**Retos al explorar la aplicación GitHub**

**Nombre:** Sareyi Yuliza Estrupiñan Ponce

**Código:** 20241135056

**Usuario de GitHub:** Yuliza18

Durante el proceso de exploración en la aplicación GitHub, experimenté una serie de retos que me permitieron desarrollar nuevas habilidades y mejorar mi comprensión sobre el manejo de repositorios, control de versiones y colaboración en proyectos de programación.

**Retos encontrados**

1. Comprender la interfaz inicial:

Uno de los primeros retos fue familiarizarme con la interfaz de GitHub, ya que cuenta con múltiples secciones como repositorios, proyectos y configuraciones, lo que al principio resultaba confuso.

2. Manejo de repositorios:

Otro desafío fue entender cómo crear y clonar repositorios, así como aprender la diferencia entre trabajar de manera local y en la nube.

3. Uso de comandos básicos:

Me costó inicialmente aprender los comandos de Git más importantes como git add, git commit y git push

**Mi conversación con chat GPT  
¡Hola, Yuliza! 🙌 Aquí tienes una guía paso a paso (clarita y sin enredos) para:**

1. **reconocer qué es cada cosa dentro de un repositorio de GitHub, y**
2. **crear y compartir tu primer proyecto con tu profe.**

**Además, te dejé un proyecto base listo para usar (con carpetas, README y .gitignore). Lo puedes descargar aquí:  
Descargar el proyecto inicial (.zip)**

**Guía 1: Reconocimiento rápido de un repositorio en GitHub**

**Cuando abres un repo en GitHub verás:**

1. **Barra superior de pestañas**

* **Code: los archivos y carpetas del proyecto. Desde aquí puedes descargar, clonar o crear archivos.**
* **Issues: lista de tareas o problemas (opcional, útil para organizarte).**
* **Pull requests: propuestas de cambios desde una rama a otra (sirve para revisar antes de mezclar).**
* **Actions: automatizaciones (no lo necesitas al inicio).**
* **Projects/Wiki: tableros y documentación (opcional).**
* **Security, Insights: información avanzada (puedes ignorarlas por ahora).**
* **Settings: configuración del repo (nombre, privacidad, colaboradores).**

1. **Botón verde “Code”  
   Desde aquí:**

* **Clone (clonar): obtienes la URL HTTPS o SSH para traer el repo a tu PC.**
* **Download ZIP: descargas todo el repositorio como archivo .zip.**

1. **Lista de archivos**

* **Muestra nombre, fecha del último cambio y mensaje del commit.**
* **El archivo README.md aparece abajo como una “portada” del proyecto.**
* **.gitignore indica qué archivos NO se subirán (por ejemplo temporales, cachés, etc.).**

1. **Commits (historial)**

* **En “<> Code”, arriba a la derecha verás el número de commits. Entras y ves el historial: qué cambió, cuándo y por quién.**

1. **Branch (rama)**

* **Suele llamarse main. Puedes crear otras ramas (por ejemplo, dev) para trabajar sin dañar la principal.**

**Guía 2: Tu primer proyecto (del cero a compartir)**

**Opción A — 100% desde la web (fácil y sin instalar nada)**

1. **Entra a GitHub y pulsa New → New repository.**
2. **Nombre: mi-primer-proyecto (o el que quieras).**
3. **Marca Public (o Private si solo quieres que la profe lo vea).**
4. **Activa “Add a README file” y .gitignore (elige el lenguaje si lo conoces; si no, lo agregas luego).**
5. **Crea el repo con Create repository.**
6. **En la página del repo, pulsa Add file → Upload files y sube tus documentos (por ejemplo, docs/informe.docx).**
7. **Al terminar, copia la URL del repo (ej.: https://github.com/tu-usuario/mi-primer-proyecto) y envíasela a tu profe.**
   * **Si el repo es privado: ve a Settings → Collaborators → Add people y agrega el usuario o correo de la profe.**

