Unidad 1 / Escenario 2

Lectura Fundamental

Modelos existentes

Contenido

Palabras clave:
SSDLC, OWASP, cascada, NIST, OPENSAMM.

Introducción

Con el entendimiento de los conceptos básicos de seguridad, es hora de entrar un poco más en materia de desarrollo de software. Para tal fin, haremos un repaso del clásico modelo de desarrollo: el modelo en cascada. Tener claridad en este modelo será fundamental para lograr el entendimiento de las temáticas a tratar en la Unidad 2 del módulo.

De igual manera se describirán de manera general los cuatro modelos de desarrollo de software más utilizados y aceptados en el mercado, con lo cual podrá tener una referencia importante en este tema.

1. Modelo de desarrollo en cascada

El modelo de desarrollo en cascada es tal vez el más conocido por los académicos y empresarios de la fabricación del software. El modelo se compone de etapas ejecutadas de manera sucesiva, motivo por el cual algunos lo asocian a la forma de cascada:

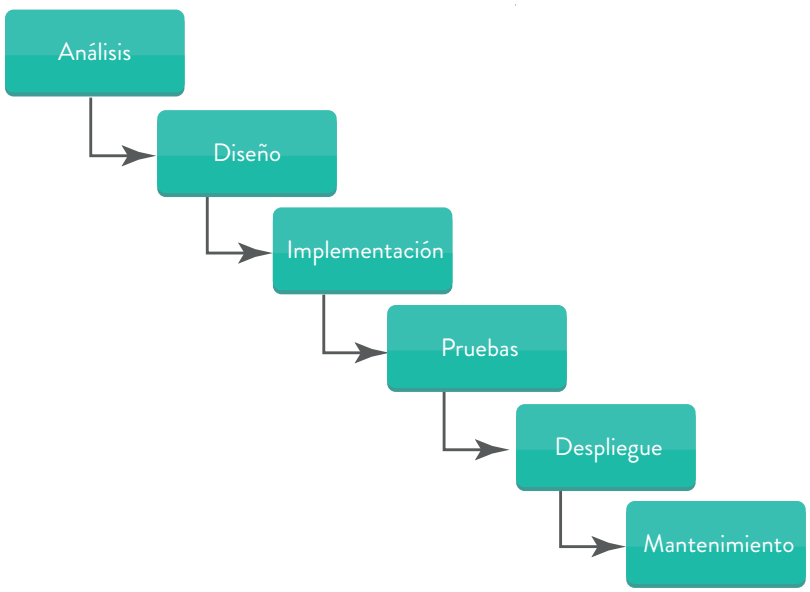


Figura 1. Etapas del modelo de desarrollo de software en cascada
Fuente: Elaboración propia

A continuación repasaremos que actividades se ejecutan en cada una de las etapas.

1.1. Análisis

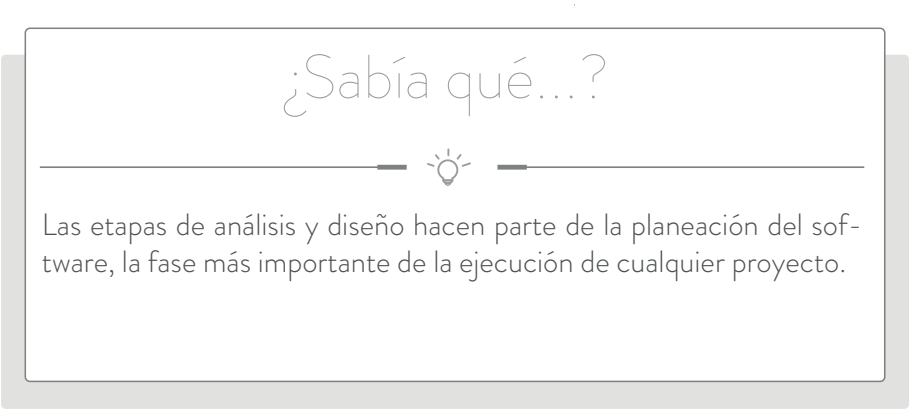
En esta etapa se definen cuáles son los requisitos de los clientes y usuarios de la aplicación. Los requisitos pueden ser de 2 tipos:

- Funcionales: lo que se espera que la aplicación realice, la función que se espera cumpla el software.
- No funcionales: propiedades que se espera cumpla el software, entre ellas velocidad, confiabilidad y capacidad.

