

**UNIVERSITARIA DE COLOMBIA**

**ESTUDIANTE**

**JUAN ESTABAN FONSECA AVILA**

**DOCENTE**

**SANDRA ESPERANZA SANCHEZ BARRAGAN**

## ACTIVIDADES DE CONTEXTUALIZACION E IDENTIFICACION DE CONOCIMIENTOS NECESARIOS PARA EL APRENDIZAJE

a. ¿Qué es una Java Virtual Machine?

La maquina java ejecuta las instrucciones generadas por un compilador java, consta de un intérprete bytecode (módulo que ejecuta las instrucciones que componen los programas Java) y un entorno de tiempo de ejecución, (un software que provee servicios para un programa en ejecución, pero no es considerado en sí mismo como parte del sistema operativo) que permita ejecutar los archivos de clase java en cualquier plataforma, sea cual sea la plataforma en la que se desarrolló originalmente.

b. ¿Qué ventaja nos da Java Virtual Machine?

- Portabilidad: Una de las principales ventajas de la JVM es que permite que el mismo programa Java se ejecute en diferentes plataformas sin necesidad de realizar modificaciones en el código fuente. Esto se debe a que la JVM proporciona una capa de abstracción entre el código Java y el sistema operativo subyacente, lo que permite que los programas Java se ejecuten de manera consistente en diferentes sistemas.
- Seguridad: La JVM proporciona un entorno de ejecución seguro para los programas Java, ya que se encarga de administrar la memoria, verificar el código antes de ejecutarlo y controlar el acceso a los recursos del sistema. Esto ayuda a prevenir errores y vulnerabilidades de seguridad en los programas Java.
- Gestión de memoria: La JVM es responsable de administrar la memoria utilizada por los programas Java, lo que incluye asignar y liberar memoria automáticamente. Esto ayuda a evitar problemas comunes asociados con la administración manual de memoria, como fugas de memoria y errores de acceso a memoria.
- Mejora de rendimiento: La JVM utiliza técnicas de optimización de código para mejorar el rendimiento de los programas Java en tiempo de ejecución. Además, la JVM puede compilar el código Java a código nativo en tiempo de ejecución, lo que también puede mejorar el rendimiento.

c. ¿Cómo funciona la JVM?

Funciona como una capa de abstracción entre el código fuente Java y el sistema operativo subyacente. Proporciona un entorno de ejecución seguro y portátil para los programas Java, gestiona la memoria, interpreta y compila el código Java y controla el acceso a los recursos del sistema. Estos procesos permiten que los programas Java se ejecuten de manera consistente en diferentes plataformas y sistemas operativos.

d. ¿Qué es un IDE?

Un entorno de desarrollo integrado (IDE) es un sistema de software para el diseño de aplicaciones que combina herramientas del desarrollador comunes en una sola interfaz gráfica de usuario (GUI). Generalmente, un IDE cuenta con las siguientes características:

- Editor de código fuente: editor de texto que ayuda a escribir el código de software con funciones como el resaltado de la sintaxis con indicaciones visuales, el relleno automático específico para el lenguaje y la comprobación de errores a medida que se escribe el código.
- Automatización de las compilaciones locales: herramientas que automatizan las tareas sencillas y repetitivas como parte de la creación de una compilación local del software para que use el desarrollador, como la compilación del código fuente de la computadora en código binario, el empaquetado de ese código y la ejecución de pruebas automatizadas.
- Depurador: programa que sirve para probar otros programas y mostrar la ubicación de un error en el código original de forma gráfica.

- e. ¿Cómo los IDE para el desarrollo de aplicaciones móviles aumentan la productividad?

Existen muchas herramientas de desarrollo de aplicaciones móviles que le brindan a usted y a su empresa la posibilidad de crear la mejor aplicación móvil. Con algo de experiencia y la voluntad de aprender, puede crear su propia aplicación desde el backend, una oportunidad gratificante, divertida y excelente para todos los propietarios de negocios. Alternativamente, puede optar por subcontratar el desarrollo de su aplicación móvil. No obstante, es probable que los desarrolladores utilicen las mismas herramientas. Entonces, con esto en mente, esta publicación de blog discutirá las diez mejores herramientas de desarrollo de aplicaciones móviles, lo que le permitirá elegir las mejores herramientas de desarrollo de software para ayudarlo a hacer realidad la visión de su aplicación móvil.

- f. ¿Qué son Aplicaciones móviles nativas?

Son aplicaciones diseñadas y desarrolladas específicamente para una plataforma móvil determinada, como Android o iOS. Estas aplicaciones se escriben utilizando lenguajes de programación y herramientas nativas de la plataforma, como Java o Kotlin para Android, o Objective-C o Swift para iOS. Las aplicaciones móviles nativas se ejecutan directamente en el sistema operativo del dispositivo móvil y tienen acceso completo a las características y funcionalidades del dispositivo, como la cámara, el micrófono y los sensores.

