

----Final----

## **1-Desarrollo de sistema donde apliquen todos los temas de la materia**

### **Generalidades**

Este trabajo puede ser individual o en grupos de a lo sumo dos alumnos (sin excepción). Para poder rendir el final necesitan **ANTES** haber aprobado un trabajo integrador. El trabajo comprende el diseño, la implementación y la documentación de un sistema en Java que utilice programación orientada a objetos, acceso a base de datos y manejo de excepciones. El sistema debe tener una interfaz gráfica y, donde aplique, usar componentes gráficas reusables tal como vimos durante la cursada.

### **Qué comprende el trabajo?**

Antes de comenzar con el desarrollo del trabajo, se debe acordar con el profesor el tema y la arquitectura propuesta.

Una vez acordada la propuesta y previo a la codificación del sistema deberá: escribir el documento de requerimientos, realizar el diagrama de casos de uso y el diseño (diagrama de clases y para las operaciones más relevantes diagrama de secuencia y de actividad).

Además del sistema funcionando y la documentación antes descrita, deberá escribir una guía de uso de la aplicación que explique cómo utilizarla y grabar un video que muestre el funcionamiento del sistema para ser fácilmente reproducible y mostrable (utilice camStudio-<http://camstudio.org/> o similar).

### **Forma de entrega y corrección**

Se debe hacer una entrega en CD/DVD con toda la documentación, la guía de uso, el video y el sistema. No necesita imprimir nada. Para la corrección se debe acordar una fecha con el profesor, previa a la mesa de finales donde se presente..

### **Ejemplos de aplicaciones**

Juego de ahorcado, gestión de pacientes de un consultorio, historia clínica electrónica, Sistema para gestión de mascotas en adopción, Sistema para manejo de empleados de aseguradora, Sistema para administración de habitaciones de hoteles, Agenda de contactos con gestión y seguimiento de eventos, Sistema para gestión de compra/venta de juegos.  
ok.

### **Backend**

El backend se debe desarrollar en Java y se promueve el uso de Hibernate para el mapeo objeto-relacional.

### **Cliente**

La aplicación puede ser una cliente:

- Desktop con Swing (como los que hicimos en clase)
- Desktop con SWT y RCP (para los que tengan ganas de investigar)
- Un applet
- Una aplicación web usando algun framework web (algunas opciones: Play, Wicket, ZK, JSF+Rich Faces).
- Otra tecnología (.NET por ejemplo).

## **Documentación**

Para la documentación, tomen como ejemplo el caso de estudio de un cajero automático que presenta el libro de la materia (que está disponible en la fotocopiadora): "Como programar java 7ma edicion, Deitel". El ejemplo está desperdigado por los siguientes capítulos: 2.9, 3.10, 4.15, 5.11, 6.14, 7.14, 8.19, 10.9. Incluso, está implementado.

En fotocopiadora están todos estos capítulos juntos bajo el título de 'caso de estudio completo'.

## **2-Exámen teórico**

Se tomarán todos los temas vistos en clase más algunos que deberán estudiar por su cuenta (descriptos más abajo). Cualquier duda me escriben a [uai.matias.butti@gmail.com](mailto:uai.matias.butti@gmail.com) y combinamos una consulta.

Los capítulos del libro de Deitel: **2, 3,4,5,6,7,8,9,10,19** tratan los temas que vimos en la teoría y aplicaron en la práctica. Sirven como repaso. En la fotocopiadora les voy a dejar estos capítulos bajo el título '**repaso**'.

Los temas extras que entrarán en el teórico son: applets, generics, manejo de excepciones. Los capítulos del libro correspondiente a estos temas son: manejo de excepciones (capitulo 13), applets (capitulo 20), y genéricos (capitulo 18). En la fotocopiadora están estos capítulos bajo el título '**temas extra**'.

Además, del libro "Patterns Of Enterprise Application Architecture" de Martin Fowler (versión digital en este mismo folder) lo siguiente:

Páginas 9-16  
 Páginas 22-27  
 Páginas 30-35  
 Páginas 88-94  
 Páginas 321-324  
 Patrón Domain Model (pg 88).  
 Patrón Unit of work (pg 285).  
 Patrón Remote Facade (pg 321).  
 Patrón Data transfer Object (pg 333).

(Para los patrones no es necesario que lo lean completo. Quiero que entiendan el concepto)

Si fuera necesario, el examen teórico será reforzado con una defensa oral.

### **3-Defensa de artículo**

Quienes pasen el examen teórico deberán defender un artículo de tecnología. Esta parte del examen es individual.

Para ello, deberán elegir dos artículos de [www.infoq.com](http://www.infoq.com) de los últimos 6 meses. Deben traerlo impreso y defenderlo ante preguntas que les voy a hacer de forma oral. Para la defensa del tema pueden utilizar una presentación powerpoint o simil.

### **----Recuperatorio de materia-**

Deben haber aprobado los TPs de programación estructurada y programación orientada a objetos.

El examen teórico tendrá el mismo contenido que para el final (ver punto 2-Examen teórico).

El día del examen implementarán uno de los ejercicios de programación estructurada ejercitados durante la primera parte del año y un ejercicio de Programación orientada a objetos.