**Opción B — Con tu carpeta local y Git (recomendado)**

1. **Descarga el proyecto base:  
   Descargar .zip → descomprímelo.**
2. **Entra a esa carpeta y revisa:**
   * **src/ (tu código o archivos principales)**
   * **docs/ (tus entregables: .docx, .pdf, etc.)**
   * **data/ (datos; evita subir archivos > 100 MB)**
   * **README.md (describe tu proyecto)**
   * **.gitignore (evita subir basura/temporales)**
3. **En GitHub crea un repo vacío (sin README, sin .gitignore).**
4. **En tu PC, abre una terminal dentro de la carpeta del proyecto y ejecuta:**
5. **git init**
6. **git add .**
7. **git commit -m "Mi primer commit"**
8. **git branch -M main**
9. **git remote add origin https://github.com/TU\_USUARIO/TU\_REPO.git**
10. **git push -u origin main**
11. **Actualiza la página del repo en GitHub: verás todos tus archivos.**
12. **Para subir cambios después:**
13. **git add .**
14. **git commit -m "Descripción corta de lo que hice"**
15. **git push**

**Opción C — Con GitHub Desktop (casi sin consola)**

1. **Instala GitHub Desktop.**
2. **File → New repository (elige la carpeta del proyecto base que descargaste).**
3. **Publish repository para subirlo a tu cuenta de GitHub.**
4. **Cada vez que cambies algo: Commit to main → Push origin.**

**Flujo de trabajo sugerido (muy simple)**

1. **Crea el repo en GitHub (o publica desde Desktop).**
2. **Pon tus documentos de entrega en docs/ (por ejemplo, docs/informe.docx).**
3. **Haz commit con un mensaje claro y push.**
4. **Comparte el enlace del repo a tu profe.**
   * **Si es privado, invítala como colaboradora (Settings → Collaborators).**

**Buenas prácticas para que todo te salga bonito**

* **Un commit por idea (mensaje claro: “Agrego introducción del informe”, “Corrijo gráficos”).**
* **No subas archivos gigantes (>100 MB). Si necesitas, pregúntame y te enseño Git LFS.**
* **Mantén un README con: objetivo, cómo usar, estructura y autora.**
* **Guarda todo lo que sea para la profe en docs/.**
* **Nunca subas contraseñas ni datos sensibles.**

**Errores típicos y soluciones rápidas**

* **“Updates were rejected because the remote contains work…”  
  Corre:**
* **git pull --rebase origin main**
* **git push**
* **“Archivo mayor a 100 MB” → No se puede subir así. Hay que usar Git LFS o comprimir/partir.**
* **“No me deja hacer push (autenticación)” → Asegúrate de iniciar sesión en GitHub Desktop o de usar Personal Access Token si la consola lo pide.**

