**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

| **PROGRAMA DE FORMACIÓN** | Seguridad en aplicaciones web |
| --- | --- |

| **COMPETENCIA** | 220501099  Probar la solución del *software* de acuerdo con parámetros técnicos y modelos de referencia. | **RESULTADOS DE APRENDIZAJE** | 220501099-01  Formular plan de actividades y artefactos de pruebas de acuerdo con los requisitos organizacionales y normatividad. |
| --- | --- | --- | --- |

| **NÚMERO DEL COMPONENTE FORMATIVO** | CF003 |
| --- | --- |
| **NOMBRE DEL COMPONENTE FORMATIVO** | Construcción del plan de pruebas con Owasporientado a aplicaciones web. |
| **BREVE DESCRIPCIÓN** | Para la construcción de un sistema de información por procesamiento electrónico de datos orientado a la web es muy importante implementar los mecanismos que brindan la seguridad de la aplicación, el presente componente tiene como objetivo elaborar un plan de pruebas de seguridad utilizando como metodología la *Guía de referencia de pruebas de Owasp* que pretende satisfacer los requerimientos de seguridad que se presentan. |
| **PALABRAS CLAVE** | Actividades, documentación, implantación |

| **ÁREA OCUPACIONAL** | 9 - Procesamiento, fabricación y ensamble |
| --- | --- |
| IDIOMA | Español |

1. **TABLA DE CONTENIDOS:**

**1. Plan de actividades**

1.1. Objetivos

1.2. Limitaciones y facilidades

1.3. Fijar las metas y objetivos particulares

1.4. Definir equipo de trabajo

1.5. Definir responsabilidades en el equipo

1.6. Crear una estrategia

1.7. Establecer los plazos

1.8. Determinar los recursos necesarios

1.9. Medir los resultados

**2. Artefactos de pruebas**

2.1 Metodología

2.2 Tipos de artefactos

2.3 Diseño de artefactos

**3. Entorno de pruebas**

3.1 Herramientas para pruebas

3.2 Herramientas de gestión

3.3 Herramientas de operación

1. **INTRODUCCIÓN**

Le damos la bienvenida al componente formativo denominado **Construcción del plan de pruebas con Owasp orientado a aplicaciones web**, el cual hace parte del programa de formación técnico en “Seguridad en aplicaciones web”, para lo cual se invita a observar el siguiente video:



1. **DESARROLLO DE CONTENIDOS:**

**1. Plan de actividades**

Es un documento que recoge un conjunto de tareas necesarias para la consecución de una acción u objetivo concreto. Antes de llevar a cabo la tarea de planificar un proyecto, es conveniente hacer un plan de actividades, en el cual se contemplará lo necesario que lleve a organizar las pruebas de seguridad de una aplicación web con base en la guía Owasp (*Open Web Application Security Project*).



Se recomienda para la elaboración del plan de actividades, la creación de un diagrama de Gantt el cual se compone de dos grandes partes de la siguiente manera:

| Figura 1  *Diagrama de Gantt generado con Open Project* | |
| --- | --- |
|  | En su parte izquierda se describe la lista de tareas a llevar a cabo. |
|  | En la parte de derecha hay un cronograma con barras que representa el trabajo a realizar. |

Así mismo, dentro del diagrama es importante definir:



La importancia de un diagrama de Gantt radica en la posibilidad de contar constantemente con una visualización global del proyecto, incluso de planes complejos, mostrando el impacto que un cambio puede generar en una sola actividad en todo el proyecto permitiendo generar retroalimentación a los roles interesados.

Para diseñar un diagrama de Gantt se recomienda la utilización de herramientas de *software* que permiten manejar de una manera ágil el seguimiento y verificación de su cumplimiento, incluso enmarcado dentro de un proyecto que puede considerarse más grande de marcando una hoja de ruta; algunas herramientas recomendadas son:

