

		Niveles de Proficiencia	Nivel 4: Accomplished	Nivel 3: Proficient	Nivel 2: Developing	Nivel 1: Beginning	Nivel 0
		Valor Numérico	5.0-4.5	4.4 - 3.8	3.7- 3.0	2.9 - 1.0	0.9 - 0.0
		Criterio					
	Analisis	Especificación de Requerimientos	La especificación de requerimientos abarca todas las necesidades planteadas por el cliente, su redacción es adecuada (claridad, precisión y separación -no solapamiento-) y en el formato apropiado	Cumple en su totalidad con dos (2) de las siguientes condiciones: completitud (todas las necesidades), redacción adecuada y formato de cada requerimiento	Incluye al menos el 50% de las necesidades del cliente y es posible reconocer las necesidades del cliente en el texto aunque puede haber deficiencias en la redacción	Incluye al menos el 50% de las necesidades del cliente o es posible reconocer las necesidades del cliente en el texto aunque puede haber deficiencias en la redacción	No cumple con ninguna de las condiciones necesarias para una adecuada especificación de requerimientos
	Diseño	Diseño de la Solución [1]	El diagrama modela todos los conceptos y sus relaciones necesarios para dar solución al problema, incluye en cada concepto y relación la información detallada necesaria para modelar adecuadamente la solución y cumple con la notación propia del lenguaje en que está soportado. El modelo plantea una adecuada separación y asignación de responsabilidades entre los conceptos (clases). Las características (atributos) y el comportamiento (métodos) de cada concepto evidencia la satisfacción de todas las funcionalidades planteadas en los requerimientos	Se modelan todos los conceptos y relaciones para dar solución al problema e incluyen adecuadamente los detalles propios de cada concepto y relación, y se utiliza completamente la notación apropiada, aunque podría no evidenciarse la separación de responsabilidades y la satisfacción de todos los requerimientos	Se modelan todos los conceptos y relaciones para dar solución al problema e incluyen adecuadamente los detalles propios de cada concepto y relación aunque no se utilice completamente la notación apropiada	Se modela al menos el 50% de los conceptos y relaciones que dan solución al problema, así como también al menos el 50% de los detalles de cada uno de ellos	El modelo entregado no cumple suficientemente con ninguna de los condiciones planteadas
		Trazabilidad entre Análisis y Diseño	Para cada requerimiento funcional evidencia todos los métodos que permiten satisfacerlo. Para cada método indica a qué clase pertenece.	Para al menos el 70% de los requerimientos funcionales evidencia todos los métodos que permiten satisfacerlo. Para cada método indica a qué clase pertenece.	Para al menos el 50% de los requerimientos funcionales evidencia al menos el 70% de los métodos que permiten satisfacerlo. Para cada método indica a qué clase pertenece	Para al menos el 25% de los requerimientos funcionales evidencia al menos el 50% de los métodos que permiten satisfacerlo	Evidencia al menos 1 método que permite satisfacer menos del 25% de los requerimientos o no entrega evidencia de la relación entre requerimientos y métodos del diagrama de clases
	Implementación	Codificación del Modelo	En el programa se codifica completamente un modelo que da solución a todas las necesidades del cliente y se corresponde completamente con el diseño planteado	El modelo implementado da solución al menos al 70% de las necesidades del cliente	El modelo implementado da solución al menos al 50% de las necesidades del cliente	El modelo implementado da solución al menos al 25% de las necesidades del cliente	No hay implementación del modelo o se da solución a menos del 30% de las necesidades o tiene errores de compilación
		Implementación estática de la Interfaz Gráfica de Usuario (FXML)	La interfaz de usuario presenta los elementos gráficos necesarios para el cumplimiento del 100% de los requerimientos funcionales	La interfaz de usuario presenta los elementos gráficos necesarios para el cumplimiento de al menos el 70% de los requerimientos funcionales	La interfaz de usuario presenta los elementos gráficos necesarios para el cumplimiento de al menos el 50% de los requerimientos funcionales	La interfaz de usuario presenta los elementos gráficos necesarios para el cumplimiento de al menos el 25% de los requerimientos funcionales	La interfaz de usuario presenta los elementos gráficos necesarios para el cumplimiento de menos del 25% de los requerimientos funcionales
		Codificación del Control de la Interfaz de Usuario	El control de la interfaz con el usuario utiliza todas las funcionalidades provistas por el modelo de la solución y se corresponde con el diseño planteado para su implementación	La codificación del control de la interfaz utiliza al menos el 70% de las funcionalidades del modelo	La codificación del control de la interfaz utiliza al menos el 50% de las funcionalidades del modelo	La codificación del control de la interfaz utiliza al menos el 25% de las funcionalidades del modelo	No hay implementación del control de la interfaz o ésta presenta menos del 25% de las funcionalidades del modelo o tiene errores de compilación
		Buenas prácticas de codificación	La implementación de modelo e interfaz de usuario cumple completamente con los estándares de codificación y la división de responsabilidades entre la interfaz y el modelo	La implementación de modelo e interfaz de usuario cumple al menos con el 70% con los estándares de codificación y la división de responsabilidades	La implementación de modelo e interfaz de usuario cumple al menos con el 50% con los estándares de codificación y la división de responsabilidades	La implementación de modelo e interfaz de usuario cumple al menos con el 25% con los estándares de codificación y la división de responsabilidades	La implementación del modelo cumple con menos del 25% de los estándares de codificación y la división de responsabilidades
		Persistencia a través de Serialización	Implementa los elementos necesarios para persistir la información del programa usando serialización. El programa desarrollado permite guardar el estado del modelo del problema y puede restaurar el estado cuando se requiera. Con esto, satisface correctamente los requerimientos del problema asociados a persistencia.	Implementa los elementos necesarios para darle persistencia a la aplicación usando serialización. Sin embargo, no satisface correctamente todos los requerimientos planteados en el problema	Implementa los elementos necesarios para darle persistencia a la aplicación usando serialización. Sin embargo, presenta algunos errores al tratar de almacenar el estado del modelo o cuando intenta restaurarlo.	La implementación satisface entre el 10% y el 30% de los requerimientos de persistencia del problema usando serialización	La implementación satisface menos del 10% de los requerimientos de persistencia del problema usando serialización

