

PRINCIPIOS DE DISEÑO DE SOFTWARE  
Rúbrica del trabajo final

Integrantes del equipo:

---



---



---



---

Elementos	Ptos.	Contenido semántico	Contenido sintáctico	Ptos. Obtenidos
1. Introducción	0.5	Explicando de que trata el documento y las partes de que está constituido.	Media hoja tamaño carta	
2. Definición de requerimientos				
2.1. Diagrama de paquetes	0.5	Diagrama de paquetes de caso de uso	Una hoja tamaño carta. . Debe elaborarse con alguna herramienta de modelado, ya sea StarUML o EA.	
2.2. Diagrama de Casos de Uso por paquete	0.5	Diagrama que contiene todos los casos de uso identificados.	Una hoja tamaño carta por diagrama. Debe elaborarse con alguna herramienta de modelado, ya sea StarUML o EA.	
2.3. Detalle de Casos de uso.	2.5	Descripción de casos de uso que incluye, para cada caso de uso: nombre, descripción, precondition(es), actor(es), flujos normal, alterno y de excepción, y postcondición(es). Todos los CU deben indicar, a lo largo de sus flujos, los objetos que en él participan y éstos deben corresponder al diagrama de clases del punto 3.1. Para los flujos normal, alterno y de excepción considerar al menos lo siguientes casos: exitoso, fallido, indeseable y un alterno.	Una hoja tamaño carta máximo por cada CU	
2.4. Diagramas de actividades	2.5	Un diagrama de actividades por CU. Un diagrama de actividades debe incluir las acciones que llevan a la realización del CU. Cada diagrama de actividades debe incluir los flujos normal, alterno y de excepción considerados en el CU. Debe elaborarse con alguna herramienta de modelado, ya sea StarUML o EA..	Debe elaborarse con alguna herramienta de modelado, ya sea StarUML o EA. Una hoja tamaño carta máximo por cada diagrama de actividades	
3. Diseño				
3.1. Diagrama de clases del modelo de dominio	2.5	Diagrama de clases del modelo realizado en java. Debe elaborarse con alguna herramienta de modelado, ya sea StarUML o EA.	El diagrama debe contener todas las clases que se emplean en las descripciones de los CU.. Una hoja tamaño carta o las que sean necesarias de manera que SE VEA	
3.2. Diagramas de secuencia por caso de uso	3	Cada diagrama debe tener a la izquierda el texto el CU. A la derecha se listan en primer lugar el actor, seguido de los objetos que corresponden a la interfaz (conforme al prototipo del punto 4) y a las clases del modelo de dominio (punto 3.1). Cada diagrama debe contener el flujo básico, así como los flujos alternos y	Deben elaborarse en alguna herramienta de modelado, ya sea EA o StarUML. Idealmente cada curso no debe exceder de una hoja tamaño carta.	

PRINCIPIOS DE DISEÑO DE SOFTWARE  
Rúbrica del trabajo final

		de excepción bien identificadlos. Debe elaborarse con alguna herramienta de modelado, ya sea StarUML o EA. Debe presentarse en el mismo orden que el punto 2.3.		
3.3. Diagramas de comunicación por caso de uso	3	Un diagrama de colaboración por CU, en el que intervengan los objetos que corresponden a las clases del modelo de dominio y a las de la interfaz. Debe presentarse en el mismo orden que el punto 2.3.	Deben elaborarse en alguna herramienta de modelado, ya sea EA o StarUML. Idealmente cada curso no debe exceder de una hoja tamaño carta	
3.4. Diagramas de estado por objeto	2	Un diagrama de estado para los objetos de aquellas clases que, de acuerdo al problema, sea necesario.	Deben elaborarse en alguna herramienta de modelado, ya sea EA o StarUML. Idealmente cada curso no debe exceder de una hoja tamaño carta	
4. Prototipo	2.5	El prototipo incluye el diseño de todas las pantallas del sistema, organizadas por CU y en el mismo orden que en el punto 2.3.	Debe desarrollarse en una herramienta como Pencil o Balsamiq	
5. Conclusiones	0.5	Donde se anota el avance logrado, lo que queda pendiente de realizar y la viabilidad del sistema.		

<b>TOTAL DE PUNTOS DE LA RÚBRICA</b>	<b>20</b>
<b>TOTAL DE PUNTOS OBTENIDOS POR EL EQUIPO</b>	