**Guía Paso a Paso para Implementar la IU de PlatSentinel con Vue.js y Tailwind CSS**

Esta guía te llevará a través del proceso de configuración, creación y prueba de la interfaz de usuario (IU) de **PlatSentinel** utilizando Vue 3, Tailwind CSS, y una estructura modular. Los pasos asumen que estás trabajando en un entorno local (e.g., Windows en C:\Users\Chema\Desktop\PlatSentinel) y que planeas integrar la IU con el backend FastAPI descrito previamente. La fecha y hora actuales (10:06 AM CEST, 11 de julio de 2025) se usarán como referencia para logs iniciales.

**Requisitos Previos**

* **Node.js**: Instala Node.js (versión 16 o superior) desde [nodejs.org](https://nodejs.org/). Verifica con node -v y npm -v en la terminal.
* **Git** (opcional): Para clonar el repositorio si ya lo tienes configurado.
* **Editor de Código**: Usa VS Code con los plugins recomendados (ms-python.python, Vue.volar, Tailwind CSS IntelliSense).
* **Conexión a Internet**: Para descargar dependencias.

**Paso a Paso**

**Paso 1: Configurar el Directorio del Proyecto**

1. **Crea la Carpeta del Proyecto**:
   * Abre una terminal (cmd, PowerShell, o VS Code Terminal).
   * Navega a tu escritorio y crea la carpeta:
   * cd C:\Users\Chema\Desktop
   * mkdir PlatSentinel
   * cd PlatSentinel
   * Crea la subcarpeta para el dashboard:
   * mkdir codigo\_fuente\dashboard
   * cd codigo\_fuente\dashboard
2. **Inicializa el Proyecto Vue**:
   * Ejecuta el siguiente comando para inicializar un proyecto básico (usaremos Vite como herramienta de construcción):
   * npm create vite@latest . -- --template vue
   * Responde a las preguntas:
     + **Project name**: Presiona Enter (usará el directorio actual dashboard).
     + **Select a framework**: Selecciona Vue.
     + **Select a variant**: Selecciona Vue (no TypeScript por simplicidad).
   * Instala las dependencias iniciales:
   * npm install

**Paso 2: Instalar Dependencias Adicionales**

1. **Instala Vue Router**:
   * Agrega el enrutador para manejar las vistas:
   * npm install vue-router@4
2. **Instala Axios**:
   * Para hacer solicitudes HTTP al backend FastAPI:
   * npm install axios
3. **Instala Tailwind CSS**:
   * Instala Tailwind y sus dependencias:
   * npm install -D tailwindcss postcss autoprefixer
   * Inicializa Tailwind:
   * npx tailwindcss init -p
   * Esto creará tailwind.config.js y postcss.config.js.

**Paso 3: Configurar Archivos Iniciales**

1. **Actualiza index.html**:
   * Reemplaza el contenido de public/index.html con el archivo proporcionado:
   * <!DOCTYPE html>
   * <html lang="es">
   * <head>
   * <meta charset="UTF-8">
   * <meta name="viewport" content="width=device-width, initial-scale=1.0">
   * <title>PlatSentinel - Dashboard</title>
   * <script src="https://cdn.tailwindcss.com"></script>
   * </head>
   * <body>
   * <div id="app">
   * <router-view></router-view>
   * </div>
   * <script type="module" src="/src/main.js"></script>
   * </body>
   * </html>
   * Nota: El CDN de Tailwind es temporal; en producción, usa la instalación local (ver Paso 5).
2. **Actualiza main.js**:
   * Reemplaza el contenido de src/main.js con:
   * import { createApp } from 'vue';
   * import App from './App.vue';
   * import router from './router';
   * import './assets/tailwind.css';
   * const app = createApp(App);
   * app.use(router);
   * app.mount('#app');
3. **Crea App.vue**:
   * Crea o reemplaza src/App.vue con:
   * <template>
   * <div>
   * <BarraNavegacion />
   * <div class="container mx-auto p-4">
   * <router-view />
   * </div>
   * </div>
   * </template>
   * <script>
   * import BarraNavegacion from './componentes/BarraNavegacion.vue';
   * export default {
   * components: {
   * BarraNavegacion,
   * },
   * };
   * </script>
4. **Crea el Enrutador (router/index.js)**:
   * Crea la carpeta src/router/ y el archivo index.js con:
   * import { createRouter, createWebHistory } from 'vue-router';
   * import InicioSesion from '../vistas/InicioSesion.vue';
   * import PanelPrincipal from '../vistas/PanelPrincipal.vue';
   * import Scans from '../vistas/Scans.vue';
   * import Reports from '../vistas/Reports.vue';
   * import Services from '../vistas/Services.vue';
   * import Learning from '../vistas/Learning.vue';
   * const routes = [
   * { path: '/login', component: InicioSesion },
   * { path: '/', component: PanelPrincipal, meta: { requiresAuth: true } },
   * { path: '/scans', component: Scans, meta: { requiresAuth: true } },
   * { path: '/reports', component: Reports, meta: { requiresAuth: true } },
   * { path: '/services', component: Services, meta: { requiresAuth: true } },
   * { path: '/learning', component: Learning, meta: { requiresAuth: true } },
   * ];
   * const router = createRouter({
   * history: createWebHistory(),
   * routes,
   * });
   * export default router;

