**Fase 1: Planificación y Diseño (1-2 meses)**

* **Definición del alcance:** Identificación de funcionalidades principales (publicaciones, hashtags, vulnerabilidades, etc.).
* **Análisis de requisitos:** Determinación de herramientas tecnológicas (PHP, MySQL, JavaScript, XAMPP).
* **Diseño inicial:** Creación de wireframes y diseño de la estructura de la base de datos.
* **Investigación:** Selección de vulnerabilidades a implementar (SQL Injection, Local File Inclusion, etc.) y referencias educativas.

**Fase 2: Desarrollo e Implementación (3-4 meses)**

* **Backend:** Desarrollo de modelos y controladores en PHP para gestionar usuarios, publicaciones y vulnerabilidades.
* **Frontend:** Implementación de la interfaz de usuario utilizando HTML, CSS y JavaScript.
* **Integración de vulnerabilidades:** Simulación de fallos de seguridad controlados en funcionalidades clave.
* **API:** Consumo de la API de vulnerabilidades para enriquecer el contenido de la plataforma.

**Fase 3: Pruebas y Correcciones (1-2 meses)**

* **Pruebas funcionales:** Verificación de todas las funcionalidades, como publicaciones, hashtags y manejo de datos.
* **Pruebas de seguridad:** Validación de las vulnerabilidades simuladas y su impacto.
* **Correcciones:** Ajustes en el código según los resultados de las pruebas para garantizar estabilidad y experiencia de usuario.

**Fase 4: Documentación y Presentación (1 mes)**

* **Documentación técnica:** Creación de manuales de uso y guías para entender las vulnerabilidades implementadas.
* **Preparación de la presentación:** Elaboración de diapositivas, demos funcionales y estadísticas del proyecto.
* **Entrega final:** Presentación del proyecto VulnSocial como herramienta educativa y de concienciación en ciberseguridad.

**Metodología del Proyecto VulnSocial**

El desarrollo de VulnSocial se llevó a cabo mediante un enfoque iterativo y estructurado, dividido en etapas que garantizan una evolución ordenada del proyecto. A continuación, se describe detalladamente cada etapa y las técnicas empleadas.

**1. Investigación y análisis previo**

Esta etapa inicial se centró en establecer las bases conceptuales y técnicas del proyecto.

* **Análisis de necesidades:**
  + Identificar el público objetivo: desarrolladores, estudiantes de ciberseguridad y usuarios interesados en entender vulnerabilidades web.
  + Definir los requisitos funcionales y no funcionales, como el manejo de publicaciones, seguridad y usabilidad.
* **Investigación técnica:**
  + Exploración de tecnologías compatibles con el objetivo del proyecto: PHP, MySQL, JavaScript, y el entorno XAMPP.
  + Selección de vulnerabilidades a implementar, destacando aquellas más comunes en aplicaciones reales (SQL Injection, LFI, XSS, etc.).
* **Referencias educativas:**
  + Análisis de plataformas existentes como PortSwigger para entender cómo presentar vulnerabilidades.

**2. Diseño del sistema**

El diseño se realizó con una visión modular para facilitar la implementación y futuras mejoras.

* **Arquitectura del proyecto:**
  + Modelo Cliente-Servidor utilizando PHP como backend, MySQL para la gestión de datos y JavaScript para la interacción en el frontend.
* **Diseño de base de datos:**
  + Creación de tablas como usuarios, posts, hashtags, y vulnerabilidades, asegurando relaciones lógicas entre ellas.
  + Ejemplo: Los hashtags se relacionan con las publicaciones mediante una tabla de asociación.
* **Diseño de interfaz:**
  + Desarrollo de wireframes para planificar las pantallas principales, como inicio de sesión, muro de publicaciones y gestión de vulnerabilidades.
  + Enfoque en un diseño intuitivo y minimalista, inspirado en redes sociales modernas.

**3. Desarrollo iterativo**

El desarrollo del proyecto se realizó en ciclos iterativos, cada uno enfocado en un módulo específico.

* **Primera iteración:**
  + Implementación del sistema de usuarios: registro, inicio de sesión y roles (admin, usuario).
  + Creación del sistema básico de publicaciones con texto e imágenes.
* **Segunda iteración:**
  + Integración de vulnerabilidades simuladas, como SQL Injection y LFI, en áreas específicas del sistema.
  + Implementación de la funcionalidad de hashtags, incluyendo autocompletado y relación con publicaciones.
* **Tercera iteración:**
  + Consumo de la API externa para enriquecer los datos mostrados en la plataforma.
  + Mejora de la seguridad del sistema, destacando las diferencias entre vulnerabilidades intencionales y protecciones reales.