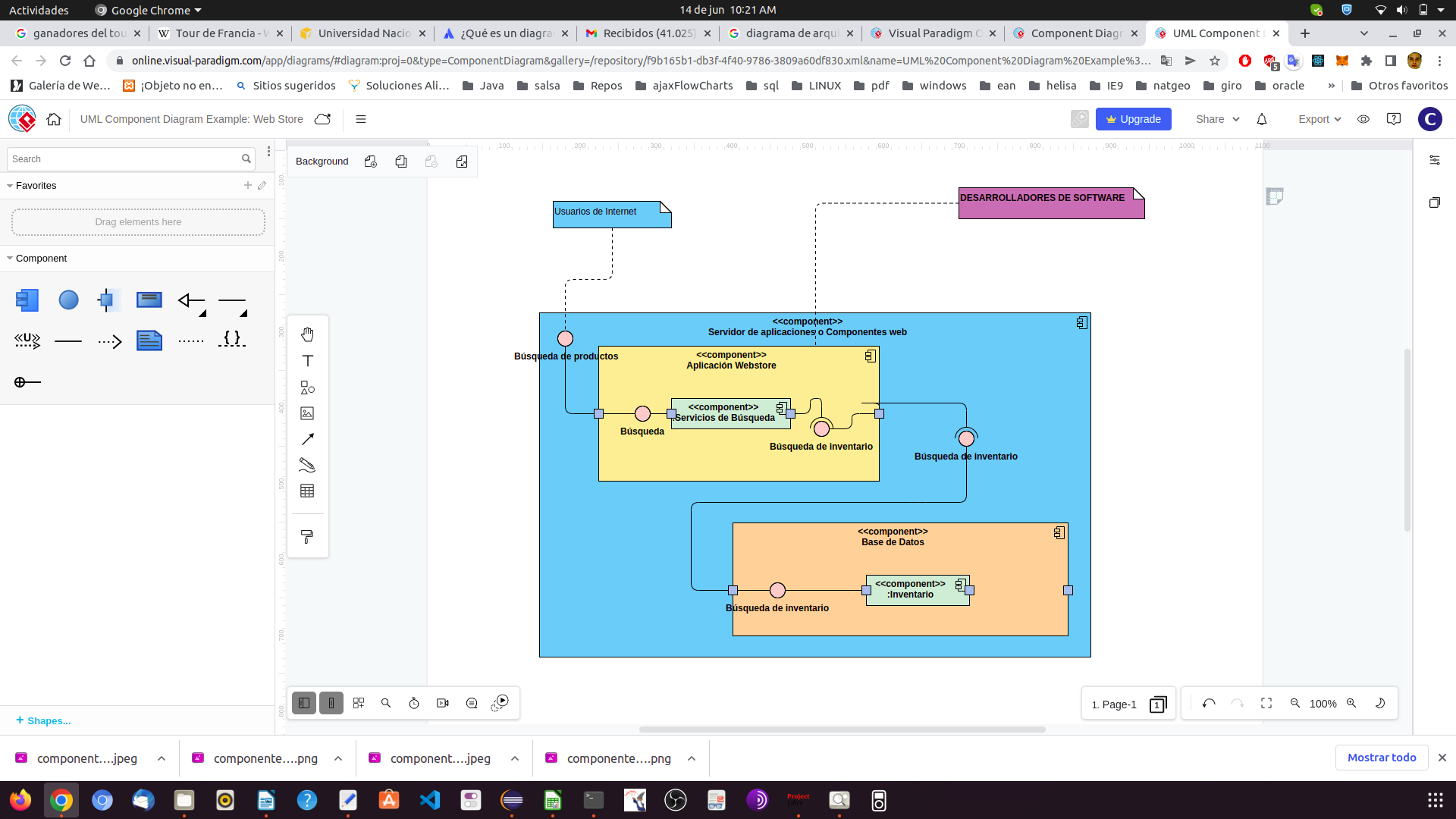


**1.1 Objetivos**

En este punto se deben plantear los objetivos generales a cumplir en el proyecto teniendo en cuenta que la guía Owasp comprende un gran número de componentes a los cuales se puede aplicar un proceso de análisis de vulnerabilidades como lo son por ejemplo, **los componentes tipo API** (interfaz de programación de aplicaciones) que normalmente se construyen bajo el concepto de *web services*, de validación de datos de autorización y autenticación, de verificación y utilización de marcos en diferentes organizaciones, hasta llegar a la verificación de infraestructura, redes con sus configuraciones y talento humano.

