 **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**

**MATERIA**

Desarrollo de Aplicaciones Web

Optativa II

Administración de Proyectos TI

**PARCIAL 1**

**PROFESOR**

Rodolfo Martínez Puente

Sergio Alan Orrala Silva

**ACTIDIDAD 2**

Estructuración de un proyecto de TI

**GRUPO:** ITI-901

|  |
| --- |
| INTEGRANTES |
| 10001609 Ana Laura Acevedo Zárraga [arieszarraga1992@outlook.com](mailto:arieszarraga1992@outlook.com) |
| 12002039 José Alejandro Téllez Aguilar [alejandrotellez9912@gmail.com](mailto:alejandrotellez9912@gmail.com) |
| 12002023 Alan Javier Córdoba Espinosa [alanozono@hotmail.com](mailto:alanozono@hotmail.com) |

Acámbaro, Gto. A 25 de Mayo 2015

Contenido

[Introducción 3](#_Toc420252563)

[Justificación del proyecto 4](#_Toc420252564)

[Descripción de la problemática 4](#_Toc420252565)

[Objetivo general 6](#_Toc420252566)

[Identificar quiénes serán los usuarios de la aplicación 6](#_Toc420252567)

[Mercado meta 6](#_Toc420252568)

[Justificar el patrón o patrones de diseño a utilizar 7](#_Toc420252569)

[Singleton 7](#_Toc420252570)

[Strategy 8](#_Toc420252571)

[Decorator 8](#_Toc420252572)

[Factory 9](#_Toc420252573)

[Por qué Singleton? 10](#_Toc420252574)

[Por qué strategy? 10](#_Toc420252575)

[Script (plantilla) del proceso de administración 12](#_Toc420252576)

[Título del documento 13](#_Toc420252577)

[Justificación de diseño web 15](#_Toc420252578)

[Estándares de calidad aplicables al proyecto de T.I. (documentación y diseño). 16](#_Toc420252579)

[Esquema del ciclo de vida 20](#_Toc420252580)

[Ciclo de vida iterativo. 20](#_Toc420252581)

[Ciclo de vida en cascada. 22](#_Toc420252582)

[Ciclo de vida prototipos. 23](#_Toc420252583)

[Justificación de las actividades del proceso 25](#_Toc420252584)

[Por qué CMMI? 25](#_Toc420252585)

[Por qué ciclo de vida iterativo? 26](#_Toc420252586)

[Planeación del Proyecto 28](#_Toc420252587)

[Carta de inicio del proyecto 28](#_Toc420252588)

[Minuta de reunión de arranque 28](#_Toc420252589)

[Organigrama 29](#_Toc420252590)

[Justificación del Organigrama 30](#_Toc420252591)

[Alcance 30](#_Toc420252592)

[Elementos clave para su planeación, monitoreo y control 31](#_Toc420252593)

[Perfiles de los participantes y del administrador 32](#_Toc420252594)

[Matriz de responsabilidades 33](#_Toc420252595)

[Lista de factores clave de desempeño 34](#_Toc420252596)

[Métodos de comunicación, justificación y formato 35](#_Toc420252597)

[Diagrama de Gantt 36](#_Toc420252598)

[Diagrama de ruta crítica 36](#_Toc420252599)

[Estructura del proyecto 37](#_Toc420252600)

[Justificación de la estructura 38](#_Toc420252601)

[Reflexiones 40](#_Toc420252602)

[Bibliografía 41](#_Toc420252603)

[Diagrama de actividades 42](#_Toc420252604)

# 

# Introducción

La Importancia de administrar un proyecto; es abrir nuevas soluciones para este, teniendo previamente un análisis, sus pros y contras nos ayudarán a salir de cualquier situación crítica presentada en el lapso de su seguimiento. Sirve para aprovechar de mejor manera los recursos sobre todo cuando están limitados en cantidad y/o tiempo de disponibilidad. También ayudará a realizar acciones concisas y efectivas para obtener el máximo beneficio.

Conforme la tecnología crece, las empresas necesitan realizar proyectos que actualicen los procesos de otras compañías para ello deben concentrarse en buscar la manera para alcanzarlo.

Como sabemos las principales funciones de la administración son:

* Planeación.
* Organización.
* Dirección.
* Control.

Estas etapas son las que nos ayudarán a sacar el máximo aprovechamiento a los recursos necesarios para el buen funcionamiento del mismo. La administración imparte efectividad a los esfuerzos humanos y ayuda a obtener mejor personal, materiales, dinero y relaciones humanas.

Si no sabemos administrar un proyecto, no solo se perdería dinero, sino también tiempo y claro productividad; siempre es necesario tener bien definido que es lo que queremos, como lo queremos y donde, si no pues todo sería un fracaso total.

# Justificación del proyecto

## Descripción de la problemática

El Colegio Teresa Martín ha trabajado con un sistema llamado Servo Escolar en donde se registran alumnos al inicio de cada ciclo escolar asignándoles una matrícula, pero debido a que este ha presentado problemas los registros como:

1. Cuando los alumnos cambiaba de grado o se gradúan se les asignaba matricula nueva.
2. Cuando realizaban las bajas, el sistema eliminaba por completo el registro sin obtener ningún respaldo.
3. No tiene forma de consultar su historial académico o los ingresos que han generado mensualmente o anualmente.
4. Los recibos impresos no llevan un folio continuo no tiene un control interno de los ingresos y a la vez saber qué cantidad de alumnos sean inscritos para primaria y preescolar.
5. Las actualizaciones para este sistema son relativamente caras y no pueden adquirirlas, debido a que no cuenta con la solvencia económica.

Lo que nosotros ofrecemos como empresa responsable es un sistema que les ayude a realizar los registros, tomando en cuenta que este actualizará de forma autónoma los grados, grupos y ciclos escolares. El administrador solo tendrá que dar de alta a los alumnos de nuevo ingreso, además de que la aplicación le permitirá imprimir recibos de los pagos y reportes por día o mes según él lo necesite, tratando de hacer un poco más fácil su trabajo.

## Objetivo general

Desarrollar un sistema web para el colegio Teresa Martín con la finalidad de publicitar la institución y administrar las colegiaturas, alumnado y padres de familia de la misma.

## Identificar quiénes serán los usuarios de la aplicación

El perfil del administrador para usar el sistema web:

* + - * Manejo básico de internet.
      * Conocimiento de los procesos internos.
      * Egresado nivel preparatorio con área en Informática.
      * Manejo de formularios.
      * Conocimientos básicos en administración.
      * Facilidad de palabra.

## Mercado meta

* Quien pagara por mi Software?
* Directores de primarias, preescolares o secundarias
* Quien va comprarlo?
* Las primarias, preescolares o secundarias
* Quien es mi competencia?
* Las desarrolladoras locales
* Donde y como vender el software?
* Ofrecerlos a escuelas privadas de manera directa

Directores de escuelas privadas o de gobierno que tengan la necesidad de administrar los ingresos por inscripciones o colegiaturas mensuales, para que de esta forma puedan llevar un control interno de los flujos de dinero, aportando una mejor administración y fácil consulta de datos que a futuro podrán ser usados para sus fines de expansión o incremento en su número de estudiantes.

## Justificar el patrón o patrones de diseño a utilizar

### Singleton

Asegurar que una clase tiene una única instancia y proporciona un punto de acceso global a la misma. A veces es importante asegurar que una clase sólo tiene una instancia (por ejemplo una sola cola de impresión, un gestor de ventanas, un sistema de ficheros).

Solución:

No impide crear múltiples objetos. Responsabilidad del constructor de la clase. Debe haber exactamente una instancia de una clase, que debe ser accesible a los clientes a través de un punto de acceso conocido. La instancia única pueda tener subclases, y los clientes deban ser capaces de usar las subclases sin modificar su propio código.

Ventajas:

* Es necesario cuando hay clases que tienen que gestionar de manera centralizada un recurso.
* Una variable global no garantiza que sólo se instancie una vez.

### Strategy

Definir una familia de algoritmos, encapsularlos y hacerlos intercambiables. Permite que el algoritmo cambie sin que afecte a los clientes que lo usan. También conocido como Policy (política) Strategy Motivación.

Ej: existen muchos algoritmos para dividir un flujo de texto en líneas.

Codificar los algoritmos en las clases que los necesitan no es deseable por lo siguiente:

* Los clientes se hacen más complejos.
* Distintos algoritmos serán apropiados en distintos momentos.
* Es difícil añadir nuevos algoritmos y modificar los existentes.
* No hay reutilización.

Solución:

Definir clases que encapsulen los distintos algoritmos.

Ventajas:

* Se permite cambiar el algoritmo dinámicamente.
* Se eliminan sentencias condicionales para seleccionar el algoritmo deseado.

### Decorator

Permite añadir responsabilidades extra a objetos concretos de manera dinámica. Proporciona una alternativa flexible a la herencia para extender funcionalidad. También conocido como wrapper. A veces se quiere añadir funcionalidad a un objeto concreto, no a una clase entera.

Ej: Un toolkit para GUIs proporciona soporte para añadir marcos, barras de desplazamiento a componentes.

Solución:

Herencia: no es flexible, la funcionalidad se añade estáticamente. Definir una clase decoradora que envuelve al componente, y le proporciona la funcionalidad adicional requerida: más flexible, transparente al cliente, se pueden anidar decoradores

### Factory

Define una interfaz para crear un objeto, pero dejando en manos de las subclases la decisión de qué clase instanciar. Permite que una clase delegue en sus subclases la creación de objetos también conocido como virtual constructor.

Ej: un framework de aplicaciones debe poder presentar distintos tipos de documentos. El framework maneja dos abstracciones:

* Documento: los distintos tipos se definen como subclases.
* Aplicación: sabe cuándo crear un documento, pero no su tipo (no puede predecir el tipo de documento que el programador definirá).

Solución:

Encapsular el conocimiento sobre qué subclase de documento crear, y mover ese conocimiento fuera del framework.

Ventajas:

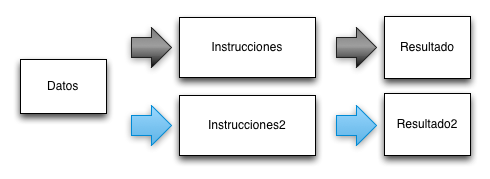
* Centralización de la creación de objetos.
* Facilita la escalabilidad del sistema.
* El usuario se abstrae de la instancia a crear.

### Por qué Singleton?

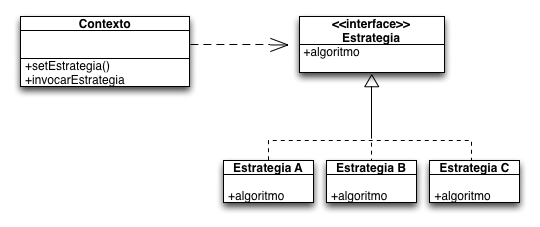
* La idea del patrón Singleton, también llamado de instancia única, es restringir la creación de objetos pertenecientes a una clase. Garantiza que sólo hay una instancia y proporciona un único punto de acceso. Con una única instancia del objeto podemos acceder a él las veces que necesitemos, eso significa que no haremos uso innecesario de recursos ya que no estaremos creando nuevos objetos cada vez que lo necesitemos.
* Su aplicación será sobre la conexión a la base de datos, pues solo requerirá una variable de acceso global para todo el proyecto, su ventaja es que sabremos cual es la única variable para hacer conexión, de esta forma no habrá confusión al momento de usarla y reducirá el margen de error en el momento de programar cualquier módulo que requiera la conexión, también representa menos código y mayor eficiencia en la implementación.

### Por qué strategy?

* Este patrón de diseño se implementará en el momento de hacer un insert o un update a la tabla alumnos, becas y administradores. Pues el patrón propone que cuando hay una entrada de datos puedes hacer una acción tras otra, dependiendo de cuál sea la condición de cada una, para que pueda tomar los datos.
* En nuestro caso se representaría de la siguiente forma como se muestra el flujo en la imagen, el cual recibe un conjunto de datos y dependiendo que características tengan estos va a realizar una acción predeterminada y devolverá un resultado.



* En un diagrama de clases, esta sería la forma en que se puede expresar el patrón de diseño, el cual es de comportamiento es por eso que hace una acción diferente según los datos revividos.



* Pues bien este patrón lo usaremos principalmente para evitar la sobre creación de archivos innecesarios, con esto se espera que el tamaño de la página reduzca y la usabilidad aumente, así se podrán realizar varios flujos con una sentencia if para comprar permitiendo una mejor legibilidad del código y una mejor estructura de este.

## Script (plantilla) del proceso de administración

**Título del documento**

|  |  |
| --- | --- |
| **Versión** | Versión del documento. |
| **Proyecto** | Nombre del proyecto. |
| **Fecha** | Fecha en que se elaboró. |
| **Elaborado por** | Quien lo realizó. |
| **Localización del Documento** |  |
| **Documento base** |  |

|  |  |
| --- | --- |
| **Autorizaciones** | |
| **Nombre** | **Fecha de Autorización** |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Distribución** | |
| **Nombre** | **Fecha de Recepción** |
| José Alejando Téllez Aguilera |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Control de Cambios** | | | |
| **Versión** | **Fecha** | **Participante** | **Descripción del Cambio** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Título del documento

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Nombre del proyecto**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**20/05/2015**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

****

**Versión:** 0.0

**Elaborado por**: nombre de quien lo elaborado por.

**CONTROL DE CAMBIOS**

|  |  |
| --- | --- |
| **Id. Proyecto** | 001 |
| **Proyecto** | Nombre del proyecto. |
| **Fecha** | Fecha en que se realizó. |
| **Elaborado por** | Quien lo elaboró. |
| **Localización del Documento** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Historial de Cambios** | | | |
| **Versión** | **Fecha** | **Participante** | **Descripción del Cambio** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### 

### Justificación de diseño web



El diseño del menú principal está basado en la rápida navegación, haciendo uso de menús verticales los cuales presentan de manera clara y listando los diferentes módulos de la aplicación, el menú abarca la mayor parte de la página por tal motivo es el elemento más resaltante. El título en la parte superior hace referencia al nombre del sistema y su versión esto con el fin de hacer saber en qué modificación o versión de este sistema, además de no quitar visibilidad a el menú que es el importante en el índex.

## Estándares de calidad aplicables al proyecto de T.I. (documentación y diseño).

**CMMI**

Es un modelo para la eficacia de procesos reflejada en una colección de mejores prácticas. También sirve como marco para la organización y priorización de mejora de procesos.

Los documentos que se solicitan dentro del modelo, son los siguientes, para los niveles 3, 4 y 5, de CMMI, deben de documentarse, todos los procesos que se realizan, de manera que, quede constancia de lo que hace cada parte del software, en el caso de ser este el objetivo, es decir, no vale, con representar la funcionalidad integrada de una aplicación, sino que hace falta documentar la de cada parte aislada de la misma, documentando cada objeto o parte de todo el proyecto, modificando las pruebas unitarias y un cuaderno de carga de trabajo del mismo.

Los niveles se utilizan en CMMI para describir un camino evolutivo recomendado para una organización que quiera mejorar los procesos que utiliza para desarrollar y mantener sus productos y servicios. Los niveles pueden también ser el resultado de la actividad como calificación de las evaluaciones.

Para alcanzar un nivel particular, una organización debe satisfacer todas las metas apropiadas del área o conjunto de áreas de procesos que son objeto de la mejora, independientemente de si es un nivel de capacidad o de madurez.

Ambas representaciones también proporcionan caminos para implementar la mejora de procesos que permiten lograr los objetivos de negocio. Ambas representaciones proporcionan el mismo contenido esencial y utilizan los mismos componentes del modelo.

**Nivel de madurez 1: Inicial**

La organización generalmente no proporciona un entorno estable para dar soporte a los procesos. El éxito en estas organizaciones depende de la competencia y heroicidad del personal de la organización y no del uso de procesos probados. A pesar de este caos, las organizaciones de nivel de madurez 1 a menudo producen productos y servicios que funcionan; sin embargo, frecuentemente exceden sus presupuestos y no cumplen sus calendarios.

**Nivel de madurez 2: Repetible**

Los proyectos de la organización han asegurado que los procesos se planifican y realizan de acuerdo a políticas, los proyectos emplean personal con habilidad que dispone de recursos adecuados para producir resultados controlados; involucran a las partes interesadas relevantes; se monitorizan, controlan y revisan.

Ayuda a asegurar que las prácticas existentes se mantengan durante tiempos de estrés. Cuando estas prácticas están en su lugar, los proyectos se realizan y gestionan de acuerdo a sus planes documentados.

**Nivel de madurez 3: Definido**

En el nivel de madurez 3, los procesos son bien caracterizados y comprendidos, y se describen en estándares, procedimientos, herramientas y métodos. El conjunto de procesos estándar de la organización, se establece y mejora a lo largo del tiempo.

Además los estándares, descripciones de proceso y procedimientos para un proyecto se adaptan para adecuarse a un proyecto particular o unidad organizativa a partir del conjunto de procesos estándar de la organización y, por tanto, son más consistentes, exceptuando las diferencias permitidas por las guías de adaptación.

**Nivel de madurez 4: Gestionado cuantitativamente**

La organización y los proyectos establecen objetivos cuantitativos en cuanto al rendimiento de calidad y del proceso, y los utilizan como criterios en la gestión de los procesos. Los objetivos cuantitativos se basan en las necesidades del cliente, usuarios finales, organización e implementadores del proceso.

Se identifican las causas especiales de variación y, donde sea apropiado, se corrigen las fuentes de las causas especiales para prevenir sus futuras ocurrencias.

**Nivel de madurez 5: En optimización**

Se centra en mejorar continuamente el rendimiento de procesos mediante mejoras incrementales e innovadoras de proceso y tecnológicas. Los objetivos cuantitativos de mejora de procesos para una organización se establecen, se revisan continuamente para reflejar el cambio a los objetivos del negocio, y se utilizan como criterios para gestionar la mejora de procesos.

**PMBOK**

Esta norma de calidad nos ayudará a la administración en cuanto a documentación se requiere, es importante que todo proyecto se desarrolle en base a sistemas que con tenga como norma principal el pmbok, además de que nos brinda el estándar de formatos para una buena documentación y dentro de esta se manejan varios aspectos a continuación se nombran algunos y su aplicación.

**Gestión de Alcance:** aquí definiremos que tan extenso estará el proyecto definiendo también los entregables encada fase. Que es lo que quiere el cliente y que es lo que se entrego, con esto se mantendrá el proyecto estable y sin alteraciones de tiempo y trabajo además es la base para las siguientes gestiones.

**Gestión de Tiempo:** esta parte de gestión es para controlar la duración de tiempos, más que nada esto se ve reflejado en el diagrama de Gantt que se realizó desde un principio para el proyecto.

**Gestión de Costos:** en esta sección del pmbok aplicaremos los gastos (entradas y salidas) y esto se ve reflejado en nuestro plan de costos del proyecto el cual ya se ha realizo, con este se pretende mantener un balance entre las fases que requieren más dinero y las que requieren menos dinero. Una de las grandes ventajas es la proyección del fin del proyecto de esta manera sabremos si es alcanzable o será un fracaso.

**Gestión de Calidad:** en este apartamento se ponen a prueba los procesos que son importantes para el desarrollo del software y en la área donde se hacen además de las condiciones y capacitación del personal para desempeñar dichos procesos de forma correcta bajo algunas normas principales.

**ISO 9000-3**

Esta norma se llevará a cabo en los procesos de elaboración de cualquier sistema que se desarrolle dentro de la empresa a demás da la introducción para aplicar otras normas como la ISO 9001 la cual está orientada la calidad del software y su desarrollo.

Al poner en marcha esta norma la calidad y la satisfacción del cliente es la mejor, por ende más clientes estarán interesados en nuestro trabajo, esto se puede tomar como una estrategia de marketing a la hora de la venta de nuestro servicio.

## Esquema del ciclo de vida

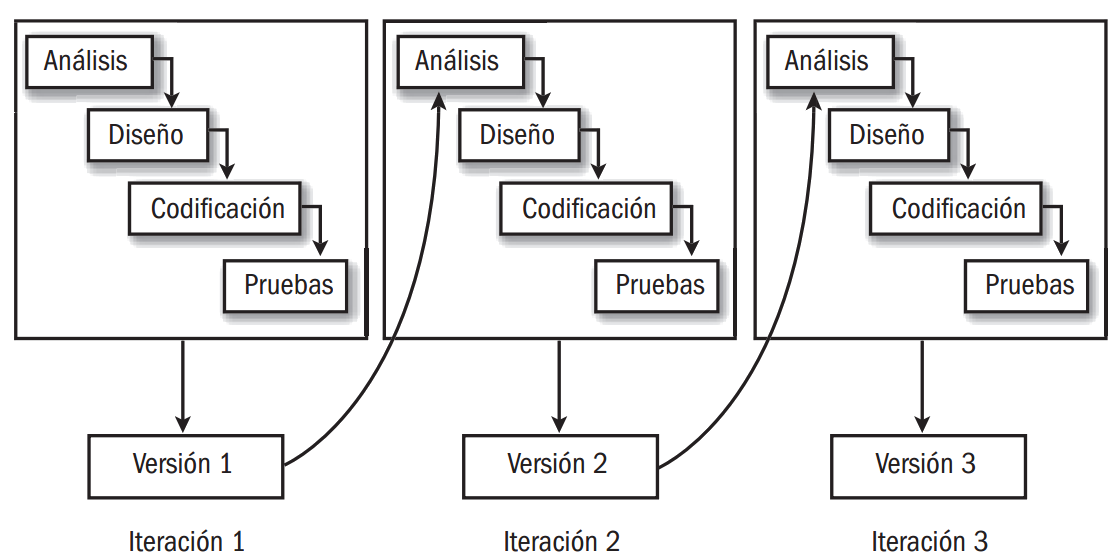
### Ciclo de vida iterativo.

Este modelo, también conocido como Evolutivo, es una derivación del ciclo de vida en cascada puro, que busca reducir el riesgo que surge entre las necesidades del usuario y el producto final por malos entendidos durante la etapa de solicitud de requerimientos.

En el ciclo de vida iterativo, en cada iteración se reproduce el ciclo de vida en cascada a menor escala. Los objetivos de una Iteración se establecen en función de la evaluación de las iteraciones precedentes. Desde el principio, al final de cada iteración se le entrega al cliente una versión completa y mejorada del producto. El cliente es quien luego de cada iteración evalúa el producto y lo corrige o propone mejoras. Estas iteraciones irán refinando el sistema y se repetirán hasta obtener un producto que satisfaga al cliente.

Las 4 fases obligatorias para cada interacción son las siguientes aun que pueden agregarse otras, más sin embargo lo correcto es representar pequeños ciclos en cascada dentro de cada iteración:

1. Análisis: Se hace el análisis de los requerimientos obtenidos en un primer intento y en base a esto se crea la primera versión del software, en las siguientes iteraciones los requerimientos pueden ir aumentando hasta un punto definido.
2. Diseño: En base a lo obtenido en el análisis se hacen los diseños, en una primera interacción los diseños se apegan a la cantidad de requerimientos e información obtenida, después de esta los diseños solo se mejoran en base a lo obtenido en cada iteración.
3. Codificación: En esta fase del proyecto se codifica todo lo obtenido del diseño en cada iteración, presentando cada vez más completo cada módulo hasta la entrega final.
4. Pruebas: Aquí se realizan y redactan los documentos de las pruebas de cada iteración, tomando como objeto de consulta para la próxima iteración, en la cual se resolverán los problemas presentados si es que hubo.
5. Implementación: En esta etapa se pone en marcha la ejecución en línea del sitio web, teniendo como objetivo entregar un sitio 100% funcional.
6. Cierre: Esta etapa se agregara una sola vez pues no se puede repetir en cada iteración pues en esta etapa abarca el cierre del proyecto en sí y la elaboración de sus documentos, cabe destacar que si se puede integrar una fase similar a cada iteración llamada aceptación donde solo se hace la aprobación de cada versión entregada al cliente y se retroalimenta para su desarrollo.



Características

* Se evitan proyectos largos y se entrega algo de valor a los usuarios con cierta frecuencia.
* El usuario se involucre más.
* Difícil de evaluar el costo total.
* Requiere gestores experimentados.
* Los errores en los requisitos se detectan tarde.
* El resultado puede ser muy positivo.

Ventajas:

* Con un paradigma incremental se reduce el tiempo de desarrollo inicial, ya que se implementa la funcionalidad parcial.
* También provee un impacto ventajoso frente al cliente, que es la entrega temprana de partes operativas del Software.
* El modelo proporciona todas las ventajas del modelo en cascada realimentado, reduciendo sus desventajas sólo al ámbito de cada incremento.
* Permite entregar al cliente un producto más rápido en comparación del modelo de cascada.
* Resulta más sencillo acomodar cambios al acotar el tamaño de los incrementos.
* Por su versatilidad requiere de una planeación cuidadosa tanto a nivel administrativo como técnico.

### Ciclo de vida en cascada.

Definición: Contiene diferentes etapas de desarrollo: especificación, diseño, implementación, pruebas y mantenimiento. En principio se debe completar una etapa antes de que se pueda avanzar a la siguiente. En la práctica existe interacción entre las etapas.

En principio, el resultado de cada fase es uno o más documentos entregables que aprueban que la fase ha concluido (firmados). La siguiente fase de debe empezar hasta que la fase previa haya finalizado.

En la práctica, estas etapas se superponen y proporcionan información a las otras. En la fase de diseño se encuentran los problemas de requerimientos, en la fase de programación se en encuentran los problemas con el código y así sucesivamente esto nos indica que en este modelo se identifican los problemas en cada fase y no desde un principio.

Esto hace que los errores sean regresados al principio para que los corrijan y de nuevo que vuelvan a pasar por todas las fases ya corregidas y a la vez representa una ventaja pues una vez identificado y solucionado ya no se propaga a otras fases. Durante la fase final del ciclo de vida (funcionamiento y mantenimiento), el software se pone en funcionamiento. Se descubren errores y omisiones de los requerimientos originales del proyecto.

Hacer estos cambios puede implicar repetir etapas previas del modelo. Su mayor ventaja es la documentación que se produce al final de cada fase para mantener controlado todos los aspectos del ciclo. Por lo tanto este modelo solo se debe de utilizar cuando los requerimientos se comprendan bien y no sea posible que cambien radicalmente durante el desarrollo del proyecto.

### Ciclo de vida prototipos.

En la práctica los prototipos se utilizan para validar los requerimientos de los usuarios en cualquier ciclo de vida. Si no se conoce exactamente como se desarrollará un determinado producto o cual son las especificaciones de forma precisa, suelen recurrirse a definir especificaciones iniciales para hará un prototipo, o un producto parcial y provisional. En este modelo, el objetivo es lograra un producto intermedio, antes de realizar un producto final, para conocer mediante el prototipo como responder las funcionalidades previstas para el producto final. Si se quiere implementar este ciclo primero hay que revisar si vale la pena realizar un prototipo.

Se utiliza mayoritariamente en desarrollos de productos con innovaciones importantes, o en el uso de tecnologías nuevas o poco probadas, en las que la incertidumbre sobre los resultados a obtener, o la ignorancia sobre el comportamiento, impiden iniciar un proyecto secuencial.

La ventaja de este ciclo se basa en que es el único apto para desarrollos en los que no se conoce a priori sus especificaciones o la tecnología a utilizar. Como contrapartida, por este desconocimiento, tiene la desventaja de ser altamente costoso y difícil para la administración temporal.

Ventajas:

* Ofrece visibilidad del producto desde el inicio del ciclo de vida con el primer prototipo. Esto puede ayudar al cliente a definir mejor los requisitos y a ver las necesidades reales del producto.
* Permite introducir cambios en las iteraciones siguientes del ciclo.
* Permite la realimentación continua del cliente.
* Este modelo reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios.

Desventajas:

* Tiene un desarrollo lento.
* Se hacen fuertes inversiones en un producto desechable ya que los prototipos se descartan.
* Aumente el coste de desarrollo del producto. Conformé los prototipos realizados.
* Pueden surgir problemas con el cliente que ve funcionando versiones del prototipo pero puede desilusionarse porque el producto final aún no ha sido construido.

El desarrollador puede caer en la tentación de ampliar el prototipo para construir el sistema final sin tener en cuenta los compromisos de calidad y de mantenimiento que tiene con el cliente.

## Justificación de las actividades del proceso

### Por qué CMMI?

Se usara el CMMI por ser uno de los indicadores de calidad de desarrollo de software más reconocidos a nivel mundial, además de disponer de 5 niveles de madures los cueles nos indican cómo es que una empresa de desarrollo de software puede obtener un máxima calidad en sus procesos.

No sólo por ser uno de los modelos más reconocidos y usados a nivel mundial, sino también porque recopila una serie de buenas prácticas.

Estas prácticas representan una guía para elevar la eficiencia y la calidad de nuestros desarrollos, consiguiendo así disminuir los costos, asegurar el tiempo de entrega y la satisfacción del Cliente, entre otros beneficios.

### Por qué ciclo de vida iterativo?

La etapa de cierre de proyecto solo se presenta una sola vez a lo largo del esquema de ciclo de vida pues como se explicó anteriormente esta no puede ser abarcada en cada iteración pues indicaría el fin de proyecto, a continuación se presenta una descripción breve de cada fase y las tareas que implica cada una.

1. Análisis: En la fase 1 se identificarán los requerimiento del sistema así también se realizarán los documentos correspondientes a este fase. Además de generar los documentos de inicio de proyecto tales como la carta de inicio y la minuta inicial, más aparte el líder será el encargado de elaborar la planeación (WBS) para todo el proyecto.
2. Diseño: En la fase número 2 cabe destacar que todo lo planteado en el análisis a manera de documentos, se interpretaran con diagramas, por ello en esta fase se realizarán todos los diagramas correspondientes a cada fase y el diseño de la página principal.
3. Codificación: En la fase número 3 está la codificación de cada módulo que tendrá el sistema web, esto también incluirá el desarrollo de la BD y algunos componentes de esta.
4. Pruebas: En la fase número 4 se engloban todas las pruebas, algunas de estas pruebas se hacen a la par de la programación y otras son realizadas después de la programación, al final de esta fase se encuentran algunos documentos que describen los resultados de las pruebas.
5. Implementación: En esta fase se encuentra la elección del hosting y subir los archivos a un servidor para almacenar el sitio web, así como montar la base de datos y hacer la conexión de esta.
6. Cierre: En la última fase se realizarán los documentos para concluir el proyecto así también se contempla la entrega el proyecto y se liberación de recursos del proyecto.

Planeación del Proyecto

## Carta de inicio del proyecto

Es un documento oficial donde se realiza la declaración del alcance, los objetivos y los participantes de un proyecto. Proporciona una delimitación preliminar de las funciones y responsabilidades, se exponen los objetivos del proyecto, identifica las principales partes interesadas, y define la autoridad del director del proyecto.

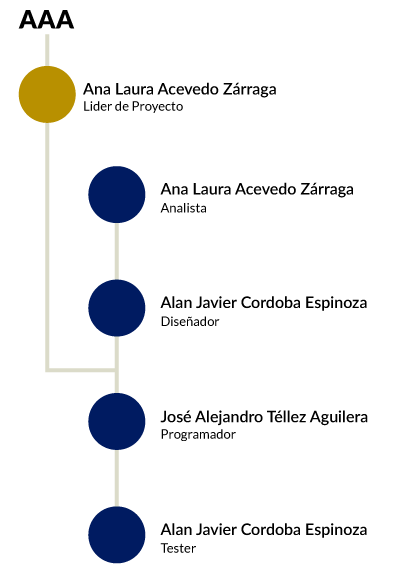
[Carta-de-inicio-del-proyecto .docx](Carta-de-inicio-del-proyecto%20.docx)

## Minuta de reunión de arranque

Sirve como documento de evidencia para un futuro debido a que incluye fecha, hora, lugar, así como los asistentes, los temas tratados de manera brevemente, las decisiones y las acciones a realizar con sus consecuencias, y deben firmarlas todos los presentes. Además de que nos ayuda a llevar el control del proyecto y a la vez a encontrar una solución a los problemas presentados de una forma más eficaz.

<Minuta.docx>

## Organigrama



### Justificación del Organigrama

El organigrama está estructurado de forma jerárquica en la cual se ordena por niveles entre más arriba o más elevado sea el nivel, más alto será el poder sobre los subordinados o niveles inferior, en este caso solo hay 2 niveles el superior, y el nivel inferior en el primero se encuentra el líder de proyecto de acuerdo a lo acordado en el equipo ella será la encargada de llevar al equipo a la recta final o al fin de proyecto, sus subordinados serán los demás roles, cabe destacar que los subordinados están en el mismo nivel y ninguno tiene poder sobre otro, teniendo esta cuestión clara los demás roles tendrán la misma carga de trabajo que sus compañeros vecinos.

Se organizó de esta manera para que todos los integrantes entregaran reportes de avances a una sola persona y esta sería la encargada de darlos a conocer a los demás.

## Alcance

Es el proceso de subdividir los entregables principales en componentes administrables con el objetivo de:

* Mejorar la exactitud de los estimados de costo y tiempo.
* Definir una línea base para medición y control del proyecto.
* Facilitar un clara asignación de roles y responsables.

