**Instituto Tecnológico Superior**

**“Nelson Torres”**

**CARRERA: DESARROLLO DE SOFTWARE**

**PROYECTO INTEGRADOR**

**“Sistema para la Automatización del proceso de seguimiento y control académico para el Instituto Tecnológico Superior “Nelson Torres” en el año 2018-2019”**

Nivel : Tercero “B”

Integrantes : TUGUNANGO FERNANDEZ JUAN WILFRIDO

2018-2019

ÍNDICE DE CONTENIDO

# 1 PROYECTO INTEGRADOR

## 1.1.-Nombre del proyecto

Sistema para la Automatización del proceso de seguimiento y control académico para el Instituto Tecnológico Superior Nelson Torres

## 1.2.- Tema del proyecto Micro

* Análisis diseño e implementación de la base de datos en un servidor central para el almacenamiento de la información.
* Desarrollo de la interfaz gráfica de usuario que sea amigable y de fácil uso
* Implementación e integración de los diversos módulos del sistema y pruebas de funcionamiento

## 1.3 Asignaturas integradoras

Asignatura Código

* Calculo diferencial e integral DS301
* Fundamentos de administración DS302
* Bases de Datos Avanzada DS303
* Diseño de Interfaz DS304
* Programación visual DS305
* Ingles V1.1 DS306

## 1.4.-Asignatura Rectora

Programación visual

## 1.4.-Nombre de los docentes tutores

* Ing. Gisel Rodríguez
* Ing. Santiago Limongi
* Ing. Viviana Zurita
* Ing. Ramiro Aguirre
* Ing. Santiago Solis
* Ing. Fernando Pavón

## 1.5.-Nombres de los estudiantes

## TUGUNANGO FERNANDEZ JUAN WILFRIDO CC. 172767772-4

Nombre N° Cédula

Nombre N° Cédula

Nombre N° Cédula

Nombre N° Cédula

**1.6.-Localización geográfica del proyecto**

Provincia : Pichincha

Cantón : Cayambe

Institución / Organización : Instituto Tecnológico Superior “Nelson Torres”

# 2 DIAGNÓSTICO SITUACIONAL

En el Instituto Superior Tecnológico Nelson Torres “INT”, se ha planteado la necesidad de gestionar datos referentes a los avances temáticos diarios en las clases dictadas, asistencia de los docentes y estudiantes, información de ingreso a las aulas, seguimiento de sílabos y otros informes.

Una solución común a este tipo de situación se ha dado a través de los sistemas de control aplicables en cualquier institución que tienen la finalidad de apoyar de manera automatizada a los usuarios el manejo de forma instantánea los procesos comunes en el aspecto académico docente.

Lo anterior conduce a la idea de que se demanda una recolección y un adecuado manejo de la información académica de los profesores del INT en las diligencias de docencia y gestión académica estudiantil que realiza diariamente, de forma que genere novedades de sus actividades.

Hoy en día el INT, no dispone un método que agilice los procesos académicos y administrativos, trayendo como consecuencia a que en cada semestre la planificación docente se vuelva una labor muy compleja, ocasionando gran pérdida de tiempo y esfuerzo.

La insuficiencia académica implica manipular una extensa cantidad de registros de datos en archivos y hojas de cálculo, y con la procreación de reportes que consumen muchos recursos, que deberían ser encaminados a otras actividades.

Al no tener procesos automatizados para el seguimiento académico de los docentes y estudiantes, existe mucha inconsistencia en la recolección de información tales como diarios temáticos y asistencias; en ocasiones, las autoridades no han dado la prioridad requerida a la inversión en la automatización de dichos procesos porque se ha podido llevar manualmente estas actividades, aunque con un efecto negativo que es una deficiente gestión de la información.

Es por ello que en la asignatura de Programación Visual se ha considerado como prioritario para los intereses institucionales, la automatización de la gestión académica docente para mejorar el control académico a la vez que sirva de soporte a la evaluación de carreras realizadas por instituciones encargadas de esta actividad.

2.1 FORMULACIÓN DEL PROBLEMA

El Instituto Superior Tecnológico Nelson Torres requiere la automatización de la gestión académica docente para mejorar el control de los avances académicos a través del seguimiento de las actividades diarias que el profesor realiza, así mismo que den soporte e información a los estudiantes de los temas tratados en cada clase y que desempeñen con las exigencias de los organismos reguladores de instituciones superiores.

