

### 3.1. Actividades de reflexión Inicial: Actividad de Reflexión 1: Taller socialización de conocimientos sobre Java

#### ¿Qué es el Frontend en el desarrollo de aplicaciones de escritorio?

El frontend es todo aquello con lo que los usuarios interactúan, como botones, pestañas y demás, es todo lo que ve el usuario.

#### ¿Cuáles son las características principales del Frontend de escritorio?

La adaptabilidad, la compatibilidad con navegadores, la accesibilidad para personas con discapacidades, que sea interactivo con el usuario, animaciones, que sea dinámico y fácil de usar, aparte de ser escalable y intuitivo.

#### ¿Qué elementos componen el Frontend de una aplicación de escritorio?

El frontend de las aplicaciones de escritorio están compuestas por elementos visuales como lo serían: menús, botones, layouts, imágenes, logotipos de la empresa, animaciones etc...

#### ¿Qué es una aplicación de escritorio y cómo se diferencia de una aplicación web?

Una aplicación de escritorio es aquella que tiene un servidor privado, y una aplicación web es aquella que combina la web con servidores externos que son propios de esta misma. Otra diferencia que se encuentra de estas, es que las aplicaciones de escritorio requieren instalación, y las web no, aparte de que las apps de escritorio están descentralizadas y las web si lo están.

## ¿Cuáles son las características principales de una aplicación de escritorio desarrollada en Java?

Las principales características de una aplicación web desarrollada en java, es que son más rápidas en comparación a otros lenguajes de programación, disponen de más seguridad, son portátiles y aceptadas en múltiples plataformas. Las aplicaciones desarrolladas en java destacan por la simplicidad y sintáxis simple de estas mismas.

## ¿Qué ventajas ofrece Java para el desarrollo de aplicaciones de escritorio?

Las principales características que ofrece java, es su simplicidad por sus comandos simples, su fácil entendimiento, su seguridad, su fluidez porque esta más entrelazado con los recursos del sistema, su portabilidad ya que es aceptado por múltiples plataformas.

## ¿Cuáles son los componentes básicos de una interfaz gráfica en una aplicación de escritorio con Java?

El frontend de las aplicaciones de escritorio están compuestas por elementos visuales como lo serían: menús, botones, layouts, imágenes, logotipos de la empresa, animaciones etc...

## ¿Qué es Java Swing y cuál es su papel en el desarrollo de aplicaciones de escritorio?

Java swing es una GUI o Graphic user interface, es una biblioteca que se utiliza para desarrollar interfaces graficas de escritorio interactivas que permite manejar eventos, modificar imágenes en 2d y hacer la experiencia del usuario mas inmersiva dentro de la app.

## ¿Qué es JavaFX y cómo se compara con Swing para el desarrollo de interfaces gráficas?

JavaFX es más moderno en comparación a java swing, este mismo fue creado en base a java swing teniendo casi las mismas funciones pero con la diferencia de que JavaFX es más atractivo para el usuario.

## ¿Qué son los contenedores y componentes en Java Swing o JavaFX? (ejemplos: JFrame, JPanel, Button, Label, etc.).

Los contenedores son las herramientas que ayudan a organizar y almacenar componentes y los componentes son las cosas que conforman la interfaz del usuario.

## ¿Cómo se crean formularios en una aplicación de escritorio con Java?

Usando el paquete Java Swing, ya que esta tiene componentes como `JPasswordField`, `JPanel`, `JTextField`, etc... que hace que el diseño de las interfaces y formularios sea más dinámico y bonito visualmente.

## ¿Qué widgets o componentes se utilizan comúnmente en formularios de aplicaciones de escritorio en Java? (ejemplos: `JTextField`, `JComboBox`, `JButton`, etc.).

Aparte de los anteriormente mencionados, existen el `JCheckBox` el cual es un cuadro de selección, el `JRadioButton` el cual es un elemento de opción única, `JPanel` el cual es un

contenedor para agrupar elementos, Jlist que muestra una lista de elementos y permite seleccionar uno o varios, etc...

## ¿Cómo se maneja la interacción del usuario con los formularios en Java (eventos, listeners)?

En java, la interacción se maneja con eventos y oyentes, los oyentes son aquellos que entre comillas, escuchan las acciones, algunos ejemplos de oyentes serian “addMouseListener” para manejar los eventos del raton, “addKeyListener” este estaría pendiente de la respuesta del teclado, “addActionListener” que este serviría para un botón. Si habláramos de eventos significa algo que ocurre en la interfaz del usuario como clics en un botón, un movimiento del mouse, etc....

## ¿Qué es la maquetación de interfaces en aplicaciones de escritorio con Java?

La maquetación es el proceso de diseño y organización de una GUI, para que sea intuitiva, dinámica y fácil de usar.

¿Qué herramientas o layouts se utilizan para organizar los componentes en una interfaz gráfica en Java? (ejemplos: BorderLayout, GridLayout, FlowLayout).