		Manejo de archivos de texto	Implementa los elementos necesarios para manipular archivos de texto. El programa permite abrir, modificar y guardar archivos de texto. Con esto, logra satisfacer todos los requerimientos del problema asociados al manejo de archivos de texto	Implementa los elementos necesarios para manipular archivos de texto. Es decir, la aplicación permite abrir, modificar y guardar archivos de texto. Sin embargo, no logra satisfacer todos los requerimientos del problema	Implementa los elementos necesarios para manipular archivos de texto, de acuerdo a los requerimientos planteados en el problema. Sin embargo, presenta errores al tratar de abrir, modificar o guardar archivos de texto.	Realiza un intento por implementar el manejo de archivos de texto. Sin embargo, no logra satisfacer los requerimientos del problema	No logra implementar los elementos necesarios para manipular archivos de texto.
		Hilos	El desarrollo del programa presenta el manejo apropiado de mas de un hilo de ejecución, incluyendo el correcto diseño del mismo en el diagrama de clases y su apropiada implementación en la codificación	El desarrollo del programa presenta el manejo de mas de un hilo de ejecución, incluyendo al menos el 75% de: (1) los elementos del diseño de hilos en el diagrama de clases y (2) su implementación en la codificación	El desarrollo del programa presenta el manejo de mas de un hilo de ejecución, incluyendo entre el 60% y el 75% de: (1) los elementos del diseño de hilos en el diagrama de clases y (2) su implementación en la codificación	El desarrollo del programa presenta el manejo de mas de un hilo de ejecución, incluyendo entre el 20% y el 60% de: (1) los elementos del diseño de hilos en el diagrama de clases y (2) su implementación en la codificación	El desarrollo del programa presenta el manejo de mas de un hilo de ejecución, incluyendo menos del 20% de: (1) los elementos del diseño de hilos en el diagrama de clases y (2) su implementación en la codificación
		Manejo de Excepciones	Implementa correctamente el manejo de las excepciones, propias y del API del lenguaje, lanzadas en la solución a este problema	Implementa correctamente al menos el 70% del manejo de las excepciones, propias y y del API del lenguaje, lanzadas en este programa	Implementa correctamente al menos el 50% del manejo de las excepciones, propias y y del API del lenguaje, lanzadas en este programa	Implementa correctamente al menos el 25% del manejo de las excepciones, propias y y del API del lenguaje, lanzadas en este programa	Implementa correctamente menos del 25% del manejo de las excepciones, propias y y del API del lenguaje, lanzadas en este programa
		Pruebas Unitarias	El desarrollo de su programa incluye al menos el 90% del (1) diseño de casos de pruebas unitarias, (2) diagrama de clases de las pruebas e (3) implementación utilizando un framework de codificación de pruebas unitarias automáticas	El desarrollo de su programa incluye entre el 75% y el 89% del (1) diseño de casos de pruebas unitarias, (2) diagrama de clases de las pruebas e (3) implementación utilizando un framework de codificación de pruebas unitarias automáticas	El desarrollo de su programa incluye entre el 60% y el 74% del (1) diseño de casos de pruebas unitarias, (2) diagrama de clases de las pruebas e (3) implementación utilizando un framework de codificación de pruebas unitarias automáticas	El desarrollo de su programa incluye entre el 20% y el 59% del (1) diseño de casos de pruebas unitarias, (2) diagrama de clases de las pruebas e (3) implementación utilizando un framework de codificación de pruebas unitarias automáticas	El desarrollo de su programa incluye menos del 20% del (1) diseño de casos de pruebas unitarias, (2) diagrama de clases de las pruebas e (3) implementación utilizando un framework de codificación de pruebas unitarias automáticas

[1] Este criterio se aplicará una vez por cada diagrama requerido en la entrega del estudiante:

- Diagrama de Clases del modelo - Diagrama de Clases de la interfaz de usuario