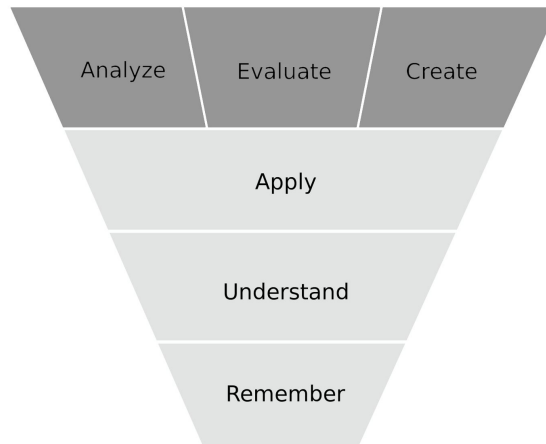


Arquitectura de Software

Trabajo Obligatorio (Marzo 2016)

Objetivo

El trabajo obligatorio tiene como finalidad diseñar y desarrollar un sistema de software a partir de una idea original propuesta por cada uno de los grupos, utilizando la tecnología estudiada en el curso y aplicando un conjunto de tácticas y/o patrones arquitectónicos.



Objetivos de aprendizaje del trabajo obligatorio

Restricciones

Un trabajo aceptable, deberá contemplar en su diseño las siguientes restricciones:

- El desarrollo se debe llevar a cabo utilizando como referencia las tecnologías provistas en la materia, basadas en la **Plataforma JEE 6** y servidor de aplicaciones **Glassfish**. Se deben utilizar al menos **4** tecnologías (API, servicios, librerías, etc.) justificadas por medio de los Requerimientos no Funcionales a definir.
- Es obligatorio el uso de **Estándares de Codificación Java** ([ver](#)), así como también el uso de **Checkstyle** en el ambiente de desarrollo, del cual se deberá presentar evidencia.
- Deberá incluir decisiones concretas y verificables para al menos **4 atributos de calidad** a elección.
- Se deberán aplicar al menos **8 tácticas de atributos de calidad** vistas en el curso.
- El documento de arquitectura debe cumplir con los lineamientos definidos en el modelo Views & Beyond visto en el curso.
- NOTA: Los docentes podrán agregar o sugerir restricciones adicionales, particulares para los distintos trabajos propuestos.

Recomendaciones generales

A continuación se describen algunas recomendaciones a tener en cuenta para la elaboración de la propuesta del obligatorio:

- Las propuestas de trabajo, así como el eventual diseño y desarrollo, deberían hacer énfasis en la implementación, investigación y uso de un conjunto de decisiones de diseño relevantes desde el punto de vista arquitectónico.
- Los aspectos **puramente funcionales** (por ejemplo ABMs) del sistema de software a desarrollar tienen una relevancia **menor** en la evaluación final, por lo cual el mayor esfuerzo no se debería concentrar en dichos aspectos. Se recomienda limitar la cantidad de ABMs implementados a **uno** solamente.
- Toda aquella funcionalidad **irrelevante** desde el punto de vista arquitectónico o tecnológico debería tener una **prioridad baja** en la planificación del trabajo.
- Al momento de evaluar el trabajo, se valorará ampliamente la **creatividad** de la propuesta desde el punto de vista técnico, así como la aplicación de diversas decisiones arquitectónicas o tecnológicas.
- El documento de arquitectura debe poder satisfacer las siguientes necesidades de información:
 - Cómo está compuesta la aplicación?
 - Cuántos y cuáles son los elementos relevantes para cumplir con la funcionalidad?
 - Cómo están organizados estos elementos y sus relaciones?
 - Cuáles son los atributos de calidad identificados como más importantes?
 - Cuáles son las funcionalidades clave en las que quedan en evidencia los atributos de calidad identificados como más relevantes?
 - Cuáles son los elementos que intervienen en la ejecución de estas funcionalidades y cómo se manifiestan los atributos de calidad?
 - Qué tácticas se aplicaron? Dónde y cómo?
 - Cómo y dónde se instala y configura la aplicación?
 - En caso de haber analizado o probado otras alternativas de solución y haberlas descartado, por qué se descartaron?
- Se deberían establecer requerimientos no funcionales claros, concretos, verificables y adecuados al alcance y complejidad del trabajo obligatorio. A continuación se presentan algunos ejemplos:
 - ▣ "El sistema debe proveer tiempos de respuesta adecuados"
 - ▣ "Los datos personales de los clientes serán consultados muy frecuentemente, por lo cuál se deberá minimizar la latencia al acceder a dicha información"
 - ▣ "El sistema debe responder con una latencia promedio menor a 1 segundo procesando 100 transacciones por segundo"
 - ▣ "Debe ser posible ejecutar reportes por medio de servicios web asíncronos"
 - ▣ "Para lograr una buena experiencia de usuario, se debe evitar que los clientes tengan que esperar que el sistema complete operaciones que insumen más de 10 segundos. En su lugar el usuario debería recibir notificaciones por parte del sistema cuando se completan dichas operaciones"

- ▣ "El sistema debe ser a prueba de fallos, brindando una disponibilidad de 100%"
- ▣ "El componente de control de inventario lleva a cabo un conjunto de operaciones críticas por lo cual se deberán auditar y registrar todos los fallos identificados durante la ejecución de dichas operaciones"
- ▣ "El componente de pagos a proveedores deberá corregir al menos el 95% de los defectos detectados"

- ▣ "Se debe desarrollar un sistema modular, aumentando la cohesión y manteniendo el bajo acoplamiento entre los módulos"
- ▣ "Los parámetros que afectan al módulo de predicción de resultados deben poder definirse en tiempo de configuración, sin impacto en el código fuente de la aplicación"

Aspectos formales

- Fecha límite para definición y modificaciones del alcance del obligatorio: **23/05/2016**
- Fecha de entrega del obligatorio: **23/06/2016**

- Las normas para la entrega del trabajo y la distribución de puntos se encuentran [aquí](#)

Propuesta del trabajo obligatorio

<Describir aquí brevemente el contexto del problema y la propuesta a desarrollar>

Requerimientos funcionales

<Enumere y describa aquí los requerimientos funcionales que cumplirá la aplicación propuesta>

Requerimientos no funcionales

<Describa aquí los requerimientos no funcionales significativos a tener en cuenta en su aplicación.NOTA: Los docentes podrán agregar o cambiar algunos de estos requerimientos>

Reglamento y criterio de corrección

https://docs.google.com/document/d/1CldHbJGtCLz8LHBTkL8bULdPBI_Ind9Xf7Wpop2OyJM/edit?usp=sharing