

## Guía de la Segunda Entrega del proyecto

### Competencias a desarrollar

- Toma en cuenta la opinión de los futuros usuarios al mostrarles la aplicación.
- Detecta las clases necesarias para modelar el sistema que resuelva el problema planteado separando claramente las clases que implementan la interfaz con el usuario y las que ofrecen funcionalidades.
- Describe el propósito de cada una de las clases modeladas, sus atributos y responsabilidades.
- Elabora el diagrama de clases utilizando UML.
- Investiga las tecnologías disponibles para implementar la solución planteada.
- Planifica las actividades a completar por el grupo de desarrollo para la culminación exitosa del proyecto.

### Identificación de requisitos funcionales:

Identifique qué debe hacer su programa. Haga una lista de cada una de las opciones con las que podrá interactuar el usuario cuando lo use. Estos son requisitos funcionales.

### Priorización de los requisitos funcionales encontrados:

Añada una prioridad a los requisitos funcionales que encontró, los que tienen prioridad 1 son los que son más importantes y sin ellos el sistema no puede funcionar. Esos son los que implementará primero.

### Identificación y descripción de clases necesarias:

Detecte las clases que necesita para modelar la solución del problema. Recuerde que debe tener la interfaz con el usuario reservada a una (o más) clases. Las clases que modelen los demás componentes del sistema y su comportamiento deben estar separadas y no interactuar directamente con (*i.e.*, recibir datos de, ni desplegar datos a) el usuario.

Para cada una de las clases elabore una descripción donde especifique para que se usará, así como el propósito de atributos y métodos.

### Diseño del sistema:

Una vez que tiene identificadas las clases, muestre en un diagrama de clases, usando UML, cómo es la relación entre las mismas. Recuerde incluir notas explicativas en sus diagramas, si lo necesita.

### Investigación de la tecnología disponible:

Investigue las herramientas, lenguajes de programación y tecnología usada para resolver problemas similares. Por ejemplo, si su solución es una aplicación *web*, investigue qué herramientas y lenguajes se emplean para hacer páginas *web*; lo mismo para aplicaciones móviles. También considere soluciones que empleen tecnologías diferentes a problemas similares al planteado por su grupo.

### Planificación y gestión:

Desglose el proyecto en tareas que luego serán asignadas a uno o más miembros del grupo. Elabore una lista con las tareas propuestas, el miembro del equipo que será responsable de cumplirla y la 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 de la entrega.
- Divida los requisitos funcionales que desarrollará en tareas más pequeñas, por ejemplo (tome en cuenta que la mención y uso de bases de datos en este ejemplo tiene usos puramente didácticos, y no plantean un requerimiento del proyecto aunque se exhorta a investigar y aplicar su uso si es posible):
  - **Requisito Autenticar (Login):**
    - Crear tabla usuario en la base de datos.
    - Hacer y probar la clase que permita conectar con la base de datos.
    - Hacer y probar la clase Usuario, agregarle los métodos buscar usuario y autenticarse.
    - Hacer y probar la clase controladora del usuario.
    - Hacer y probar la clase GUI que permita autenticarse.
- Especificar cada tarea:
  - Nombre de la tarea
  - Descripción de la tarea
  - Horas estimadas de desarrollo
  - Responsable de desarrollarla
  - 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 encontrados **por miembro del equipo** de desarrollo. 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/>) como controlador de versiones. Cada uno de los miembros debe crear un usuario si no tiene ya uno, y todos deben trabajar sobre el mismo repositorio de código. Debe haber más de 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

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

**Tabla 2.** Objetivos e instructivo para llenar el formulario 1

### Material a entregar en canvas

- Archivo .pdf que incluya:
  - o Los requisitos funcionales.
  - o La identificación y descripción de las clases.
  - o La investigación de la tecnología usada y los problemas similares que se han resuelto.
  - o La planificación de las tareas
  - o 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 final.
- Archivos de código (extensión .java).
- Link del repositorio del proyecto en github.

### Evaluación Grupal

- **Requisitos funcionales (15 puntos):** se elaboró una lista de los requisitos funcionales a desarrollar en la construcción del sistema. Las funcionalidades descritas resuelven, o fundamentan una solución para, el problema que se detectó.
- **Identificación y descripción de clases (20 puntos):** el grupo de desarrollo describe correctamente cada una de las clases que identificó, se comprende el propósito de cada clase, de cada atributo y de cada método. La cantidad de clases detectadas es suficiente para resolver el problema planteado. Tiene correspondencia con las clases implementadas en su código.
- **Diseño del sistema (20 puntos):** el sistema fue diseñado separando la interfaz de las funcionalidades. Se elaboró el diagrama de clases usando la notación del estándar UML. Las relaciones entre las clases son las adecuadas. Se incluyeron en el diagrama todos los métodos y atributos detectados en la etapa de análisis, incluidos los *setters*, *getters*, constructores y `toString()` de cada clase. Están presentes TODAS las clases identificadas.
- **Implementación del sistema (20 puntos):** el grupo programó un prototipo del sistema, implementando el diseño planteado y desarrollándolo de forma incremental con apoyo del controlador de versiones.
- **Investigación de la tecnología a usar (10 puntos):** Se investigó sobre las tecnologías existentes para resolver el problema y selección de la tecnología a usar, así como soluciones a problemas similares que hayan usado (o no) la tecnología investigada.
- **Planificación (15 puntos):** Se elaboró una lista de tareas en la que se reparte de forma equitativa el trabajo entre todos los integrantes del grupo de desarrollo. Todos los 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.

### **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.

$$N_i = [(A+C1+C2+C3) / 4 ] * N_p$$

**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.