Guía Choni - Cambios y Cosas Hechas en el Proyecto (Desde las 22h de Ayer hasta Ahora)

==========

1. Problema Inicial:

- No se mostraban las imágenes específicas de cada restaurante; todas tenían las mismas fotos.
- Se permitía subir imágenes duplicadas para un restaurante.
- Las solicitudes de usuarios y restaurantes no aparecían en el panel admin.
- El panel admin mostraba "No hay solicitudes" incluso cuando la base de datos tenía registros.
- Al intentar acceder a ciertas APIs con SSL daba error de "wrong version number".

2. Cambios y Soluciones Implementadas:

A. Imágenes en Restaurantes:

- Ajustamos la API backend para que devuelva las imágenes específicas por restaurante.
- En el frontend Angular, corregimos la ruta base y la carga de imágenes para que se lean desde la carp
- Añadimos validaciones estrictas en el componente de subida para evitar que se repitan imágenes (por

B. Solicitudes de Modificación y Bajas en Admin:

- Cambiamos en el servicio AdminService que solo devuelva solicitudes no gestionadas (gestionada == f
- Corregimos el método en AdminController que obtiene las solicitudes de modificación de usuarios y res
- Añadimos logs para verificar que la llamada a la API sí devuelve datos.
- En el frontend Angular, nos aseguramos que el componente admin carga correctamente las solicitudes

C. Problema SSL (Error: wrong version number):

- Se produjo al hacer peticiones HTTPS a localhost con proxy inverso o configuración incorrecta.
- Se recomendó revisar la configuración del servidor backend, asegurarse de que HTTPS está bien conf

3. Archivo AdminController.java:

- Controlador REST con endpoints para administrar denuncias, bajas y modificaciones.
- Métodos para aceptar/rechazar denuncias, eliminar restaurantes o usuarios que pidieron baja.

- Endpoints para listar solicitudes de modificación pendientes y aceptarlas/rechazarlas.
- Permite modificar datos de usuarios y restaurantes desde admin.

4. Archivo AdminService.java:

- Lógica de negocio para manejar las operaciones administrativas.
- Métodos para obtener denuncias, aceptar/rechazar denuncias, obtener bajas y solicitudes de modificació
- Se modificó para devolver solo solicitudes no gestionadas, evitando ocultar datos.
- Métodos para resolver (aceptar o rechazar) modificaciones y notificar a usuarios/restaurantes.

5. Componente AdminPanelComponent.ts (Angular):

- Carga datos de denuncias, bajas, modificaciones y notificaciones vía HTTP desde API backend.
- Muestra panel de administración con acordeones para cada tipo de petición.
- Permite aceptar, rechazar, eliminar o modificar registros directamente desde la UI.
- Controla estados de UI para mostrar/ocultar secciones y manejar formularios de edición.

6. Templates HTML y CSS del AdminPanel:

- Diseño sencillo y responsivo con acordeones para organizar secciones.
- Muestra mensajes de alerta si no hay datos pendientes.
- Botones para ejecutar acciones sobre denuncias, bajas y modificaciones.

7. Problemas Detectados y Soluciones Propuestas:

- Asegurarse que la base de datos tiene los registros con el campo gestionada == false para que aparezca
- Verificar que las rutas API usadas en Angular coinciden con las expuestas por Spring Boot.
- Comprobar que el usuario admin tiene permisos y está autenticado correctamente.
- Para error SSL, revisar configuración de servidor y certificados locales.

8. Sugerencias para el Futuro:

- Añadir paginación para las listas largas en panel admin.

- Mejorar validaciones y mensajes de error en frontend.
- Usar servicios WebSocket para notificaciones en tiempo real.
- Integrar logs y monitorización para detectar problemas tempranos.

¡Y eso es todo, choni! Si quieres te paso también el código exacto de cada archivo, o te ayudo a montar u

Solo dime. ??