# 3.-JUSTIFICACIÓN

El presente proyecto se justifica ya que es difícil el control y seguimiento de la información académica curricular de las carreras del Instituto Tecnológico Superior “Nelson Torres” lo que señala la conveniencia de crear un sistema automatizado que permita gestionar eficientemente el avance académico curricular de cada una de las asignaturas en cada carrera y obtener reportes de asistencia estudiantil, temáticas de la asignatura para cada usuario registrado con el correspondiente manejo de roles de acceso y restricción en los cambios de la información.

El sistema se encargará del almacenamiento de datos relacionados con el problema planteado, lo que garantizará que el proceso de control de esta información se lo realice eficientemente por lo que es relevante desde el punto de vista social y por el aporte práctico que tendrá.

# 4.-OBJETIVOS

## 4.1.- Objetivo General

Crear un sistema para la Automatización del proceso de seguimiento y control académico para el Instituto Tecnológico Superior Nelson Torres”

## 4.2.- Objetivos Específicos

* Analizar, diseñar e implementar la base de datos en un servidor central para el almacenamiento de la información.
* Desarrollar la interfaz gráfica de usuario que sea amigable y de fácil uso
* Implementar e integrar los diversos módulos del sistema y pruebas de funcionamiento

# 5.- MARCO TEÓRICO

# 5.1.- Ingeniería de Software

La Ingeniería de Software es una disciplina que trata sobre el análisis, diseño y construcción de aplicaciones informáticas a través de la utilización de metodologías, herramientas, técnicas de desarrollo y documentación para una correcta gestión del software en una organización.

**Esta ingeniería trata con áreas muy diversas de la**informática y **de las**[**ciencias de la computación**](http://es.wikipedia.org/wiki/Ciencias_de_la_computaci%C3%B3n)**, tales como construcción de compiladores, sistemas operativos, o desarrollos páginas web, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información y aplicables a infinidad de áreas pudiendo ser negocios, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, derecho, Internet, entro otras.** (UDIMA, 2018)

# 5.2.- Lenguajes de programación

# **Un lenguaje de programación es básicamente un sistema estructurado de comunicación, similar al humano,** el cual nos permite comunicarnos por medio de signos, ya sean palabras, sonidos o gestos. Refiriéndonos a los aparatos, este sistema está organizado para que se entiendan entre sí y a su vez interprete las instrucciones que debe ejecutar.

El término programación se define como un **conjunto de instrucciones consecutivas y ordenadas que llevan a ejecutar una tarea específica.** Dichas instrucciones se denominan “código fuente”, el cual es único para cada lenguaje y está diseñado para cumplir una función o propósito específico. Usan diferentes normas o bases para controlar el comportamiento de un dispositivo y también pueden ser usados para crear programas informáticos. En la actualidad, hay más de cien lenguajes de programación diferentes. (blogvelneo, 2014)

# 5.3.-UML

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación.

Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene. (unad, 2015)

# 5.3.1.- Herramientas UML

# UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Recordemos que un modelo es una representación simplificada de la realidad.

A continuación, se describirán los diagramas más comunes del UML

* Diagrama de Clases
* Diagrama de Objetos
* Diagrama de Casos de Uso
* Diagrama de Estados
* Diagrama de Secuencias
* Diagrama de Actividades
* Diagrama de Colaboraciones
* Diagrama de Componentes
* Diagrama de Distribución (unad, 2015)

Diagrama de Casos de Uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario. Los diagramas de caso de uso modelan la funcionalidad del sistema usando actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios. (Cevallos, 2015)

