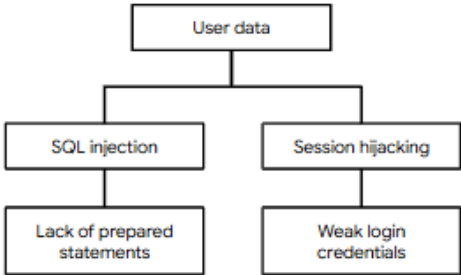


# PASTA

Etapas	Empresa de Zapatillas Online
1. Definir objetivos comerciales y de seguridad.	<p>Requisitos comerciales que se analizarán:</p> <ul style="list-style-type: none"><li>• <i>Los usuarios pueden crear perfiles de miembros internamente o conectando cuentas externas.</i></li><li>• <i>La aplicación debe procesar transacciones financieras.</i></li><li>• <i>La aplicación debe cumplir con PCI-DSS.</i></li></ul>
2. Definir el alcance técnico	<p>Lista de tecnologías utilizado por la aplicación:</p> <ul style="list-style-type: none"><li>• <i>Interfaz de programación de aplicaciones (API)</i></li><li>• <i>Infraestructura de clave pública (PKI)</i></li><li>• <i>SHA-256</i></li><li>• <i>SQL</i></li></ul> <p><i>Las API facilitan el intercambio de datos entre clientes, socios y empleados, por lo que se les debe dar prioridad. Manejan una gran cantidad de datos confidenciales mientras conectan varios usuarios y sistemas. Sin embargo, se deben considerar detalles como qué API se utilizan antes de priorizar una tecnología sobre otra. Por lo tanto, pueden ser más propensos a sufrir vulnerabilidades de seguridad porque hay una superficie de ataque más grande.</i></p>
3. Descomponer la aplicación	<p><i>Diagrama de flujo de datos de muestra</i></p> <p><b>Data flow diagram</b></p> <p><b>Note:</b> This data flow diagram represents a single process. Data flow diagrams for an application like this are normally much more complex.</p> <pre>graph LR; User[User] -- "Searching for sneakers for sale." --&gt; Process((Product search process)); Process -- "Listings of current inventory." --&gt; Database[Database];</pre>
4. Análisis de amenazas	<ul style="list-style-type: none"><li>• <i>Inyección</i></li><li>• <i>Secuestro de sesión</i></li></ul>
5. Análisis de vulnerabilidad	<ul style="list-style-type: none"><li>• <i>Falta de declaraciones preparadas.</i></li><li>• <i>Token de API roto</i></li></ul>

6. Modelado de ataques	<p><i>Diagrama de árbol de ataque de muestra</i></p> <p style="text-align: center;"><b>Sample attack tree</b></p> <p><b>Note:</b> Applications like this normally have large, complex attack trees with many branches.</p>  <pre> graph TD     A[User data] --&gt; B[SQL injection]     A --&gt; C[Session hijacking]     B --&gt; D[Lack of prepared statements]     C --&gt; E[Weak login credentials] </pre>
7. Análisis de riesgos e impacto.	<p>4 controles de seguridad que pueden reducir el riesgo son:  <i>SHA-256, procedimientos de respuesta a incidentes, política de contraseñas, principio de privilegio mínimo</i></p>

## Etapa 1: Definir los objetivos empresariales y de Seguridad

**Resumen:** Estos objetivos se definen al principio formulando preguntas generales sobre la finalidad de la aplicación. Por ejemplo, ¿cómo hace ganar dinero la aplicación al negocio? Comprender la respuesta a estas preguntas ayuda a guiar el trabajo detallado que vendrá a continuación.

**Recomendaciones:** Una aplicación de compras como ésta necesitará procesar pagos. Basándonos en esta descripción, sabemos que se necesitan ciertas tecnologías para mantener la información privada y segura, y que todo tendrá que cumplir las normas PCI-DSS.

## Etapa 2: Definir el alcance técnico

**Resumen:** El objetivo aquí es comprender la superficie de ataque identificando las tecnologías que utiliza la aplicación y comprendiendo sus dependencias.

**Recomendaciones:** Las API facilitan el intercambio de datos entre clientes, socios y empleados, por lo que deben ser prioritarias. Además, manejan muchos datos sensibles mientras conectan a varios usuarios y sistemas entre sí. Sin embargo, hay que tener en cuenta detalles como qué API se están utilizando antes de priorizar una tecnología sobre otra. Así, pueden ser más propensas a las vulnerabilidades de Seguridad porque hay una mayor superficie de ataque.

## Etapa 3: Descomponer la aplicación

**Resumen:** El objetivo es revisar cómo funciona la aplicación y cómo se implementan actualmente los Controles de seguridad.

**Recomendaciones:** El diagrama de flujo de datos muestra cómo una solicitud de búsqueda típica pasa a través de múltiples capas. Algo que se podría revisar es que la base de datos MySQL está utilizando sentencias preparadas cuando se introducen las consultas.

## **Etapla 4: Análisis de amenazas**

**Resumen:** El objetivo principal de la cuarta etapa es considerar los tipos de amenazas que podrían afectar a su aplicación. Otra cosa a tener en cuenta son los tipos de datos que procesa la aplicación.

**Recomendaciones:** Los ataques de inyección son habituales en las bases de datos SQL. El secuestro de sesión es posible porque la aplicación comunica las cookies entre varias capas. Es importante tener en cuenta la superficie de ataque tecnológico y cualquier amenaza relevante para su producto para implementar eficazmente sus responsabilidades en materia de Seguridad de la Información.

## **Etapla 5: Análisis de vulnerabilidad**

**Resumen:** consiste en asociar las vulnerabilidades de los recursos con las amenazas potenciales. El objetivo es identificar qué falla en el diseño de la aplicación o en su código base basándose en sus pruebas de Seguridad.

**Recomendaciones:** la falta de sentencias preparadas puede hacer que nuestra base de datos SQL sea vulnerable a ataques de inyección. Y el secuestro de sesión es posible si las cookies se manejan mal entre las fuentes de entrada y salida.

## **Etapla 6: Modelo de ataque**

**Resumen:** En esta etapa, el objetivo es vincular las amenazas y vulnerabilidades identificadas en los pasos anteriores utilizando la estructura de un árbol de ataque, lo cual permite demostrar que las amenazas potenciales que se han identificado son realmente viables. Recursos como MITRE ATT&CK y la Lista de CVE® son referencias útiles para ello.

**Recomendaciones:** Este ejemplo muestra cómo los datos del usuario son vulnerables a los ataques que se identificaron anteriormente. Al igual que el diagrama de flujo de datos de muestra, un árbol de ataque real para una aplicación para dispositivos móviles sería mucho más complejo que éste.

## **Etapla 7: Análisis de riesgos e impacto**

**Resumen:** El objetivo de la etapa final de PASTA es identificar formas de mitigar los riesgos que se identificaron de las etapas 4 a 6 y planificar los riesgos restantes que no se puedan remediar.

**Recomendaciones:** SHA-256, los Procedimientos de respuesta ante incidentes, la política de contraseñas y el principio de privilegio mínimo son algunos ejemplos de controles técnicos, operativos y de gestión que pueden implementarse antes del Lanzamiento para reducir el Riesgo.