**IDENTIFICACIÓN**



|  |  |
| --- | --- |
| **Versión** | [1.0] |
| **Proyecto** | [Sitio web Eddy Burguer] |
| **Fecha** | [27/05/2017] |
| **Elaborado por** | [Miguel Ángel Franco Ramírez] |
| **Localización del Documento** | [Github] |
| **Documento base** |  |
|  |  |
| **Autorizaciones** | |
| **Nombre** | **Fecha Autorización** |
| Miguel Ángel Franco Ramírez | 01/06/2017 |
| Alejandro Onofre Cornejo | 01/06/2017 |
|  |  |
|  |  |
| **Distribución** | |
| **Nombre** | **Fecha Recepción** |
| Alejandro Onofre Cornejo | 01/06/2017 |
|  |  |
|  |  |

**Control de cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Control de Cambios** | | | |
| **Versión** | **Fecha** | **Participante** | **Descripción del Cambio** |
| 1.0 | 27/05/2017 | Miguel Ángel Franco Ramírez | Recién creado |
|  |  |  |  |
|  |  |  |  |



***SITIO WEB EDDY BURGUER***

Versión: [ 1.0 ]

Elaborado por: [Miguel Ángel Franco Ramírez]

Fecha de revisión: [01/05/2017]

**CONTROL DE CAMBIOS**



|  |  |
| --- | --- |
| **Id. Proyecto** | EB\_001 |
| **Nombre del Proyecto** | Sitio web Eddy Burguer |
| **Fecha** | 27/05/2017 |
| **Elaborado por** | Miguel Ángel Franco Ramírez |
| **Localización del Documento** | Github |

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

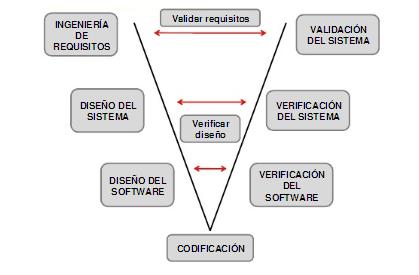
**CICLO DE VIDA A IMPLEMENTAR**

**Ciclo de Vida en V**

El ciclo de vida en V será el que se va a implementar en el desarrollo del proyecto “sitio web Eddy Burguer”, este ciclo de vida deriva directamente del modelo en cascada, y se usa como base de procesos dentro del ciclo de vida de software. El modelo considera el testing como una actividad paralela al SDLC (Software development Life Cycle) y no como una actividad aislada que se realiza al final del desarrollo. La ventaja principal con respecto al modelo en cascada es simple, ya que este modelo involucra revisiones de cada una de las etapas del modelo de cascada.

Se eligió el ciclo de vida en V, debido a que, este es adecuado para desarrollar proyectos de aplicaciones web sencillas, pero que estas necesitan una fiabilidad muy alta, esto se logra gracias a que las pruebas de cada fase ayudan a corregir posibles errores sin tener que esperar a que sean rectificados en la etapa final del proceso.

Por otro lado, con las pruebas unitarias y de integración podemos conseguir y obtener una exactitud en la aplicación que se está desarrollando, esto nos va a ayudar en revisar los documentos de las actividades y verificar si estas cumplen con los estándares y normas de calidad que se eligieron para el desarrollo del presente proyecto.



**Plan preliminar**

Generar el WBS del proyecto, con las actividades que se desarrollarán durante la realización del mismo, también se da inicio formal al proyecto mediante un acta de inicio.

**Análisis**

Se identificarán los requerimientos funcionales y los requerimientos no funcionales que contendrá el sistema, después se estos serán evaluados para saber si agregan o no valor al proyecto.

**Planificación**

Se establecerá las actividades de cada integrante del equipo de trabajo también se analizará los riesgos que puedan presentarse a lo largo del proyecto y también establecerá una estimación aproximada. Al igual, se planifica cómo será la administración de los documentos generados en cada una de las etapas del proyecto.