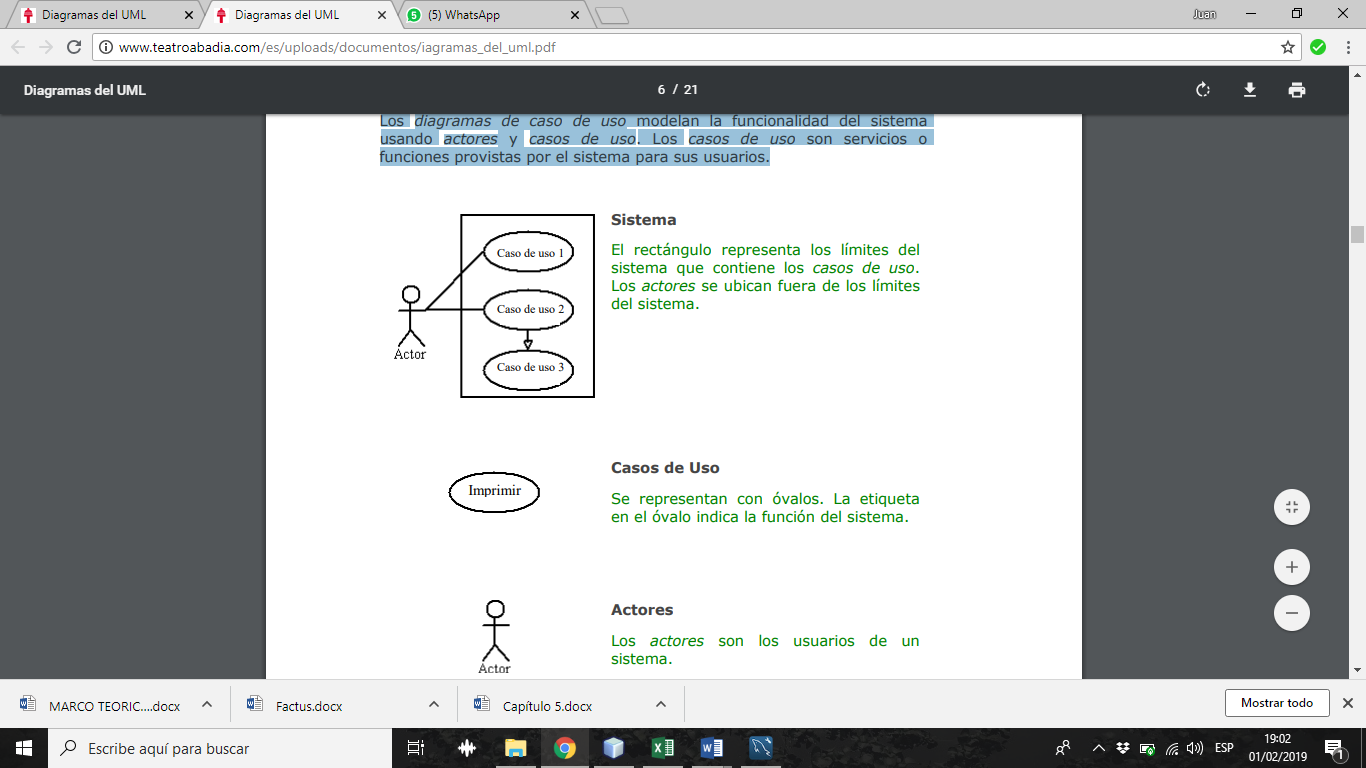


Ilustración 2 Casos de uso

Diagrama de Clases

Los diagramas de clases describen la estructura estática de un sistema. Las cosas que existen y que nos rodean se agrupan naturalmente en categorías. Una clase es una categoría o grupo de cosas que tienen atributos (propiedades) y acciones similares.

Un rectángulo es el símbolo que representa a la clase, y se divide en tres áreas. Un diagrama de clases está formado por varios rectángulos de este tipo conectados por líneas que representan las asociaciones o maneras en que las clases se relacionan entre sí.