- g. Ventajas y Desventajas Aplicaciones móviles nativas.

#### VENTAJAS

Te ofrece la mejor experiencia de usuario, mejor rendimiento y mayor integración con el sistema operativo y el hardware del dispositivo. Al ser desarrolladas específicamente para una plataforma determinada, las aplicaciones móviles nativas pueden aprovechar al máximo las características y funcionalidades de cada plataforma y ofrecer una experiencia de usuario más fluida y personalizada.

#### DESVENTAJAS

Puede ser más costoso y requiere de un mayor tiempo de desarrollo en comparación con las aplicaciones híbridas o basadas en la web. Además, cada plataforma móvil requiere de un conjunto de habilidades y conocimientos de programación específicos, lo que puede requerir de diferentes equipos de desarrollo para cada plataforma.

#### h. Ejemplos de Aplicaciones móviles nativas.

- Facebook
- Instagram
- Uber
- WhatsApp
- Snapchat
- Spotify
- Gmail
- Google Maps
- LinkedIn
- YouTube

#### i. ¿Qué son Aplicaciones móviles híbridas?

Las aplicaciones móviles híbridas son aplicaciones móviles que se construyen utilizando una combinación de tecnologías web (como HTML, CSS y JavaScript) y tecnologías nativas (como Java para Android o Swift para iOS). Estas aplicaciones se ejecutan en una plataforma nativa a través de un envoltorio o contenedor que proporciona acceso a características y capacidades de la plataforma subyacente, como la cámara, la ubicación y los sensores.

#### j. Ejemplos de Aplicaciones móviles híbridas

**Basecamp:** La aplicación móvil de Basecamp es una aplicación híbrida que permite a los usuarios gestionar proyectos y tareas, compartir archivos y comunicarse con miembros del equipo. La aplicación ofrece una experiencia de usuario similar a la de una aplicación nativa y está disponible en iOS y Android.

**Evernote:** La aplicación móvil de Evernote es una aplicación híbrida que permite a los usuarios tomar notas, crear listas de tareas, guardar imágenes y compartir archivos. La aplicación ofrece una experiencia de usuario similar a la de una aplicación nativa y está disponible en iOS y Android.

**Amazon:** La aplicación móvil de Amazon es una aplicación híbrida que permite a los usuarios comprar productos en línea, ver recomendaciones de productos y hacer seguimiento de los pedidos. La aplicación ofrece una experiencia de usuario similar a la de una aplicación nativa y está disponible en iOS y Android.

## k. Ventajas y Desventajas de Aplicaciones móviles híbridas

### VENTAJAS

**Ahorro de tiempo y dinero:** Las aplicaciones móviles híbridas permiten a los desarrolladores construir una única aplicación que se pueda implementar en múltiples plataformas, lo que reduce el tiempo y los costos de desarrollo en comparación con el desarrollo de aplicaciones nativas separadas para cada plataforma.

**Interfaz de usuario amigable:** Las aplicaciones híbridas pueden ofrecer una interfaz de usuario intuitiva y fácil de usar, lo que mejora la experiencia de usuario y aumenta la retención de los usuarios.

**Funcionalidades nativas:** Las aplicaciones híbridas pueden acceder a las funcionalidades y capacidades de las plataformas nativas, como la cámara, el GPS, el micrófono y los sensores, lo que permite ofrecer una experiencia de usuario similar a la de las aplicaciones nativas.

**Mayor alcance:** Como las aplicaciones híbridas pueden funcionar en múltiples plataformas, pueden llegar a un público más amplio y aumentar el alcance de la aplicación.

## 1. Frameworks para Aplicaciones móviles híbridas

**React Native:** Es un framework de código abierto basado en React que permite crear aplicaciones móviles híbridas con una interfaz de usuario nativa y un alto rendimiento.

**Xamarin:** Es un framework de código cerrado que permite crear aplicaciones móviles híbridas utilizando C# y .NET. Es compatible con múltiples plataformas y permite acceder a funcionalidades nativas del dispositivo.

**Flutter:** Es un framework de código abierto basado en Dart que permite crear aplicaciones móviles híbridas con una interfaz de usuario atractiva y un alto rendimiento. Flutter utiliza su propio motor de renderizado y no depende de los componentes de la plataforma subyacente.

**PhoneGap:** Es un framework de código abierto que permite crear aplicaciones móviles híbridas utilizando HTML, CSS y JavaScript. Es compatible con múltiples plataformas y permite acceder a funcionalidades nativas del dispositivo.