Los Layout Managers son las herramientas que se utilizan para organizar los componentes en una interfaz gráfica, por ejemplo, el box layout organiza los componentes en filas y columnas, grid layout organiza los elementos en un número fijo de filas y columnas, etc...

¿Cuáles son las mejores prácticas para diseñar interfaces de usuario eficientes y atractivas en aplicaciones de escritorio con Java?

Las mejores prácticas que se pueden realizar al crear interfaces, serían: la claridad de los elementos, simplicidad para que los usuarios sepan usar de forma intuitiva la interfaz, la flexibilidad para que la interfaz se adapte a las necesidades y habilidades de los usuarios, y que la plataforma tenga los mínimos errores posibles .

¿Qué frameworks o bibliotecas son populares para el desarrollo de aplicaciones de escritorio en Java? (ejemplos: Swing, JavaFX, SWT).

Aparte de las ya mencionadas, existen AWT o abstract window toolkit, es el kit de herramientas de ventanas abstracto original de Java, pero es más antiguo y no compete con las opciones más modernas, además que una de sus ventajas es que permite hacer una GUI de forma rápida y sencilla además de que permite la creación de aplicaciones web, etc...

## ¿Cómo se compara JavaFX con Swing en términos de funcionalidad y facilidad de uso?

JavaFX es más moderno y mas atractiv, con mejores graficos y CSS para hacer interfaces mas bonitas y detalladas, Swing es mas antiguo y ligero, es sencillo y compatible, pero un poquito viejo. JavaFX es para aplicaciones con diseños modernos en cambio Swing es para las más simples o para que funcionen con versiones antiguas de Windows o así.

## ¿Qué herramientas de desarrollo (IDEs) son recomendadas para crear aplicaciones de escritorio en Java?

De herramientas, estaría netbeans ya que es el IDE oficial que tiene java, después visual studio code que seria un editor de código, pero con las extensiones correctas es una opción muy buena, y también existe eclipse, que es un IDE de código abierto y muy personalizable.

## ¿Cuáles son las ventajas de desarrollar aplicaciones de escritorio con Java frente a otros lenguajes?

Las principales ventajas serian, su flexibilidad, permitiendo que una misma aplicación funcione en diferentes sistemas operativos sin cambiar. Sus multiples frameworks, y disponer de una comunidad muy buena y grande, además java tiene bastante seguridad y buen rendimiento.

## ¿Cuáles son las desventajas de desarrollar aplicaciones de escritorio con Java?

Pues, en rendimiento aunque ahora es mejor que antes, no se puede comparar a otros lenguajes de programación como C# o c++, también que para realizar algo, a veces se necesita muchísimo mas código comparado a otros lenguajes, y también que puede gastar mas memoria comparado a otros.

## ¿Cómo se maneja la portabilidad de aplicaciones de escritorio desarrolladas en Java?

En java, la portabilidad se basa en la JVM o java virtual machine, que traduce el código a bytecode y este bytecode se ejecuta en cualquier sistema con una JVM, permitiendo que la aplicación funcione en múltiples plataformas sin cambiar mucho, aunque puede que la interfaz cambie un poquito.

## ¿Cómo se gestiona la persistencia de datos en aplicaciones de escritorio con Java? (ejemplos: uso de bases de datos como SQLite, MySQL, o archivos locales).

Pues, en java es mediante el uso de bases de datos como SQLite o MySQL, que permite recuperar y ingresar información de forma sencilla, o a través de archivos locales que sirven para guardar datos simples en el pc.

¿Qué bibliotecas o frameworks se utilizan para conectar una aplicación de escritorio en Java con una base de datos? (ejemplos: JDBC, Hibernate).

Para vincular una base de datos a java, se podría usar la JDBC o java database connectivity para vincular directamente a sql, o un framework como JPA para transformar de manera fácil un objeto de java a una tabla.

¿Cómo se empaqueta y distribuye una aplicación de escritorio desarrollada en Java?

Pues, sería creando un Java archive, que esto tendría todo lo necesario para que la app funcionara, después sería crear un instalador y al final solo subirla a internet.

¿Qué herramientas se utilizan para crear instaladores o ejecutables de aplicaciones de escritorio en Java? (ejemplos: JPackage, Launch4j).

se puede utilizar el jpackage que viene con el JDK, o Launch4j, que convierte JARS en instaladores .exe para Windows. También existen otras opciones como Install4j, que ofrecen más funciones o inno setup que hace instaladores para windows.



¿Cómo se asegura la compatibilidad de una aplicación de escritorio en Java en diferentes sistemas operativos?

Aunque esta pregunta ya la había respondido anteriormente, se logra usando una JVM que traduce el código a bytecode y este bytecode se ejecuta en cualquier sistema con una JVM, permitiendo que la aplicación funcione en plataformas sin cambiar mucho.

¿Cuáles son algunos ejemplos de aplicaciones de escritorio populares desarrolladas con Java?

