Trabajo Práctico Nro 4

Maximiliano Alderete

Tema: Reutilización de Software

Consignas:

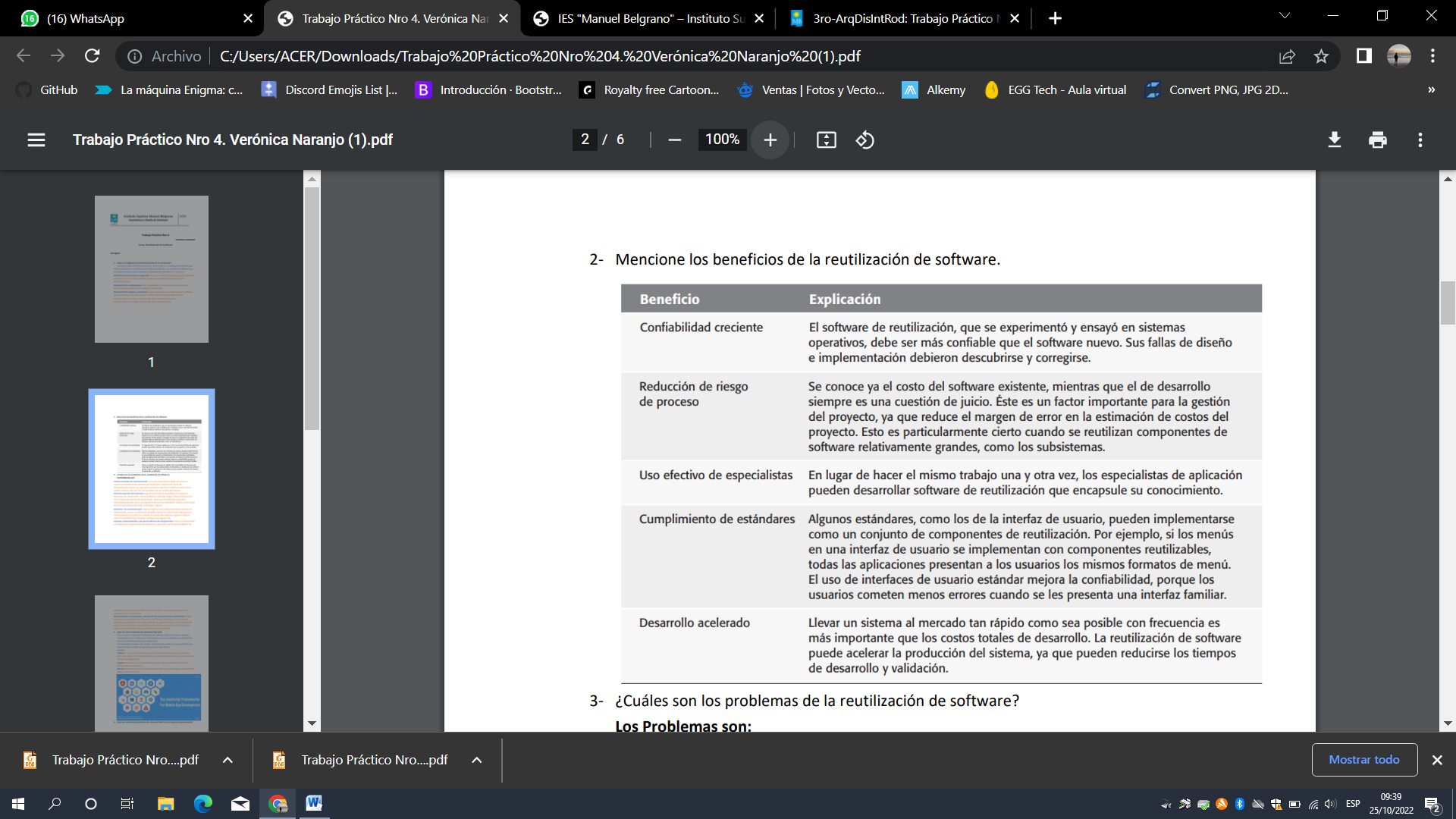
1. ¿Qué es la ingeniería de software basada en la reutilización?
2. Mencione los beneficios de la reutilización de software.
3. ¿Cuáles son los problemas de la reutilización de software?
4. ¿Qué son los frameworks de aplicación? Ejemplos
5. ¿Qué son las líneas de producto de software? Mencione los tipos de especialización.
6. - Explique que es la reutilización de productos COTS. Describa beneficios y desventajas
7. ¿Qué son los sistemas COTS integrados?
8. ¿Qué son los sistemas de solución COTS?
9. Diferencias entre sistemas d Los sistemas integrados COTS implican la integración de dos o más sistemas COTS e solución COTS y sistemas COTS integrados

Respuestas

1. La ingeniería de software basada en la reutilización es un enfoque al desarrollo que trata de maximizar la reutilización del software existente. Las unidades de software que se reutilizan pueden ser de tamaños sustancialmente diferentes. Por ejemplo:

**Reutilización del sistema de aplicación:** Todo un sistema de aplicación puede reutilizarse al incorporarlo sin cambios en otros sistemas o al configurar la aplicación para diferentes clientes.

**Reutilización de componentes**: Los componentes de una aplicación, que varían en tamaño desde subsistemas hasta objetos individuales.

 **Reutilización de objetos y funciones**: Pueden reutilizarse los componentes de software que implementan una sola función, tal como una función matemática, o una clase de objeto. Las clases y funciones en dichas librerías se reutilizan al vincularlas con un código de aplicación de desarrollo reciente.

