**Actividad de investigación.**

1. Requisitos indispensables de un editor de código. Explicar que debe tener al menos un editor para considerarse una herramienta adecuada para crear código.
2. Diferencias entre un EC y un IDE. Poner en una tabla al menos 3 ejemplos de cada.
3. **Apache Maven:**
   1. ¿Cuál es el propósito principal de Apache Maven en el desarrollo de software?
   2. ¿Cómo ayuda Maven en la gestión de dependencias en un proyecto?
   3. ¿En qué situaciones sería preferible utilizar Maven en lugar de otras herramientas de construcción?
4. **Apache Ant:**
   1. ¿Qué función cumple Apache Ant en el proceso de construcción de software?
   2. ¿Cómo se diferencia Apache Ant de otras herramientas de construcción, como Maven o Gradle?
   3. ¿En qué tipo de proyectos o escenarios sería más adecuado utilizar Apache Ant?
5. **npm (Node Package Manager):**
   1. ¿Cuál es el propósito principal de npm en el ecosistema de desarrollo de JavaScript?
   2. ¿Cómo utiliza npm la gestión de dependencias en proyectos basados en Node.js?
   3. ¿En qué medida npm facilita la automatización de tareas en el desarrollo web?
6. **Bazel:**
   1. ¿Cuál es el objetivo principal de Bazel en el desarrollo de proyectos a gran escala?
   2. ¿Cómo aborda Bazel los desafíos específicos asociados con la construcción de proyectos de gran envergadura?
   3. ¿En qué situaciones sería recomendable elegir Bazel sobre otras herramientas de construcción?
7. **CMake:**
   1. ¿Cuál es el propósito de CMake en el desarrollo de proyectos basados en C y C++?
   2. ¿Cómo aborda CMake la generación de archivos de construcción para diferentes plataformas?
   3. ¿En qué escenarios o proyectos CMake podría ser la opción preferida en comparación con otras herramientas de construcción?
8. **Gradle:**
   1. ¿Qué es Gradle y para qué se utiliza en el desarrollo de software?
   2. ¿Cuál es la principal ventaja de utilizar Gradle en comparación con otras herramientas de construcción de proyectos?
   3. ¿En qué tipo de proyectos y lenguajes de programación es comúnmente utilizado Gradle?
   4. ¿Cómo facilita Gradle la automatización de tareas en el ciclo de vida del desarrollo de software?
9. **JUnit:**
   1. ¿Qué es JUnit y cuál es su propósito en el desarrollo de software?
   2. ¿Cómo ayuda JUnit a los desarrolladores a realizar pruebas unitarias de manera efectiva?
   3. ¿En qué entornos de desarrollo y proyectos es comúnmente utilizado JUnit?
   4. ¿Cuáles son los beneficios clave de implementar pruebas unitarias con JUnit en un proyecto de software?
   5. ¿Puede proporcionar un ejemplo práctico de cómo se utiliza JUnit para realizar una prueba unitaria simple?