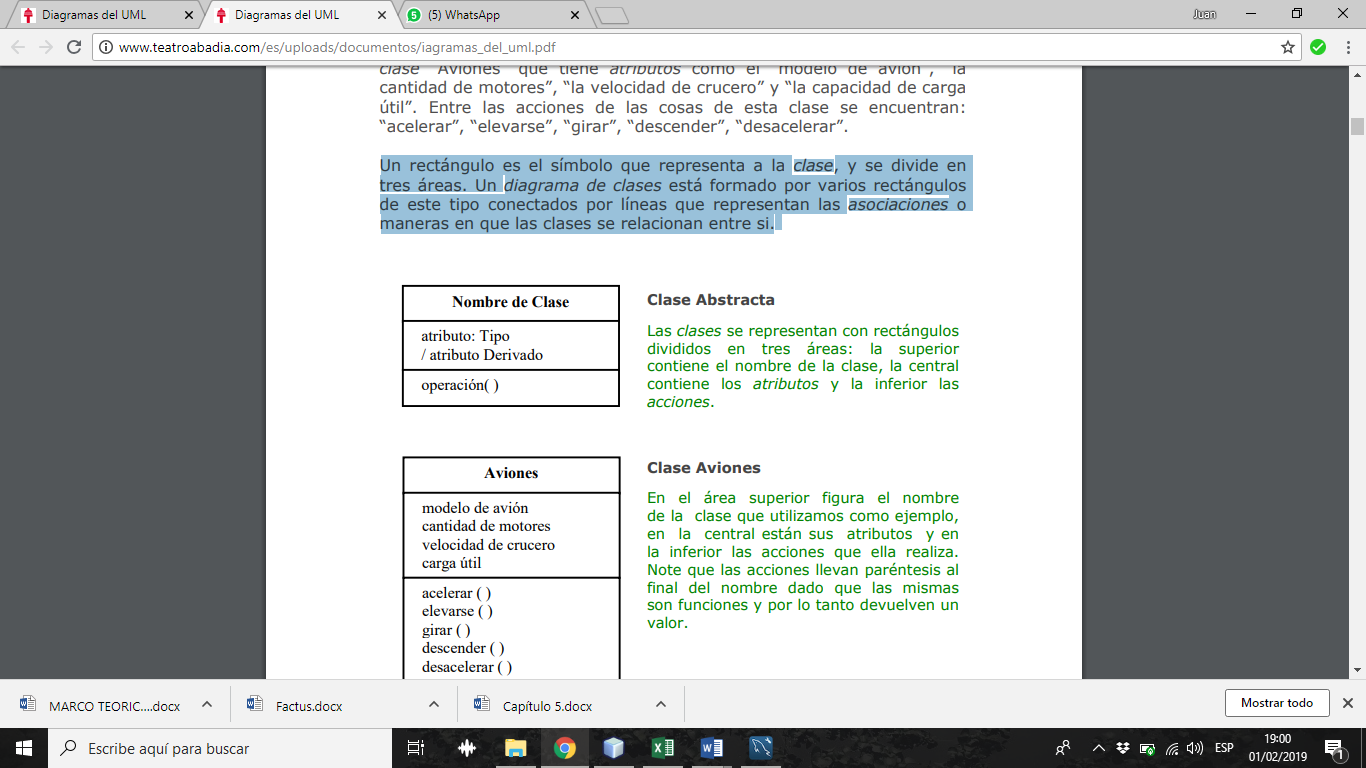


Ilustración 3 Diagramas de clases

Diagramas de secuencia

Estos diagramas en el UML se utilizan principalmente para la modelación de aspectos dinámicos, o sea, de las interacciones entre los objetos y los actores en un sistema, de igual manera que las interacciones de los objetos entre sí. Habitualmente, sirven para mostrar como interaccionan unos objetos con otros en un caso de uso o un escenario de ejecución. (Patricia Lopez, 2014)

# 5.4.- DBMS

Es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos, está compuesto por DDL (Lenguaje de Definición de Datos), DML (Lenguaje de Manipulación de Datos) y SQL (Lenguaje de Consulta).

* Definir una base de datos consiste en especificar los tipos de datos, estructuras y restricciones para los datos que se almacenarán.
* Construir una base de datos es el proceso de almacenar los datos sobre algún medio de almacenamiento.
* Manipular una base de datos incluye funciones como consulta, actualización, entre otros.

# 5.5.- Base de Datos

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.

Las bases de datos tradicionales se organizan por campos, registros y archivos. Un campo es una pieza única de información; un registro es un sistema completo de campos; y un archivo es una colección de registros.

# 5.6.- Estructura Cliente-Servidor

Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras. La interacción cliente-servidor es el soporte de la mayor parte de la comunicación por redes. Ayuda a comprender las bases sobre las que están construidos los algoritmos distribuidos. (Juan Francisco Montalvo, 2015)

# 5.7.- Java

# Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

# 5.8.- NetBeans

# **NetBeans** es un entorno de desarrollo gratuito y de código abierto, permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones Web, o para dispositivos móviles. Da soporte a las siguientes tecnologías, entre otras: **Java**, **PHP**, **Groovy**, **C/C++**, **HTML5** entre otras. Además, puede instalarse en varios sistemas operativos: Windows, Linux, Mac OS, etc.