**Diseño**

Se diseñará la arquitectura que el sistema tendrá, también se construirán los diagramas UML que ayudarán a los programadores a codificar, al igual, serán construidos los diagramas para el desarrollo de la base de datos.

**Codificación**

Se programará el sistema con las especificaciones de la fase anterior, codificando los requerimientos funcionales y los requerimientos no funcionales ya descritos anteriormente.

**Pruebas**

Se harán las pruebas necesarias para comprobar el funcionamiento correcto del sistema, dichas pruebas deben cumplir con las especificaciones descritas por el cliente además las se tendrá dos tipos de pruebas unitarias y de modulo.

**Implementación**

Una vez finalizado el sistema, este se debe de poner en marcha para que el cliente pueda usarlo, realizando una vez más pruebas al software para verificar que no haya anomalías, por otro lado, se desarrollarán manuales en los cuales se describe como es la función de sistemas y como debe usarse.

**Cierre del proyecto**

Una vez implementado el sistema se realizan los documentos necesarios que avalen que el proyecto ha sido terminado y entregado en tiempo y forma al cliente. También se realiza se realiza una reunión al culminar con el proyecto para saber qué fue lo que salió bien y lo que salió mal así con estos dos puntos al desarrollar otro proyecto se sabrá que se tiene que hacer y que no hacer.

## **CICLOS DE VIDA APLICABLES**

## **Ciclo de vida en v**

* **Ventajas**
* La relación entre las etapas de desarrollo y los distintos tipos de pruebas facilitan la localización de fallos.
* Es un modelo sencillo y de fácil aprendizaje.
* Hace explícito parte de la iteración y trabajo que hay que revisar.
* Especifica bien los roles de los distintos tipos de pruebas a realizar.
* Involucra al usuario en las pruebas.
* **Desventajas**
* Es difícil que el cliente exponga explícitamente todos los requisitos.
* El cliente debe tener paciencia pues obtendrá el producto al final del ciclo de vida.
* Las pruebas pueden ser caras y, a veces, no lo suficientemente efectivas.
* El producto final obtenido puede que no refleje todos los requisitos del usuario.

## **Ciclo de vida en cascada**

* **Ventajas**
* Es un modelo sencillo y disciplinado.
* Es fácil aprender a utilizarlo y comprender su funcionamiento.
* Está dirigido por los tipos de documentos y resultados que deben obtenerse al final de cada etapa.
* Ha sido muy usado y, por tanto, está ampliamente contrastado.
* Ayuda a detectar errores en las primeras etapas a bajo costo.
* Ayuda a minimizar los gastos de planificación.
* **Desventajas**
* Los proyectos raramente siguen el proceso lineal tal como se definía originalmente el ciclo de vida.
* Es difícil que el cliente exponga explícitamente todos los requisitos al principio.
* El cliente debe tener paciencia pues obtendrá el producto al final del ciclo de vida.
* No refleja exactamente cómo se programa realmente el sistema, en el que suele haber un gran componente iterativo.
* Puede resultar complicado regresar a etapas anteriores (ya acabadas) para realizar correcciones.
* El producto final obtenido puede que no refleje todos los requisitos del usuario.

## **Ciclo de vida lineal**

* **Ventajas**
* Suministra una plantilla en la que pueden colocarse los métodos para cada una de las fases.
* Pasos similares a los pasos genéricos.
* Modelo procedimental más utilizado.
* **Desventajas**
* Gran énfasis en la producción de documentos completamente elaborados, producto de las fases de análisis y especificación de requerimientos y de diseño.
* No muy aplicable a productos de software altamente interactivos.
* Es difícil tener todos los requerimientos, bien definidos al principio, como lo requiere el modelo y además presenta dificultades para acomodar posibles incertidumbres existentes al comienzo de los proyectos.
* Los productos de software raramente siguen el flujo secuencial que propone el modelo. Siempre hay iteraciones y se crean problemas en la aplicación del paradigma.
* Un error importante no detectado al principio puede ser desastroso.
* Se requiere mucha paciencia por parte del cliente, porque solo hasta las etapas finales del desarrollo podrá tener una versión operativa del producto.