**Paso 4: Crear Componentes y Vistas**

1. **Crea la Carpeta de Componentes**:
   * Crea src/componentes/ y añade:
     + BarraNavegacion.vue:
     + <template>
     + <nav class="bg-gray-800 text-white p-4">
     + <div class="container mx-auto flex justify-between items-center">
     + <div class="text-xl font-bold">PlatSentinel</div>
     + <ul class="flex space-x-6">
     + <li><router-link to="/" class="hover:text-gray-300">Inicio</router-link></li>
     + <li><router-link to="/scans" class="hover:text-gray-300">Escaneos</router-link></li>
     + <li><router-link to="/reports" class="hover:text-gray-300">Informes</router-link></li>
     + <li><router-link to="/services" class="hover:text-gray-300">Servicios</router-link></li>
     + <li><router-link to="/learning" class="hover:text-gray-300">Aprendizaje</router-link></li>
     + <li><a href="/login" class="hover:text-gray-300">Salir</a></li>
     + </ul>
     + </div>
     + </nav>
     + </template>
     + <script>
     + export default {
     + name: 'BarraNavegacion',
     + };
     + </script>
     + TarjetaResumen.vue:
     + <template>
     + <div class="bg-white p-4 rounded-lg shadow-md text-center">
     + <h3 class="text-lg font-semibold">{{ title }}</h3>
     + <p class="text-2xl mt-2">{{ value }}</p>
     + </div>
     + </template>
     + <script>
     + export default {
     + props: {
     + title: String,
     + value: String,
     + },
     + };
     + </script>
     + TablaResultados.vue:
     + <template>
     + <div class="mt-4">
     + <table class="min-w-full bg-white border border-gray-300">
     + <thead>
     + <tr>
     + <th v-for="header in headers" :key="header" class="py-2 px-4 border-b">{{ header }}</th>
     + </tr>
     + </thead>
     + <tbody>
     + <tr v-for="(row, index) in data" :key="index" class="hover:bg-gray-100">
     + <td v-for="cell in row" :key="cell" class="py-2 px-4 border-b">{{ cell }}</td>
     + </tr>
     + </tbody>
     + </table>
     + </div>
     + </template>
     + <script>
     + export default {
     + props: {
     + headers: Array,
     + data: Array,
     + },
     + };
     + </script>
2. **Crea la Carpeta de Vistas**:
   * Crea src/vistas/ y añade:
     + InicioSesion.vue:
     + <template>
     + <div class="flex items-center justify-center min-h-screen bg-gray-100">
     + <div class="bg-white p-6 rounded-lg shadow-lg w-full max-w-md">
     + <h2 class="text-2xl font-bold mb-4 text-center">Iniciar Sesión</h2>
     + <form @submit.prevent="iniciarSesion" class="space-y-4">
     + <div>
     + <label for="username" class="block text-sm font-medium text-gray-700">Usuario</label>
     + <input v-model="username" type="text" id="username" class="mt-1 block w-full p-2 border rounded" required>
     + </div>
     + <div>
     + <label for="password" class="block text-sm font-medium text-gray-700">Contraseña</label>
     + <input v-model="password" type="password" id="password" class="mt-1 block w-full p-2 border rounded" required>
     + </div>
     + <button type="submit" class="w-full bg-blue-500 text-white p-2 rounded hover:bg-blue-600">Iniciar Sesión</button>
     + </form>
     + <p v-if="error" class="text-red-500 text-center mt-2">{{ error }}</p>
     + </div>
     + </div>
     + </template>
     + <script>
     + import axios from 'axios';
     + export default {
     + data() {
     + return {
     + username: '',
     + password: '',
     + error: '',
     + };
     + },
     + methods: {
     + async iniciarSesion() {
     + try {
     + const response = await axios.post('http://localhost:8000/auth/login', {
     + username: this.username,
