

DAW, DAM . ENTORNOS DE DESARROLLO

Jose Alba Arrufat

Índice

Actividad de investigación.....	3
1. Apache Maven.....	3
2. Apache Ant.....	5
3. npm (Node Package Manager).....	6
4. Bazel.....	7
5. CMake.....	8
6. Gradle.....	9
7. JUnit.....	10

Actividad de investigación.

- **Requisitos indispensables de un editor de código. Explicar que debe tener al menos un editor para considerarse una herramienta adecuada para crear código.**

Como mínimo, que puedas manipular la estructura de un archivo.

Luego, los editores de texto suelen tener algunas cosas extra integradas, veáse una sección de plugins adicionales para integrar cosas útiles que quieran aportar la comunidad, o sin ir más lejos y mirando un poco más por el lado de los entornos gordos de desarrollo (IDE), estos suelen integrar mucha cosa interna para comodidad a la hora de escribir código del calibre de sugerencias de que hacer o cosas adicionales como copilot.

- **Diferencias entre un EC y un IDE. Poner en una tabla al menos 3 ejemplos de cada.**

EC – Son programas de edición de código simples

Algunos ejemplos de EC son

SublimeText | Notepad | Atom

IDE – Son programas de edición de código con cosas especializadas integradas.

Algunos ejemplos de IDE son:

Visual Studio Code | IntelliJBrains | NetBeans.

1. Apache Maven

1. *¿Cuál es el propósito principal de Apache Maven en el desarrollo de software?*

Es el de poder hacer un esquema a la hora de realizar construcción de proyectos (muy específicamente, de Java).

2. *¿Cómo ayuda Maven en la gestión de dependencias en un proyecto?*

A través de un archivo XML que utiliza para guardar todos los datos de la estructura del proyecto.

3. *¿En qué situaciones sería preferible utilizar Maven en lugar de otras herramientas de construcción?*

Cuando estas haciendo un proyecto de Java, y me imagino que más específicamente si es un proyecto no muy grande, ya que por lo visto, hay otro framework llamado Grandle que tiene la misma concepción que Apache Maven pero a mayor escala, para proyectos más gordos.

2. Apache Ant

1. *¿Qué función cumple Apache Ant en el proceso de construcción de software?*

Es una herramienta el cuál sirve para la automatización de tareas repetitivas a la hora de programar

2. *¿Cómo se diferencia Apache Ant de otras herramientas de construcción, como Maven o Gradle?*

Por lo que leí, que es un software multiplataforma.

3. *¿En qué tipo de proyectos o escenarios sería más adecuado utilizar Apache Ant?*

Me imagino que cuando queramos hacer algo más de nuestra propia cosecha, al final el apache ant es una herramienta que viene vacía, has de rellenarla tu con cosas.

3. npm (Node Package Manager)

1. *¿Cuál es el propósito principal de npm en el ecosistema de desarrollo de JavaScript?*

Es una herramienta que nos permite desarrollar archivos javascript a través de archivos .json (rollo, son los gestores de proyectos enteros en javascript).

2. *¿Cómo utiliza npm la gestión de dependencias en proyectos basados en Node.js?*

En la carpeta node_modules se guarda todo lo que instalas a través del npm.

3. *¿En qué medida npm facilita la automatización de tareas en el desarrollo web?*

A la hora de automatizar tareas a través de scripts dentro de nuestro proyecto.

4. Bazel

1. *¿Cuál es el objetivo principal de Bazel en el desarrollo de proyectos a gran escala?*

Es una herramienta de código abierto en el que bajo un proyecto, se permite aunar múltiples proyectos de múltiples idiomas.

2. *¿Cómo aborda Bazel los desafíos específicos asociados con la construcción de proyectos de gran envergadura?*

Entiendo que validando la integridad de cada archivo de todos los proyectos individualmente para que no explote a la hora de ejecutar un gran proyecto.

3. *¿En qué situaciones sería recomendable elegir Bazel sobre otras herramientas de construcción?*

Me imagino que cuando sea un penta-proyecto giga-enorme.

5. CMake

1. *¿Cuál es el propósito de CMake en el desarrollo de proyectos basados en C y C++?*

Pos pa' hacer proyectos basados en estos de la misma manera que npm, gestionar archivos de C.

2. *¿Cómo aborda CMake la generación de archivos de construcción para diferentes plataformas?*

Entiendo yo que a través de C++, pues en la wiki ponía que utiliza este como dependencia mínima y necesaria para utilizar el Cmake en tema de librerías.

3. *¿En qué escenarios o proyectos CMake podría ser la opción preferida en comparación con otras herramientas de construcción?*

Pos cuando buscar lo que te ofrece CMake, muy en concreto. Me imagino que pa' cosas sencillas por lo que leí por encima ahí en la wiki.

6. Gradle

1. *¿Qué es Gradle y para qué se utiliza en el desarrollo de software?*

Es el mismo propósito que el Apache Maven pero para proyectos de mayor escala.

2. *¿Cuál es la principal ventaja de utilizar Gradle en comparación con otras herramientas de construcción de proyectos?*

Por lo que leí, tiene pinta de que es lo de que puedes hacer múltiples versiones de un mismo proyecto.

3. *¿En qué tipo de proyectos y lenguajes de programación es comúnmente utilizado Gradle?*

.DSL (Lenguaje Específico de Dominio) | groovy | Kotlin

4. *¿Cómo facilita Gradle la automatización de tareas en el ciclo de vida del desarrollo de software?*

A través del archivo .dsl entiendo yo, al igual que en el Apache Maven utilizan el XML para ello.

7. JUnit

1. ¿Qué es JUnit y cuál es su propósito en el desarrollo de software?

JUnit es un framework de Java, full dedicado a este para realizar testing de si al ejecutar un archivo Java, este se ejecute correctamente.

2. ¿Cómo ayuda JUnit a los desarrolladores a realizar pruebas unitarias de manera efectiva?

A la hora de ejecutar el código en pantalla te dice si las task que realiza se hacen correctamente o no

3. ¿En qué entornos de desarrollo y proyectos es comúnmente utilizado Junit?

Entiendo yo que, en los que se utiliza principalmente Java.

4. ¿Cuáles son los beneficios clave de implementar pruebas unitarias con JUnit en un proyecto de software?

A mi entender, entiendo que para ver que métodos funcionan y cuales no.

5. ¿Puede proporcionar un ejemplo práctico de cómo se utiliza JUnit para realizar una prueba unitaria simple?

No lo mire mucho, es probable que no sea así, pero, creo que has de instalarte la librería de JUnit y tras esto, en una carpeta creas un archivo java para testear, te metes ahí, y metes syntaxis de test

(sriry, no miré mucho, pero por lo que vi, se representan a través de los '@' dentro del archivo o algo del estilo).

