

Guía de la Tercera Entrega del proyecto

Competencias a desarrollar:

- Toma en cuenta la opinión de los futuros usuarios al mostrarles la aplicación.
- Desarrolla las clases sobre el patrón Modelo, Vista, Controlador para separar las responsabilidades básicas de interfaz, control y modelación entre las clases del sistema.
- Aplica programación defensiva con *try-catch* y aplica los principios de desarrollo orientado a objetos vistos en clase
- Amplía el o los diagramas de clases UML de la entrega anterior para incluir las funcionalidades nuevas.
- Aplica las tecnologías disponibles o desarrolla las propias para implementar la solución planteada.
- Planifica las actividades a completar por el grupo de desarrollo para la culminación exitosa del proyecto.

Profundización sobre requisitos funcionales:

Para la entrega anterior desarrolló una lista de cada de opciones con las que podrá interactuar el usuario cuando use su sistema. Los requisitos funcionales allí listados fueron priorizados y la entrega consistió en un prototipo que satisficiera los requisitos de más alta prioridad. Para esta entrega debe continuarse trabajando sobre el listado de requerimientos. Esto implica revisar el listado de requerimientos pendientes, añadir y/o eliminar los elementos que sea necesario; y efectuar una nueva fase de priorización para establecer las metas de esta entrega.

El prototipo de la entrega anterior debe ser ampliado para presentar al usuario la experiencia diseñada con el sistema, en su totalidad. Esto significa que todas las opciones de interacción para el usuario deben estar habilitadas y el sistema debe funcionar correctamente ante pruebas con usuarios. Este prototipo debería, por tanto, presentar persistencia de datos en donde sea necesario y como el grupo lo considere adecuado (bases de datos, archivos de texto, archivos JSON, etc.).

No se espera que este prototipo presente todos los aspectos estéticos a la perfección ni que la experiencia con el usuario sea la mejor, todavía. Tampoco se espera que el prototipo sea demasiado robusto contra ingresos inválidos y potenciales errores. El objetivo de este prototipo es tener un sistema con todas las funcionalidades, para trabajarle mejoras, correcciones, programación defensiva y aspectos estéticos en las próximas dos entregas.

Identificación y descripción de clases necesarias:

Identifique las clases que necesiten modificación para completar la solución al problema. Aplique las modificaciones necesarias respetando el patrón Modelo-Vista-Controlador y los principios de programación orientada a objetos. Agregue las clases que considere necesario. Aplique los cambios correspondientes a su diagrama UML antiguo.

Para cada uno de los cambios introducidos y las clases nuevas desarrolladas, elabore una descripción donde especifique qué cambio se ha hecho, dónde, por qué y cuál será su impacto en el resto del sistema existente.

Aplicación de la tecnología disponible:

De las tecnologías investigadas para la entrega anterior se espera que se apliquen o desarrollen aquellas necesarias para completar los requisitos funcionales identificados como necesarios para esta fase. Por el requerimiento de persistencia de datos, cualquier tecnología para almacenamiento de información, como bases de datos, debería ser implementada en el sistema para este prototipo, por poner un ejemplo.

Planificación y gestión:

Elabore una lista con las tareas propuestas para completar los requisitos funcionales especificados por el grupo, junto con el miembro del equipo que será responsable de cumplir cada tarea y su fecha probable de terminación. Siga los siguientes pasos:

- Liste los requisitos que se implementarán en esta entrega con su respectiva prioridad con respecto al desarrollo del proyecto.
- Divida los requisitos funcionales que desarrollará en tareas más pequeñas.
- Especifique cada tarea:
 - o Nombre de la tarea
 - o Descripción de la tarea
 - o Horas estimadas de desarrollo
 - o Responsable de desarrollarla
 - o Fecha probable de terminación de la tarea
- Haga un calendario de planificación donde incorpore todas las tareas que redactó con su respectiva fecha de terminación.

Cada integrante del grupo debe llenar el formulario 1 con la información de las tareas que le fueron asignadas y el tiempo que le tomó elaborarlas. El formulario está lleno con información hipotética para que sirva como ejemplo del contenido que debe llevar. En la tabla 2 puede encontrar las instrucciones y el propósito del formulario.

Implementación:

- Implemente como mínimo 1 requisito funcional de los determinados **por miembro del equipo**. La implementación debe estar completa, de manera que quede listo para mostrar a usuarios.
- En el caso de que no lograra terminar algunos de los requisitos funcionales propuestos, haga una lista de las tareas que le quedaron pendientes.

Control de versiones:

- Utilice Github (<https://github.com/>) o el repositorio Git en línea de su preferencia como controlador de versiones. Cada uno de los miembros debe crear un usuario si no tiene ya uno, y tod@s deben trabajar sobre el mismo repositorio de código. Debe haber, al menos 2 *commits* semanales por miembro del grupo de desarrollo.

