

**Final 2 - Fecha 07/07/2026**

Resolvé los siguientes problemas en forma clara y legible. Si te parece que el código, diagrama o justificación no son suficientemente explicativos, podés agregar una descripción más detallada que lo complementa.

**Las secciones tachadas de la resolución y las escritas en lápiz serán ignoradas**

Todas las hojas que entregues deben tener nombre, apellido, padrón, número de hojas y cantidad total de hojas.

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | <p>"Has de favorecer la composición sobre la herencia." - Gang of Four.</p> <p>¿Qué opinas de esta afirmación? ¿A qué se refiere la frase? Responder estas preguntas asociándolo con los conceptos vistos en clase.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 2 | <p>Suponiendo que se tiene la siguiente función en un lenguaje de programación arbitrario:</p> <pre>func gana_arg(z, empata_messi, lo_gana_enzo) {     if (!empata_messi(z) or lo_gana_enzo(z))         return z     else         return NONE }</pre> <p>Definir la misma funcionalidad en la sintaxis del Cálculo Lambda, utilizando la cantidad de expresiones lambda que considere.</p> <p><i>NOTA: cualquier elemento que se necesite del encoding de Church (Números, Booleanos, operadores, etc) deben ser definidos.</i></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 3 | <p>Una plataforma de venta de paquetes turísticos, escrita en Java, gestiona la creación de viajes personalizados. Actualmente cuenta con las siguientes entidades:</p> <ul style="list-style-type: none"> <li>- <b>PaqueteTuristico</b>: es la entidad central que representa el viaje. Tiene más de 15 atributos, incluyendo destino, fechas, vuelos (ida y vuelta), categoría de hotel, traslados, alquiler de auto, tipo de seguro médico, lista de excursiones incluidas, régimen de comidas y plan de pago.</li> <li>- <b>ArmadorDeViajes</b>: es la clase encargada de ensamblar y devolver los paquetes. Para crear las distintas ofertas comerciales, tiene métodos como <code>armarPaqueteAventura()</code>, <code>armarPaqueteCaribeAllInclusive()</code>, <code>armarPaqueteEjecutivo()</code>, entre otros.</li> </ul> <p>En cada uno de estos métodos, el <b>ArmadorDeViajes</b> instancia la clase <b>PaqueteTuristico</b> utilizando un único constructor que recibe todos los parámetros posibles. Cuando un componente no corresponde al paquete (por ejemplo, no hay alquiler de auto en el paquete Caribe, o no hay excursiones en el Ejecutivo), se le pasa null o listas vacías en la llamada al constructor. Además, las reglas de validación complejas están escritas directamente dentro de los métodos del <b>ArmadorDeViajes</b>. Periódicamente, el área comercial solicita agregar nuevos servicios opcionales.</p> <ol style="list-style-type: none"> <li>1. Indicar qué problemas tiene el diseño actual del sistema.</li> <li>2. <b>Proponer cambios en el diseño</b> para que los submódulos queden lo más abstraídos que se pueda de la lógica de envíos. Mostrar un diagrama de clases del nuevo diseño.</li> </ol> |
| 4 | <p>Se desea desarrollar un sistema central de gestión para una red eléctrica inteligente (Smart Grid) a nivel provincial. La red está compuesta por miles de nodos heterogéneos (plantas solares, turbinas eólicas, subestaciones transformadoras y bancos de baterías de respaldo). Cada nodo físico posee un estado altamente dinámico y crítico (nivel de carga, temperatura, desgaste, voltaje actual).</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | <p>El sistema monitorea la red recibiendo decenas de miles de lecturas por segundo desde los medidores inteligentes y debe reaccionar de manera inmediata a picos de demanda imprevistos, caídas de tensión o alertas meteorológicas (como ráfagas de viento que afectan a las turbinas).</p> <p>A su vez, para decidir cómo balancear la carga y qué nodos activar o desactivar, el sistema central ejecuta de forma continua un modelo matemático de optimización. Este algoritmo toma el flujo masivo de datos como entrada y, mediante cálculos complejos sobre matrices de distribución, genera el esquema de ruteo energético más eficiente sin interrumpir el servicio.</p> <p><b>¿Cuál(es) paradigma(s) de programación crees más conveniente(s) de los siguientes para resolver este problema?</b> Explicar qué ventajas y desventajas tendría utilizar cada uno de estos <b>para este problema</b>:</p> <ul style="list-style-type: none"><li>- Orientado a Objetos (POO)</li><li>- Funcional</li><li>- Lógico</li><li>- Orientado a Eventos (POE)</li></ul> <p>Tip: Mencionar ventajas/desventajas de cada paradigma aplicado al problema propuesto, no mencionar ventajas/desventajas genéricas de ellos.</p> |
| 5 | <p>En Scala, escribir las siguientes funciones:</p> <ul style="list-style-type: none"><li>• <b>curry</b>: dada una función de dos argumentos, devuelve su equivalente currificada.</li><li>• <b>uncurry</b>: dada una función currificada de dos argumentos, devuelve su versión no currificada equivalente. Es la inversa de la anterior.</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |