



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA DE SISTEMAS

TEMA: DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL DE GRUADOS PARA LA UNIVERSIDAD DE LAS FUERZAS ARMADAS- ESPE, ENMARCADO EN EL “MODELO - VISTA - CONTROLADOR (MVC)” Y UTILIZANDO LA HERRAMIENTA “YII” COMO FRAMEWORK DEL LENGUAJE PHP

AUTORA: ADELA SOFIA GALLARDO CUEVA

DIRECTOR: ING. EDGAR MONTALUISA
CODIRECTOR: ING. XIMENA LÓPEZ

LATACUNGA

2015

Marco Teórico

PATRONES DE DISEÑO

Gamma en su obra manifiesta que los patrones de diseño son “descriptores de clases y objetos relacionados que están particularizados para resolver un problema de diseño general en un determinado contexto” (Erich, Helm, Johnson, & Vlissides, 2003).

En base al propósito; es decir, en función de lo que cada patrón realiza, tenemos la siguiente clasificación:

- creacionales
- estructurales
- comportamiento.

Patrones Creacionales: Están relacionados con los procesos de creación de objetos. Sin descuidar la instancia de creación como su configuración, es decir, ayudan a encapsular y abstraer dicha creación.

Patrones Estructurales: Tratan sobre la composición de clases u objetos, separan la interfaz de la implementación, se ocupan de como las clases y objetos se agrupan para formar estructuras más grandes.

Patrones de Comportamiento: Describen el modo en que las clases y objetos interactúan y distribuyen sus responsabilidades, así como los algoritmos que encapsulan: es decir, muestran la comunicación entre los objetos.

		Propósito		
		De creación	Estructurales	De comportamiento
Ámbito	Clase	Factory Method	Adapter (de clases)	Interpreter Template Method
	Objeto	Abstract Factory Builder Prototype Singleton	Adapter (de objetos) Bridge Composite Decorator Facade Flywiegth Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

Ventajas	Desventajas
Permite reducir operaciones complejas presentando son soluciones concretas: Los patrones son recetas de diseño, se pueden clasificar, manteniendo independencia para exponerlos como simple.	No posee ingeniería inversa
Las soluciones técnicas dada una determinada situación presenta un escenario interesante de desacoplamiento entre los objetos utilizando un lenguaje específico y otros de carácter general.	Es difícil interpretar el patrón: los patrones no se reflejan en el código empleado.
Se aplican en soluciones muy comunes: Proceden de la experiencia y tiene utilidad demostrada en problemas del diseño orientado a objetos.	El patrón de diseño es sumamente costoso y difícil en lenguajes que no se alinean al paradigma de desarrollo.
Presenta soluciones simples: Resuelven un problema particular en la mayoría de aplicaciones utilizando un pequeño número de clases con flexibilidad y finamente con calidad de software.	Su utilización es complicada por el hecho que las tecnologías actuales incentivan a los desarrolladores para particionar la aplicación en la fase de diseño
La utilidad se vuelve cada vez más importante: Nos permite reducir tiempo de desarrollo por lo tanto costo incrementando calidad	Tiempo: El tiempo dedicación en la fase inicial es elevado siendo relativo ya que su ventaja de mantenimiento es productiva
Estructura general: Su estructura es conocida por todos los programadores, de tal manera que su producto al ser independiente no resulte distinta de su forma de trabajar.	
Permiten tener una estructura de código común a todos los proyectos que implemente una funcionalidad genérica.	
Una aplicación con MVC es mucho más mantenible, extensible y modificable	

Modelo Vista Controlador (MVC)



La interacción que se muestra en la Figura es la siguiente:

Solicitud: El usuario ejecuta un navegador, pide un URL solicitando a la Aplicación.

Despacha: La Aplicación delega a la capa **Controladora** la petición del usuario.

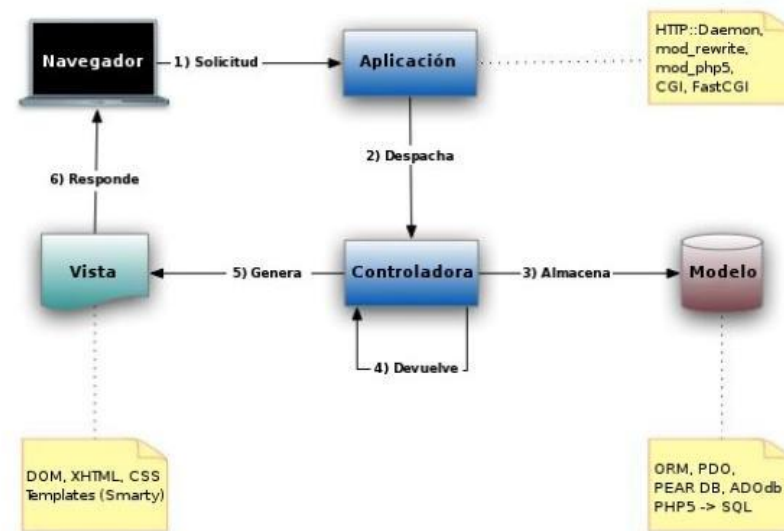
Almacena: Consulta en la capa **Modelo** por medio de la capa de datos del manejador de base de datos existente.

Devuelve: La capa de Datos devuelve los resultados en datos puros a la capa **Controladora**.

Genera: La capa **Controladora** genera la capa de **Vista** en base a los tipos de datos generados.

Responde: La **Vista** generada se envía como respuesta a la solicitud del navegador.

MVC con PHP5.



Con lo expuesto se resume las responsabilidades básicas de cada componente del patrón MVC de la siguiente manera:

- El **modelo** es responsable de:
 - Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
 - Define las reglas de negocio.
- El **controlador** es responsable de:
 - Recibe los eventos de entrada.
 - En base a reglas de gestión de eventos se realizan peticiones al modelo o a las vistas.
- Las **vistas** son responsables de:
 - Recibir datos del modelo y los muestran al usuario.
 - Tienen un registro de su controlador asociado (normalmente porque además lo instancia).

Arquitectura de Software

The background features a series of overlapping, semi-transparent green triangles and polygons of various shades, ranging from light lime green to dark forest green. These shapes are primarily concentrated on the right side of the image, creating a dynamic, layered effect. The rest of the background is plain white.

La arquitectura de software de un programa o de un sistema computacional está definida por la estructura, comprendida por los elementos de software, las propiedades visibles de esos elementos y las relaciones entre ellos. (Gomaa, 2011)

Incluyendo:

- La descripción de los componentes con los cuales se construyen los sistemas
- Las interacciones entre esos componentes
- Patrones para guiar la composición
- Restricciones sobre dichos patrones
- Componentes: servidores, clientes, bases de datos, filtros, capas en un sistema jerárquico, etc.
- Interacciones: llamadas a procedimientos, protocolos C/S, protocolos de acceso a BD, etc.

La arquitectura de software se encarga de:

- Diseño preliminar o de alto nivel.
- Organización a alto nivel del sistema, incluyendo aspectos como la descripción y análisis de propiedades relativas a su estructura y control global, los protocolos de comunicación y sincronización utilizados, la distribución física del sistema y sus componentes, etc.
- Otros aspectos relacionados con el desarrollo del sistema y su evolución y adaptación al cambio:

Composición, reconfiguración, reutilización, escalabilidad, mantenibilidad, etc.

Los aspectos que la arquitectura de software no se ocupa:

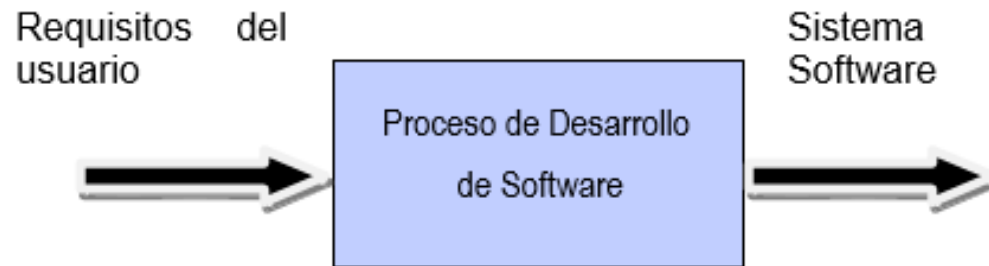
- Diseño detallado.
- Diseño de algoritmos.

Diseño de estructuras de datos

Abstracción de URLs y sesiones.	No es necesario manipular directamente las URLs ni las sesiones, el framework ya se encarga de hacerlo.
Acceso a datos	Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en Base de Datos, XML, etc.
Controladores.	La mayoría de frameworks implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.
Autenticación y control de usuarios de acceso.	Incluyen mecanismos para la identificación de usuarios mediante login y password y permiten restringir el acceso a determinadas páginas a determinados usuarios.
Internacionalización.	
Separación entre diseño y contenido.	

Procesos de Desarrollo

Un proceso de desarrollo de software o llamado ciclo de vida es la estructura básica para el desarrollo del producto software, existes varios modelos a seguir para establecer el proceso de desarrollo de software de manera que éste cumpla los requisitos del cliente tanto en tiempo, costo, alcance y calidad. Este proceso es afectado a las habilidades e ingenio del grupo de trabajo.



Metodología de las tic's de la
Universidad de las Fuerzas Armadas
ESPE extensión Latacunga

Las fases de la Metodología de Desarrollo de Software propuesta que plantea son las siguientes:

- Formalización de la Necesidad
- Requerimientos
- Diseño
- Implementación
- Pruebas
- Puesta en Marcha

La metodología propicia el desarrollo en un entorno en el cual los requerimientos se presentan de forma urgente y en la cual el contacto con el dueño del procesos suele ser dificultoso por cuestiones de la organización y los procesos internos, así como la urgencia de contar con la herramienta de software que ayude a acelerar el trabajo de forma automatizada.

METODOLOGÍA PARA DESARROLLO DE SOFTWARE



Formalización de la Necesidad

La fase de Formalización de la Necesidad constituye el primer proceso de la metodología que se ejecutará por única vez al inicio del proceso metodológico.

Esta fase determina el inicio del proyecto, propiciada por la iniciativa de desarrollo de una aplicación de software como respuesta a una necesidad. Considerando las circunstancias que rodean el entorno en el cual se originan estas necesidades, en esta fase se establece el inicio del proyecto de desarrollo mediante una disposición que es propiciada por la Unidad Operativa Requirente, esta necesidad se modelará mediante un pedido formal que se denominará “Formalización de la Necesidad”. La Formalización de la Necesidad establecerá los responsables y una primera indagación de lo requerido, citando las leyes, reglamentos y procesos establecidos que respalden dicha necesidad.

La Formalización de la Necesidad tendrá como finalidad dar inicio formalmente al proceso de desarrollo de software, comprometer a los actores a aportar en todas las fases requeridas para el desarrollo, e identificar el entorno en el cual se desarrollará y se ejecutará la aplicación de software desarrollada.

En esta fase se establecerá un cronograma de entregas parciales que determinará el alcance del proyecto y de las versiones de software a desarrollarse, estas versiones tendrán autonomía y que en su conjunto conformarán la aplicación de software resultante del proyecto.

Se propicia garantizar los compromisos adquiridos tanto por parte del desarrollador como del requirente.

1. Requerimientos

La fase de Requerimientos es la segunda fase del proceso metodológico, y se constituye como un componente que es parte del proceso repetitivo para el desarrollo de las versiones de software. Esta fase se ejecutará cada vez que se repita el proceso de desarrollo de una nueva versión, en esta fase se detallará la descripción formal de los requerimientos logrando plasmar el alcance del sistema.

Se considera en la metodología que el desarrollo del proyecto se ejecuta en un entorno complejo que dificulta la obtención de los requerimientos y debido al corto tiempo que se establece para el desarrollo de los proyectos se propone la esquematización de historias de usuarios con la variante que el técnico responsable de la elaboración del sistema será el encargado de recopilar y plasmar en el formato la información necesaria para detallar el requerimiento y luego deberá confrontarlo con el usuario responsable del proceso a fin de legalizarlo. Para esta fase deberá aprovechar al máximo el tiempo disponible del personal operativo responsable de los procesos y recopilar la información necesaria aplicando técnicas como:

- Entrevistas al personal involucrado con los procesos a automatizar
- Recopilación de documentos de apoyo y formatos que hayan sido establecidos para el trabajo diario
- Observación de reglamentos e instructivos desarrollados para la ejecución de proyectos que están relacionados con el requerimiento

Los requerimientos describirán de la mejor forma posible la información necesaria para obtener versiones de software de rápida entrega y que contemplen funcionalidades que el usuario final pueda iniciar a ejecutar, los requerimientos serán agrupados y levantados desde dos puntos de vista:

- La necesidad del usuario
- El criterio del técnico que permitirá estructurar paquetes de software de uso autónomo.

En la definición de los requerimientos se establecerá una sección que permita establecer los aspectos necesarios que serán validados en las siguientes fases del desarrollo a fin de establecer si el software desarrollado cumple las expectativas del usuario que lo requiere.

Diseño

La fase de diseño constituye la tercera fase y es parte del proceso iterativo de desarrollo de la aplicación de software, su ejecución se refleja en los modelos producto de los requerimientos establecidos para la versión de software en ejecución.

El proceso de diseño contemplará la esquematización de los requerimientos agrupados por funcionalidades, por consiguiente generará modelos parciales que provean funcionalidad independientes.

La fase de diseño considerará los siguientes aspectos que deberán ser modelados:

- **Diseño de Datos.**- El diseño establecerá el modelo de datos que dará soporte al sistema en desarrollo, se realizará mediante la aplicación de software relacionado directamente con el manejador de base de datos que se haya definido para el proyecto. Por ejemplo para MySQL se usará MySQL Workbench.
- **Diseño de Interfaces.**- El Diseño de interfaces se establecerá de acuerdo al Framework a utilizarse, se realizará un único diseño base para todas las interfaces de mantenimiento de las tablas en la cual se establecerá el posicionamiento para:
 - Texto en el formulario
 - Cajas de entrada de datos (texto, combos, casillas de selección, etc.)
 - Botones de comando y enlaces de funcionalidad
 - Gráficos y logos

- Se personalizará aquellas interfaces que requieran de complejidad en la implementación y que difieren del esquema general de entradas y presentación de datos. El diseño de interfaces determina en forma gráfica mediante esquemas la distribución y operatividad deseada de la interface la que concordará con los requerimientos establecidos.

Implementación

La fase de Implementación es la cuarta fase del proceso iterativo de desarrollo y se enfocará en obtener código de forma rápida, con la mayor fiabilidad posible y con un alto grado de calidad.

Para lograr un estilo homogéneo y eficiente se deberá observar los diseños establecidos de interfaces y se utilizará técnicas de programación que permitan acelerar la generación de código, tal como Frameworks que posibiliten generar código fuente con una arquitectura de software MVC (Modelo Vista Controlador).

El objetivo al integrar el Framework y la arquitectura MVC es permitir un desarrollo rápido de cada módulo y las versiones establecidas en el cronograma de entregas, pasando de la necesidad al producto software de la forma más rápida posible y la entrega al usuario final luego de las pruebas y validaciones necesarias.

La Implementación generará versiones de software que serán autónomos y fácilmente acoplables en un producto final.

El proceso para la generación de código establece seguir varios pasos de forma metódica, los que tomando como insumo los diseños de datos e interfaces generarán el código:

- Creación de Tablas de Datos en el manejador de base de datos
- Uso del generador de código
- Personalización del Modelo
- Personalización de la Vista
- Personalización del Controlador
- Pruebas Unitarias (Funcionalidad del código)

Para la implementación, así como para el resto de fases se recomienda el uso de herramientas de software dentro del marco que regula a las instituciones públicas. El decreto 1014 recomienda el uso del Software Libre, y en este marco podemos usar herramientas de desarrollo como NetBeans, así como herramientas de gestión de versiones como SubVersion (SVN)

Pruebas

La fase Pruebas es la quinta fase y es parte del proceso iterativo de desarrollo. Una vez finalizada la generación del código se realizarán pruebas del software a fin de garantizar la calidad mediante la concordancia entre los requerimientos, el diseño y el producto resultante.

Las pruebas se establecerán en 2 puntos a fin de minimizar los fallos residuales en el software:

- **Pruebas de Desarrollo.-** Las realizará el técnico a cargo del desarrollo y se basa en examinar de forma detallada los objetos creados y su funcionalidad. Propicia la eliminación de errores a nivel de código la funcionalidad individual del módulo.
- **Pruebas de Función.-** Las realizará un técnico a fin encontrar residuos de fallos que no han sido detectados y también establecerá la correcta relación entre los módulos desde el punto de vista funcional del software, de esta manera se asegura que la versión de software sea un elemento entregable al usuario final. En esta evaluación se contrapondrá los requerimientos establecidos para la versión de software contra el desempeño de la aplicación desarrollada.

Conseguiremos entonces una versión que pasará a estado de producción y podrá ser implantada.

Puesta en Marcha

La Fase de puesta en marcha se ejecutará al finalizar cada una de las versiones de software planificadas. La ejecución de esta fase requiere la puesta en marcha de la versión de software, realizando paralelamente la capacitación sobre las funcionalidades incluidas en la determinada versión.

Se orientará al usuario final a utilizar las funcionalidades implementadas y se determinará el grado de satisfacción sobre los requerimientos.

Se realizará un Acta de Conformidad en la cual se indicará las funcionalidades implementadas en la versión y se registrarán las novedades suscitadas en la capacitación.

Mientras se avanza en el proceso de Puesta en Marcha por parte de un técnico, el grupo de trabajo iniciará inmediatamente a trabajar en la nueva versión.

Los fallos residuales detectados en la Puesta en Marcha serán incluidos como parte de la nueva versión a fin de viabilizar de forma fluida el desarrollo del software.

Artefactos y Manuales

Para lograr la operatividad de la metodología se plantea un conjunto de artefactos de apoyo, que permitirán documentar el proceso de desarrollo e integrar el esfuerzo de las personas involucradas en el proyecto. Estos artefactos se establecerán en cada una de las fases de la metodología.

Se requiere además de los formatos de base para los manuales que estarán a disposición del cliente o usuario final para poder implantar el sistema o capacitar a nuevos funcionarios que no fueron parte del grupo de personal capacitado durante la ejecución del proyecto.

Para poder organizar los entregables que se definen en la metodología se ha establecido un árbol de directorios que contendrán cada uno de los entregables de forma organizada y nos permitirán que todos los miembros del equipo de desarrollo sean capaces de encontrar fácilmente los artefactos que han desarrollado otros colaboradores.

- Raíz [Repositorio de Proyectos de Desarrollo]
 - Proyecto [Nombre Corto del Proyecto]
 - Entregables [Documentos de Texto y Modelos]
 - Código [Versiones de Código Fuente]
 - Sistema Versión 1.0 [Primera Versión]
 - Sistema Versión 2.0 [Segunda Versión]
 - Manuales [Versiones de Manuales]

Una vez establecido el repositorio para los artefactos que serán desarrollados como producto de la metodología procederemos a detallar cada uno de ellos de acuerdo a la fase en la cual se elaborarán.

Plantilla:**“FORMALIZACIÓN DE DESARROLLO DE SOFTWARE”****Función de la Plantilla:**

Formalizar el pedido de desarrollo de software, logrando la identificación de las entidades participantes en el proceso de desarrollo, y realizando acuerdos que permitirán una adecuada planificación y garanticen la participación activa tanto del grupo de desarrollo como del requirente.

Objetivos de la Plantilla:

- Formalizar la necesidad presentada mediante un pedido verbal o mediante una disposición documentada.
- Identificar las personas que participarán en el proceso de desarrollo tanto de técnicos como de funcionarios requirentes.
- Comprometer la participación del grupo de desarrollo y del requirente mediante el suministro de información y la entrega en los plazos establecidos del producto de software.

Alcance de la Plantilla:

Esta plantilla se usará al inicio del proyecto para formalizar el pedido.

Notas:

Esta plantilla se guardará como un archivo en Formato de Texto (MS Word 2007-2010) y su nombre seguirá la nomenclatura siguiente:
Formalizacion.[nombre_de_proyecto].v[número_version(0.0)].docx

Versiones de la Plantilla:

Versión:	Fecha:	Responsable:
1.0	10-noviembre-2014	Ing. Edgar Montaluisa Ing. Ximena López

FORMALIZACIÓN DE DESARROLLO DE SOFTWARE

Proyecto:

[Nombre del Proyecto]

Fecha de Reunión:

[Fecha de Reunión entre las partes]

Documento de Requerimiento:

[No. de documento de requerimiento]

Unidad Requirente:

[Nombre de la Unidad Requirente]

Integrantes Unidad Requirente:

[Personal que participará directamente en el proyecto]

Unidad Desarrollo:

[Nombre de la Unidad Desarrollo]

Integrante Grupo de Desarrollo:

[Personal del Grupo de Desarrollo que participará directamente en el proyecto]

Descripción:

[Relato detallado del requerimiento, que permita establecer el objetivo principal que se persigue con su desarrollo; No incluye requisito solo una descripción global del proyecto a desarrollarse; Debe ser lo suficientemente claro para establecer el alcance global y las iteraciones que serán necesarias para la implementación]

Nombre del Responsable

Nombre del Líder del Proyecto

RESPONSABLE UNIDAD REQUIRENTE

RESPONSABLE DEL GRUPO DE DESARROLLO

Plantilla:**“PLANIFICACIÓN DE ENTREGAS”****Función de la Plantilla:**

Esta plantilla establecerá los tiempos en los cuales se realizará la entrega de las versiones de software planificadas en el desarrollo.

Objetivos de la Plantilla:

- Especificar los tiempos estimados en los cuales se entregarán las versiones de software.
- Preparar a la Unidad Requirente para el proceso de inducción y organizar su trabajo para dar tiempo a la capacitación del nuevo sistema.
- Organizar el trabajo del personal técnico para el cumplimiento de los tiempos estimados en el desarrollo de software.

Alcance de la Plantilla:

Esta plantilla se usará luego de haber formalizado el desarrollo de software y previo al proceso iterativo de desarrollo de la aplicación.

Notas:

Esta plantilla se guardará como un archivo en Formato de Texto MS Word 2007-2010) y su nombre seguirá la nomenclatura siguiente:

Planificacion.[nombre_de_proyecto].v[número_version(0.0)].docx

Versiones de la Plantilla:			
Versión:	Fecha:	Responsable:	
1.0	10-noviembre-2014	Ing. Edgar Montaluisa Ing. Ximena López	
PLANIFICACIÓN DE ENTREGAS			
Proyecto: [Nombre del Proyecto]			
Fecha Elaboración: [Fecha de Elaboración del Documento]		Tiempo Total Entrega: [Tiempo Total de Desarrollo en días]	
Cronograma de Entrega de Versiones:			
Versión:	Tiempo Desarrollo	Fecha Entrega:	Resumen de la Entrega:
[#Versión]	[En días]	[Fecha]	[Breve descripción de la Funcionalidad a entregar en la versión]
Nombre del Responsable RESPONSABLE UNIDAD REQUIRENTE		Nombre del Líder del Proyecto RESPONSABLE DEL GRUPO DE DESARROLLO	

Plantilla:**“REQUISITOS ESPECÍFICOS”****Función de la Plantilla:**

Esta plantilla permitirá realizar el levantamiento de los requisitos funcionales detallando de la mejor manera cada uno de los requerimientos del usuario.

Objetivos de la Plantilla:

- Especificar los requisitos funcionales de cada una de las versiones a desarrollar de la aplicación.
- Definir las pruebas a realizarse como comprobación del software realizado, a fin de determinar el grado de implementación del requisito en el sistema.

Alcance de la Plantilla:

Esta plantilla se usará de forma iterativa por cada ciclo que se realice en el desarrollo de una versión específica y será el primero en ser diseñado a fin de proporcionar los parámetros para el desarrollo de la aplicación en las siguientes fases.

Notas:

Esta plantilla se guardará como un archivo en Formato de Texto (MS Word 2007-2010) y su nombre seguirá la nomenclatura:

Requisitos.[nombre_de_proyecto].v[número_version(0.0)].docx

Versiones de la Plantilla:

Versión:	Fecha:	Responsable:
1.0	10-noviembre-2014	Ing. Edgar Montaluisa Ing. Ximena López

Plantilla:

"DISEÑO DE INTERFACES"

Función de la Plantilla:

Esta plantilla detallará la ubicación básica de los componentes en las interfaces a fin de mantener una armonía en el diseño. Los procesos que son recurrentes se modelarán una sola vez, dando la opción de modelar de forma detallada solo las interfaces que requieren mayor complejidad.

Objetivos de la Plantilla:

- Especificar el diseño de las interfaces a utilizar, usando diagramas de bloques indicando el **posicionamiento** de elementos como texto, cajas de entrada de datos, cajas de selección, imágenes, etc.
- Detallar aquellas interfaces que requieran mayor complejidad de interpretación.
- Guardar armonía en el diseño de las interfaces

Alcance de la Plantilla:

Esta plantilla se usará de forma iterativa en cada ciclo que se realice en el desarrollo de todas las versiones. Se diseñará luego de identificar si el requisito requiere un nuevo diseño de los anteriores ya existentes. El diseño es general y no específico a cada requerimiento.

Notas:

Esta plantilla se guardará como un archivo en Formato de Texto (MS Word 2007-2010) y su nombre seguirá la nomenclatura: Interfaces. [nombre_de_proyecto]. v[número_version(0.0)].docx

Versiones de la Plantilla

Versión:	Fecha:	Responsable:
1.0	10-noviembre-2014	Ing. Edgar Montaluisa Ing. Ximena López

DISEÑO DE INTERFACES

Proyecto:

[Nombre del Proyecto]

Fecha Elaboración:

[Fecha de Elaboración del Documento]

Técnico:

[Nombre del técnico que propuso el uso de la interface]

Título:

[Nombre que resumen la funcionalidad de la interface]

Descripción:

[Relato que detalla la concepción de la interface, especificando en qué casos se debe utilizar este diseño. Se acompañará con una imagen que represente en Bloques el diseño]

Nombre del Técnico Responsable

TÉCNICO RESPONSABLE DEL DESARROLLO

Plantilla:

“DISEÑO DE BASE DE DATOS”

Función de la Plantilla:

Detallar la funcionalidad de la base de datos que soportará la versión correspondiente de software desarrollado.

Objetivos de la Plantilla:

- Especificar el modelo de datos que soportará la implementación de una versión completa desarrollada.
- Especificar los requerimientos que se han considerado en la versión de la Base de Datos.
- Permitir la trazabilidad en la evolución de la Base de Datos

Alcance de la Plantilla:

Esta plantilla se elaborará como resumen del modelamiento de datos, indicando los requisitos implementados y la versión final de la base de datos a ser entregada en la versión del Software.

Notas:

Esta plantilla se guardará como un archivo en Formato de Texto (MS Word 2007-2010) y su nombre seguirá la nomenclatura siguiente:
Base.[nombre_de_proyecto].v[número_version(0.0)].docx

Versiones de la Plantilla:

Versión:	Fecha:	Responsable:
1.0	10-noviembre-2014	Ing. Edgar Montaluisa Ing. Ximena López

DISEÑO DE BASE DE DATOS

Proyecto:

[Nombre del Proyecto]

Fecha Elaboración:

[Fecha de Elaboración del Documento]

Versión del Software:

[Número de versión del software a la cual da soporte la Base de Datos]

Requisitos Implementados:

[ID de los Requisitos implementados en la base de datos]

Técnico:

[Nombre del técnico que realizó el modelo de la Base de Datos]

Descripción:

[Relato que detalla las funcionalidades a las cuales da soporte la Base de Datos, especificando las implementaciones de la nueva versión. Será necesario anexar la referencia del archivo de modelo de ser necesario, usando el mismo nombre que este documento]

Nombre del Técnico Responsable

TÉCNICO RESPONSABLE DEL DESARROLLO

Plantilla:**“VERSIÓN DE SOFTWARE”****Función de la Plantilla:**

Detallar la funcionalidad que se ha implementado en una versión específica de software, especificando la operatividad y las correcciones realizadas sobre la versión anterior.

Objetivos de la Plantilla.

- Detallar las nuevas funcionalidades de la versión desarrollada.
- Detallar las correcciones realizadas sobre una versión anterior.
- Definir el entorno necesario para la puesta en marcha de la versión de software desarrollada.

Alcance de la Plantilla:

Esta plantilla se elaborará como resumen de la implementación de cada versión de software desarrollada, recopilando las funcionalidades y requerimientos implementados.

Notas:

Esta plantilla se guardará como un archivo en Formato de Texto (MS Word 2007-2010) y su nombre seguirá la nomenclatura siguiente: Software.[nombre_de_proyecto].v[número_version(0.0)].docx

Versiones de la Plantilla:

Versión:	Fecha:	Responsable:
1.0	10-noviembre-2014	Ing. Edgar Montaluisa Ing. Ximena López

VERSIÓN DE SOFTWARE

Proyecto:

[Nombre del Proyecto]

Fecha Elaboración:

[Fecha de Elaboración del Documento]

Versión del Software:

[Número de versión del software]

Requisitos Implementados:

[ID de los requisitos que componen esta nueva versión de software]

Requisitos Corregidos:

[ID de los requisitos revisados en esta nueva versión de software]

Descripción:

[Relato que detalla las funcionalidades implementadas en la versión de software desarrollado. Descripción de los requerimientos técnicos y operativos para la implantación de la versión. Detallará el repositorio en el cual se encontrará la versión del código fuente de la aplicación]

Nombre del Técnico Responsable

TÉCNICO RESPONSABLE DEL DESARROLLO

Plantilla:**“PRUEBAS DE SISTEMA”****Función de la Plantilla:**

La función de esta plantilla es detallar los fallos detectados durante la revisión de funcionalidad del software y los resultados de la retroalimentación hacia el grupo de desarrollo, así como las soluciones establecidas.

Objetivos de la Plantilla:

- Detallar los fallos detectados sobre la versión final desarrollada.
- Retroalimentar las soluciones aplicadas para solventar los fallos detectados.
- Establecer el nivel de cumplimiento del proceso de desarrollo de software y la obtención de una versión de software de calidad.

Alcance de la Plantilla:

Esta plantilla se elaborará al final de haber realizado las pruebas de funcionalidad del sistema y deberá mostrar un balance positivo sobre los fallos detectados y solucionados.

Notas:

Esta plantilla se guardará como un archivo en Formato de Texto (MS Word 2007-2010) y su nombre seguirá la nomenclatura siguiente: Pruebas.[nombre_de_proyecto].v[número_version(0.0)].docx

Versiones de la Plantilla:

Versión:	Fecha:	Responsable:
1.0	10-noviembre-2014	Ing. Edgar Montaluisa Ing. Ximena López

PRUEBAS DE SISTEMA

Proyecto:

[Nombre del Proyecto]

Fecha Elaboración:

[Fecha de Elaboración del Documento]

Versión del Software:

[Número de versión del software]

Requisitos Validados:

[ID de los requisitos que componen esta nueva versión de software y que han sido validados]

Técnico Revisor:

[Nombre y Firma del Técnico encargado de la Revisión del software]

Tiempo de Revisión:

[Tiempo en número de horas que se utilizaron para la revisión]

Tiempo de Soluciones:

[Tiempo reportado por el Grupo de Desarrollo sobre los fallos reportados]

Descripción de Fallos:

[Relato que detalla los fallos encontrados y como este afecta al desempeño del sistema, las acciones tomadas para solucionarlas y el estado actual del sistema revisado]

No.

[Secuencial de Fallos]

Fallo:

[Descripción del Fallo Reportado]

Solución:

[Descripción de la Solución Aplicada por el grupo de desarrollo]

Nombre del Técnico Responsable

TÉCNICO RESPONSABLE DEL DESARROLLO

Plantilla:

“ENTREGA DEL SISTEMA”

Función de la Plantilla:

Detallar los aspectos más importantes del proceso de entrega de la versión, indicando al requirente la operatividad implementada en esta versión del software y contando con la aceptación del mismo.

Objetivos de la Plantilla:

- Detallar las funcionalidades entregadas en la versión del software.
- Detallar las novedades de capacitación, tales como: personal capacitado y novedades de la implantación.
- Propiciar la continuidad del desarrollo del proyecto, evidenciado la aceptación por parte del responsable del proceso y del técnico responsable del proyecto.

Alcance de la Plantilla:

Esta plantilla se elaborará al final del ciclo de desarrollo de la versión de software, y en forma posterior al proceso de puesta en marcha y capacitación. De esta manera se finaliza el proceso de desarrollo de la versión y propicia la nueva iteración.

Notas:

Esta plantilla se guardará como un archivo en Formato de Texto (MS Word 2007-2010) su nombre seguirá la nomenclatura siguiente: Entrega.[nombre_de_proyecto].v[número_version(0.0)].docx

Versiones de la Plantilla:

Versión:	Fecha:	Responsable:
1.0	10-noviembre-2014	Ing. Edgar Montaluisa Ing. Ximena López

ENTREGA DEL SISTEMA

Proyecto:

[Nombre del Proyecto]

Fecha Elaboración:

[Fecha de Elaboración del Documento]

Versión del Software:

[Número de versión del software]

Personal Capacitado:

[Nombre de las personas y número de horas usadas en la inducción de la nueva versión del software, sumilla del Personal Capacitado]

Técnico Capacitador:

[Nombre del técnico encargado de la implantación y Capacitación]

Novedades de Puesta en Marcha y la Capacitación:[Detalle de las novedades suscitadas en la capacitación y el estado de ejecución del sistema por parte de los capacitados]

Detalle de la Versión:[Funcionalidades implantadas en la versión de Software]

Nombre del Responsable
RESPONSABLE UNIDAD REQUIRENTE

Nombre del Líder del Proyecto
RESPONSABLE DEL GRUPO DE DESARROLLO

DESARROLLO DEL PRODUCTO SOFTWARE

The background features abstract, overlapping geometric shapes in various shades of green, ranging from light lime to dark forest green. The shapes are primarily triangles and polygons, creating a dynamic, layered effect. The overall composition is clean and modern, with the text centered on the left side of the frame.

