int xValue;

private int yValue;

public Point(int x, int y)

{

this.xValue = x;

this.yValue = y;

}

public int X

{

get

{

return xValue;

}

set

{

xValue = value;

}

}

public int Y

{

get

{

return yValue;

}

set

{

yValue = value;

}

}

* **Utilizar una protected property que retorne el valor del atributo para exponerlo a una clase derivada.**

Ejemplo:

public class Control: Component

{

private int handle;

protected int Handle

{

get

{

return handle;

}

}

}

* **Utilizar atributos public static read-only para instancias predefinidas de los objetos de la clase.**

Debe utilizarse nomenclatura Pascal, pues los atributos son públicos. El siguiente ejemplo muestra el uso correcto de los atributos declarados como public static read-only.