**Grupo 1: Conceptos Fundamentales de Verificación de Algoritmos**

Este grupo se encargará de investigar y explicar los conceptos básicos de la verificación de algoritmos y programas. Entre los temas que deben abordar están:

* Definición de verificación de algoritmos.
* Diferencia entre verificación y validación.
* Importancia de la verificación en el ciclo de desarrollo de software.
* Casos en los que es crucial verificar algoritmos.

**Grupo 2: Técnicas de Verificación Estática**

Este grupo investigará las técnicas de verificación estática de algoritmos y programas. Algunos de los temas a cubrir incluyen:

* Análisis estático de código.
* Inspección y revisión de código.
* Herramientas utilizadas para la verificación estática.
* Ejemplos de verificación estática aplicada.

**Grupo 3: Técnicas de Verificación Dinámica**

Este grupo se enfocará en las técnicas de verificación dinámica, que se realizan durante la ejecución del programa. Algunos temas clave son:

* Pruebas unitarias.
* Pruebas funcionales.
* Técnicas de depuración y pruebas en tiempo real.
* Comparación entre verificación estática y dinámica.

**Grupo 4: Métodos Formales para la Verificación de Programas**

Este grupo analizará los métodos formales que se utilizan para garantizar la corrección de los algoritmos. Estos métodos son fundamentales en software crítico. Temas a investigar:

* Lógica de Hoare.
* Pruebas de corrección formal.
* Modelos matemáticos para la verificación.
* Ejemplos de uso en sistemas críticos (ej.: aviones, hospitales).

**Grupo 5: Herramientas y Casos Prácticos de Verificación de Programas**

Este grupo será responsable de estudiar las herramientas más populares de verificación y mostrar ejemplos de casos reales donde se han utilizado. Deben incluir:

* Herramientas de verificación automática (ej.: VeriFast, Dafny).
* Comparación entre diferentes herramientas.
* Ejemplos de verificación de programas en proyectos reales.
* Desafíos y limitaciones de las herramientas de verificación.