**Checklist de entrega (marca y listo ✅)**

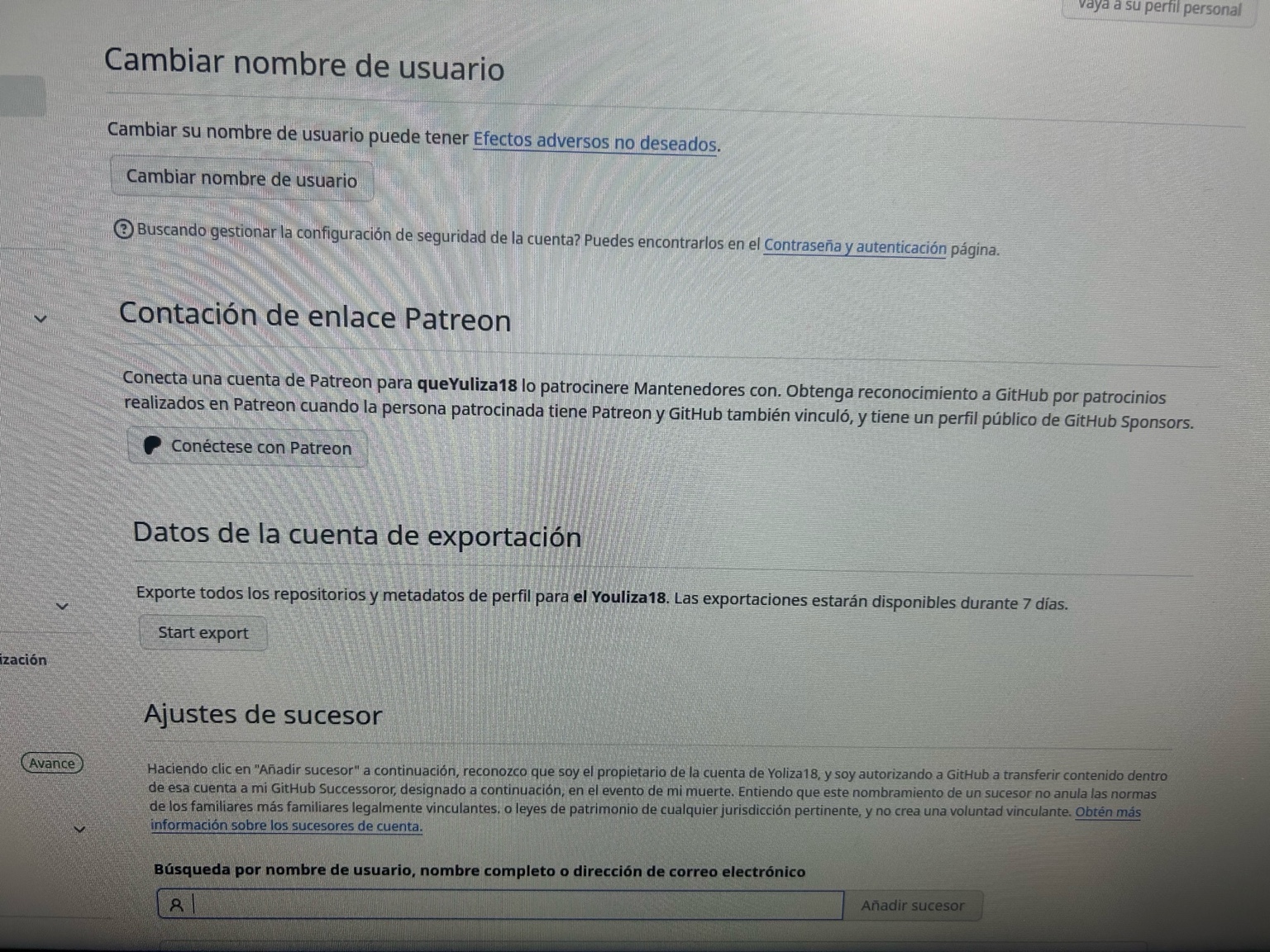
* **El repo tiene README.md con descripción.**
* **Los archivos para la profe están en docs/.**
* **Hice commit y push sin errores.**
* **Pegué el link del repo en el medio de entrega (correo/Aula/WhatsApp).**
* **Si el repo es privado, ya agregué a la profe como colaboradora.**

# **1. Configuración de Cuenta**

• Cambiar nombre de usuario: Permite modificar tu usuario en GitHub.

• Conexión con Patreon: Opción para conectar tu cuenta de Patreon y recibir reconocimiento en GitHub si alguien te patrocina.

• Exportar datos de la cuenta: Descarga todo tu contenido de GitHub (repositorios, configuraciones, etc.)

• Ajustes de sucesor: Te deja asignar a alguien para heredar tu cuenta en caso de que no puedas usarla más.  


# **2. Perfil Público**

• Nombre: Tu nombre real o el que quieras mostrar.

