Definición de Frameworks

Al haber seleccionado los lenguajes JavaScript y Motoko como principales para el desarrollo del software y considerando las herramientas disponibles en la web3 e internet computer, hemos definido como la mejor opción para un desarrollo nativo, descentralizado y eficiente los siguientes frameworks que se adaptan perfectamente al desarrollo en web3.

DFINITY

Para el backend se seleccionó DFINITY, una herramienta proporcionada directamente por los desarrolladores de internet computer que se compone de las siguientes características:

Herramientas de Desarrollo: Incluye herramientas como dfx que permiten compilar, desplegar y gestionar canisters (unidades de cómputo) en la red de Internet Computer.

Lenguaje de Programación: Facilita el desarrollo en Motoko, el lenguaje de programación específico para IC, proporcionando soporte para escribir y compilar código Motoko.

Integración con IDEs: Proporciona integración con entornos de desarrollo integrados (IDEs) para facilitar la escritura, depuración y gestión de proyectos en IC.

Depuración y Testeo: Incluye herramientas para depurar y probar aplicaciones en el entorno de IC, permitiendo a los desarrolladores verificar y optimizar el rendimiento de sus aplicaciones.

Desarrollo de Canisters: Permite la creación de canisters, que son las unidades de cómputo en IC que encapsulan lógica de negocio y datos.

Gestión de Identidad: Ayuda en la gestión de identidades y autorización para interactuar con canisters y otros servicios en la red.

Interfaz con la Red de IC: Facilita la comunicación con la red de IC para desplegar y operar aplicaciones descentralizadas de manera eficiente y segura.

AZLE

Al tener conocimiento de JavaScript y ser novatos en el tema de internet computer, Azle se postula como un framework ideal al simplificar el desarrollo mediante la implementación de typescript para el desarrollo de canisters sus beneficios al proyecto serían los siguientes:

1. Facilita la Integración de Desarrolladores de JavaScript

Azle está diseñado específicamente para permitir a los desarrolladores de JavaScript trabajar de manera eficiente en Internet Computer. Esto significa que proporciona una capa de abstracción y herramientas que facilitan la escritura de código y la interacción con los canisters utilizando JavaScript.

2. Integración con TypeScript

Azle ofrece soporte para TypeScript, lo cual es beneficioso al requerir características que TypeScript proporciona sobre JavaScript. Esto mejora la calidad del código, reduce errores y facilita el mantenimiento a medida que el proyecto crece en complejidad.

3. Simplifica la Interacción con Canisters

Azle proporciona abstracciones y utilidades que simplifican la interacción con los canisters en Internet Computer. Esto incluye métodos para la gestión de identidades, manejo de autorización, y facilita la llamada a funciones y métodos en los canisters de manera más intuitiva y estructurada desde JavaScript.

4. Mejora la Productividad del Desarrollador

Al utilizar Azle, los desarrolladores pueden enfocarse más en la lógica de producto y menos en los detalles técnicos específicos de Internet Computer. Esto puede acelerar el desarrollo, reducir el tiempo de aprendizaje necesario para trabajar en IC y permitir a los equipos de desarrollo aprovechar al máximo las capacidades de la plataforma.

5. Documentación y Comunidad

Azle es respaldado por una comunidad activa y una documentación adecuada que facilita el aprendizaje y la resolución de problemas. Esto es crucial para los desarrolladores que novatos en Internet Computer.

REACT

Al trabajar de manera activa con DFX, React es una de las primeras y mas atractivas opciones que nos proporcionan al momento de estipular con que framework se desea trabajr el frontend del proyecto y no es para menos ya que react contiene una cantidad significativa de beneficios, usar React en Internet Computer ofrece varias ventajas significativas que hacen que sea una opción atractiva para desarrollar aplicaciones descentralizadas en esta plataforma como lo son las siguientes:

1. Componentes y reutilización de código:

React se basa en el concepto de componentes, lo que permite dividir la interfaz de usuario en pequeños fragmentos reutilizables de código. Esto facilita el modularidad y la mantenibilidad del código, lo cual es crucial al desarrollar aplicaciones complejas en IC.

2. Virtual DOM y Rendimiento Optimizado

React utiliza un Virtual DOM (DOM virtual), que es una representación ligera del DOM real. Esto permite que React realice actualizaciones eficientes y minimice las manipulaciones directas del DOM, lo cual es beneficioso para aplicaciones que deben ejecutarse de manera eficiente en Internet Computer, optimizando el uso de recursos.

3. Ecosistema y Librerías

React tiene un ecosistema robusto y activo de librerías y herramientas que facilitan el desarrollo de aplicaciones frontend modernas. Esto incluye herramientas de estado global como Redux, enrutamiento con React Router, y librerías de componentes como Material-UI, entre otras. Estas librerías pueden ser utilizadas también en proyectos de IC para acelerar el desarrollo y mejorar la calidad de las aplicaciones.

4. Flexibilidad y Adaptabilidad

React es conocido por su flexibilidad y capacidad de adaptarse a diferentes necesidades y entornos de desarrollo. Esto es importante en IC, donde las aplicaciones deben estar preparadas para escenarios descentralizados y manejar la lógica de proyecto de manera eficiente.

5. Comunidad y Soporte

React cuenta con una comunidad grande y activa de desarrolladores y empresas que respaldan su desarrollo y mejora continua. Esto asegura que haya una abundante cantidad de recursos educativos, tutoriales, y soluciones para resolver problemas que puedan surgir al desarrollar en IC con React.