

Criterio / Tarea	It.1	It.2	It.3	Observaciones / Mejoras realizadas
Tiempo empleado (mins.)	10 mins	30 mins	35 mins	El tiempo empleado vario según la dificultad de las iteraciones y los elementos nuevos a integrar
Ventana funcional con JFrame y visibilidad	X	X	X	Se logro desde la primera iteración
Tiempo empleado (mins.)	10 mins	30 mins	35 mins	El tiempo empleado vario según la dificultad de las iteraciones y los elementos nuevos a integrar
Componentes añadidos: JTextField, JLabel, JButton	X	X	X	Los componentes se añadieron correctamente y se ajustó su posición.
Evento del botón funcionando correctamente	X	X	X	Se implementó el ActionListener para el botón 'Saludar'.
Validación de entrada implementada		X	X	Se agregó la validación para nombres vacíos en la segunda iteración.
Modularización (métodos o clases separadas)			X	Se refactorizó el código para separar la lógica de la GUI en una clase y métodos dedicados.
Uso de clase auxiliar (ej. Usuario)			X	Se creó la clase Usuario para manejar la lógica del saludo, mejorando la modularidad.
Soporte para tecla ENTER		X	X	Se añadió un KeyListener al JTextField para activar la acción del botón.
Estilo del código claro y comentado		X	X	Se aplicaron nombres claros a las variables y se organizó el código
Botón limpiar, nueva funcionalidad			X	Se agregó un nuevo botón para limpiar el campo de texto y la etiqueta de saludo.
Reflexión final del estudiante:	<p>Esta actividad me permitió aplicar los principios de la programación orientada a objetos (POO) en la creación de una interfaz gráfica (GUI) funcional con Java Swing. Al refactorizar el código, logré modularizar las responsabilidades, separando la lógica de la interfaz.</p> <p>Los objetivos principales, como la gestión de eventos de usuario (clic de botón y tecla Enter) y la validación de entrada, se cumplieron con éxito. La creación de la clase auxiliar Usuario para generar el saludo y el nuevo botón de "Limpiar" demuestran un claro entendimiento de las buenas prácticas de programación.</p>			