

FACULTAD: INFORMÁTICA Y ELECTRÓNICA**CARRERA: SOFTWARE**

ESTUDIANTE:	CÉSAR GINES ARTURO BADILLO	CÓDIGO:	7237 6857
ASIGNATURA:	APLICACIONES INFORMÁTICAS II	PAO:	OCTAVO
PERIODO ACADÉMICO:	OCTUBRE 2024 – MARZO 2025		
FECHA DE ENTREGA:	18 DE NOVIEMBRE DEL 2024		

Actividad 5: Definición de tecnologías, herramientas, frameworks y lenguajes de programación

En nuestra red social basada en blockchain con un sistema de incentivos de tokens por cada interacción de usuario “Reppost”, estamos utilizando una combinación de lenguajes de programación, tecnologías, frameworks y herramientas para desarrollar tanto frontend como backend.

Lenguajes de programación

- HTML y PHP: Para la estructura frontend (formularios, vistas, etc.) y para el desarrollo del backend, incluida la lógica del sistema y la conexión con la base de datos
- CSS: Para el diseño y estilo del frontend
- JavaScript (JS): Para la interactividad en el frontend y funcionalidades avanzadas en el navegador.
- Solidity: Ya que es requerido para los contratos inteligentes para otorgar valor a los tokens de recompensa, aún no confirmado, pero es común su uso para blockchain.

Tecnologías y Frameworks

- Laravel: Para estructurar el backend en PHP
- Fabric.js: Para la funcionalidad avanzada de dibujo en la pizarra de tu red social.
- phpMyAdmin: Como herramienta de administración de gestión para la base de datos MySQL
- MySQL: Para el manejo de la base de datos relacional que almacena información de usuarios, interacciones y blockchain.
- API de Canvas de HTML5 para la pizarra de creación de dibujos
- API para la generación de imágenes a través de IA implementado en la pizarra
- Bootstrap: Framework css que facilita paginas adaptables y estilizadas
- Fluent icons y similares: librerías de iconos para personalizar mejor la pagina

Entorno de desarrollo

- XAMPP: Entorno de desarrollo local que incluye Apache (servidor web) y MySQL.
- VS Code: Como IDE para escribir y gestionar el código.
- Ganache: Permitirá probar y simular las interacciones de usuario, creando y validando los bloques para otorgar los tokens, además nos sirve como un servidor para pruebas de la aplicación.

Herramientas de apoyo

- Git/GitHub: Para control de versiones y colaboración entre los integrantes del grupo

Promovedor de servicios de Host (por elegir)

- Hostinger
- Vercel
- AWS
- GoDaddy
- Arsys

Cuadros comparativos

Lenguaje de programación	Ventajas Elegidas	Comparación con Otros Lenguajes	Razón de Elección
HTML	Esencial para la estructura del frontend, permite la integración con CSS y JS.	Markdown: Más sencillo para documentación, pero no adecuado para vistas interactivas.	Necesario para cualquier desarrollo web, se complementa perfectamente con CSS y JS.
PHP	Integración eficiente con bases de datos como MySQL. Laravel optimiza la gestión de proyectos en PHP.	Python (Django): Más moderno y versátil, pero tiene curva de aprendizaje mayor para proyectos como este.	Su combinación con Laravel y el conocimiento previo del equipo lo hace más rápido de implementar.
CSS	Permite diseño adaptable y estilizado con frameworks como Bootstrap.	SASS: Más potente para personalización avanzada, pero añade complejidad en proyectos pequeños o medianos.	Ideal para proyectos que buscan un diseño limpio y funcional sin complicaciones extra.
JavaScript	Agrega interactividad avanzada y es compatible con	TypeScript: Más robusto, pero requiere más configuración inicial.	JS es suficiente para la interactividad del proyecto y tiene

	frameworks populares.		mayor soporte en navegadores.
Solidity	Específico para contratos inteligentes en blockchain, ampliamente adoptado en Ethereum y otras cadenas.	Rust: Más seguro y eficiente, pero menos usado en blockchain comercial y con menor documentación.	Solidity tiene una curva de aprendizaje más baja y es ampliamente utilizado en proyectos similares.