     + password: this.password,
     + });
     + this.$router.push('/');
     + } catch (err) {
     + this.error = 'Credenciales inválidas';
     + }
     + },
     + },
     + };
     + </script>
     + PanelPrincipal.vue:
     + <template>
     + <div>
     + <h1 class="text-3xl font-bold mb-4">Panel Principal</h1>
     + <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
     + <TarjetaResumen title="Escaneos Realizados" value="25" />
     + <TarjetaResumen title="Informes Generados" value="15" />
     + <TarjetaResumen title="Servicios Activos" value="10" />
     + </div>
     + </div>
     + </template>
     + <script>
     + import TarjetaResumen from '../componentes/TarjetaResumen.vue';
     + export default {
     + components: {
     + TarjetaResumen,
     + },
     + };
     + </script>
     + Scans.vue:
     + <template>
     + <div>
     + <h1 class="text-3xl font-bold mb-4">Escaneos</h1>
     + <button @click="iniciarEscaneo" class="bg-green-500 text-white p-2 rounded hover:bg-green-600">Iniciar Nuevo Escaneo</button>
     + <TablaResultados :headers="['ID', 'Fecha', 'Estado']" :data="scans" />
     + </div>
     + </template>
     + <script>
     + import TablaResultados from '../componentes/TablaResultados.vue';
     + import axios from 'axios';
     + export default {
     + components: {
     + TablaResultados,
     + },
     + data() {
     + return {
     + scans: [
     + { id: 1, fecha: '2025-07-11', estado: 'Completado' },
     + { id: 2, fecha: '2025-07-10', estado: 'En Progreso' },
     + ],
     + };
     + },
     + methods: {
     + async iniciarEscaneo() {
     + const response = await axios.post('http://localhost:8000/scans', {});
     + this.scans.push({ id: this.scans.length + 1, fecha: new Date().toISOString().split('T')[0], estado: 'En Progreso' });
     + },
     + },
     + };
     + </script>
     + Reports.vue:
     + <template>
     + <div>
     + <h1 class="text-3xl font-bold mb-4">Informes</h1>
     + <TablaResultados :headers="['ID', 'Fecha', 'Descarga']" :data="reports" />
     + </div>
     + </template>
     + <script>
     + import TablaResultados from '../componentes/TablaResultados.vue';
     + export default {
     + components: {
     + TablaResultados,
     + },
     + data() {
     + return {
     + reports: [
     + { id: 1, fecha: '2025-07-11', descarga: 'Descargar' },
     + { id: 2, fecha: '2025-07-10', descarga: 'Descargar' },
     + ],
     + };
     + },
     + };
     + </script>
     + Services.vue:
     + <template>
     + <div>
     + <h1 class="text-3xl font-bold mb-4">Servicios</h1>
     + <TablaResultados :headers="['ID', 'Nombre', 'Estado']" :data="services" />
     + </div>
     + </template>
     + <script>
     + import TablaResultados from '../componentes/TablaResultados.vue';
     + export default {
     + components: {
     + TablaResultados,
     + },
     + data() {
     + return {
     + services: [
     + { id: 1, nombre: 'OWASP ZAP', estado: 'Activo' },
     + { id: 2, nombre: 'ClamAV', estado: 'Inactivo' },
     + ],
     + };
     + },
     + };
     + </script>
     + Learning.vue:
     + <template>
     + <div>
     + <h1 class="text-3xl font-bold mb-4">Aprendizaje</h1>
     + <p class="mb-4">Recursos para aprender sobre ciberseguridad:</p>
     + <ul class="list-disc pl-5">
     + <li><a href="#" class="text-blue-500 hover:underline">Guía de Escaneos</a></li>
     + <li><a href="#" class="text-blue-500 hover:underline">Seguridad en PDFs</a></li>
     + </ul>
     + </div>
     + </template>
     + <script>
     + export default {
     + name: 'Learning',
     + };
     + </script>