1. **Costos crecientes de mantenimiento:** Si no está disponible el código fuente de un sistema o componente de software de reutilización, entonces los costos de mantenimiento podrían ser superiores, porque los elementos reutilizados del sistema pueden volverse cada vez más incompatibles con los cambios del sistema.

**Falta de apoyo de herramientas:** Algunas herramientas de software no apoyan el desarrollo con reutilización. Tal vez sea difícil o imposible integrar dichas herramientas con un sistema de librería de componentes. Esto es particularmente cierto para herramientas que dan apoyo a la ingeniería de sistemas embebidos, aunque menos cierto para herramientas de desarrollo orientadas a objetos.

**Síndrome “no se inventó aquí”:** Algunos ingenieros de software prefieren rescribir los componentes, porque consideran que pueden mejorarlos. Esto en parte tiene que ver con la confianza y en parte con el hecho de que escribir software original se observa como más desafiante que reutilizar el software de alguien más.

**Creación, mantenimiento y uso de una librería de componentes:** Suele ser costoso dotar a una librería de componentes de reutilización y garantizar que los desarrolladores de software puedan utilizar esta librería. Hay que adaptar procesos de desarrollo para asegurar que se use la librería.

**Descubrimiento, comprensión y adaptación de componentes de reutilización:** Deben descubrirse componentes de software en una librería, entenderse y, en ocasiones, adaptarse para trabajar en un nuevo entorno. Los ingenieros deben estar ampliamente seguros de encontrar un componente en la librería antes de incluir una búsqueda de componentes como parte de su proceso de desarrollo normal.

1. Es un conjunto integrado de artefactos de software (tales como clases, objetos y componentes), que colaboran en la facilitación de una arquitectura de reutilización para una familia de aplicaciones relacionada. Los frameworks brindan soporte para características genéricas que es probable que sean utilizadas en todas las aplicaciones de un tipo similar.

Ejemplo:

Angular: ideal para crear aplicaciones web a gran escala y de alto rendimiento, manteniéndolas fácilmente.

Meteor: Permite el desarrollo rápido de prototipos y produce código multiplataforma (Web, Android, iOS, etc.)

1. Una línea de productos de software es un conjunto de aplicaciones con una arquitectura común y componentes compartidos, con cada aplicación especializada para reflejar diferentes requerimientos.

* Especialización de plataforma Se elaboran versiones de la aplicación para diferentes plataformas. Por ejemplo, pueden existir versiones de aplicación para plataformas Windows, Mac OS y Linux. 2.
* Especialización de entorno Se crean versiones de aplicación para manejar entornos operacionales particulares y dispositivos periféricos. Por ejemplo, un sistema para los servicios de emergencia puede existir en diferentes versiones, dependiendo del sistema de comunicaciones en los vehículos.
* Especialización funcional Se crean versiones de aplicación para clientes específicos que tengan diferentes requerimientos. Por ejemplo, un sistema de automatización bibliotecario se podrá modificar en función de si se utiliza en una biblioteca pública, una biblioteca de consulta o una biblioteca universitaria.
* Especialización de proceso El sistema se adapta para hacer frente a los procesos empresariales específicos. Por ejemplo, un sistema de pedidos puede adaptarse para enfrentarse con un proceso de pedidos centralizado en una compañía y con un proceso distribuido en otra.

1. Un producto COTS es un sistema de software que puede adaptarse a las necesidades de diferentes clientes sin cambiar el código fuente del sistema. El beneficio que da es rápido en desarrollar aplicaciones y menores costos de desarrollo.

Beneficios

* .Al igual que sucede con otros tipos de reutilización, es posible la implementación más rápida de un sistema fiable.
* Es posible ver qué funcionalidad ofrece la aplicación, de manera que es más fácil juzgar si es probable que sea adecuada o no.
* Se evitan algunos riesgos de desarrollo al usar software existente. Sin embargo, este enfoque tiene sus propios riesgos, como se verá más adelante.
* Las empresas pueden enfocarse en su actividad central sin tener que dedicar muchos recursos al desarrollo de sistemas TI.
* Conforme evolucionan las plataformas operativas, las actualizaciones de tecnología se pueden simplificar, pues éstas son responsabilidad del proveedor del producto COTS y no del cliente.

Desventajas

* Tienen que adaptarse los requerimientos para reflejar la funcionalidad y el modo de operación del producto COTS. Esto puede conducir a cambios bruscos en los procesos empresariales existentes.
* El producto COTS puede basarse en suposiciones que sean casi imposibles de cambiar. Por lo tanto, el cliente debe adaptar su empresa para reflejar dichas suposiciones.
* Elegir el sistema COTS correcto para una empresa puede ser un proceso difícil, en especial porque muchos productos COTS no están debidamente documentados. Tomar la decisión equivocada podría ser desastroso, ya que tal vez sea imposible hacer funcionar el nuevo sistema como se requiere.
* Quizá no haya experiencia local para apoyar el desarrollo de los sistemas. En consecuencia, el cliente deberá apoyarse en el proveedor y en consultores externos para obtener consejos de desarrollo.
* Los proveedores de productos COTS controlan el soporte y la evolución del sistema. Pueden salir del negocio, perder el control o incluso hacer cambios que ocasionen dificultades a los clientes.

1. Los sistemas integrados COTS implican la integración de dos o más sistemas COTS (quizás de diferentes proveedores) para crear un sistema de aplicación.
2. Los sistemas de solución COTS son sistemas de aplicación generales que pueden diseñarse para dar apoyo a un tipo de empresa particular, actividad empresarial o, en ocasiones, a toda la empresa.

