**Identificación**



|  |  |
| --- | --- |
| **Versión** | 1.0 |
| **Proyecto** | [Sitio web Sexy Lady] |
| **Fecha** | [29/05/2017] |
| **Elaborado por** | [Lilia Isabel Rojas Noguéz] |
| **Localización del Documento** | [GitHub] |
| **Documento base** |  |
|  |  |
| **Autorizaciones** | |
| **Nombre** | **Fecha Autorización** |
| Lilia Isabel Rojas Noguéz | 01/06/2017 |
| María Guadalupe Caballero Cruz | 01/06/2017 |
|  |  |
|  |  |
| **Distribución** | |
| **Nombre** | **Fecha Recepción** |
| María Guadalupe Caballero Cruz | 01/06/2017 |
|  |  |
|  |  |

**Control de cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Control de Cambios** | | | |
| **Versión** | **Fecha** | **Participante** | **Descripción del Cambio** |
| 1.0 | 29/05/2017 | Lilia Isabel Rojas Noguéz | Recién creado |
|  |  |  |  |
|  |  |  |  |



***Esquema del ciclo de vida***

Versión: [1.0]

Elaborado por: [Lilia Isabel Rojas Noguéz]

Fecha de revisión: [1/06/2017]

**Control de cambios**



|  |  |
| --- | --- |
| **Id. Proyecto** | SL\_001 |
| **Nombre del Proyecto** | Sitio web Sexy Lady |
| **Fecha** | 29/05/2017 |
| **Elaborado por** | Lilia Isabel Rojas Noguéz |
| **Localización del Documento** | GitHub |

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

**CICLOS DE VIDA APLICABLES AL PROYECTO**

**En cascada**

Este es el más básico de todos los modelos, y sirve como bloque de construcción para los demás modelos de ciclo de vida. La visión del modelo cascada del desarrollo de software es muy simple; dice que el desarrollo de software puede ser a través de una secuencia simple de fases. La flecha de avance muestra el flujo normal. Las flechas hacia atrás representan la retroalimentación.

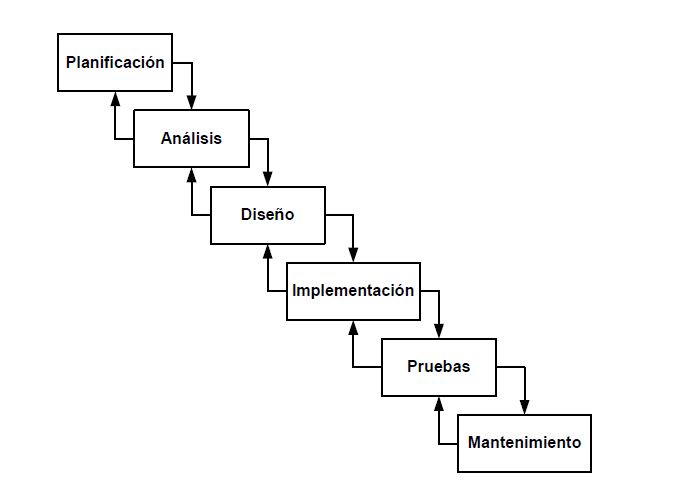
“Se inicia con la especificación de requerimientos del cliente y continúa con la planeación, el modelado, la construcción y el despliegue para culminar en el soporte del software terminado” (Pressman, 2006).

**Ventajas**

* Planear un proyecto antes de embarcarse en él.
* Documentar los resultados de cada actividad.
* Diseñar un sistema antes de codificarlo.

**Desventajas**

* En la vida real, un proyecto rara vez sigue una secuencia lineal, esto crea una mala implementación del modelo, lo cual hace que lo lleve al fracaso.
* El proceso de creación del software tarda mucho tiempo ya que debe pasar por el proceso de prueba y hasta que el software no esté completo no se opera. Esto es la base para que funcione bien.



**Modelo v**

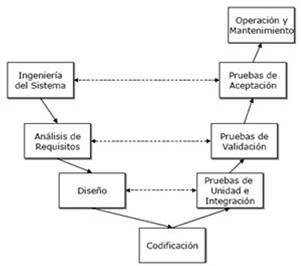
El modelo en V es una variación del modelo en cascada que muestra cómo se relacionan las actividades de prueba con el análisis y el diseño. El modelo en V se centra en las actividades y la corrección.

