**Gestión de Versiones y Uso de Ramas en el Proyecto**

Para el manejo de versiones del proyecto, se utilizó **Git** y se trabajó con varias ramas de manera estructurada para permitir el desarrollo colaborativo sin interferencias entre los integrantes del equipo.

**1. Estructura de ramas:**

* master: Rama principal que contiene la versión estable y lista para entrega del proyecto.
* develop: Rama de integración donde se fusionan los cambios de las ramas de desarrollo (feature/\*) antes de pasar a master.
* feature/diego, feature/mauricio, feature/sebastian: Ramas individuales asignadas a cada desarrollador para trabajar en nuevas funcionalidades o correcciones. Cada desarrollador trabajó en su propia rama para mantener el trabajo independiente.

**2. Flujo de trabajo:**

* Cada desarrollador clonaba el repositorio y trabajaba en su rama local (feature/\*).
* Los cambios locales se realizaban mediante commits frecuentes, siguiendo buenas prácticas de documentación de cada modificación.
* Cuando una funcionalidad estaba lista, se realizaba un **pull request (PR)** hacia la rama develop en GitHub.
* Todos los PR eran revisados por los demás integrantes antes de realizar el merge, asegurando que los cambios fueran revisados y aprobados.
* Una vez que develop contenía las funcionalidades validadas y probadas, se realizaba un merge hacia master, generando la versión estable del proyecto.

