

## **Seguimiento del cronograma**

**¿Se cumplen los requisitos mínimos y funcionalidades propuestas al inicio del proyecto?**

**Registro de usuarios (creación de cuentas y contraseñas).**

**Comentario:** La forma de uso de las claves o usuarios es de una clave de acceso y se almacena en un csv.

**Guardar información de usuarios.**

**Comentario:** La aplicación guarda los siguientes registros de las interacciones de los usuarios con la plataforma:

Para registro: Clave

Para bodega: distribuidor, producto, referencia, precio, unidades y fecha de compra.

Para compra: Empresa, producto, precio, unidades y fecha de compra.

Para clientes: Cédula, nombre, apellidos, celular y fijo.

Para costos: Nómina, Servicios, arriendo, seguros, mantenimiento, impuestos, transporte, otros gastos.

**Acceso a funcionalidades básicas de la aplicación sin registro (público en general)**

**Comentario:** Un usuario sin registro no puede acceder a ninguna funcionalidad. Debe ser usuario registrado.

**Acceso a funcionalidades avanzadas de la aplicación con registro (usuarios registrados)**

**Comentario:** Un usuario registrado tiene todas las funcionalidades exclusivas.

**Funcionalidades específicas según el tema del proyecto.**

- Vista de bodega y dataframe de bodega
- Vista de compras y dataframe de compras
- Vista de costos y dataframe de costos, con opción de filtro por fechas
- Vista de clientes y dataframe de clientes

En general, las diferentes vistas del proyecto llevan la misma lógica y funcionan adecuadamente. Sin embargo, se recomienda la creación de un manual de usuario y reparar los errores que se presentan en la vista de mostrador y cambio y creación de contraseñas.

## **Organización del repositorio de GitHub**

**¿Se organizó en carpetas, de manera adecuada, el repositorio de GitHub?**

**Comentario: Repositorio organizado.**

## **Integración entre el repositorio de GitHub y el teams de replit**

**¿Funciona de manera adecuada la integración entre el repositorio de GitHub y el teams de replit?**

**Comentario: Integración actualizada y funcional.**

## **Comprensión del código**

**¿Se escribieron comentarios en el código de forma que se pueda concluir que los estudiantes comprendieron el trabajo realizado?**

**Comentario: Código comentado correctamente.**

## **Legibilidad del código**

**¿Se presenta código limpio, es decir, se presenta el código organizado dentro de los ficheros necesarios y la lectura de este se hace de manera fluida sin realizar extensas sesiones de depuración o lectura. Este componente se estudia más desde lo organizacional que desde lo técnico?**

**Comentario: Código legible.**

## **Gestión de asuntos (issues) y tareas (cards) en GitHub**

**¿Se deja registro mediante el manejo de asuntos (issues) de los ajustes que requiere el código? ¿Se actualiza el estado de las tareas (cards) según lo avanzado en el proyecto?**

**Comentario: Cards desactualizadas y no se han presentado nuevos issues.**

## **Trabajo en equipo**

**¿Se realizó un trabajo en grupo de manera equitativa de tal manera que el desarrollo de la aplicación esté distribuida entre los integrantes?**

**Comentario: Los commits se reparten entre Juan Carlos Velásquez y Hernán Cardona. No hay evidencia de la participación de Santiago Cortés.**

## **Anotaciones o comentarios respecto a la entrega**

**Comentario: Se recomienda escribir un manual de usuario y revisar las fallas en el código.**