1.2. Diseño

En la etapa de diseño la aplicación se estructura o se divide en partes más pequeñas, lo que comúnmente se conoce como módulos.

Durante la etapa de diseño también se establecen las relaciones entre los diferentes módulos que tendrá la aplicación.



1.3. Implementación

En la etapa de implementación se realiza la codificación según el diseño, en el lenguaje de programación seleccionado para el desarrollo de la aplicación.

Adicionalmente a la codificación, durante la implementación se integran los módulos desarrollados. El producto de esta etapa es conocido como el código fuente de la aplicación.

1.4. Pruebas

En esta etapa se pone a prueba que el código fuente, verificando que los requisitos funcionales y no funcionales se cumplan de acuerdo con la definición.

Existen varios tipos de pruebas, entre ellas pruebas unitarias y de integración entre módulos.

¿Sabía qué...?

Las pruebas son el primer control de calidad, donde se pueden tomar las acciones correctivas necesarias para cumplir con los requerimientos.

1.5. Despliegue

La etapa de despliegue se consiste en salida o paso a producción de la aplicación, es decir la instalación y puesta en marcha de la aplicación en servidores productivos.

El despliegue de la aplicación debe cumplir los lineamientos establecidos por una empresa en cuanto al proceso de gestión de cambios.

1.6. Mantenimiento

La etapa final del modelo en cascada tiene como objetivos principales:

- Corrección de errores identificados en el software.
- Implementación del “roadmap” o mapa de ruta de la aplicación.
- Implementar mejoras.

Como se pudo ver, las etapas de desarrollo en el modelo de cascada incluyen una serie de actividades. Una representación simplificada del modelo y sus actividades se observa a continuación:

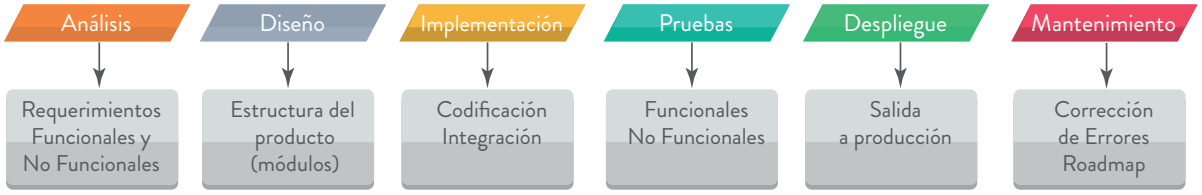


Figura 2. Actividades para cada una de las etapas del modelo de desarrollo en cascada

Fuente: Elaboración propia

La implementación de seguridad en el ciclo de desarrollo que veremos en la siguiente unidad, básicamente consiste en la implementación de actividades de seguridad en cada una de las etapas. Observemos los nombres de las actividades que aprenderemos más adelante a profundidad:

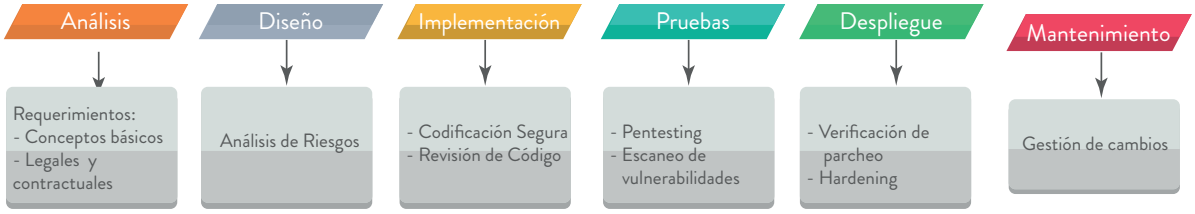


Figura 3. Actividades de seguridad para cada una de las etapas del modelo en cascada