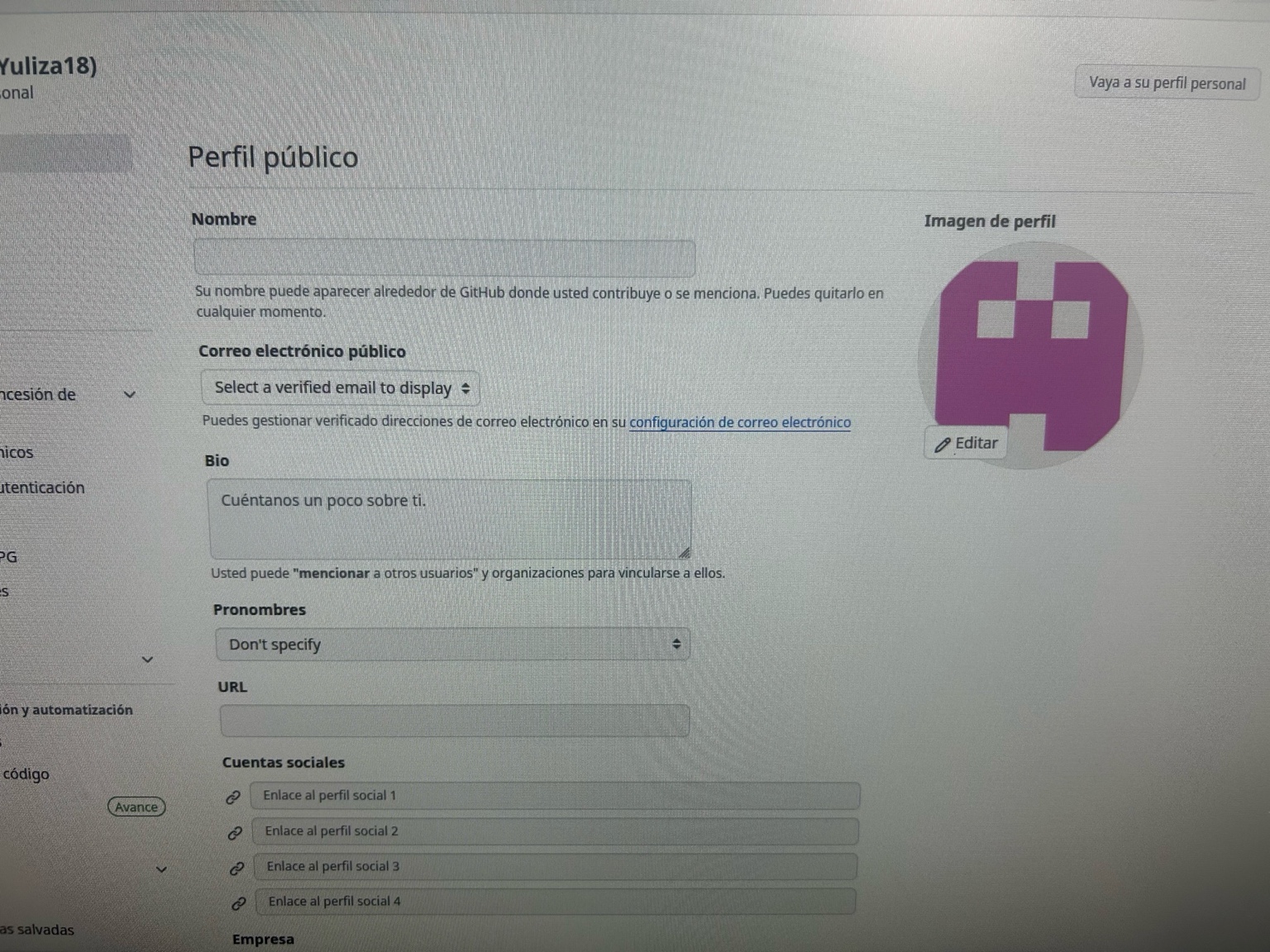
• Correo electrónico público: Decide qué correo mostrar en tu perfil.

• Bio: Espacio para escribir un resumen sobre ti

• Pronombres: Escoge cómo quieres que se refieran a ti

• URL: Puedes añadir un enlace a tu página web, portafolio o LinkedIn.

• Cuentas sociales: Enlaza tus otras redes sociales.

• Empresa: Nombre de la institución o empresa en la que trabajas.  