* Características De NetBeans
* Suele dar soporte a casi todas las novedades en el lenguaje Java. Cualquier preview del lenguaje es rápidamente soportada por NetBeans.
* Asistentes para la creación y configuración de distintos proyectos, incluida la elección de algunos frameworks.
* Buen editor de código, multilenguaje, con el habitual coloreado y sugerencias de código, acceso a clases pinchando en el código, control de versiones, localización de ubicación de la clase actual, comprobaciones sintácticas y semánticas, plantillas de código, coding tips, herramientas de refactorización, etc.
* Simplifica la gestión de grandes proyectos con el uso de diferentes vistas, asistentes de ayuda, y estructurando la visualización de manera ordenada, lo que ayuda en el trabajo diario.
* Herramientas para depurado de errores: el debugger que incluye el IDE es bastante útil para encontrar dónde fallan las cosas
* Optimización de código: por su parte el Profiler nos ayuda a optimizar nuestras aplicaciones e intentar hacer que se ejecuten más rápido y con el mínimo uso de memoria.
* Acceso a base de datos: desde el propio Netbeans podemos conectarnos a distintos sistemas gestores de bases de datos, como pueden ser Oracle, MySQL y demás, y ver las tablas, realizar consultas y modificaciones, y todo ello integrado en el propio IDE.

# 5.9.- MySQL

# MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo ​ y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web. Usada para mover grandes cantidades de información, muchos proyectos en internet como Facebook o Twitter nacieron con este SGBD. (Ramirez, 2018)

# 6.-MARCO LEGAL

# 7.- METODOLOGÍA

## 7.1.- Tipos de Investigación

Aquí se detalla los tipos de investigación que se van a emplear para el desarrollo de la temática y de que manera aportan.

## 7.2.- Métodos de Investigación

Describir cada uno de los métodos utilizados para alcanzar los objetivos deacuerdo al tipo de investigación empleada.

## 

## 7.3.- Técnicas de Investigación

Describir cada uno de los instrumentos utilizados para el levantamiento de información ya sean encuestas, entrevista, focus group, observación directa o indirecta, entre otras y como apoyan en la investigación

# 8.- POBLACIÓN Y MUESTRA

Aquí se debe describir cual es la población objeto de estudio y el cálculo utilizado para determinar la muestra, si el proyecto lo amerita.

Usted utilizara el cálculo de la población o muestra si el proyecto lo amerita previa autorización de su docente tutor.

# 8.1 Cálculo del tamaño de la muestra

**Población Infinita Población Finita**

**Simbología:**

N = Tamaño de la población

n = Tamaño de la muestra

p = Probabilidad de aceptación - 0,50

q = Probabilidad de No Aceptación - 0,50

z= Nivel de Confianza - 1,96

Σ = Error muestral - 0,05

9.-Desarrollo o propuesta

En esta sección se aborda el diseño del sistema propuesto; así como las tecnologías y metodologías que se usaron para la implementación, de igual manera también se describe detalladamente la propuesta técnica del proyecto con los artefactos principales que intervienen.

* Diseño de la propuesta

El corazón de un sistema de información académico administrativo es la base de datos que lo sostiene para lo cual se utiliza el modelo Entidad-Relación como referencia, en el esquema conceptual de la base de datos se representan tipos de entidades e interrelaciones que soportan los procesos que intervienen en el negocio.

Un tipo de entidad puede ser un actor o la información obtenida de aplicar un proceso o caso de uso; mientras que las interrelaciones pueden expresar restricciones de dominio, asociaciones, especificaciones, generalizaciones, entre otros.

La base de datos facilitará la búsqueda y la navegación dentro del sistema a través de las consultas en lenguaje SQL. Las aplicaciones desarrolladas en casi todos los lenguajes contemporáneos incluyen herramientas de conexión e interacción completa con los DBMS más comunes.

Es importante destacar que al realizar un diseño propio de la base de datos este permitirá que los requerimientos de un sistema pueden cambiar fácilmente con la aparición de nuevas características y que debe estar acorde con los casos de uso propuestos en sistema académico, así la base de datos contiene tipos de entidades centrales para establecer un orden lógico en el soporte de estos procesos: Docente, Estudiante, Materia, entre otros.

El tipo de entidad Docente agrupa en su entorno la información personal del docente, y así sucesivamente con cada tipo de entidad cuya semántica está implícita a partir de su nombre.

Los distintos tipos de entidades e interrelaciones que se modelan en el esquema Entidad-Relación generan tablas a nivel físico de la base de datos implementada en el DBMS. Además, las interrelaciones contribuyen a la implementación de características automáticas en el sistema al manejar la información, pues muchos componentes de los formularios se construyen siguiendo los patrones establecidos en el diseño de la base de datos.

