

## Tarea 7: Identificación de Aspectos de Control de Calidad del Proyecto

### 1. Calidad del Código

- **Revisión del Código:**
  - Se realizarán revisiones periódicas del código en GitHub.
- **Criterios de calidad:**
  - Código limpio y comprensible (sin redundancias).
  - Organización en paquetes y clases correctamente estructuradas.
- **Uso de Librería Jackson para JSON:**
  - Pruebas de serialización y deserialización de datos.
  - Validar que los objetos Java se conviertan correctamente a JSON y viceversa.
- **Control de Versiones:**
  - El proyecto se almacenará y gestionará en GitHub.
  - Posibles ramas:
    - main: Código estable.
    - develop: Funcionalidades en desarrollo.
    - Ramas individuales para cada tarea.

Si el código está bien estructurado, será más fácil mantener y actualizar la aplicación. Además, revisar el código y hacer pruebas unitarias garantiza que la aplicación no tenga errores críticos que puedan afectar a los usuarios finales.

### 2. Calidad de la Funcionalidad

- **Validación de Requisitos:**
  - Comprobar que las funcionalidades implementadas cumplen con los requisitos iniciales.
  - Métodos: Formulario de ayuda a víctimas claro y fácil de usar.
- **Pruebas de Integración:**
  - Verificar que los diferentes módulos de la aplicación trabajen de forma conjunta sin fallos.
  - Herramientas: Postman (para probar APIs), simulación de flujos de usuario.

Es crucial verificar que la aplicación cumpla con sus objetivos: recoger información de los afectados, mostrar datos útiles (como ayudas disponibles, organizaciones involucradas, etc.) y facilitar la comunicación.

### **3. Calidad de la Experiencia de Usuario (UX/UI)**

- **Evaluación de Usabilidad:**
  - Probar si la aplicación es intuitiva y fácil de usar.
  - Métodos: Encuestas o feedback de usuarios finales.
- **Compatibilidad con Dispositivos y Navegadores:**
  - Pruebas en diferentes tamaños de pantalla (responsive design).
  - Herramientas: BrowserStack, pruebas manuales.

Las víctimas de la DANA y los voluntarios necesitan una interfaz clara, rápida y fácil de usar. En esta situación de emergencia, no tienen tiempo para navegar por una aplicación compleja.

### **4. Rendimiento y Seguridad**

- **Optimización del Rendimiento:**
  - Medir tiempos de carga, respuestas de formularios, etc.
  - Herramientas: Lighthouse, Google PageSpeed.
- **Seguridad de Datos:**
  - Validar que los datos sensibles de los usuarios están protegidos.
  - Acciones: Cifrado de datos, protección de contraseñas.

La aplicación debe funcionar rápido y sin errores, incluso si muchas personas la usan al mismo tiempo. Además, la seguridad es fundamental porque se manejan datos sensibles de personas afectadas.

### **5. Organización del Trabajo con Trello**

En Trello, organiza las tareas en **listas** como:

- 1. Por Hacer**
  - a. Crear pruebas unitarias.
  - b. Revisar diseño UX.
  - c. Realizar pruebas de rendimiento.
- 2. En Progreso**
  - a. Revisar código en GitHub.
  - b. Realizar pruebas de integración.
- 3. Hecho**

- a. Subida de documentación en GitHub.
- b. Verificación de seguridad de datos.

## **6. Presentación y Almacenamiento**

- **GitHub:**
  - Subir la versión final del documento en formato PDF.
  - Mantener el repositorio organizado con etiquetas de versiones y ramas (versionado).