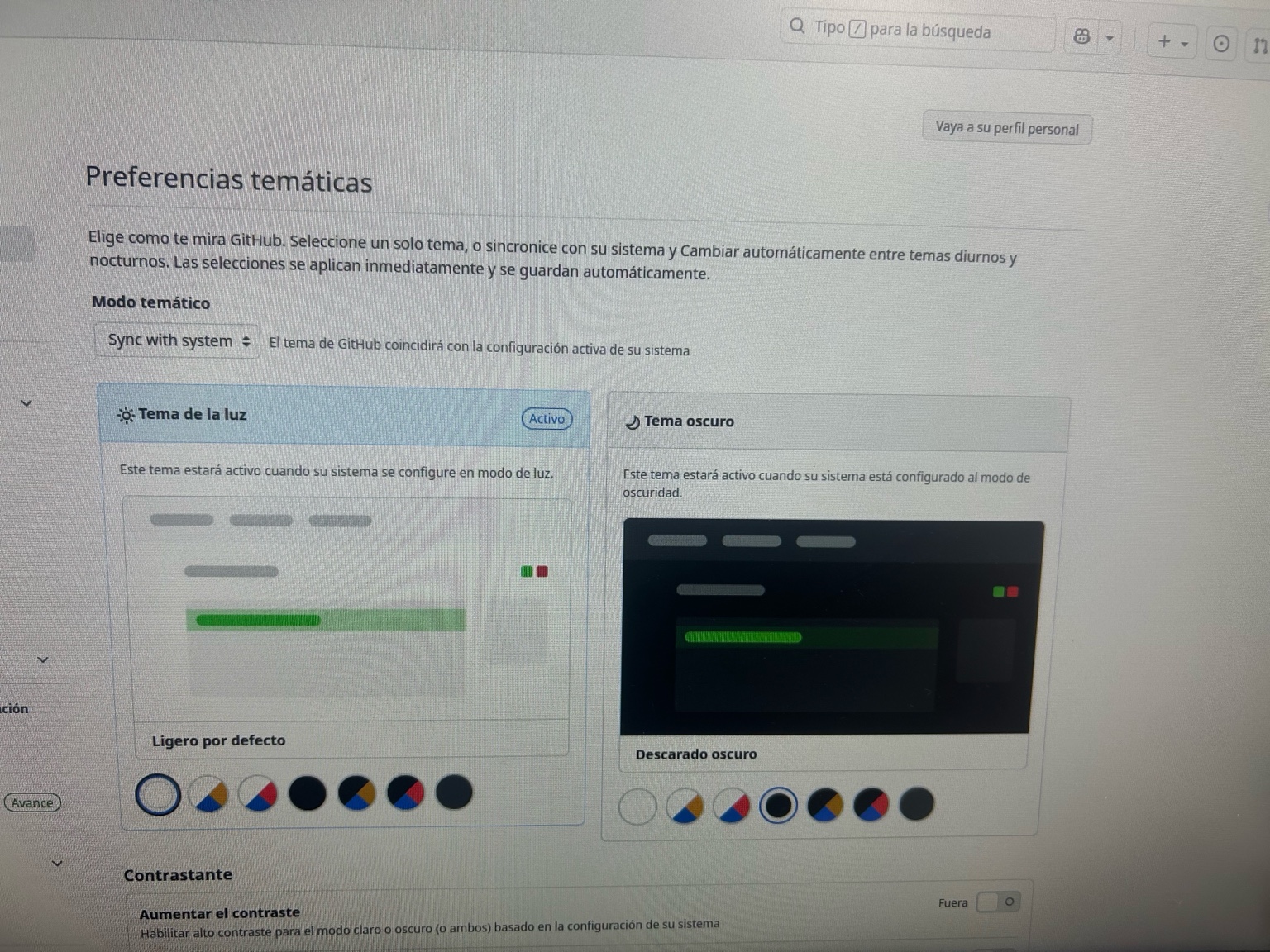
# **3. Preferencias Temáticas**

• Tema de la luz: Pantalla clara.