**Figura 2**

*Objetivos del plan de pruebas*



Los objetivos se pueden plantear de la siguiente manera:

| Servicio o plataforma de análisis de sitios web en línea | Vector Premium | Análisis del componente de aplicaciones web que corresponden a herramientas de *software* que funcionan vía internet y se convierten en soluciones que están *“expuestas”* al público de manera masiva. |
| --- | --- |
| Personaje de mujer de desarrollo de sitios web y análisis de contenido web  icono de línea aislado | Vector Premium | Establecer hacia quién va dirigido dicho análisis, entendiéndose que esas pruebas según la guía pueden estar orientadas a *testers* de *software* que son personas que trabajan en el marco de metodologías y pretenden dar cumplimiento a estándares de calidad unidos a una metodología en el marco del desarrollo de *software.* |

Estos objetivos están orientados a:

Especialistas de seguridad que, en su gran mayoría, se encuentran en la capa de administración de la infraestructura que soporta el funcionamiento del componente web en un ecosistema tecnológico y también a desarrolladores de *software* que son las personas encargadas de escribir las sentencias de código fuente que conforman el componente.

**1.2. Limitaciones y facilidades**

Cuando se aborda la creación de un plan de actividades es muy importante recopilar la mayor cantidad posible de información de una organización. Esta debe estar orientada hacia la consecución de un contexto general de la situación actual de la empresa que va a utilizar o que está utilizando la aplicación web y que redunda en un análisis organizacional global que permita ubicar de una manera ágil los posibles recursos o la carencia de ellos.



Es importante listar e identificar esas limitantes y facilidades en los siguientes temas:

* Documentación existente y facilidad de acceso.
* Tecnologías.
* Infraestructura.
* Redes.
* Facilidad de acceso.
* Componentes.
* Identificación del personal encargado con sus respectivos roles y responsabilidades.
* Grado de conocimiento en la organización acerca del proceso de seguridad informática.

La importancia de este análisis radica en la posibilidades de recomendar a futuro, o durante la ejecución del plan de pruebas, mecanismos, herramientas de *software*, prácticas, personal necesario, estrategias, tecnologías de carácter libre o incluso de pago que ayuden en la consecución de los objetivos y que eliminen la existencia de brechas que antes no se han tenido en cuenta.

El siguiente gráfico representa lo anteriormente mencionado.



**1.3. Fijar las metas y objetivos particulares**

Para este apartado es importante definir que los objetivos serán guiados por la metodología Owasp y las metas a cumplir en este proceso son:

* Aplicar el estándar Owasp y el manual de pruebas de *pentesting* propuesto para determinar el plan de pruebas que se aplicarán sobre el sistema con el enfoque de análisis de seguridad en las aplicaciones web.
* Determinar las pruebas de la metodología Owasp de seguridad en aplicaciones web para los 10 casos de vulnerabilidades más conocidos y definidos en el punto de la estrategia de este proceso.
* Determinar los artefactos de pruebas a utilizar de acuerdo con la metodología Owasp de seguridad en aplicaciones web.



**1.4. Definir equipo de trabajo**

Los mejores equipos de trabajoson los quese estructuran correctamente, saben comunicarse e interaccionan entre sí; la motivación, participación, organización, compromiso, confianza, objetivos comunes y resolución de problemas son las siete característicasdel trabajoen equipofundamentales. De ahí la importancia de tomarse el tiempo necesario para listar a las personas indicadas a quienes se les van a asignar las actividades en el plan de trabajo para interactuar con ellas.

No obstante que este tipo de pruebas están orientadas a desarrolladores de *software,* es importante tener en cuenta otros tipos de roles que pueden interactuar dentro de un equipo de trabajo como los administradores de red y plataformas, puesto que estos son los encargados de otorgar permisos en usuarios, definir tipologías de red, manejo de configuración de reglas de *firewalls*, acceso a servidores de aplicaciones sistemas de autenticación entre otros, sobre todo para la parte en la cual consiste en contrastar los resultados obtenidos de las pruebas contra lo que realmente existe en el ecosistema tecnológico analizado en el que funciona una aplicación web.



**1.5 Definir responsabilidades en el equipo**

Los roles y responsabilidades se definen a nivel de la organización y del proyecto y el trabajo será conocer estos dos ítems del equipo en diferentes momentos de desarrollo de las actividades. Para ello, contar con una estructura de equipo sólida en el nivel organizacional es fundamental y es lo primero que se debe apropiar para apoyar en la resolución de posibles brechas en esta área.