Fuente: Elaboración propia

Modelos de seguridad en el ciclo de desarrollo de software

Como en todos los campos de las tecnologías de la información y las comunicaciones, las empresas, investigadores y los académicos reúnen sus prácticas y las documentan para construir modelos.

La seguridad en el desarrollo de software no es la excepción, por lo que podemos encontrar diversos modelos ampliamente utilizados en por las empresas. En las siguientes secciones conoceremos cuatro de los más conocidos.

1.1. Open Web Application Security Project - OWASP

OWASP es una organización que desarrolla proyectos orientados a la seguridad en el software. Uno de sus proyectos es conocido como CLASP (OWASP, 2016) Comprehesive, Lightweight Application Security Process, el cual propone la implementación de 7 buenas prácticas:

1. Implementación de programas de sensibilización: programas para generar conciencia de seguridad en los desarrolladores.
2. Evaluación de la aplicación: evaluación de vulnerabilidades comunes de una aplicación.
3. Definición de requisitos de seguridad: requisitos que debe cumplir la aplicación, generalmente basado en los conceptos básicos de seguridad.
4. Implementación de prácticas de desarrollo seguro: codificación segura y análisis de código en busca de vulnerabilidades de seguridad.
5. Procesos de remediación de vulnerabilidades: cierre de vulnerabilidades identificadas.
6. Definición y monitoreo de métricas: definición de métricas de seguridad y su correspondiente monitoreo.
7. Publicación de políticas de seguridad operativa: documentación, aprobación y formalización de las políticas de seguridad operativa en temas de desarrollo seguro.

OWASP está en proceso de desarrollar el modelo OPENSAMM en reemplazo de CLASP. Dicho modelo lo veremos a continuación.

2.2. Open Software Assurance MAturity Model – OPENSAMM

OPENSAMM (Pravir Chandra, 2017) es una guía para integrar seguridad en el desarrollo del software. La guía está estructurada en cuatro funciones de negocio, para cada una de las cuales se definen tres prácticas de seguridad, como se muestra en la siguiente tabla:

¿Sabía qué...?

Dos de los cuatro modelos de seguridad en el ciclo de desarrollo son desarrollados por la organización OWASP.

Tabla 1. Descripción de funciones y prácticas

Función	Prácticas de seguridad	Explicación
1. Gobierno	Estrategia y métricas	Dirección estratégica del aseguramiento del software y sus correspondientes métricas.
	Política y cumplimiento	Establecimiento de la estructura de control, auditoría y cumplimiento.
	Educación y orientación	Incremento del conocimientode seguridad en los desarrolladores.
2. Construcción	Evaluación de amenazas	Identificación y caracterización de los eventuales ataques contra el software.
	Requisitos de seguridad	Inclusión de las necesidades de seguridad.
	Arquitectura de seguridad	Fortalecimiento del proceso de diseño bajo una óptica de seguridad permanente.
3. Verificación	Revisión de diseño	Revisión del diseño para asegurar la provisión de mecanismos de seguridad pertinentes.
	Revisión de código	Evaluación del código fuente en busca de vulnerabilidades.
	Pruebas de seguridad	Pruebas de la aplicación en busca de vulnerabilidades.
4. Implementacion	Administración de vulnerabilidades	Tratamiento de las vulnerabilidades y mejora del programa de aseguramiento.
	Fortalecimiento de ambientes	**Implementación de controles de seguridad en el ambiente donde está instalada la aplicación.**
	Habilitación operativa	**Identificación de información de seguridad pertinente para la implementación y puesta en marcha de programas.**

Fuente: Elaboración propia

2.3. Security considerations in the system development life cycle NIST 800-64

El National Institute of Standards and Technology NIST, desarrolló y publicó un documento que establece las consideraciones de seguridad a tener en cuenta en el ciclo de desarrollo de sistemas, lo anterior a través del documento NIST 800-64 (Kissel R. ET al., 2008).

