

Pauta Entregable I (Informe + Modelos + Software)  
Análisis y Diseño de Software / Fundamentos de Ingeniería de Software

Integrantes:

|  |  |  |
| --- | --- | --- |
| Nombre | Email | Teléfono |
|  |  |  |
|  |  |  |
|  |  |  |

**Elimine las siguientes dos páginas y las secciones en rojo antes de entregar el informe.**

# Pauta para Entregable/Informe I

## Objetivo

Presentar las consideraciones iniciales respecto al proyecto a desarrollar y los elementos requeridos para el Entregable I.

## Consideraciones generales

-Todos los grupos de trabajo **(3 personas)** trabajan en el mismo proyecto. Sin embargo, se espera y acepta diversidad en las formas de trabajo e implementación.

-Se ha establecido como obligatorio el uso de las siguientes herramientas para **modelado y gestión del desarrollo de software**:

* **Visual Paradigm CE** (modelado, UML)
* **Trello** (kanban)
* **Github** (issue tracker)
* **Github** (scm)

-Se evaluará **buenas prácticas** del desarrollo. Así también buenas prácticas en el uso de las herramientas mencionadas (mayores detalles en siguiente sección). Todos los proyectos deben ser *públicos* en las herramientas mencionadas.

-Se realizarán **revisiones por pares**. Los ayudantes informarán el *match* entre grupos. El resultado de esta revisión debe concluir en un reporte conciso de 1 página (máximo 2 si es necesario). El reporte de revisión debe ser entregado a no más de una semana a contar de la fecha límite para el Entregable. La evaluación de este reporte tiene una ponderación de 20% en la nota del “Proyecto”.

-Su grupo de trabajo tendrá un ayudante asignado por todo el semestre.

-Todos los diagramas, a excepción del modelo de la base de datos, deben ser realizados utilizando **UML 2.0**. Debe usar una herramienta acorde a esta consideración.

## Consideraciones sobre las herramientas

Las herramientas mencionadas serán utilizadas siguiendo buenas prácticas del desarrollo de software. En particular:

* **Trello**: la herramienta implementa un Kanban que debe tener, al menos, tres columnas: Backlog, In Progress, Done. Si usted juzga necesario mantener más columnas, puede hacerlo. Los elementos a incluir en las columnas deben ser **minimal marketeable features (MMF)**. No está permitido el uso del Kanban como issue tracker, es decir, no se aceptan ítems como “corregir defecto en código xxxxx”.
* **Issue Tracker de Github**: la herramienta implementa un **issue tracker** o gestor de incidencias. En esta herramienta deben ir todas las incidencias encontradas. Se incluye aquí elementos como “corregir acento en palabra xxxxx de página inicial”, “aplicar refactoring a código en archivo xxxx.xxxx”, “corregir defecto en algoritmo xxxx”, etc. No deben ir en esta herramienta elementos de tipo MMF.
* **Github**: la herramienta es para **gestionar las versiones de su proyecto**. Todo el código y la documentación relacionada son ítems de configuración. Es de gran importancia que usted utilice esta herramienta para **gestionar versiones y NO como backup**. Si usted tiene código que no compila o con errores, o documentación incompleta, NO debe subirla a Github.

*Tanto los ayudantes como el profesor estarán revisando periódicamente el trabajo sobre estas herramientas. El uso de estas herramientas también es evaluado.*

## Contenido del Entregable I

1. Este informe (Informe I)
2. Modelos
3. Primera parte del software.

## Contenido del Informe I (ver detalles más abajo)

1. Listado de requerimientos

## Contenido de los Modelos (ver detalles más abajo)

1. Casos de uso
2. Diagrama de secuencia del sistema
3. Modelo (ER) de base de datos

## Contenido del Software (primera parte)

1. Base de datos *funcionando* (acorde al modelo entregado en el documento)
2. Funcionalidad que será indicada por su profesor y/o ayudantes en su respectivo paralelo

## Entrega

El Entregable I completo (informe + modelos + software) debe ser subido a GitHub (ver notas más arriba) **hasta el día 11/09/2016 a las 23:55 horas**. El informe debe ser subido en formato PDF. Los modelos deben ser subidos en el archivo VPP del software Visual Paradigm. Por cada día de atraso existe una penalización de 20 puntos. El ayudante asignado a su grupo de trabajo coordinará, posterior a esta fecha, una demostración breve en horario a convenir (ayudantía).

Importante: el archivo VPP debe contener todos los modelos, es decir, es uno por proyecto/equipo.

# 1- Listado de requerimientos

Listado de todos los requerimientos funcionales. Se acepta requerimientos escritos en “formato de escenario” (ej: como usuario normal, deseo poder ver mis archivos propios) y en “formato tradicional” (ej: el sistema debe proveer al usuario la opción de almacenar fotografías”).

Indique, para cada requerimiento funcional, si es obligatorio o deseable.

Utilice una tabla con tres columnas: id de requerimiento (ej: FR1), requerimiento, obligatoriedad.

Desde los requerimientos funcionales, indique 4 requerimientos no funcionales.

Utilice una tabla con tres columnas: id de requerimiento (ej: NFR1), requerimiento no funcional, lista de FRs asociados (utilice el id de requerimiento funcional para indicar a cuáles se asocia cada uno de estos 4 requerimientos no funcionales).

# 2- Casos de uso

IMPORTANTE: Escriba cada caso de uso en un documento Word y enlácelo al Caso de Uso (UML) respectivo en la herramienta Visual Paradigm CE. Los diagramas de casos de uso deben ser entregados en el archivo de modelo de Visual Paradigm CE.

Seleccione 8 requerimientos **complejos** del listado de la sección anterior y escriba casos de uso breve para cada uno de ellos (utilice “negrita” para destacar actores).

De los 8 casos de uso en formato breve, seleccione los 4 más complejos y escríbalos en formato expandido identificando claramente actores (primarios, secundarios, offstage), precondiciones, postcondiciones, cursos normales y alternativos.

Construya 1 diagrama de casos de uso (considere todos los casos de uso del sistema). Utilice, cuando corresponda, las relaciones de extensión, inclusión y generalización.

# 3- Diagramas de secuencia del sistema

IMPORTANTE: Los diagramas de secuencia deben ser entregados en el archivo de modelo de Visual Paradigm CE.

Elija 2 de los casos de uso en formato expandido y construya diagramas de secuencia **del sistema** para ellos. Recuerde notar claramente cursos alternativos y repeticiones.

Para todos los efectos, considere al sistema como caja negra (i.e., no descomponga el sistema en componentes).

# 4- Modelo relacional de la base de datos

IMPORTANTE: El modelo de la base de datos debe ser entregado en el archivo de modelo de Visual Paradigm CE.

Presente una versión inicial lo más completa posible del modelo relacional para la base de datos que utilizará su sistema.