Java ha sido usado para múltiples propósitos, desde juegos hasta apps empresariales, de los juegos destaca sobre todo minecraft, el cual demostró que java era más versátil de lo que se creía, etc...

¿En qué tipos de proyectos o industrias es común el uso de aplicaciones de escritorio en Java?

Se usan para IDEs, para apps empresariales, para juegos, para comercios, para gestionar inventarios, etc...

¿Qué es la maquetación de interfaces en el desarrollo de aplicaciones de escritorio?

Es básicamente organizar como se verá una aplicación, su diseño.

¿Cuáles son los elementos clave que se deben considerar al maquetar una interfaz?

La claridad visual, la adaptabilidad para que se vea bien en diferentes dispositivos, que la interfaz sea intuitiva y fácil de usar, que sea accesible, osea, que la puedan usar todos y que el diseño sea llamativo y consistente.

¿Qué recomendaciones se deben seguir para lograr una maquetación óptima en el diseño de interfaces?

Hay que mantener un diseño simple, intuitivo y fácil de usar, contar con accesibilidad multiplataforma, diseño responsive, mejorar continuamente la interfaz, minimizar errores y escuchar la opinión de los usuarios.

¿Cuáles son los lenguajes de programación más utilizados para el desarrollo de aplicaciones de escritorio?

Los principales lenguajes serian C#, C++, Python, java y javascript.

¿Cuáles son los lenguajes de programación más utilizados para el desarrollo de aplicaciones de escritorio?

Los principales lenguajes serian C#, C++, Python, java y javascript.

¿Qué lenguajes de programación son compatibles con Java para el desarrollo de aplicaciones web y de escritorio?

Para apps web se estaría html, css, Python, sql y php, y para apps de escritorio estarían C, C++ y Python como opciones principales.

¿Cuáles son las ventajas de desarrollar aplicaciones web frente a aplicaciones de escritorio?

En primer lugar, estaría la facilidad de actualizaciones y escalabilidad, la relación directa entre el server y el usuario, hay menor costo de mantenimiento ya que todo se haría en el servidor y todos los usuarios tendrían la versión mas reciente sin necesidad de descargar nada extra, y la accesibilidad multiplataforma.

¿Cuáles son las desventajas de desarrollar aplicaciones web frente a aplicaciones de escritorio?

Pues, tiene varias, en principal porque las web necesitan de acceso a internet siempre, la seguridad es un tema delicado porque, aunque son seguros, son mas vulnerables a ataques, también pierden rendimiento si las comparamos a las de escritorio y hay limitaciones de funciones porque no se puede acceder directamente a los recursos del sistema.

¿Cuáles son las ventajas de desarrollar aplicaciones de escritorio frente a aplicaciones web?

Las apps de escritorio cuentan con mayor seguridad al ser mas centralizadas, también cuentan con mayor rendimiento ya que pueden acceder directamente a los recursos del sistema, sirve sin conexión y el usuario tiene control total sobre la app.

¿Cuáles son las desventajas de desarrollar aplicaciones de escritorio frente a aplicaciones web?

la facilidad de actualizaciones, su escalabilidad, menor costo de mantenimiento ya que la actualización se haría en el servidor y todos los usuarios tendrían la versión mas reciente sin necesidad de descargar nada extra, y la accesibilidad multiplataforma.

¿Qué lenguajes de programación son más adecuados para el desarrollo de aplicaciones web?

Pues estarían, de lado del front-end: javascript, html y css, para el back end estarían java, php, Python, C# y node.js.

¿Qué lenguajes de programación son más adecuados para el desarrollo de aplicaciones de escritorio?

Los principales serian C#, C++, Java y Python debido a su versatilidad y su usabilidad.

¿Cuál es el papel de Java en el desarrollo de aplicaciones web y de escritorio?

El papel de java esta en su capacidad para crear apps empresariales complejas y escalables, así como servidores y servicios. En el de escritorio, Java permite crear aplicaciones multiplataforma con GUIs buenas, aparte de destacar por su seguridad, portabilidad y sus frameworks.

¿Qué frameworks y herramientas de Java son más utilizados para el desarrollo web?

Como anteriormente ya lo dije, serian hibernate que facilita el acceso a la base de datos transformando objetos en tablas y spring que es una gran opción para seguridad, desarrollo de interfaces y el acceso a datos mediante spring MVC.

¿Qué frameworks y herramientas de Java son más utilizados para el desarrollo de aplicaciones de escritorio?

Para apps de escritorio, estaría Java FX y java swing, aunque esta ultima sea un poco antigua, sigue siendo usada y rentable debido a su buen rendimiento.

¿Cómo se compara Java con otros lenguajes de programación en términos de desarrollo web y de escritorio?

Java es un lenguaje potente para grandes proyectos web y de escritorio, es seguro y funciona en muchos sistemas, esta misma usa Spring para aplicaciones webs complejas y JavaFX o Swing para apps

de escritorio. Aunque a veces Java puede llegar a ser más complejo que otros lenguajes de programación, es muy bueno para proyectos de software grande y confiable.