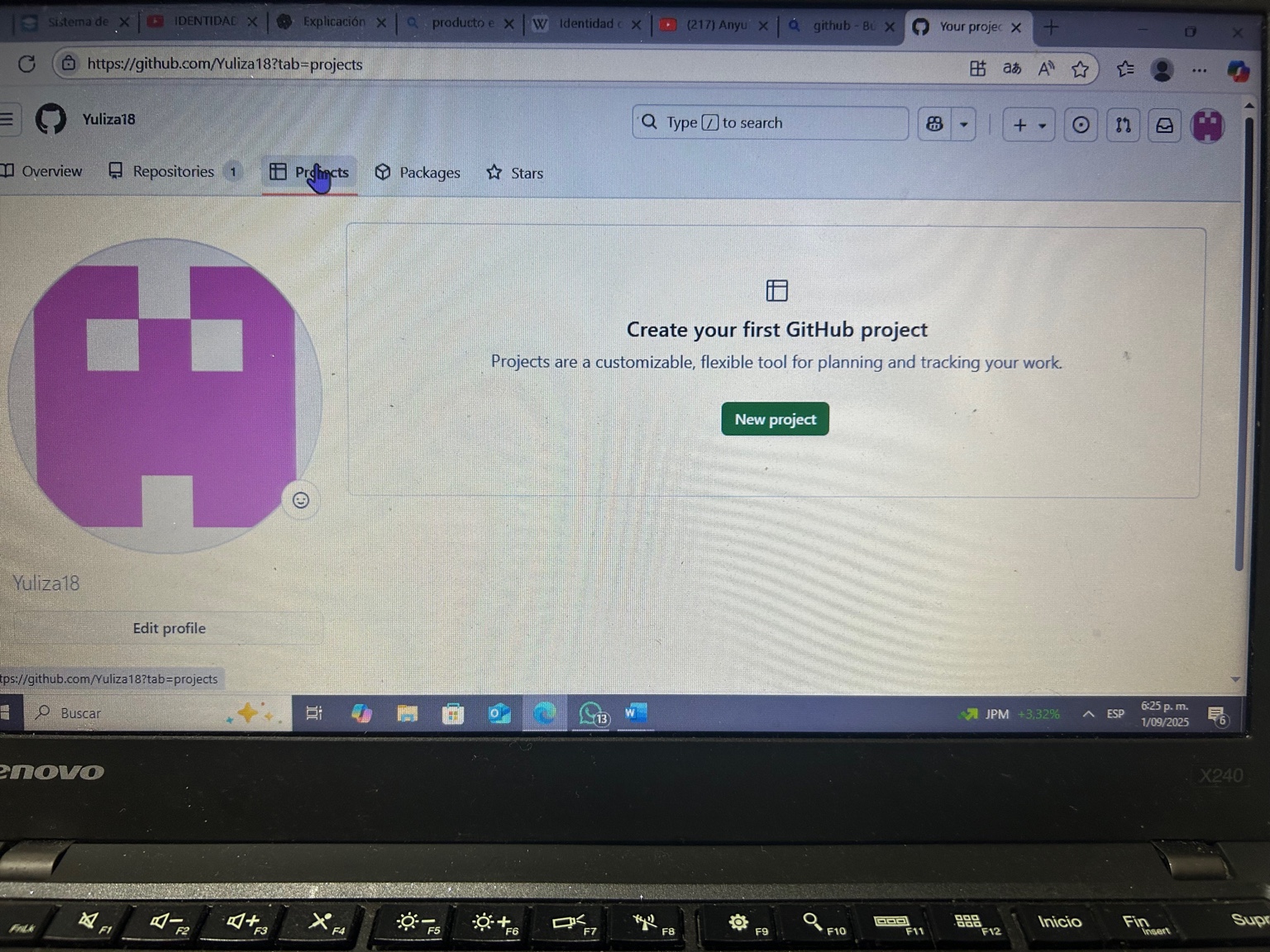
• Tema oscuro: Pantalla negra o gris oscura.

• Sync with system: GitHub copia el tema de tu computador (si está en oscuro, GitHub también).

• Colores de tema: Personalización de colores.

• Contraste: Mejora la visibilidad del texto y botones.  


