Introducción a las recetas de Dory web app

¡Bienvenido al conjunto de documentos de recetas para la operación y mantenimiento de nuestra aplicación web! Este conjunto de documentos ha sido diseñado para proporcionar instrucciones claras y detalladas sobre cómo realizar diferentes procedimientos en nuestra aplicación, con el fin de garantizar su correcto funcionamiento y permitir que cualquier persona encargada pueda llevar a cabo modificaciones de manera eficiente.

En este documento principal, encontrarás una descripción general del contenido de las diferentes recetas, así como información útil sobre cómo utilizarlas. Desde cómo preparar el entorno de desarrollo, hasta cómo agregar una nueva opción de menú en el navbar para consultar las granjas mejor puntuadas, pasando por cómo cambiar globalmente los colores de Dory y crear un nuevo módulo para uso de lazy load, cada una de las recetas ha sido cuidadosamente elaborada para proporcionar la información necesaria de manera clara y concisa.

Las recetas específicas que encontrarás en este conjunto de documentos son:

Receta 1: Preparar entorno de desarrollo

Receta 2: Descargar o clonar código de Dory web app

Receta 3: Cómo ejecutar Dory web app en un servidor de desarrollo

Receta 4: Configurar variables de entorno en el ambiente de desarrollo

Receta 5: Cómo desplegar Dory web app en Heroku

Receta 6: Cómo configurar las variables de entorno en Heroku

Receta 7: Cómo desplegar el Api rest de Dory en Heroku

Receta 8: Cómo modificar los componentes fijos e intercambiables de Dory web app

Receta 9: Cómo desplegar cambios nuevos a Heroku

Receta 10: Agregar una nueva opción de menú en el navbar para consultar las granjas mejor puntuadas

Receta 11: Agregar un nuevo servicio

Receta 12: Cómo cambiar globalmente los colores de Dory

Receta 13: Cómo cambiar la fuente primaria

Receta 14: Creación de un proyecto en Firebase y generar el objeto "firebaseConfig" para configurar las variables de entorno en Heroku

Receta 15: Creación de un nuevo módulo para uso de lazy load

Esperamos que este conjunto de documentos sea una herramienta valiosa para todos aquellos que trabajan en el mantenimiento y la operación de nuestra aplicación web.

Glosario

**Node.js** es un intérprete que ejecuta código JavaScript fuera del navegador web, es decir, en el lado del servidor. Node.js utiliza el motor de JavaScript V8 de Google Chrome para interpretar el código JavaScript y proporciona una amplia gama de funciones y bibliotecas para la creación de aplicaciones de servidor y de línea de comandos.

**Angular CLI** es una interfaz de línea de comandos (CLI) para Angular, que es un marco de trabajo (framework) de JavaScript utilizado para crear aplicaciones web del lado del cliente. Angular CLI proporciona un conjunto de herramientas de línea de comandos para crear, compilar, probar y desplegar aplicaciones Angular de manera más eficiente. Con Angular CLI, puedes crear fácilmente componentes, módulos, servicios y otros artefactos de Angular, y también pueden utilizar bibliotecas externas para mejorar la funcionalidad de sus aplicaciones. En resumen, es una herramienta que facilita la creación y mantenimiento de proyectos de Angular.

**GIT** es un sistema de control de versiones distribuido de código abierto, que se utiliza para el desarrollo de software y la gestión de código fuente. Permite un seguimiento de los cambios realizados en el código de la aplicación a lo largo del tiempo, así como colaborar con otros desarrolladores en un proyecto. Con GIT, se pueden crear ramas (branches) para trabajar en diferentes versiones del código, realizar cambios y combinaciones (merges) de ramas, y mantener un registro detallado de los cambios realizados en cada archivo.

**Visual Studio Code** es un editor de código fuente desarrollado por Microsoft, que se utiliza para programar en diversos lenguajes de programación como JavaScript, TypeScript, Python, PHP, C#, entre otros. Visual Studio Code es un editor de código gratuito y de código abierto, que se ejecuta en Windows, macOS y Linux. El editor cuenta con una amplia variedad de características, como la finalización de código (code completion), el resaltado de sintaxis, la depuración de código, la integración con control de versiones como Git, la extensibilidad a través de una gran cantidad de complementos y la posibilidad de personalizar la interfaz de usuario a través de temas y configuraciones. Visual Studio Code también cuenta con una gran comunidad de desarrolladores que contribuyen con extensiones y herramientas para el editor.