**Anteproyecto: Plataforma Web para Localización de Spots Fotográficos**

**1. Título**

**SpotMap: Plataforma Web Colaborativa para Descubrir y Compartir Spots Fotográficos Urbanos y Naturales**

**2. Introducción**

**2.1 Contexto**

En la actualidad, el uso de redes sociales como Instagram, TikTok y Facebook ha generado una tendencia creciente en la búsqueda de lugares fotogénicos. Estos sitios, conocidos popularmente como *spots*, se convierten en atractivos no solo para influencers, sino también para turistas y ciudadanos que desean descubrir nuevos entornos para capturar recuerdos.

A pesar de esta demanda, no existe una plataforma centralizada que reúna spots fotográficos categorizados, geolocalizados y con la posibilidad de ser alimentados por la propia comunidad. La mayoría de usuarios recurre a blogs, hilos en redes sociales o recomendaciones personales.

El desarrollo de una aplicación web que permita identificar, compartir y valorar estos spots podría cubrir un vacío en el mercado digital y servir como recurso útil tanto para particulares como para organismos turísticos.

**2.2 Situación actual**

Actualmente existen plataformas como:

* **Google Maps**: permite marcar lugares, pero no está especializado en spots fotográficos.
* **Pinterest**: ofrece imágenes, pero sin geolocalización precisa.
* **Aplicaciones de turismo locales**: en muchos casos carecen de usabilidad y actualización constante.

Esto demuestra una oportunidad de crear una aplicación centrada exclusivamente en spots visuales, con un enfoque colaborativo.

**2.3 Fortalezas del proyecto**

* Uso de **tecnologías web modernas** escalables (frameworks, bases de datos geoespaciales).
* **Contribución comunitaria**, lo que garantiza constante actualización.
* Potencial de **integración con redes sociales** para viralización.
* Puede convertirse en una **herramienta turística oficial** para municipios.

**2.4 Vulnerabilidades del proyecto**

* Riesgo de que los usuarios suban contenido irrelevante o inapropiado.
* Dependencia de la masa crítica de usuarios para que el sistema sea útil.
* Competencia indirecta con grandes plataformas.
* Necesidad de políticas de moderación de contenidos y protección de datos personales.

**3. Objetivo e Innovaciones**

**3.1 Objetivo general**

Desarrollar una aplicación web colaborativa que permita a los usuarios descubrir, compartir y valorar spots fotográficos geolocalizados, con un diseño escalable, accesible y adaptable a distintos dispositivos.

**3.2 Objetivos específicos**

* Implementar un sistema de registro e inicio de sesión seguro.
* Integrar un mapa interactivo con marcadores de spots.
* Permitir a los usuarios subir spots con coordenadas, fotos y descripciones.
* Incluir un sistema de valoración y comentarios.
* Incorporar categorías de spots (naturaleza, urbano, panorámico, nocturno, etc.).
* Preparar la aplicación para futuras integraciones con IA (reconocimiento automático de lugares a partir de fotos).

**3.3 Innovaciones**

* **Mapa colaborativo especializado** en spots fotográficos, no en lugares generales.
* Posibilidad de **gamificación**: recompensas a usuarios que más contribuyan.
* **Escalabilidad**: diseño inicial web, pero adaptable a aplicación móvil.
* **Interfaz optimizada para fotografía**: vista previa de imágenes directamente en el mapa.

**4. Medios de HW y SW**

**4.1 Hardware**

* **Servidor Cloud** (ej. AWS, Azure o Google Cloud):
  + CPU: 4 cores
  + RAM: 8 GB
  + Almacenamiento: 200 GB SSD (escalable)
* Ordenador personal para desarrollo:
  + CPU i5 o superior
  + RAM 8 GB
  + 500 GB HDD/SSD
* Dispositivos móviles para pruebas (Android/iOS).

**4.2 Software**

* **Frontend**: React, TailwindCSS, Leaflet (mapas).
* **Backend**: Node.js con Express.
* **Base de datos**: MongoDB (con soporte para geolocalización).
* **Control de versiones**: GitHub/GitLab.
* **Gestión de proyecto**: Trello/Jira.
* **Herramientas de diseño**: Figma, Canva.
* **Testing**: Jest, Cypress.

**5. Planificación**

**5.1 Metodología**

Se seguirá una metodología **ágil (Scrum)**, con iteraciones de 2 semanas y entregas funcionales.

**5.2 Fases**

1. **Análisis y diseño (2 semanas)**
   * Estudio de requisitos.
   * Diseño de arquitectura.
   * Wireframes iniciales.
2. **Desarrollo inicial (4 semanas)**
   * Implementación del backend.
   * Configuración de base de datos.
   * API REST para spots.
3. **Desarrollo frontend (4 semanas)**
   * Implementación de interfaz.
   * Integración del mapa interactivo.
   * Registro/login.
4. **Pruebas y validación (2 semanas)**
   * Tests unitarios y de integración.
   * Corrección de errores.
5. **Optimización y despliegue (2 semanas)**
   * Despliegue en servidor cloud.
   * Documentación técnica.
   * Presentación del proyecto.

Duración total estimada: **14 semanas (3,5 meses)**.

**6. Bibliografía**

* Sommerville, I. (2011). *Ingeniería del Software*. Pearson.
* Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach*. McGraw-Hill.
* Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
* LeafletJS Documentation: https://leafletjs.com/
* MongoDB Documentation: <https://www.mongodb.com/docs/>
* React Documentation: https://react.dev/
* Scrum Guides: https://scrumguides.org/