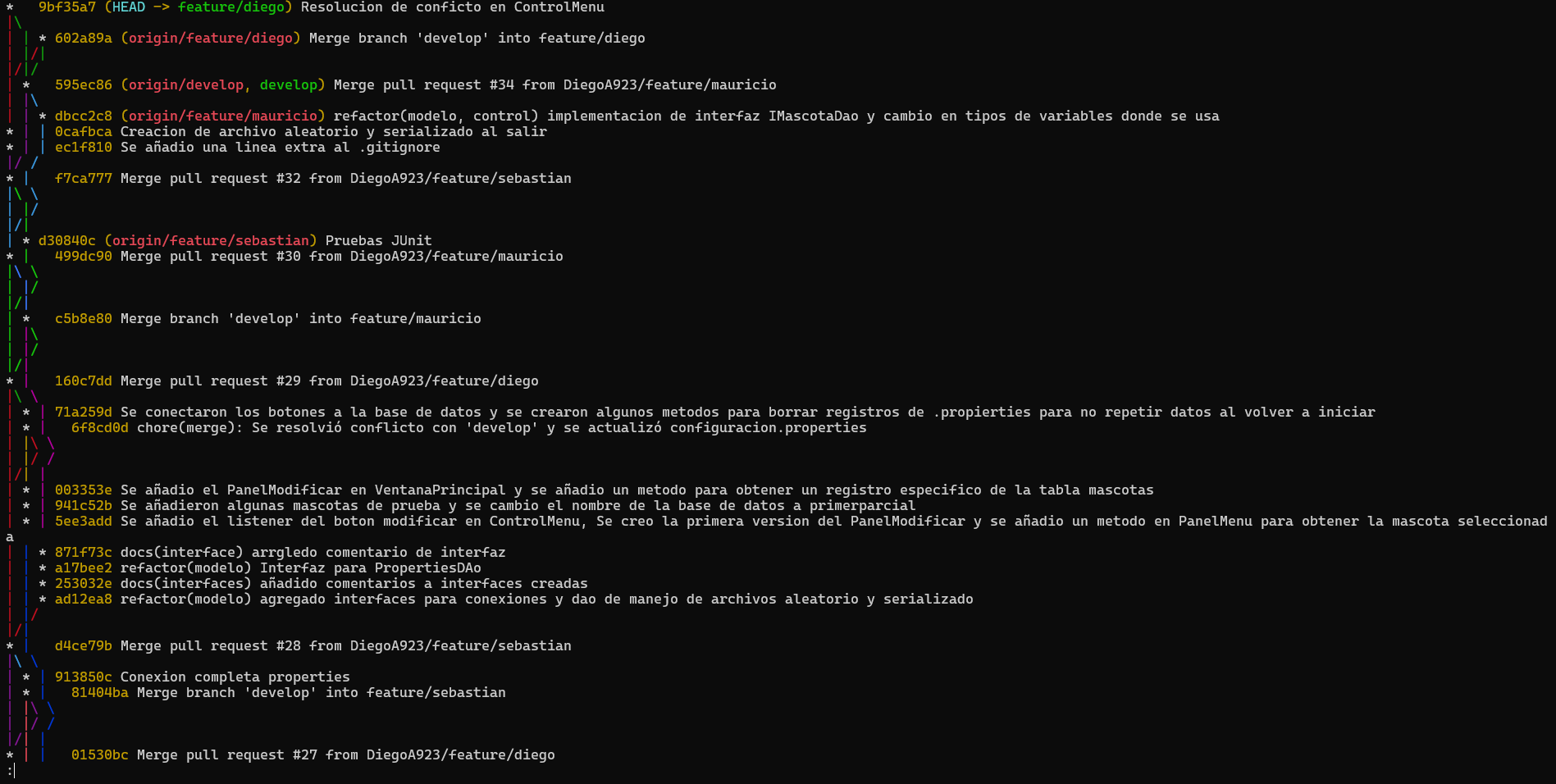
**3. Sincronización local y remota:**

* Cada desarrollador mantenía su repositorio local actualizado con git pull origin feature/<nombre> antes de iniciar nuevos cambios.
* Después de realizar commits locales, sincronizaban su trabajo con el repositorio remoto usando git push origin feature/<nombre>.
* La integración a develop se hacía siempre mediante PR para mantener el control de versiones y permitir la revisión de código.

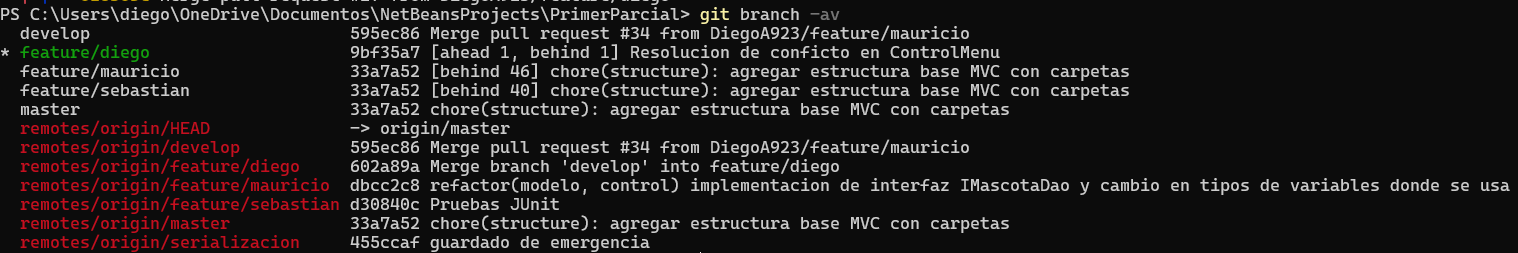
**4. Visualización de ramas y fusiones:**

Para evidenciar la gestión de versiones, se recomienda incluir capturas de pantalla de las siguientes visualizaciones de Git:

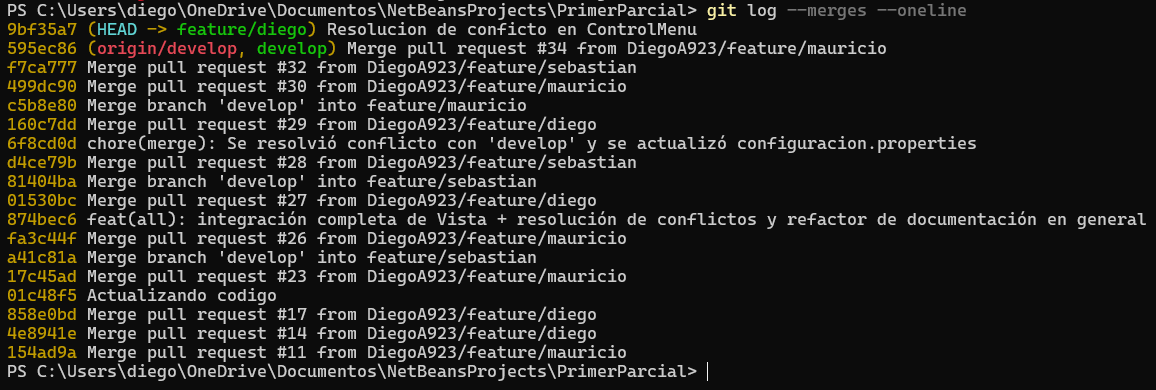
* **Gráfico de ramas y merges:**



* Estado de ramas locales y remotas



* Historial de merges hacia develop y master



**Observaciones finales:**

* Todos los desarrolladores trabajaron de manera colaborativa, sin asignación de componentes específicos, contribuyendo a todas las partes del proyecto.
* El uso de ramas individuales y PR permitió mantener la **estabilidad de master** y un control claro del desarrollo en equipo.
* Las capturas y los logs incluidos en este documento evidencian la correcta gestión de versiones y la estrategia de desarrollo adoptada.