El principal rol que es el del desarrollador, debe contar como mínimo con los siguientes conocimientos:

,



**1.6 Crear una estrategia**

El procedimiento mediante el cual se determina para este caso es el análisis de vulnerabilidades de una aplicación web, y este corresponde a la verificación punto por punto de los 10 casos de ataques más comunes reportados por los informes que cada cierto tiempo destaca Owasp en dichos ambientes web, los cuales son:



**1.7 Establecer los plazos**

Mediante un cronograma se deben establecer los tiempos para cada actividad con fecha de inicio y fecha de finalización y cada una debe ser organizada en orden lógico de aplicación.

En el siguiente video se conocerá más detalladamente qué se requiere establecer en un cronograma:



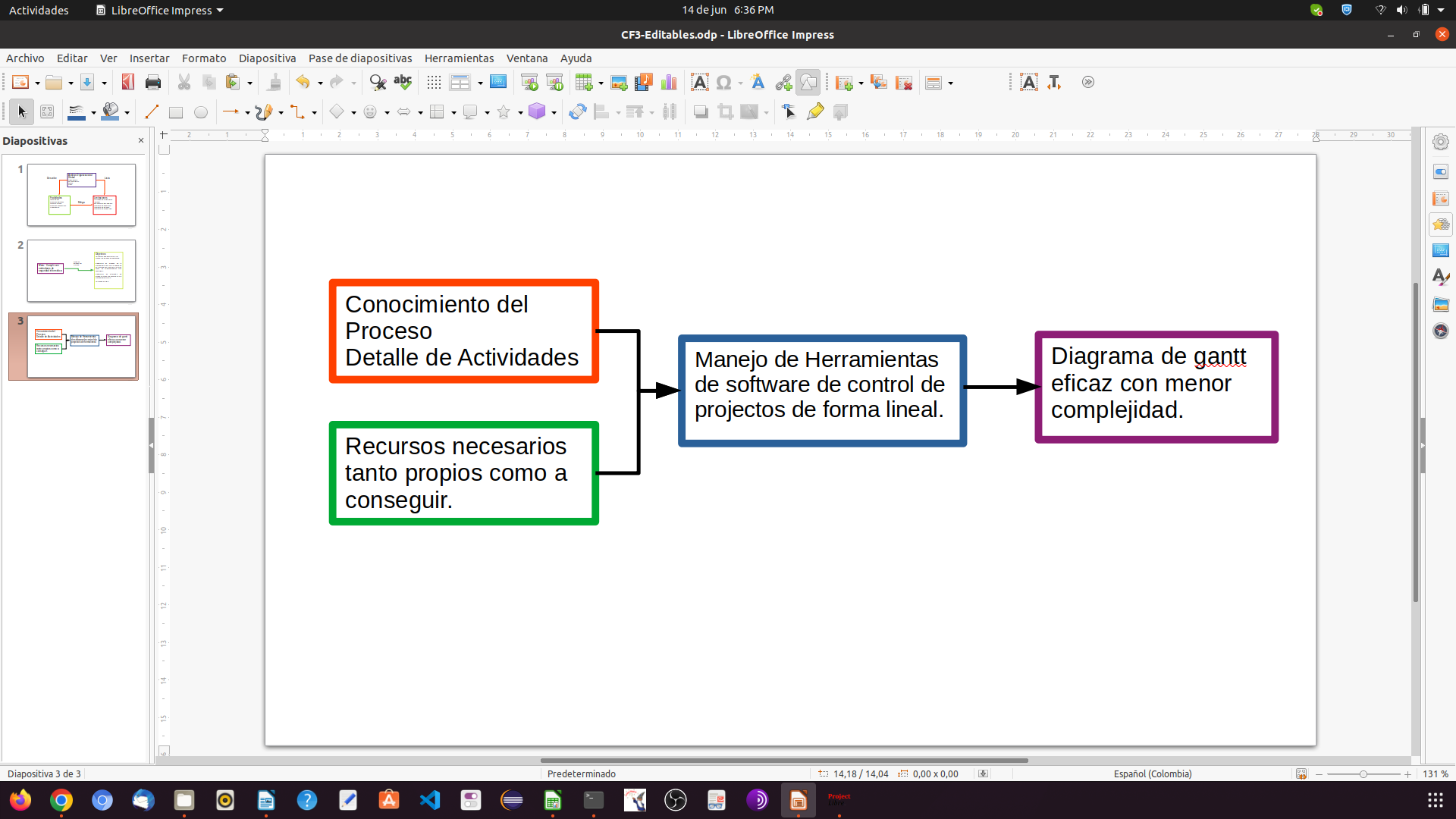
**1.8 Determinar los recursos necesarios**

