Rúbrica

Prueba - Sistema de Clientes

- La evaluación tiene un total de 10 puntos.
- En caso de que el/la estudiante no entregue ningún proyecto, recibirá el puntaje mínimo.

Crear y cargar el proyecto Java	0,25 Puntos	0,5 Puntos	0,75 Puntos	1 Punto
	El código no es ejecutable según los requerimientos entregados.	*El código incorpora todos los paquetes y clases solicitadas, sin embargo, no inicia el menú en consola con los métodos requeridos.*	*El código incorpora todos los paquetes y clases solicitadas, además inicia el menú en consola con los métodos requeridos y es ejecutable desde Eclipse.*	*El código incorpora todos los paquetes y clases solicitadas, además inicia el menú en consola con los métodos requeridos y es ejecutable desde otros programas (Eclipse, NetBeans, entre otros).*
Utilizar Ciclos y métodos	0,5 Puntos	1 Punto	1,5 Puntos	2 Puntos
	*El código incorpora al menos **1** flujo de sentencias repetitivas, integrando los conceptos de iteración, contador y acumulador (for, if, switch, while,do, try, etc) . Sin embargo, no utiliza de manera correcta el ciclo para la recepción de parámetros y sobrecargas.*	*El código incorpora al menos **2** flujos de sentencias repetitivas, integrando los conceptos de iteración, contador y acumulador (for, if, switch, while, do, try, etc). Utiliza al menos un ciclo anidado o no anidado para recibir parámetros y sobrecargas.*	*El código incorpora al menos **3** flujos de sentencias repetitivas, integrando los conceptos de iteración, contador y acumulador (for, if, switch, while, do, try, etc) . Utiliza ciclos anidados y/o no anidados para recibir parámetros y sobrecargas.*	*El código incorpora al menos **4** flujos de sentencias repetitivas, integrando los conceptos de iteración, contador y acumulador (for, if, switch, while,do, try, etc) . Utiliza ciclos anidados y/o no anidados para recibir parámetros y sobrecargas.*
Utilizar Arreglos y archivos	0,5 Puntos	1 Punto	1,5 Puntos	2 Puntos
	El código incorpora la librería Collections y Streams. Sin embargo, no trabaja de manera correcta con agrupaciones de variables (arreglos), no utiliza iteraciones para recorrer elementos y tampoco incorpora un archivo de texto para la obtención de datos.	*El código incorpora la librería Collections y Streams. Trabaja con agrupaciones de variables en forma de arreglos estáticos y/o dinámicos. Sin embargo, no utiliza iteraciones para recorrer elementos y tampoco incorpora un archivo de texto para la obtención de datos.*	*El código incorpora la librería Collections y Streams. Trabaja con agrupaciones de variables en forma de arreglos estáticos y/o dinámicos. Utiliza iteraciones para recorrer elemento a elemento, arrojando y rescatando excepciones. Sin embargo, no incorpora un archivo de texto simple para la obtención y manejo de*	*El código incorpora la librería Collections y Streams. Trabaja con agrupaciones de variables en forma de arreglos estáticos y/o dinámicos. Utiliza iteraciones para recorrer elemento a elemento, arrojando y rescatando excepciones. Además, incorpora un archivo de texto para la obtención y manejo de datos.*

			datos.	
Aplicar instancias al Paradigma Orientado a Objetos (POO)	0,5 Puntos	1 Punto	1,5 Puntos	2 Puntos
	Mantiene un código limpio, legible y ordenado. Sin embargo, no crea e instancia clases de manera correcta. No utiliza estos métodos de instancia para un objeto. No diferencia entre variables de instancia de variables locales y tampoco crea alguna clase que herede de otra.	*Mantiene un código limpio, legible y ordenado. Crea e instancia clases de manera correcta. Utiliza estos métodos de instancia para un objeto. Sin embargo, no sabe diferenciar entre variables de instancia de variables locales. No crea alguna clase que herede de otra.*	*Mantiene un código limpio, legible y ordenado. Crea e instancia clases de manera correcta. Utiliza métodos de instancia para un objeto. Diferencia variables de instancia de variables locales. Sin embargo, no crea alguna clase que herede de otra.*	*Mantiene un código limpio, legible y ordenado. Crea e instancia clases de manera correcta. Utiliza métodos de instancia para un objeto. Diferencia variables de instancia de variables locales. Crea al menos una clase que herede de otra.*
Aplicar polimorfismo y herencias al Paradigma Orientado a Objetos (POO)	0,5 Punto	1 Punto	1,5 Puntos	2 Puntos
	Las clases creadas no presentan cohesión, ni acoplamiento (atomización). No se generan interfaces, ni clases abstractas con los principios de diseño. No utiliza el POO para resolver un problema de baja complejidad con polimorfismo y herencia.	*Las clases creadas presentan componentes de alta cohesión y acoplamiento (atomización). Sin embargo, no genera interfaces, ni clases abstractas con los principios de diseño. No utiliza el POO para resolver un problema de baja complejidad con polimorfismo y herencia.*	*Las clases creadas presentan componentes de alta cohesión y acoplamiento (atomización). Genera interfaces y clases abstractas utilizando los principios de diseño. Sin embargo, no utiliza el POO para resolver un problema de baja complejidad con polimorfismo y herencia.*	*Las clases creadas presentan componentes de alta cohesión y acoplamiento (atomización). Genera interfaces y clases abstractas utilizando los principios de diseño. Además, utiliza el POO para resolver un problema de baja complejidad con polimorfismo y herencia.*
Aplicar pruebas unitarias	0,25 Puntos	0,5 Punto	0,75 Puntos	1 Punto
	El set de pruebas no integra de manera correcta JUnit.	*El set de pruebas integra JUnit para verificar el correcto funcionamiento de métodos en el software, sin embargo, no realiza ninguna prueba unitaria de manera correcta.*	*El set de pruebas integra JUnit para verificar el correcto funcionamiento de métodos en el software y se realizan al menos 1 prueba unitaria de manera correcta.*	*El set de pruebas integra JUnit para verificar el correcto funcionamiento de métodos en el software y se realizan al menos 2 pruebas unitarias de manera correcta.*