

Consideraciones sobre el trabajo práctico

Introducción

Para aprobar la materia, además de mantener una comunicación activa con el docente, participar en los foros temáticos y resolver las actividades semanales, es necesario presentar y aprobar un trabajo práctico, cuyo tema será de elección del alumno o grupo, previo acuerdo con el equipo docente, utilizando el lenguaje de programación Groovy y el framework Grails, de una aplicación de tipo empresarial que integre tecnologías y conceptos innovadores. El alcance será definido por el equipo docente en función de la cantidad de integrantes del grupo, que podrá ser de a lo sumo cuatro personas.

Entrega parcial

Objetivo

Seleccionar un espacio de problemas o dominio de aplicación lo suficientemente complejo como para plantear una serie de [historias de usuario](#) no triviales que tengan lugar en el mismo y acordar con el equipo docente cuáles resolver en la entrega final.

Alcance

Esta primera entrega condiciona la entrega final, de forma tal que las historias de usuario presentadas durante el coloquio son las propues en esta instancia. A criterio del equipo docente, según la complejidad y la cantidad de integrantes del grupo, pueden ser más o menos que las propuestas originalmente. Además, si bien no deben ser triviales, es recomendable que no sean tan complejas que no puedan ser implementadas en un cuatrimestre.

¿Cómo es la primera entrega?

Consiste en compartir, con permisos de edición al docente asignado y en la fecha que será oportunamente comunicada durante la cursada, un único documento de [Google Docs](#) que incluya al menos las siguientes secciones:

Carátula

- Título o nombre de fantasía del trabajo práctico
- Año y cuatrimestre de cursada
- Nombre, apellido, patrón y carrera de cada uno de los integrantes del grupo.

Introducción

Un texto breve de entre 500 y 1000 palabras, que describa a grandes rasgos el dominio elegido y cuáles son las problemáticas que podrían presentarse en él. Debe ser lo suficientemente claro como para que una persona que no es experta en ese dominio en particular pueda entender de qué se trata.

Historias de usuario

Una serie de historias de usuario no triviales, escritas en formato [Behavior-Driven Development](#) que sean propias del dominio descrito en la introducción y no sean comunes a cualquier aplicación. En particular, se consideran triviales las historias de usuario relacionadas con la administración de entidades en términos de altas, bajas y modificaciones.

Modelo del dominio

Diagrama de clases del [modelo del dominio](#), indicando relaciones de cardinalidad, herencia, composición y métodos públicos.

Repositorio de código

El código fuente que implementa el modelo del dominio y las pruebas unitarias del mismo debe estar disponible en un repositorio público de [GitHub](#) o [Bitbucket](#).

Entrega final

Objetivo

Mostrar un prototipo funcional de aplicación que implemente los casos de éxito de las historias de usuario presentadas en la entrega final y acordadas con el equipo docente, haciendo uso de los conceptos, técnicas y tecnologías explicados durante el desarrollo de la cursada.

Alcance

Como mínimo, cada una de las historias de usuario presentadas en la entrega parcial y acordadas con el equipo docente debe resultar funcional. No es necesario considerar validaciones obvias o historias de usuario triviales, pero sí las historias secundarias relacionadas con las reglas de negocio que se están tratando. No se trata de desarrollar un sistema completo en cada una de sus características sino un prototipo funcional de algunos de sus aspectos centrales. De esta forma, debe calcularse el alcance teniendo en cuenta que debería poder desarrollarse durante un cuatrimestre.

Restricciones

Los lenguajes, plataformas y herramientas utilizados en el desarrollo deben ser de [código abierto](#).

¿Cómo es la entrega final?

La presentación final del trabajo práctico consiste en una exposición oral que cuente con una introducción al tema utilizando [diapositivas](#) y una demostración del funcionamiento de la aplicación, siguiendo cada

una de las historias desde la interfaz de usuario hasta el mecanismo de persistencia, explicando las tecnologías usadas, problemas y soluciones encontrados arquitectura propuesta, modularización usada y descomposición funcional del trabajo. en particular se hará hincapié en la distancia usada entre el lenguaje del espacio de la solución implementada y el usado en el espacio real del problema. Esto es: qué tan cercanos son la redacción de la introducción, la de las historias de usuario y el código fuente que las implementa.

Criterios de evaluación

- General
 - Existencia de una entrega parcial previa
 - Calidad de la presentación: ortografía, sintaxis, estética...
 - Participación de cada integrante del grupo
- Exposición
 - Cumplimiento de las precondiciones: conectividad, datos de prueba ambientes...
- Implementación
 - Distancia entre el lenguaje usado en la introducción, las historias de usuario y el código fuente.
 - Modularización
 - Claridad
 - Dependencias

Lecturas complementarias

[Git Handbook](#)