Esta publicación se compone de 5 fases o etapas, para la cual se definen una serie de actividades a desarrollar. En la siguiente ilustración se observa un resumen de cada una de estas fases:

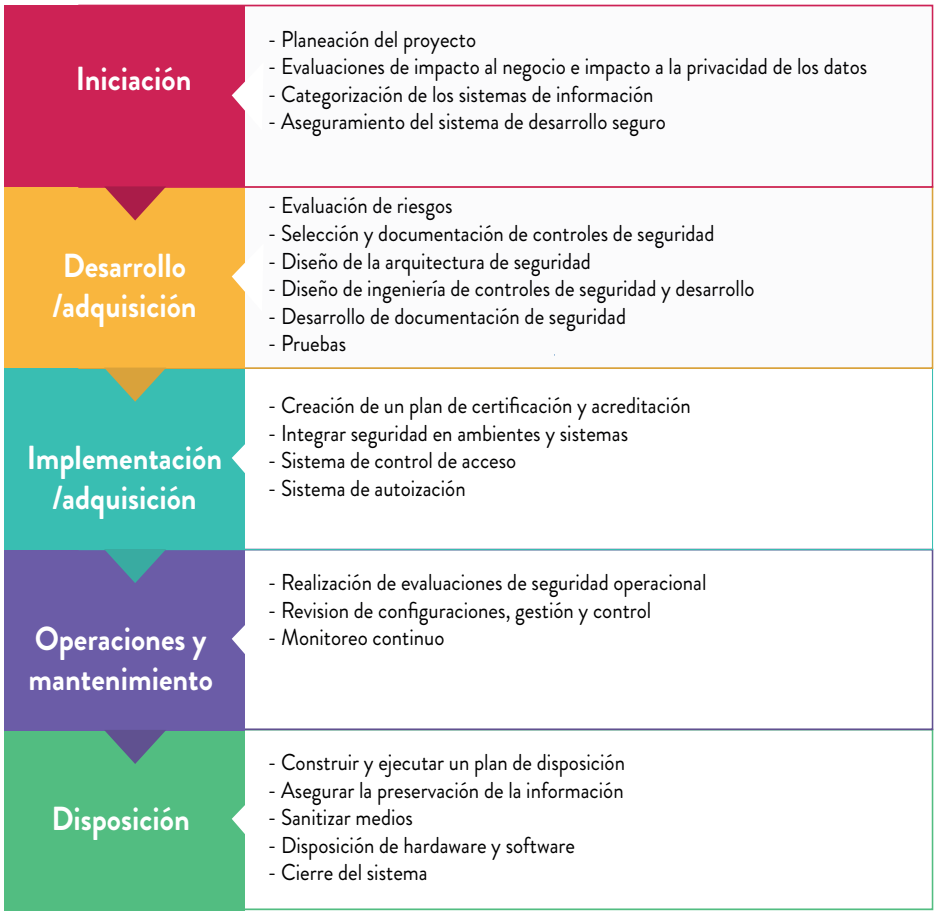


Figura 4. Resumen de las 5 fases de NIST 800-64

Fuente: Elaboración propia

2.4. Microsoft security development lifecycle

El modelo de seguridad en el ciclo de vida del desarrollo (Microsoft Corporation, 2010) es un proceso cuyo objetivo es construir software más seguro y estable.