**4. Pruebas y validación**

La fase de pruebas incluyó una combinación de técnicas manuales y automáticas para garantizar la funcionalidad y la experiencia del usuario.

* **Pruebas funcionales:**
  + Verificación de funcionalidades básicas, como la creación de publicaciones, gestión de hashtags y consumo de la API.
* **Pruebas de seguridad:**
  + Evaluación de las vulnerabilidades implementadas y su impacto dentro del entorno controlado.
  + Validación de las medidas de protección para evitar exploits fuera del alcance previsto.
* **Correcciones iterativas:**
  + Ajustes realizados según los resultados obtenidos, priorizando estabilidad y usabilidad.

**5. Documentación y presentación**

* **Documentación técnica:**
  + Se elaboró un manual detallado para entender el código, la base de datos y las vulnerabilidades implementadas.
  + Guías educativas sobre cómo funcionan las vulnerabilidades y cómo mitigarlas.
* **Preparación de la presentación:**
  + Elaboración de diapositivas y una demo funcional para exponer el proyecto.
  + Creación de ejemplos prácticos que muestren las vulnerabilidades en acción y su impacto.

**Metodologías empleadas**

* **Desarrollo Ágil:**
  + División del trabajo en iteraciones, permitiendo evaluar y ajustar el proyecto continuamente.
* **Técnica de prueba de caja negra y caja blanca:**
  + Para evaluar la funcionalidad y entender el flujo interno de los datos.
* **Documentación continua:**
  + Registros constantes del progreso y los cambios realizados durante el desarrollo.

Esta metodología permitió que VulnSocial evolucionara desde una idea inicial hasta una herramienta funcional que combina aspectos educativos y técnicos, destacando la importancia de la ciberseguridad en aplicaciones web.

**Objetivos del Proyecto VulnSocial**

**Objetivos Principales**

1. **Crear una red social funcional con un enfoque educativo en ciberseguridad:**
   * Diseñar y desarrollar una aplicación que permita a los usuarios interactuar mediante publicaciones y hashtags.
   * Integrar vulnerabilidades intencionales, como SQL Injection, Local File Inclusion (LFI) y Cross-Site Scripting (XSS), para demostrar los riesgos de seguridad en aplicaciones web reales.
2. **Promover la concienciación sobre vulnerabilidades web:**
   * Mostrar cómo estas vulnerabilidades pueden ser explotadas dentro de un entorno controlado.
   * Explicar métodos de mitigación para cada vulnerabilidad implementada.
3. **Ofrecer una herramienta educativa práctica:**
   * Proveer a estudiantes y desarrolladores un entorno seguro donde puedan aprender a identificar y solucionar vulnerabilidades web.
   * Asegurar que los usuarios comprendan tanto la explotación como la protección de estas fallas de seguridad.

**Objetivos Secundarios**

1. **Fomentar la interacción social simulada:**
   * Implementar funcionalidades como hashtags, publicaciones con imágenes y comentarios para recrear una experiencia de red social moderna.
   * Introducir autocompletado de hashtags relacionado con vulnerabilidades existentes o creadas por el usuario.
2. **Incorporar una API externa para ampliar la funcionalidad:**
   * Conectar VulnSocial con una API externa (como FreeToGame) para demostrar cómo se manejan y muestran datos externos en la aplicación.
3. **Explorar buenas prácticas de desarrollo web:**
   * Documentar las diferencias entre código seguro e inseguro para resaltar la importancia de una buena programación.
   * Asegurar que la aplicación mantenga medidas reales de protección en las áreas fuera del alcance de las vulnerabilidades simuladas.
4. **Facilitar la escalabilidad del proyecto:**
   * Diseñar una arquitectura que permita integrar futuras funcionalidades, como estadísticas de uso, nuevos tipos de vulnerabilidades y roles avanzados para los usuarios.
5. **Demostrar habilidades técnicas y metodológicas:**
   * Evidenciar la capacidad para gestionar un proyecto completo, desde la planificación hasta la entrega.
   * Incluir documentación técnica y educativa para complementar la herramienta.