Tecnología Framework	Ventajas Elegidas	Comparación con Otros	Razón de Elección
Laravel	Estructura modular, robustez, y soporte extenso para proyectos PHP.	Symfony: Más complejo de aprender, pero con mayor personalización.	Laravel es más intuitivo y tiene mayor soporte comunitario para proyectos de tamaño medio como <i>Reppost</i> .
Fabric.js	Ideal para manejar elementos de dibujo en canvas con gran interactividad.	Konva.js: Más avanzado en animaciones, pero con menos documentación en comparación a Fabric.js.	Perfecto para funcionalidades avanzadas de dibujo sin añadir complejidad innecesaria.
phpMyAdmin	Herramienta gráfica fácil de usar para administrar bases de datos MySQL.	Adminer: Más ligero, pero con menos opciones gráficas y menos intuitivo para usuarios nuevos.	phpMyAdmin tiene mejor documentación y soporte para equipos en formación.
Bootstrap	Facilita la creación de diseños responsivos con gran cantidad de componentes prediseñados.	TailwindCSS: Más flexible, pero requiere más tiempo para configuraciones personalizadas.	Bootstrap es más fácil de usar y ofrece resultados rápidos para diseño web.
API Canvas HTML5	Potente para integrar gráficos en navegadores de manera nativa.	SVG: Mejor para gráficos estáticos, pero menos eficaz para animaciones dinámicas.	La API Canvas es perfecta para dibujos dinámicos y herramientas creativas como la pizarra en <i>Reppost</i> .

Entorno	Ventajas Elegidas	Comparación con Otros	Razón de Elección
XAMPP	Combina Apache, MySQL y PHP en un solo paquete fácil de instalar y configurar.	MAMP: Similar a XAMPP pero con enfoque en macOS, menos soporte multiplataforma.	Es multiplataforma y adecuado para proyectos basados en PHP como <i>Reppost</i> .
VS Code	Gran soporte de extensiones, depurador integrado y fácil integración con GitHub.	IntelliJ IDEA: Más robusto, pero requiere licencia paga para todas sus funcionalidades.	VS Code es gratuito, ligero y extensible, ideal para el equipo en formación.
Ganache	Simulación rápida de blockchain y pruebas de contratos inteligentes en un entorno local.	Hardhat: Más avanzado para desarrolladores experimentados en blockchain, pero con curva de aprendizaje más pronunciada.	Ganache ofrece todo lo necesario para validar la lógica blockchain sin complicaciones adicionales.

Herramientas de apoyo	Ventajas Elegidas	Comparación con Otros	Razón de Elección
Git/GitHub	Control de versiones en la nube, ideal para colaboración en equipos distribuidos.	GitLab: Más personalizable, pero requiere mayor configuración y recursos en servidores propios.	GitHub es más accesible para equipos pequeños y proporciona integración inmediata.

Proveedor	Ventajas	Desventajas	Recomendación para Reppost
Hostinger	Económico, panel intuitivo, y soporte 24/7.	Rendimiento limitado para proyectos de alto tráfico.	Ideal para comenzar el proyecto con un presupuesto ajustado.
Vercel	Optimizado para aplicaciones modernas basadas en JavaScript y frameworks como Next.js.	Limitado en soporte para tecnologías como PHP y MySQL.	Bueno para un frontend moderno, pero insuficiente para el backend de <i>Reppost</i> .

AWS	Escalabilidad, servicios avanzados, y soporte para blockchain y bases de datos relacionales.	Costos altos y mayor complejidad en configuración.	Ideal si se prevé un crecimiento rápido en usuarios e interacciones.
GoDaddy	Fácil configuración inicial y gran reconocimiento.	Menor rendimiento en planes básicos y opciones limitadas para tecnologías modernas.	Útil para alojar páginas informativas, pero no recomendado para una red social compleja.
arsys	Soporte en español, planes dedicados y buena estabilidad.	Menos integración con tecnologías emergentes como blockchain.	Puede ser una opción si se busca soporte local, pero no es óptimo para proyectos avanzados.

Aquí en este último cuadro, podemos decir que, para la red social “*Reppost*”, Hostinger es una excelente opción para iniciar el proyecto, mientras que AWS es más adecuado para una etapa de crecimiento donde se necesite escalabilidad y soporte avanzado, esto como un vistazo a futuro sin inclinarnos por uno más que otro.