Formulario

Nombre: **Pepito Pérez**

Carné: **16789**

Fecha	Inicio	Fin	Tiempo Interrupción (min)	Delta Tiempo (min)	Tarea	Comentarios
05/08/2016	08:00	10:00	5	115	Diagrama de clases	Me paré a tomar un café
06/08/2016	14:00	15:00	30	30	Análisis de posibles errores	Vi un pokémon y tuve que ir a la poképarada por pokébolos

Formulario 1. Gestión del tiempo en el cumplimiento de las tareas planificadas

Objetivo	<ul style="list-style-type: none"> • Registrar el tiempo gastado en cada fase del proyecto • Usar estos datos para completar el Resumen Plan Proyecto
General	<ul style="list-style-type: none"> • Registrar todos los tiempos gastados en el proyecto • Registrar el tiempo en minutos • Ser lo más exacto posible • Si requiere espacio adicional usar otra copia del formulario
Fecha	Fecha en que se registra el tiempo
Inicio	Hora en que comienza a trabajar en la tarea
Fin	Hora en que deja de trabajar en la tarea
Tiempo Interrupción	Cualquier interrupción que ocurre durante la tarea y la razón de esta. Ej. Teléfono, baño. Si son varias entrar el tiempo total en minutos
Delta tiempo	Tiempo en minutos real gastado en la tarea (Fin – Inicio) – Interrupción
Tarea	Entrar el nombre o sigla de la fase, paso o tarea en la que esté trabajando
Comentarios	Cualquier comentario interesante

Tabla 2. Objetivos e instructivo para llenar el formulario 1

Material a entregar en Canvas:

- Archivo .pdf que incluya:
 - Los requisitos funcionales que se desarrollarán en esta entrega.
 - La identificación y descripción de las modificaciones a clases y/o clases nuevas.
 - La planificación de las tareas.
 - Registro de retroalimentación de usuarios potenciales (al menos un usuario por miembro del grupo)
 - Los formularios de registro de trabajo de cada uno de los integrantes del grupo.
- Archivo de imagen (.jpg o .png) con el diagrama de clases incluyendo cambios, clases nuevas y comentarios al diseño.
- Archivos de código (extensión .java).
- Link del repositorio del proyecto en github.

Evaluación Grupal:

- **Requisitos funcionales (15 puntos):** se revisó la lista de requisitos funcionales ya desarrollada para identificar, añadir y/o eliminar los requisitos para esta entrega, y se presentó el listado de estos requerimientos, ordenados por una nueva fase de priorización.
Los requisitos para esta entrega deben completar la funcionalidad esperada del sistema diseñado, aunque no resultar necesariamente eficiente o elegante en ello. Véase como el primer avión en despegar del suelo. Ciertamente cumplió con la meta de transportar a una persona por el aire, pero definitivamente requirió muchas mejoras para prestar el servicio que hoy tenemos en aerolíneas. Del mismo modo, su sistema necesitará mejoras como éstas a lo largo de las próximas dos entregas, no ésta.
- **Descripción de cambios y clases nuevas (10 puntos):** el grupo de desarrollo describe a detalle cada una de las clases a añadir y/o cambios a realizar al sistema existente. El objetivo es que se entienda dónde se hará cada cambio, por qué y cuál será su efecto sobre el resto del sistema.
- **Retroalimentación de usuarios (15 puntos):** evidencia de la retroalimentación obtenida al probar (virtualmente) el sistema con usuarios potenciales (personas que estén relacionadas con la problemática elegida por el grupo). Recordar que la obtención de retroalimentación no es una encuesta ni una entrevista, sino una observación del sistema en uso para saber qué podría pasar con el usuario cuando “nosotr@s l@s desarrollador@s no estamos allí con ell@s”. El apoyo provisto a los usuarios potenciales durante la obtención de retroalimentación debe reducirse al mínimo necesario.
- **Diseño del sistema (15 puntos):** el sistema aplica el patrón Modelo-Vista-Controlador y los principios de programación OO vistos en clase. Se modificó el diagrama de clases entregado en la fase anterior, usando la notación del estándar UML. Las relaciones entre las clases son las adecuadas. Se incluyeron, en el diagrama, todas las clases, métodos y atributos necesarios para los requisitos de esta entrega sean cumplidos.
- **Código y programa funcionando (30 puntos):** el código entregado cumple con los requisitos funcionales, sigue los principios de programación OO, está debidamente documentado y aplica los conceptos y

herramientas vistos en clase hasta ahora. Debe entregarse el código ADEMÁS del enlace al repositorio en github.

- **Planificación (15 puntos):** se elaboró una lista de tareas en la que se reparte de forma equitativa el trabajo entre tod@s l@s integrantes del grupo de desarrollo. Tod@s l@s integrantes tienen tareas en las que tienen que programar y elaborar parte del análisis y el diseño. Se incluyen las tareas de muestra del sistema a usuarios finales en la planificación de la segunda fase del proyecto. Se evidencia en el registro de contribuciones al repositorio en GitHub el trabajo equitativo que ha contribuido cada miembro del grupo.

Evaluación individual:

Cada estudiante tendrá una nota individual (**Ni**) del proyecto, que se calcula de la siguiente manera:

1. Se califica el proyecto en general, de acuerdo a la distribución de puntos descrita en esta guía.
2. Se realiza un promedio del porcentaje de autoevaluación (**A**) y coevaluación (**C**) de sus compañeros de grupo. La autoevaluación y coevaluación son actividades que cada miembro del grupo debe realizar para calificar su propio desempeño y el de sus compañer@s en esta fase del proyecto.
3. El porcentaje promedio individual multiplica la nota del proyecto (**Np**) en general.

$$Ni = [(A+C1+C2+C3) / 4] * Np$$

Importante: se ignorarán, del cálculo de nota individual, las auto y coevaluaciones que no sean realizadas. Si ningún miembro del grupo realiza su auto y coevaluaciones, la nota de la entrega será 0 para todo el grupo.