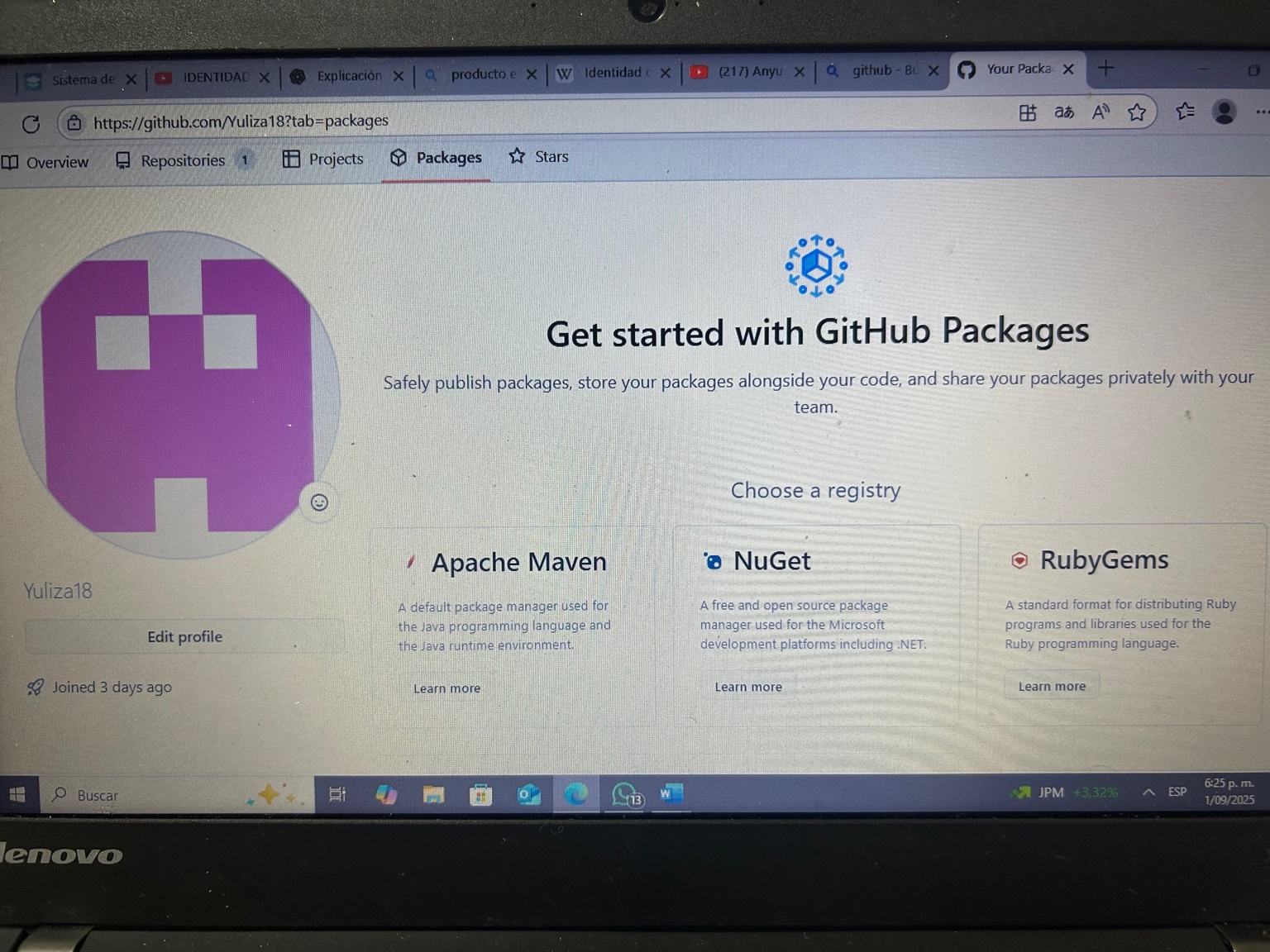
# **4. Projects**

• Funciona como un tablero de tareas   


# **5. Packages**

• Un package es un conjunto de archivos o librerías que otros programadores pueden usar.

• GitHub te deja publicar, guardar y compartir paquetes.



# **6. Stars**

• Las estrellas sirven para guardar proyectos que te gustan o que quieres revisar más tarde.

• También puedes organizar listas con tus repositorios favoritos.  
