

Actividad evaluable UT3A1 – Generación de informes

RA5

En esta actividad evaluable deberás modificar tu aplicación para que el usuario con rol *admin* sea capaz de generar informes a partir de la base de datos. Además, el usuario con rol *user* ya no podrá borrar elementos de la base de datos.

¿Qué tengo que entregar? (Requisito)

- Enlace al github en la rama **informes**.
- Añadir a tu manual de usuario (el que hiciste en UT2A4) la parte de informes y las diferentes tareas que son capaces de realizar usuarios con diferentes roles (el usuario con rol *user* y el usuario con rol *admin*).
- El proyecto será descargado por la profesora desde el github: es requisito indispensable que la aplicación se pueda ejecutar en cualquier equipo para poder ser corregida.
- No se admite código generado por IA (y sí, se nota ...)

¿Cómo identifico mi trabajo? (Requisito)

- Debes poner tu nombre y apellidos en el fichero README.md
- Debes nombrar el PDF como sigue:

PrimerApellido_SegundoApellido_Nombre_UT3A1

Desarrollo de la actividad

1. Creación de una nueva rama en el git llamada informes

git checkout -b informes

Trabajarás en esta nueva rama para esta actividad.

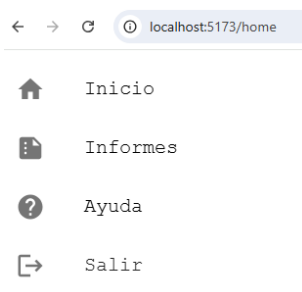
2. Gestión de permisos usando el renderizado condicional

Vamos a gestionar los permisos del usuario con rol *user* y del usuario con rol *admin* usando el renderizado condicional que ya has aprendido en unidades anteriores.

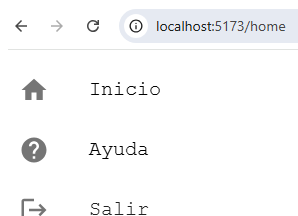
2.1 Gestión de permisos: sólo el usuario con rol *admin* va a poder generar informes

Usando el renderizado condicional, haz que sólo cuando el rol sea *admin* se muestre *Informes* en el menú.

Cuando ingreso en la aplicación con el usuario patricia, cuyo rol es *admin* veré esto en el menú:



Cuando ingreso en la aplicación con el usuario user, cuyo rol es *user* veré esto en el menú;



2.2 Gestión de permisos: el usuario con rol *user* no va a poder eliminar registros

Usando el renderizado condicional, hacer que el usuario con rol *user* **no** pueda eliminar registros de la tabla colección: es decir, el botón-icono de borrar no le aparecerá.

Si ingresamos con el usuario **con rol *admin***, podremos ver el botón-icono de Eliminar en la tabla del Dashboard:

☰
Patricia 

Nombre *	Marca *	Tipo *	Precio * 0
----------	---------	--------	---------------

+ INSERTAR DATOS

	Nombre	Marca	Tipo	Precio
	fsdfs	dfd	dfs	0.9

Si ingresamos con el usuario **con rol *user* no tendremos** el botón-icono de Eliminar en la tabla del Dashboard:

☰
usuario 

Nombre *	Marca *	Tipo *	Precio * 0
----------	---------	--------	---------------

+ INSERTAR DATOS

Nombre	Marca	Tipo	Precio
fsdfs	dfd	dfs	0.9

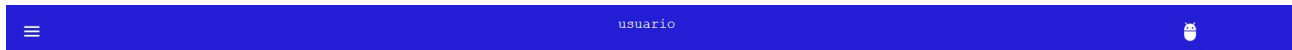
2.3 Diferenciación de iconos de usuario según el rol

Si no lo has hecho ya en prácticas anteriores, usando el renderizado condicional, hacer que el usuario con rol *admin* tenga un icono diferente al usuario con rol *user*. Por ejemplo:

Vemos a la derecha el icono para el usuario con rol *admin*:



Vemos a la derecha el icono para el usuario con rol *user*:



3. Limpiar datos de los TextField

Si no lo has hecho ya, cuando se apriete el botón INSERTAR se deben limpiar los datos de los TextField. Impleméntalo en tu código. Recuerda que tienes la variable **item** ...

4. Creación del informe

Por ahora tu página de informes `frontend/src/pages/Reports.tsx` solo tiene el menú. Añade ahora un botón con el texto INFORME COLECCION.

Nuevo aspecto de la página <http://localhost:5173/reports>



Al picar en el botón INFORME COLECCIÓN, en la función que maneja el evento, vamos a:

- 1) Obtener los datos de la base de datos que necesitemos para el informe y almacenarlos en una variable. En este caso vamos a obtener todos los datos de la tabla *coleccion*.
- 2) Usar una variable de control que se ponga a true para indicar que hemos picado el botón.
- 3) Si esa variable de control está a true, dentro del return vamos a renderizar un nuevo componente, que es el que nos va a mostrar la tabla del informe. El componente que nos genera el informe lo tendrás que crear y se llamará: `frontend/src/components/InformeColeccion.tsx`

¿Hay que pasarle algo al componente <InformeColeccion />?

Pues sí, los datos que se obtuvieron de la consulta a la base de datos. Se lo pasas mediante props, algo que ya sabes hacer.

5. Creación de un informe básico dentro del fichero `frontend/src/components/InformeColeccion.tsx`

Antes de avanzar, si no lo has hecho ya, debes instalar en el frontend las librerías necesarias para crear y exportar informes en formato PDF y CSV: `@material-table/core` y `@material-table/exporters`

```
npm install @material-table/core
npm install @material-table/exporters
```

Ahora, ayudándote del documento guía “*Uso de informes en REACT - Librería @Table-Core.pdf*” que está colgado en la moodle y ha sido explicado en clase realiza, dentro del fichero `InformeColeccion.tsx`, un informe que tenga:

- una tabla con los datos siguientes: nombre, marca, tipo y precio.
- debemos poder mover las columnas a nuestro antojo
- se debe poder exportar a PDF y a CSV
- personalizar la tabla con un nombre adecuado y con colores personalizados que estén acorde a la paleta de colores que has usado en tu aplicación.
- se debe permitir que el usuario elija qué columnas mostrar
- realizar filtrado por columnas: sólo las columnas marca y tipo.
- realizar la suma de los precios de nuestra colección y mostrarlo en la tabla.

Añade al manual de usuario todas las nuevas funcionalidades que has realizado en esta práctica: gestión de permisos y cómo funciona la generación de informes.

PUNTUACIÓN

Entrega	Nota MÁXIMA
1) El ejercicio se entrega en plazo y resuelto correctamente.	10
2) Igual que el 1) pero fuera de plazo. Máximo 24 horas	6
3) No se admiten entregas fuera de plazo cuando se han pasado más de 24 horas de la fecha límite de entrega.	0
4) Si no se cumplen las especificaciones de entrega la tarea irá directamente a recuperación.	0

En la tabla se indican las notas máximas en cada situación.