La unión mediante líneas discontinuas entre las fases de la parte izquierda y las pruebas de la derecha representa una doble información. Por un lado sirve para indicar en qué fase de desarrollo se deben definir las pruebas correspondientes. Por otro sirve para saber a qué fase de desarrollo hay que volver si se encuentran fallos en las pruebas correspondientes.

Ventajas

* La relación entre las etapas de desarrollo y los distintos tipos de pruebas facilitan la localización de fallos.
* 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
* 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

**Prototipos**

Un prototipo puede ser un modelo del ciclo de vida del Software, tal como desarrollo en el desarrollo en cascada. Éstos permiten testar el objeto antes de que entre en producción, detectar errores, deficiencias, etcétera. Cuando el prototipo está suficientemente perfeccionado en todos los sentidos requeridos y alcanza las metas para las que fue pensado, el objeto puede empezar a producirse.

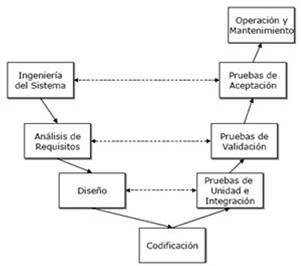
Ventajas

* Permite la construcción del sistema con requisitos poco claros o cambiantes
* El cliente recibe una versión del sistema en muy poco tiempo, por lo que lo puede evaluar, probar e, incluso, empezar a utilizarlo
* Se pueden introducir cambios en las funcionalidades del sistema en cualquier momento
* Involucra al usuario en la evaluación de la interfaz de usuario
* Se reduce el riesgo y la incertidumbre sobre el desarrollo
* Genera signos visibles de progreso, que se utilizan cuando existe una demanda en la velocidad del desarrollo
* Permite entender bien el problema antes de la implementación final

Desventajas

* Requiere trabajo del cliente para evaluar los distintos prototipos y traducirlo en nuevos requisitos
* Requiere un tiempo adicional para definir adecuadamente el sistema
* No se sabe exactamente cuánto será el tiempo de desarrollo ni cuantos prototipos se tienen que desarrollar
* Si un prototipo fracasa, el coste del proyecto puede resultar muy caro

**CICLO DE VIDA A IMPLEMENTAR**

**Modelo v**

Para el desarrollo del sitio web se decidió utilizar el modelo de ciclo de vida en V, algunas de las razones por las cuales se eligió esta metodología fueron:

El modelo en V brinda las pautas necesarias que nos permiten una correcta organización en el desarrollo del software a través de la implementación y verificación de cada una de sus etapas.

La figura anterior muestra el Modelo en V, que representa las relaciones temporales entre las distintas fases de este ciclo de vida a utilizar para el desarrollo del proyecto Sexy Lady, y en cada etapa o fase se tendrán las actividades correspondientes al proyecto junto con las pruebas necesarias para culminar la etapa de una manera más óptima, puesto que estas nos ayudan a encontrar y corregir posibles fallas o errores sin tener que esperar a que la etapa llegue final del proceso.

Por otra parte, las pruebas que se deben ejecutar en cada etapa nos sirven para verificar y validar la calidad de los procesos que se realizan dentro del proyecto ya que en conjunto con los responsables de realizar dichas pruebas, nos proporcionan mayor fiabilidad en los productos que se desarrollan.

Cabe mencionar que esta metodología es ideal, por su fácil entendimiento; puesto que está orientado para proyectos pequeños y además se puede implementar en proyectos con equipos de desarrollo desde una hasta cinco personas, en este caso nuestro equipo consta de 3 integrantes.

Las etapas de este ciclo de vida son:

Plan preliminar: En el cual se encuentra la mejora de los procesos que se realizan durante el proyecto.

Análisis: En esta fase se analizan las necesidades (requerimientos) de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge un documento llamado SRS (documento de especificación de requisitos).

Planificación: En esta etapa se realiza la factibilidad del proyecto para ver si el proyecto es viable realizarlo y además de la estimación de costos.

Diseño: Se plasman las características que llevara el software y se empieza a diseñar y a realizar el proyecto de acuerdo a las especificaciones que se obtuvieron.

Codificación: Es la fase en donde se implementa el código fuente, haciendo uso de prototipos así como de pruebas y ensayos para corregir errores.

Pruebas: Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente y que cumple con los requisitos, antes de ser entregado al usuario final.

Implementación: Es la fase en donde el usuario final ejecuta el sistema.

Cierre del proyecto

Se realizan los documentos necesarios para terminar el proyecto de manera formal, entregado en tiempo y con las especificaciones del cliente.