Formalización de la Necesidad

Mediante reuniones de trabajo y observación directa, se establece el compromiso para el desarrollo de este producto software en la metodología detallada en el capítulo anterior y en cumplimiento de la legislación, procedimiento y artefactos de desarrollo se obtiene el artefacto de formulación de software para el sistema de control de graduados que podemos evidenciar en Anexo 1.

Requerimientos

Mediante reuniones de trabajo y observación directa, se establece los requisitos para el desarrollo de este producto software y cumpliendo con la metodología se aplica los artefactos que se encuentran en Anexo 2.

Diseño

Mediante reuniones de trabajo y observación directa, se establece los requisitos para el desarrollo de este producto software y cumpliendo con la metodología se aplica los artefactos que se encuentran en Anexo 3.

Implementación

Mediante reuniones de trabajo y observación directa, se establece los requisitos para el desarrollo de este producto software y cumpliendo con la metodología se aplica los artefactos que se encuentran en Anexo 4.

Pruebas

Mediante reuniones de trabajo y observación directa, se establece los requisitos para el desarrollo de este producto software y cumpliendo con la metodología se aplica los artefactos que se encuentran en Anexo 5.

Liberación

Mediante reuniones de trabajo y observación directa, se establecen los requisitos para el desarrollo de este producto software y cumpliendo con la metodología se aplica los artefactos que se encuentran en Anexo 6.

CONCLUSIONES

The background features abstract, overlapping geometric shapes in various shades of green, ranging from light lime to dark forest green. These shapes are primarily located on the right side of the slide, creating a modern, layered effect. The rest of the slide is a plain white background.

- Se cumplió el objetivo del proyecto de crear un sistema para la administración de Graduados que nace de la necesidad de contar con datos reales para estadísticas requeridas en el departamento de Electrónica y Eléctrica.
- En el presente tema investigativo el uso del patrón MVC mediante el framework Yii permitió optimizar el tiempo de desarrollo utilizando métodos propios del framework.
- Con la metodología de las TIC's de la Universidad de las Fuerzas Armadas ESPE extensión Latacunga, se mejora el tiempo de diseño y desarrollo porque es una metodología ágil y se puede usar en cualquier proyecto de éstas características.
- Se aplicaron cada una de las fases de desarrollo la cual requiere de la documentación de cada uno de los artefactos que intervinieron en el desarrollo, se llevaron los lineamientos de programación y base de datos garantizando el cumplimiento de los mismos.
- La documentación de cada uno de los artefactos usados en la metodología de las TIC'S, permite que se pueda realizar a futuros mejoras, modificaciones, crear nuevas opciones según crezcan las necesidades de los usuarios del sistema.

RECOMENDACIONES

The background features abstract, overlapping geometric shapes in various shades of green, ranging from light lime to dark forest green. These shapes are primarily located on the right side of the page, creating a modern, layered effect. The rest of the page is plain white.

- Se recomienda usar software libre para desarrollos de este tipo de proyectos, ya que cualquier persona puede usar bajo condiciones de la licencia, es de bajo costo y existe libertad de conocimiento y trabajo cooperativo a través de blogs interactivos en la web.
- Se sugiere aplicar la metodología de las TIC's de la Universidad de las Fuerzas Armadas ESPE extensión Latacunga por ser una metodología ágil y se puede usar en cualquier proyecto de estas características.
- Se recomienda tener un repositorio único de código fuente para evitar que se sobrescriban archivos.
- Se recomienda tener un servidor de compilación continua, de ésta forma garantizar las publicaciones de versiones con todas las clases, servicios y dll actualizadas.
- Se recomienda el uso de base de datos libre como MySQL o Postgresql ya que es posible desarrollar propios tipos de datos, permite la innovación y libertad en la programación, velocidad al realizar las operaciones lo que hace uno de los gestores con mejor rendimiento, no se requiere de un equipo de costosas características . Su conectividad, velocidad y seguridad hacen que MySQL Server sea muy apropiado para acceder bases de datos en Internet.

- Se recomienda contratar un certificado de seguridad SSL (Secure Socket Layer) con la finalidad de complementar la seguridad de la aplicación en la transferencia de datos entre el navegador y el servidor Web.
- Se recomienda actualizar la malla de formación de tal forma que los estudiantes perciban y vivan una experiencia real del desarrollo de software en ingeniería y con buenas prácticas.

Muchas gracias por su atención

Preguntas?