

## Seguimiento del cronograma

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

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

**Comentario:** El aplicativo crea cuentas de usuario con contraseñas, valida que las nuevas cuentas no tengan los mismos datos claves (correo electrónico) que las cuentas ya creadas.

**Guardar información de usuarios.**

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

Para registro: Nombre, apellido, correo electrónico, confirmar correo electrónico, contraseña, confirmar contraseña y nombre de usuario.

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

**Comentario:** Un usuario sin registro puede acceder a las siguientes funcionalidades: Gráficas de temperatura máxima y de histórico de lluvias de datos internos del servidor.

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

**Comentario:** Un usuario registrado tiene acceso a dos gráficas exclusivas.

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

- **Página de pronósticos:** No se comprende de qué tratan las gráficas.
- **Página estaciones:** Los seleccionadores funcionan, pero hay años que no tienen datos. Además no se comprende de qué tratan las gráficas.
- **Página general:** No se comprende de qué tratan las gráficas.

## 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 adecuadamente.

## **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: Issues y cards actualizados.

## **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: Commits distribuidos satisfactoriamente.

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

**Comentario:** Las gráficas deberían tener al menos títulos para comprender como usuario de qué se tratan.