Diagrama de clases

Esquema de la base de datos

Diagramas de secuencia

Construcción de la propuesta

Diagrama de implementación

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Categoría | Ponderación % | (5) Sobresaliente 10 | (4) Muy Bueno 8.5 | (3) Bueno 7 | (2) Regular 4 | (1) Insuficiente 1 |
| Diseño del programa bajo el paradigma a objetos | 30 | Se ha realizado un análisis de entidades y funcionalidades y se han identificado los módulos del programa 3 pts. | Se utilizan correctamente interfaces o relaciones de herencia para hacer polimorfismo donde fuese necesario 2,55 pts. | Se utilizan clases y objetos de librerías. Se evita repetir implementaciones de funcionalidades sin justificación 2,1 pts. | Utiliza clases, pero no instanciación de objetos de librerías. 1,2 pts. | No utiliza clases ni instanciación de objetos de librerías. Repite implementaciones de funcionalidades sin justificación 0,3 pts. |
| Implementación y desarrollo ( máximo 3 pts.) | 30 | El formato del código fuente es correcto conforme a algún estándar (preferible al estilo de Java) 3 pts. | Se utilizan clases para representar cada una de las entidades del programa 2,55 pts. | El programa tiene un diseño sencillo y eficiente, pero no con un suficiente detalle 2,1 pts. | Las estructuras del programa no es la adecuada debido a la falta de creación de clases 1,2 pts. | No hay coherencia en el código no manipula los conceptos del paradigma a objetos. 0,3 pts. |
| Visualización de resultados ( máximo 2 pts.) | 20 | La salida es creativa e innovadora y cumple con todos los requerimientos 2 pts. | La salida tiene elementos adicionales a los elementos básicos 1,7 pts. | La salida es legible, ordenada y contempla los elementos necesarios 1,4 | La salida es legible pero no muestra orden 0,8 pts. | La salida no es legible ni ordenada 0,2 pts. |
| Estructuración del código ( máximo 1 pts) | 10 | Se utilizan correctamente relaciones de agregación o composición para estructurar el programa 1 pts. | Se evita la repetición de código o la falta de estructura. Por ej. se añaden las clases necesarias, aunque sean pequeñas 0,85 pts. | Maneja conceptos de clases e instanciación de objetos sin el manejo de herencia 0,7 pts. | No existe un concepto claro en el manejo de herencia de clases e instanciación de objetos con la invocación de métodos 0,4 pts. | No maneja conceptos de herencia y en ningún caso existe la creación de clases y la declaración de sus miembros (atributos, métodos) 0,1 pts. |
| Documentación ( máximo 1 pts.) | 10 | La presentación incluye el nombre, los apellidos, la carrera, nombre del profesor especialidad, nombre de la institución, objetivos bien definidos documentados y partes del código bien documentado. 1 pts. | Le faltan algunos datos presentación, nombre del profesor especialidad, y nombre de la institución, los objetivos no están completos define bien el código en la documentación. 0,85 pts. | La presentación está incompleta, la carrera, nombre del profesor especialidad, y nombre de la institución, objetivos no define bien el código en la documentación. 0,7 pts. | No tiene presentación no tiene objetivos no define bien el código en la documentación. 0,4 pts. | No es aceptable la documentación. 0,1 pts. |
|  |  |  |  |  |

# 10.- CRONOGRAMA DE ACTIVIDADES (en Project)

En este apartado se describe cada una de las actividades a realizarse para la consecución del proyecto integrador esto se deberá realizar en el programa Project

# 11.- BENEFICIARIOS

Se indica si fuera el caso quienes será las personas beneficiarias del estudio o ejecución del proyecto

Directos:

Indirectos:

# 12.- RECURSOS

Se da a notar todos los recursos utilizados en el proyecto de una manera detallada

* Recursos Humanos.-
* Recursos Materiales,
* Recursos tecnológicos.-

# 

# 13.- COSTOS DE LA ELABORACIÓN DEL PRODUCTO O SERVICIO

En este campo se detalla de manera específica todos los costos y gastos para la utilización de los recursos detallados en el numeral 12. Es importante indicar que aquí se darán a conocer todos los costos incurridos para la construcción del prototipo.

### **Tabla No. Costos**

|  |  |  |  |
| --- | --- | --- | --- |
| Detalle | CANTIDAD | VALOR UNIT | VALOR TOTAL |
| Papel | 4 |  |  |
| Gasolina | 4 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| TOTAL |  |  |  |
|  |  |  |  |

Fuente:

Elaborado por:

# 17 CONCLUSIONES

# 18 RECOMENDACIONES

# 19 REFERENCIAS BIBLIOGRÁFICAS

# 20 ANEXOS

Notas se debe borrar todo lo que este marcado de color rojo