Centrado en la estrategia definida en el punto anterior para lograr la determinación de los recursos necesarios, se deben observar las condiciones dentro y fuera de la organización, a saber:



Se debe considerar que la determinación de los recursos necesarios para un proyecto debe contemplar cierto margen de flexibilidad y evolucionar a la par de la intervención, el comportamiento del entorno y las decisiones del cliente.

Dicho de otro modo, lo que no parece indispensable en un principio puede serlo más adelante.



**1.9 Medir los resultados**

Con el ánimo de evaluar el éxito o fracaso del proyecto se proponen, como mínimo, medir los siguientes indicadores:



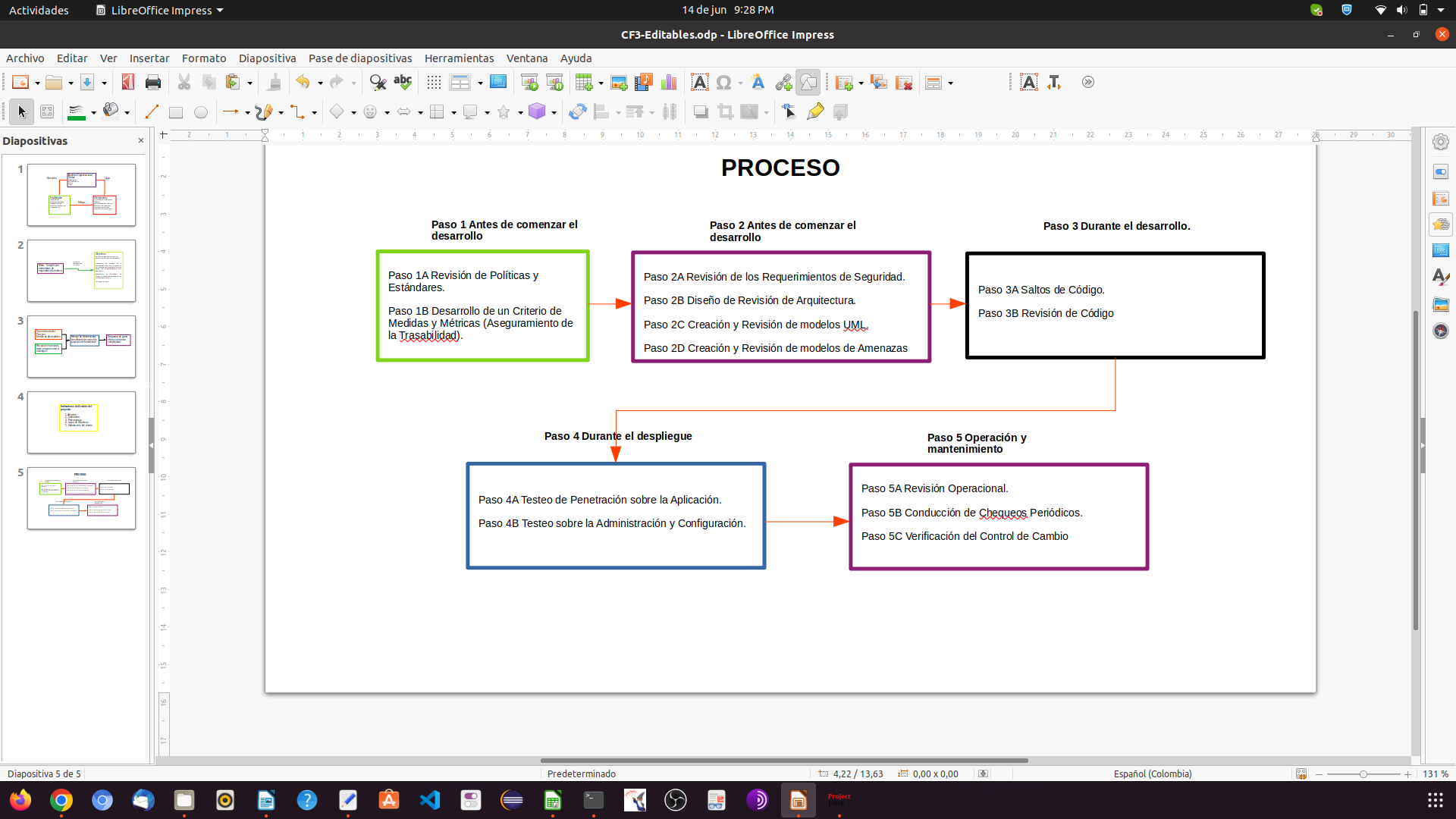
**2. Artefactos de pruebas**

Son los mecanismos que se centran en recoger toda la información como sea posible sobre la aplicación web objetivo y conocer la manera cómo opera. La recopilación de información es un paso fundamental y necesario en una prueba de intrusión.

Esta prueba corresponde al reconocimiento pasivo de información que se debe realizar sin intervenir los servidores o que estos generen logs de seguridad. Esta tarea normalmente se puede llevar a cabo de muchas formas. Utilizando herramientas de acceso público motores de búsqueda, *scanner*, enviando peticiones HTTP simples, o peticiones especialmente diseñadas, es posible forzar a la aplicación a filtrar información al exterior con mensajes de error devueltos, o revelar las versiones y tecnología en uso por la aplicación web.

**2.1 Metodología**

La metodología a utilizar es Owasp la que comprende las siguientes fases enmarcadas dentro del ciclo de vida del desarrollo de *software* para lo que se recomienda reconocer las pruebas dentro del siguiente proceso, en especial los pasos 4 y 5.



**2.2 Tipos de artefactos**

Como ya se ha mencionado, los artefactos tecnológicos son aquellos dispositivos concebidos y creados para ayudar a resolver necesidades o facilitar ciertas tareas, empleando para su construcción y funcionamiento las virtudes de la [técnica](https://www.ejemplos.co/20-ejemplos-de-tecnicas/) y la [ciencia](https://www.ejemplos.co/20-ejemplos-de-ciencia/).

