Guión de Inspección para el Fragmento de Código- vm´s

1. Planeamiento y Documentos de Entrada

- **Líder de la Inspección**: Asignado a una persona con conocimiento profundo de React, Next.js, y TypeScript (puede ser un líder técnico).
- **Inspectores**: Se designarán entre 3-4 inspectores. Es importante que los inspectores incluyan al menos:
    - Un desarrollador con experiencia en React y Next.js.
    - Un desarrollador experimentado en buenas prácticas de TypeScript.
    - Un miembro de Aseguramiento de Calidad (QA).
- **Documentación Necesaria**:
    - Especificaciones del proyecto, que incluya la estructura esperada para la funcionalidad del formulario y la API.
    - Estándares de codificación definidos para el equipo (estilo de código, estructura de carpetas, manejo de errores, validación, etc.).
    - Reglas para la interacción cliente-servidor usando Axios y Next.js.
- **Estándares de Revisión**:
    - Buenas prácticas de React/TypeScript.
    - Normas de manejo de estado y efectos secundarios en React.
    - Requisitos de seguridad en las solicitudes HTTP (manejo de errores, validaciones, etc.).
- **Programa**: El líder define una fecha para la reunión rápida y la inspección, y distribuye el código entre los inspectores.

Producto de esta Fase: Plan maestro de la inspección, con asignación de roles y un cronograma claro.

2. Reunión Rápida (15 minutos)

- **Objetivo**: Explicar a los inspectores los objetivos de la inspección y proporcionar las instrucciones necesarias.
- **Tareas**:
    - El líder define los plazos y objetivos de la inspección.
    - Se entregan los criterios de evaluación.
    - Se explican los aspectos principales del código y la estructura del mismo:
        1. El fragmento evalúa un formulario que permite al usuario agregar o editar información.
        2. Usa React, Next.js y Axios para la interacción entre frontend y backend.
        3. Tiene que cumplir con buenas prácticas de manejo de estado, validación, y manejo de errores.
- **Instrucciones para los Inspectores**: Deben centrarse en encontrar problemas de eficiencia, correctitud y completitud en el código.

Producto de esta Fase: Alineamiento del equipo en los objetivos y proceso de inspección.

3. Inspección o Comprobación

- **Método**: Los inspectores realizan la inspección individualmente, registrando los defectos en una tabla que se entregará al líder. Deben buscar problemas en las siguientes áreas:
    1. **Correctitud**: Verificar si el código cumple con los requerimientos funcionales.
    2. **Completitud**: Asegurarse de que no faltan elementos importantes, como manejo de errores o validaciones.
    3. **Eficiencia**: Revisar si hay mejoras de rendimiento que se puedan realizar.
    4. **Seguridad**: Verificar la correcta validación de los datos ingresados y la protección contra ataques (p.ej., inyecciones o solicitudes maliciosas).
- **Ejemplos de Defectos**:
    1. Falta de manejo de errores en las solicitudes a la API.
    2. Falta de validación adecuada en los campos de entrada del formulario.
    3. Importaciones duplicadas que podrían simplificarse.
    4. Estado de "loading" ausente, lo que puede llevar a una mala experiencia de usuario.
- **Registro de Defectos**: Cada inspector registra cada defecto encontrado y lo clasifica según su severidad (crítico, mayor, menor).

Producto de esta Fase: Tabla con los defectos identificados y clasificados.

4. Reunión de Registro (2 horas máximo)

- **Objetivo**: Discutir los defectos encontrados y decidir su aceptación o rechazo.
- **Participantes**: Todos los inspectores, el líder y el autor del código.
- **Agenda**:
    1. Cada inspector presenta sus hallazgos de manera organizada.
    2. El líder organiza los defectos en un registro general y decide si el defecto es aceptado o rechazado.
    3. Se clasifica cada defecto como **critico**, **mayor** o **menor**, con base en su impacto en la funcionalidad y el rendimiento.
    4. Se acuerda el orden de corrección de los defectos aceptados.

Producto de esta Fase: Registro general de defectos aceptados y plan de corrección.

5. Tormenta de Ideas (5-30 minutos)

- **Objetivo**: Generar soluciones para los defectos identificados.
- **Participantes**: Los inspectores, el autor del código y el líder.
- **Tareas**:
    1. Presentar sugerencias para corregir los defectos.
    2. Debatir si hay mejores enfoques para manejar ciertas funcionalidades, como el uso de estados, validaciones y la estructura del código.
    3. Evaluar soluciones para mejorar la seguridad, rendimiento o experiencia del usuario.
    4. El anotador realiza el registro de una lista de soluciones según los acuerdos tomados en esta sesión para ser compartidos posteriormente

Producto de esta Fase: Lista de soluciones y mejoras sugeridas para el código.

6. Edición (Acción de Corrección)

- **Objetivo**: El autor del código toma las acciones necesarias para corregir los defectos encontrados.
- **Tareas**:
    - Implementar las soluciones acordadas en la tormenta de ideas.
    - Asegurarse de que los cambios realizados no rompan ninguna otra parte del sistema.
    - Actualizar la documentación si es necesario.

Producto de esta Fase: Código corregido y listo para revisión.

7. Seguimiento

- **Objetivo**: Asegurar que las correcciones se han implementado correctamente.
- **Tareas**:
    - El líder verifica con el autor del código que los defectos se han corregido de manera satisfactoria.
    - Los inspectores pueden realizar una segunda revisión rápida para validar que los cambios sean correctos y no introduzcan nuevos problemas.

Producto de esta Fase: Confirmación de que las correcciones se han realizado correctamente.

8. Salida

- **Objetivo**: Asegurar que el código está listo para producción.
- **Tareas**:
    - Revisión final por parte del líder y el equipo de inspección.
    - Generación de un informe de inspección final que documente el proceso, los defectos corregidos, y las soluciones aplicadas.

Producto de esta Fase: El código está listo para ser considerado como parte de la base de código principal (o para ser implementado en producción).