<Alcance.docx>

## Elementos clave para su planeación, monitoreo y control

Los elementos de monitoreo y control comparan continuamente el desempeño real con el plan de implementación del proyecto (análisis de variación). Si se encuentra variación, los integrantes del proyecto deben analizar su causa, identificar posibles acciones correctivas e implementar los cambios para realinear el modelo (el plan para implementar el proyecto) con la realidad del contexto del proyecto. Los cambios se hacen primero en el plan del proyecto para que sus objetivos en otros aspectos del proyecto puedan considerarse. Cuando el equipo del proyecto y otros interesados estiman que las acciones propuestas producirán el efecto deseado, el plan del proyecto revisado se aprueba y comunica. El trabajo continúa conforme al plan revisado.

<Elementos-clave-para-su-planeación-monitoreo-y-control.docx>

## Perfiles de los participantes y del administrador

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Bases** | **Departamento** | **Principales Competencias** | **Correo electrónico** |
| Ana Laura Acevedo Zárraga.  Alan Javier Córdoba Espinosa.  José Alejandro Téllez Aguilar. | Planificación de proyectos, análisis de sistemas, ordenado capacidad de liderazgo en equipos de alto rendimiento.  Diseño de sistemas con HTML5, CSS3, construcción, implementación.  Diseño de bases de datos, conocimientos de SQL, phpmyadmin, JavaScript. | Jefe del proyecto, Analista.    Diseñador, Tester.    Programador. | * Responsabilidad. * Habilidad Analítica. * Trabajo en equipo. * Capacidad de planificación y organización. * Pro activo. * Búsqueda de información. * Preocupación por el orden y la claridad. * Tolerancia a la presión. * Comunicación asertiva. | [arieszarraga1992@outlook.com](mailto:arieszarraga1992@outlook.com)  [alanozono@hotmail.com](mailto:alanozono@hotmail.com)  [alejandrotellez9912@gmail.com](mailto:alejandrotellez9912@gmail.com) |

## Matriz de responsabilidades

Se utiliza para ilustrar las relaciones entre las actividades, los paquetes de trabajo y los miembros del equipo. El formato matricial muestra todas las actividades asociadas con una persona y todas las personas asociadas con una actividad. Esto asegura que haya una sola persona encargada de rendir cuentas por una tarea determinada a fin de evitar confusiones.

[Matriz de responsabilidades.xlsx](Matriz%20de%20responsabilidades.xlsx)

## Lista de factores clave de desempeño

|  |  |  |
| --- | --- | --- |
| **Objetivo** | **Indicador/ Factor** | **Obtener** |
| Cubrir las necesidades del cliente. | Calidad de trabajo. | El cliente estará satisfecho con el trabajo. |
| Cada uno de los elementos del equipo domina en 90% su área de trabajo. | Conocimientos | Evitar capacitación del personal. |
| Los participantes toman una actitud positiva ante los problemas. | Iniciativa. | El trabajo será más fácil de realizar y su entrega sea tiempo. |
| Estar en contacto por posibles cambios. | Comunicación. | Una comunicación eficiente para evitar con funciones en la información recibida. |
| Que cada uno de los integrantes se responsabilicé del área a cubrir. | Responsabilidades. | Integrantes comprometidos con sus labores. |
| Que el proyecto no sea más caro de lo planeado. | Costos y tiempo. | El proyecto terminado en el tiempo establecido. |

Existen muchos factores a tener en cuenta para la evaluación del desempeño de los colaboradores, dependiendo del método de evaluación que se emplee, algunos objetivos, factores y lo que deseamos obtener con ellos son:

## Métodos de comunicación, justificación y formato

Matriz de comunicación es una herramienta que contiene una descripción detallada de todos los requisitos y necesidades de información de todos los participantes del proyecto y cuya responsabilidad de elaborar e implementar recae sobre el líder del proyecto.

Una buena matriz de comunicaciones permitirá al proyecto y a sus integrantes recibir información práctica, confiable y sobre todo pertinente en relación con las necesidades de los involucrados.

El propósito es definir los requerimientos de comunicación para el proyecto y cómo serán distribuidos. Los requerimientos para su elaboración son:

1. Los requerimientos de comunicación basados en los roles del proyecto.
2. Qué información será comunicada.
3. Cómo la información será comunicada.
4. Cuándo será distribuida la información.
5. Quién lleva a cabo la comunicación.
6. Quién recibe la información.

Los formatos de comunicación algunos serán entregados de forma impresa y otros se enviarán por correo electrónico, por ejemplo el acta de inicio se entregará a cada uno de los participantes del proyecto y al cliente de forma física para que la firmen de enterados, a lo igual el reporte semanal, los SRS y la carta de aceptación, estos ultimo no se entregan a todos solo a los interesados.

[Matriz de comunicacion.xlsx](Matriz%20de%20comunicacion.xlsx)

## Diagrama de Gantt

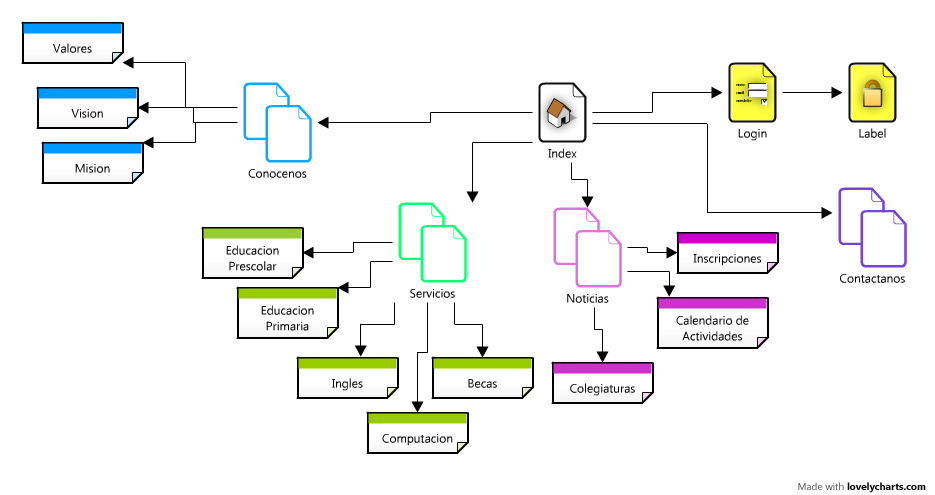
Es una útil herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

[Teresamartin (Wbs).mpp](Teresamartin%20(Wbs).mpp)

## Diagrama de ruta crítica

Es la secuencia de los elementos terminales de la red de proyectos con la mayor duración entre ellos, determinando el tiempo más corto para completar el proyecto. Es la relación de actividades físicas o mentales que forman procesos interrelacionados en un proyecto total.