**Paso 5: Configurar Tailwind CSS**

1. **Edita tailwind.config.js**:
   * Reemplaza el contenido con:
   * module.exports = {
   * content: [
   * './index.html',
   * './src/\*\*/\*.{vue,js,ts,jsx,tsx}',
   * ],
   * theme: {
   * extend: {},
   * },
   * plugins: [],
   * };
2. **Crea tailwind.css**:
   * Crea la carpeta src/assets/ y el archivo tailwind.css con:
   * @tailwind base;
   * @tailwind components;
   * @tailwind utilities;

**Paso 6: Probar la Aplicación**

1. **Inicia el Servidor de Desarrollo**:
   * En la terminal, desde C:\Users\Chema\Desktop\PlatSentinel\codigo\_fuente\dashboard, ejecuta:
   * npm run dev
   * Esto iniciará Vite en http://localhost:5173. Abre este enlace en tu navegador.
2. **Verifica las Páginas**:
   * Navega manualmente a:
     + http://localhost:5173/login para la página de inicio de sesión.
     + http://localhost:5173/ para el Panel Principal (requiere autenticación simulada).
     + http://localhost:5173/scans, /reports, /services, /learning para las otras vistas.
   * Nota: La autenticación no está implementada en el frontend; ajusta router/index.js con un middleware si lo necesitas.
3. **Conecta con el Backend**:
   * Asegúrate de que el backend FastAPI esté corriendo (ver docker-compose.yml de la consulta previa):
   * docker-compose -f C:\Users\Chema\Desktop\PlatSentinel\docker\docker-compose.yml up --build
   * La solicitud en InicioSesion.vue usa http://localhost:8000/auth/login. Si usas un VPS, reemplaza la URL con la IP del servidor (e.g., http://<IP\_VPS>:8000/auth/login).

**Paso 7: Depuración y Mejoras**

1. **Configura VS Code**:
   * Instala los plugins Vue - Official, Tailwind CSS IntelliSense, y ESLint.
   * Crea o edita .vscode/settings.json con:
   * {
   * "editor.formatOnSave": true,
   * "[vue]": {
   * "editor.defaultFormatter": "esbenp.prettier-vscode"
   * },
   * "tailwindCSS.includeLanguages": {
   * "vue": "html"
   * }
   * }
2. **Agrega Logging (Opcional)**:
   * Integra el logger del backend (logging\_config.py) en las vistas Vue usando un plugin como vue-logger:
   * npm install vue-logger
   * Añade al main.js:
   * import VueLogger from 'vue-logger';
   * app.use(VueLogger, { level: 'debug' });
3. **Prueba Responsividad**:
   * Redimensiona la ventana del navegador o usa las herramientas de desarrollo (F12) para verificar que los diseños Tailwind sean responsivos.

**Paso 8: Despliegue (Opcional)**

1. **Construye para Producción**:
   * Ejecuta:
   * npm run build
   * Esto genera una carpeta dist/ con los archivos estáticos.
2. **Despliega en VPS**:
   * Sube dist/ al VPS usando SCP o FTP.
   * Sirve los archivos con un servidor web (e.g., Nginx) o intégralo en docker-compose.yml:
   * dashboard:
   * build: ./codigo\_fuente/dashboard
   * ports:
   * - "8080:80"
   * volumes:
   * - ./dist:/usr/share/nginx/html
   * image: nginx:latest

**Resultados Esperados**

* Al visitar http://localhost:5173/login, verás un formulario de inicio de sesión.
* Tras "iniciar sesión" (simulado), navega a / para el Panel Principal con tarjetas de resumen.
* Las rutas /scans, /reports, /services, y /learning mostrarán tablas o listas con datos estáticos.
* Los estilos Tailwind (colores, grids, hover) estarán aplicados.

**Notas Adicionales**

* **Timestamp Actual**: Los logs iniciales (si integras vue-logger) registrarán 10:06 AM CEST, 11 de julio de 2025.
* **Integración con Backend**: Implementa la lógica JWT en autenticacion.py para validar las credenciales de InicioSesion.vue.
* **Imágenes**: Si deseas un logo o gráficos, confirma y puedo generarlos.
* **Estructura**: Esta IU se alinea con codigo\_fuente/dashboard/ de la estructura previa; ajusta rutas si usas un ZIP existente.

Si encuentras errores (e.g., puertos ocupados, dependencias faltantes) o necesitas más detalles (e.g., autenticación, Dockerización), ¡avísame! También puedo generar un diagrama Mermaid del flujo de navegación o un archivo ZIP con esta estructura.