Se basa en siete fases las cuales se presentan a continuación:

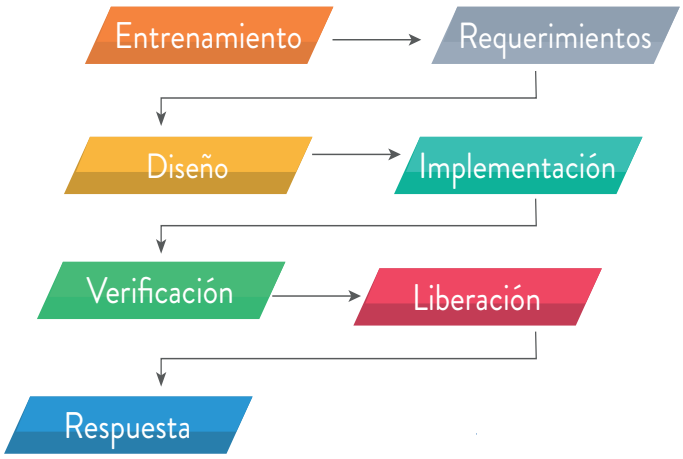


Figura 5. Secuencia de las fases del modelo de Microsoft
Fuente: Elaboración propia (2017)

Para cada una de estas fases, Microsoft define una serie de prácticas que se deben cumplir para garantizar la seguridad en el ciclo de desarrollo. Observemos las prácticas que hacen parte de cada fase:

1. Entrenamiento
    - SDL Práctica # 1: Capacitación de Seguridad
2. Requerimientos
    - SDL Práctica # 2: Establecer los requisitos de seguridad y privacidad
    - SDL Practica # 3: Crear puntos de calidad
    - DL Práctica # 4: Evaluaciones de riesgo de seguridad y privacidad

3. Diseño
    - SDL Práctica # 5: Establecer los requisitos de diseño
    - SDL Práctica # 6: Análisis de la superficie de ataque
    - DL Práctica # 7: Modelamiento de Amenazas
4. Implementación
    - SDL Practica # 8: Utilización de herramientas aprobadas
    - SDL Practica # 9: Desaprobar funciones no seguras
    - SDL Practica # 10: Realizar análisis estático de código
5. Verificación
    - SDL Practica # 11: Realizar análisis dinámico
    - SDL Práctica # 12: Pruebas Fuzz
    - SDL Practica # 13: Revisión de superficie de ataque
6. Liberación
    - SDL Práctica # 14: Crear un Plan de Respuesta a Incidentes
    - SDL Práctica # 15: Revisión de la Conducta Final de Seguridad
    - SDL Practica # 16: Certificar lanzamiento y Archivo
7. Respuesta
    - SDL Práctica # 17: Ejecutar el Plan de Respuesta a Incidentes

Referencias bibliográficas

OWASP. (2016). *CLASP Concepts*. Recuperado de https://www.owasp.org/index.php/CLASP_Concepts

Pravir Chandra, (2017). *Software Assurance Maturity Model.* Recuperado de https://www.owasp.org/images/a/a9/SAMM-1.0-es_MX.pdf, (1-9)

Kissel Richard, Stine Kevin, Scholl Matthew, Rossman Hart, Fahlsing Jim, Gulick Jessica. (2008). *Security Considerations in the System Development Life Cycle*. Recuperado de http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-64r2.pdf

Microsoft Corporation. (2010). *Security Development Lifecycle*. Recuperado de https://www.microsoft.com/en-us/sdl

INFORMACIÓN TÉCNICA

FACULTAD DE
INGENIERÍA, DISEÑO
E INNOVACIÓN

Módulo: Seguridad en el Ciclo de Desarrollo

Unidad 1: Conceptos básicos y modelos existentes

Escenario 2: Modelos existentes

Autor: Miguel Ángel Zambrano Puentes

Asesor Pedagógico: Edwin Mojica Quintero

Diseñador Gráfico: Brandon Steven Ramírez Carrero

Asistente: Ginna Quiroga