## **Estructura del proyecto**

Apartados

* Índex: este es la página de inicio donde se presenta el menú del sitio web publicitario así como un banner para hacerlo más atractivo y armónico para la vista de visitante.
* Módulo conócenos: El apartado contendrá la filosofía de la institución para que los padres de familia conozcan un poco más sobre la institución y su ética profesional, de esta manera los padres de familia se verán más interesados y se sentirán más confianza hacia la institución.
* Módulo de servicios: Este tiene la finalidad de dar a conocer los servicios que ofrece la institución así como cursos o talleres para los 2 niveles de educación que ofrece, el objetivo es dar a conocer detalles de cada uno de los servicios.
* Módulo noticias: En este apartado se publicaran noticias relevantes a la institución ya sean fechas futuras o pasadas siempre con el fin de poner la información más destacada para los padres de familia.
* Contáctanos: En este apartado se presentara un formulario con el fin de que todo aquel que visite el sitio web de la institución pueda dejar una opinión o solicitar información de manera directa con la escuela.
* Login: Es el apartado para acceder al administrador del sitio web en el solo aparecerá el formulario y el menú.

### Justificación de la estructura

La estructura manejada para distribuir la información es la más usual en páginas web publicitarias y tomando en cuenta de que es para una institución educativa el contenido se presenta de tal que los padres u otra persona pueda navegar cómodamente en el sitio, encontrando en cada sección información relevante, además de proponer una navegabilidad entre paginas todas partiendo desde el índex o página principal.

El login se representa en este diagrama solo para indicar que estará presente en algún lugar de las carpetas más sin embargo en el menú principal no estará de forma directa.

La estructura propone un desplazamiento entre archivos y carpetas, al estar de esta manera estructurada internamente, la seguridad se incrementara pues al no tener archivos raíz más que el índex, y todos los demás encarpetados se tendrán que saber las rutas completas para poder entrar a un archivo en específico.

# Reflexiones

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Integrantes** | 1 | 2 | 3 | 4 | Total |
| 1. Ana Laura Acevedo Zárraga. | 5 | 5 | 5 |  | 15 |
| 2. Alan Javier Córdoba Espinosa. | 5 | 5 | 5 |  | 15 |
| 3. José Alejandro Téllez Aguilar. | 5 | 5 | 5 |  | 15 |
| 4. |  |  |  |  |  |

# Bibliografía

(www.sei.cmu.edu)

(www.arantxa.ii.uam.es)

(www.arantxa.ii.uam.es)

(www.siul02.si.ehu.es)

(www.donnierock.com)

(www.www.fcarrizalest.com)

(Casanova)

(Torres Hernández Zacarías)

(Benet)

# Diagrama de actividades

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Actividad | Responsable | 11/05/2015 | 12/05/2015 | 13/05/2015 | 14/05/2015 | 15/05/2015 | 16/05/2015 | 17/05/2015 | 18/05/2015 | 19/05/2015 | 20/05/2015 | 21/05/2015 | 22/05/2015 | 23/05/2015 | 24/05/2015 | 25/05/2015 |
| 1 | Dividir las actividades | Ana Laura Acevedo Zárraga |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Realizar introducción | Ana Laura Acevedo Zárraga |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Recopilar información | Todos |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Estandarizar documento | José Alejandro Téllez Aguilera |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Redactar documentos | Todos |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Realizar conclusión | Todos |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Revisión de orografía | José Alejandro Téllez Aguilera |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Realizar el diagrama de actividades | Alan Javier Córdoba Espinosa |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | Juntar toda la actividad | Ana Laura Acevedo Zárraga |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | Entregar actividad | Ana Laura Acevedo Zárraga |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Nuestra Organización está compuesta por un líder de proyecto, Analista, Diseñador, Codificador o Programador y un Tester. Donde cada uno de ellos será responsable de su área, para empezar el líder y analista son los encargados de asignar las actividades, realizar la introducción, apoyar en la recopilación de información, redactar la misma, realizar conclusión, juntar la actividad y enviarla. El diseñador y tester deberán recolectar información, redactar documentos, realizar conclusión y hacer el diagrama de actividades. Por su parte el programador también recopilara información, estandarizar el documento, redactar los documentos, realizar conclusión y revisar ortografía.

### Áreas de mejora

* Tiempo de dicado a las actividades.
* Comunicación entre los integrantes de equipo.
* Calidad de trabajo entregado.
* Dar la oportunidad de que todos opinen.
* La forma de organizarnos para las actividades.

### Soluciones para las áreas de mejora

* Organizar nuestras actividades personales y laborales, para que estas no se conviertan en presión o estrés.
* Estar en contacto con los integrantes del equipo por futuros cambios en las actividades asignadas.
* Entregar el trabajo con buena redacción y checar la ortografía para obtener una mejor calificación.
* Realizar una lluvia de ideas donde todos den su punto de vista.
* Analizar con que habilidades cuenta cada uno y así realizar la asignación de tareas.

# 