De acuerdo con la estrategia definida y según la metodología seleccionada, se describen los tipos de artefactos a construir para esta prueba, así:



**2.3 Diseño de artefactos**

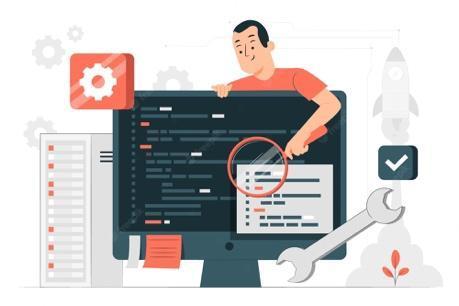
Con base en la metodología y los tipos de artefactos a utilizar se construye un modelo de implementación de estos:



**3. Entorno de pruebas**

Un ambiente de pruebas es un entorno recomendado que pretende emular, ojalá al 100%, el ecosistema tecnológico en el que funciona un aplicación en este caso una aplicación web que ya se encuentra en uso por parte de una organización.

Es decir, ya se encuentra funcionando en un ambiente productivo, de ahí la necesidad de identificar claramente de ese entorno productivo los siguientes ítems para replicarlos en el entorno de pruebas: sistemas operativos, servidores de aplicaciones, sistemas de bases de datos, mecanismos de interoperabilidad con otras aplicaciones, entornos de red, documentación correspondiente a manuales técnicos y de usuario, marcos de trabajo (*frameworks*) y demás *software* necesario para ejecutar las pruebas; para cada uno de estos es imprescindible la identificación de sus versiones y realizar el seguimiento a posibles cambios y sus dependencias.



El entorno de pruebas recomendado para este tipo de plan que se va a realizar del lado servidor se basa en sistemas operativos de la familia Unix (Linux en diferentes versiones Centos, Red Hat, Ubuntu, FreeBSD o m[acOS](https://www.gcfaprendelibre.org/tecnologia/curso/curso_de_mac_os/que_es_mac_os/1.do)) sin desconocer la necesidad de realizarlos en ambientes Windows, entre otros sistemas; el objetivo es ejecutar las pruebas sin temor a equivocarse para luego realizarlas en ambientes de producción.

A continuación se muestra un ejemplo de implementación de ambiente de pruebas recomendado que consta de cuatro pasos.



