

Pregunta 1: Evolución natural del desarrollo web (60%)

- a. El desarrollo de sitios web con contenido dinámico se apoyó inicialmente en la tecnología CGI. Un tiempo después, cuando la tecnología **JAVA** comenzaba a ser aceptada se produjo una fuerte transición hacia esta tecnología para la utilización de **SERVLETS**. ¿Cuáles eran las principales ventajas que validaban tal transición?

La principal ventaja es el hecho de que los CGI se ejecutan en el servidor como procesos pesados para el sistema operativo, mientras que los Servlets se ejecutan en multi-threads como procesos livianos dentro de un solo proceso pesado, que corresponde a la máquina virtual de java (JVM).

Otra ventaja importante es que los programas asociados al servidor pueden tener acceso a variables globales tales como request, sesión, etc.

Otras ventajas podrían ser las ventajas asociadas a Java en sí, tales como la independencia de la plataforma, modelo de seguridad asociado, etc.

- b. No mucho después de haberse establecido el desarrollo de aplicaciones en el lado servidor con **SERVLETS** apareció la primera versión de **JAVA SERVER PAGES**. ¿Cuál fue el principal aporte de esta tecnología?

El principal aporte de JSP es que simplifica la generación de la capa de presentación, particularmente respecto a los Servlets donde el código asociado a la capa de presentación debe inyectarse en el código de la clase. Esto por supuesto, se expresa en una mejora importante en la mantenibilidad de la aplicación.

- c. ¿Cuál fue el aporte que introduce el desarrollo de **CUSTOM TAG LIBRARIES** respecto a la tecnología **JSP**?

Los Custom Tag Libraries permiten encapsular funcionalidad a ser utilizada en la capa de presentación, y por ende mejora de manera importante la reutilización de dichas funcionalidades. Por otro lado, al encapsular las funcionalidades en etiquetas (TagLibs) mejora igualmente la legibilidad y mantenibilidad de la aplicación.

- d. ¿Considera natural la aparición de cuadros de aplicaciones que implementan el patrón arquitectural **MVC**, tales como **STRUTS** o **JSF**? ¿Cuáles serían los 'pro' y los 'contra' de utilizar estos cuadros de aplicaciones?

La tendencia actual de las aplicaciones es la implementación de un patrón arquitectural de N-capas. En este contexto, el patrón MVC facilita la implantación de este patrón arquitectural, permitiendo principalmente mantener separadas las capas de presentación, negocio y persistencia. Estos cuadros de aplicación pueden tener sus propias dificultades, sin

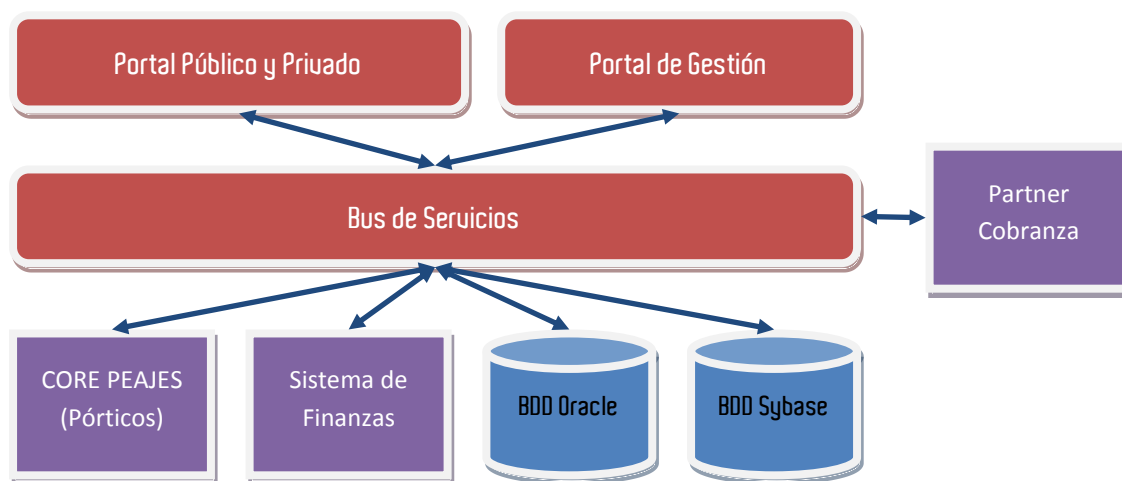
embargo podríamos decir que son menores. Tienen asociado un costo de aprendizaje, y aplicaciones de buenas prácticas, pero en general los beneficios superan ampliamente sus costos.

Pregunta 2: Evaluando una arquitectura de TI

TELEPEAJE METROPOLITANO es una empresa de administración de autopistas en la región metropolitana cuyo cobro funciona en base a peajes inalámbricos establecidos sobre pórticos por cada tramo de la autopista. Actualmente existe un sistema de TI separado e independiente para cada autopista. Esta compañía está considerando modernizar su arquitectura de TI a una arquitectura que tiende a adoptar los principios de una arquitectura orientada a servicios. Las principales directrices que la gerencia de TI tiene en mente para esta evolución son:

- Permitir aprovechar y reutilizar los sistemas existentes que procesan las 2 autopistas actualmente administradas por la compañía. En particular estos sistemas permiten administrar la parte financiera de la compañía y la interacción con una compañía externa (*Business to Business*) de cobro a los clientes.
- Extender la arquitectura de manera que nuevas autopistas administradas puedan adherirse a esta arquitectura tecnológica de manera transparente y sin que esto signifique un rediseño de la misma.
- Escalabilidad horizontal y vertical de la arquitectura, teniendo en cuenta que se piensa adquirir nuevas autopistas, y alta disponibilidad para soportar momentos de alta demanda.

La arquitectura propuesta se ha ilustrado en la siguiente figura:



Comente esta arquitectura, sus beneficios y sus posibles defectos. Proponga una mejora.

El principal problema que presenta la arquitectura propuesta es la agrupación de los portales público y privado. Al estar agrupados, un problema en un portal afecta al otro, no así al de gestión, que se encuentra arquitecturalmente separado. Los beneficios de la arquitectura son simples: el bus de servicios los deja disponibles a cualquier aplicación que los requiera, haciendo transparente la implementación subyacente. En particular, esto es aprovechado por sistemas externos, tales como el Partner de Cobranza. Al estar separado cada componente, es posible escalarlos por separado en función de las necesidades puntuales.