**3.1 Herramientas para pruebas**

Para complementar las herramientas de un entorno de pruebas asociadas a la metodología utilizada, a continuación se mencionan algunas de las herramientas más conocidas para ejecutar este tipo de pruebas.



Además, también se pueden utilizar las siguientes:



**3.2 Herramientas de gestión**

Para realizar labores de control de operaciones, registro, administración, documentación y seguimiento a casos se describen las siguientes posibilidades:



**3.3 Herramientas de operación**

Una vez se cuente con un ambiente de producción, se pretende implementar el seguimiento y monitoreo a dicho ambiente para lo cual se recomienda el uso de estas herramientas:



Hemos llegado a la finalización de este componente, pero antes es importante que observar las videoclases que se presentarán a continuación y que servirán como apoyo al ejercicio práctico que debe realizar en la elaboración de un plan de pruebas.





1. **Síntesis**

En el siguiente recurso se expone un resumen de este componente formativo:



1. **ACTIVIDAD DIDÁCTICA**

| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| --- | --- |
| Nombre de la actividad | Conceptos básicos de pruebas de seguridad de aplicaciones web |
| Objetivo de la actividad | Afianzar los conocimientos para el proceso de pruebas de seguridad de aplicaciones web con Owasp. |
| Tipo de actividad sugerida | Cuestionario. |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | Anexo3\_ActividadDidactica1\_CF03 |

1. **MATERIAL COMPLEMENTARIO**

| Tema | Referencia APA del Material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del Recurso o  Archivo del documento o material |
| --- | --- | --- | --- |
| 1. Plan de Objetivos | Caballero, Q., A. E. (2019). *Webinar gratuito: guía de pruebas de Owasp.* [Video]. YouTube. https://www.youtube.com/watch?v=kXfZqQY0rcg&ab\_channel=AlonsoCaballero | Video | https://www.youtube.com/watch?v=kXfZqQY0rcg&ab\_channel=AlonsoCaballero |

1. **GLOSARIO:**

| **TÉRMINO** | **SIGNIFICADO** |
| --- | --- |
| ***Exploits*** | parte de un *software* o una secuencia de comandos que se aprovecha de un error o vulnerabilidad. |
| **Netcat** | herramienta de línea de comandos que sirve para escribir y leer datos en la red. |
| **Owasp** | *Open Web Application Security Project.* |
| ***Pentesting*** | proceso que imita posibles ataques a una red informática e intenta robar datos. |
| ***Checklist*** | lista de chequeo que sirve para registrar un proceso de auditoría. |

1. **REFERENCIAS BIBLIOGRÁFICAS:**

Amor. F. (2020). *Introducción a Owasp.* <https://www.adictosaltrabajo.com/2016/03/07/introduccion-a-owasp/>

Asana, (2020). *4 técnicas eficaces para definir roles y responsabilidades.* <https://asana.com/es/resources/roles-and-responsibilities>

Diaz, M., y Marulanda, M. F. (2018). *Proyecto de aplicación de Owasp.*<https://repository.unad.edu.co/bitstream/handle/10596/20479/1060648494.pdf?sequence=3&isAllowed=y#page=57&zoom=100,148,204>

Junta de Andalucía. (2019). *Owasp Testing Project.* Junta de Andalucía. <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/553>

**CONTROL DEL DOCUMENTO**

|  | **Nombre** | **Cargo** | **Dependencia**  ***(Para el SENA indicar Regional y Centro de Formación)*** | **Fecha** |
| --- | --- | --- | --- | --- |
| **Autor (es)** | Carlos Hernán Muñoz | Experto Temático | Regional Cauca, Centro de teleinformática y producción industrial | Junio de 2022 |
|  | Paula Andrea Taborda Ortiz | Diseñadora Instruccional | Regional Norte de Santander, Centro de la Industria, la Empresa y Los Servicios CIES | Julio de 2022 |
| Carolina Coca Salazar | Asesora Metodológica | Regional Distrito Capital- Centro de Diseño y Metrología | Julio de 2022 |
| José Gabriel Ortiz Abella | Corrector de estilo. | Regional Distrito Capital - Centro de Diseño y Metrología. | Julio del 2022. |

1. **CONTROL DE CAMBIOS**

|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** | **Razón del Cambio** |
| --- | --- | --- | --- | --- | --- |