

	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					
Método de la Ingeniería	Aplicación del Método de la Ingeniería para la solución del problema	Desarrolla de manera completa y correcta cada una de las fases del Método de la Ingeniería para dar solución al problema.	Desarrolla cada una de las fases del Método de la Ingeniería para dar solución al problema. Al menos un 70% de los componentes desarrollados en cada fase es claro, conciso y correcto.	Desarrolla al menos un 70% de las fases del Método de la Ingeniería para dar solución al problema. Al menos un 70% de los componentes desarrollados en cada fase es claro, conciso y correcto.	Desarrolla al menos un 70% de las fases del Método de la Ingeniería para dar solución al problema. Al menos un 50% de los componentes desarrollados en cada fase es claro, conciso y correcto.	Desarrolla menos de un 50% de las fases del Método de la Ingeniería para dar solución al problema. Menos de un 50% de los componentes desarrollados en cada fase es correcto.
PSP 0	Medición del tiempo	Mide el tiempo de cada fase de desarrollo. Los tiempos reportados en cada una de las etapas son consistentes en cantidad (tiempo en minutos) con el trabajo mínimo inherente a la actividad.	Mide el tiempo de cada fase de desarrollo. Los tiempos reportados en cada una de las etapas son consistentes en al menos un 70%.	Mide el tiempo de al menos un 70% de las fases de desarrollo. Los tiempos reportados en cada una de las etapas son consistentes en al menos un 50%.	Mide el tiempo de al menos un 50% de las fases de desarrollo. Los tiempos reportados en cada una de las etapas son consistentes en al menos un 25%.	Mide el tiempo de menos del 25% de las fases de desarrollo. Los tiempos reportados en cada una de las etapas son consistentes en en menos de un 25%.
	Registro de defectos	Registra los defectos y el tiempo de su reparación. La cantidad de defectos y el tiempo tomado para cada uno es consistente con el trabajo propio de la actividad.	Registra los defectos y el tiempo de su reparación. La cantidad de defectos y el tiempo tomado para cada uno es consistente en al menos un 70%.	Registra al menos un 70% de los defectos y el tiempo de su reparación. La cantidad de defectos y el tiempo tomado para cada uno es consistente con al menos un 50%.	Registra al menos un 50% de los defectos y el tiempo de su reparación. La cantidad de defectos y el tiempo tomado para cada uno es consistente con al menos un 25%.	Registra menos de un 25% de los defectos y el tiempo de su reparación. La cantidad de defectos y el tiempo tomado para cada uno es consistente con menos de un 25%.
	Registro de comentarios	Registra comentarios que resumen las actividades realizadas en cada etapa.	Registra comentarios que resumen las actividades realizadas en al menos un 70% de las etapas.	Registra comentarios que resumen las actividades realizadas en al menos un 50% de las etapas.	Registra comentarios que resumen las actividades realizadas en al menos un 25% de las etapas.	Registra comentarios que resumen las actividades realizadas en menos de un 25% de las etapas.
Diseño	Definición de Tipos Abstractos de Datos	Define todos los TAD a utilizarse, incluyendo, de manera correcta y completa todos sus componentes (nombre, descripción del objeto abstracto, invariante, y declaración de operaciones, con nombre, entradas y salida).	Define todos los TAD a utilizarse, incluyendo, al menos un 70% de sus componentes desarrollados de manera correcta.	Define al menos el 70% de los TAD a utilizarse, incluyendo, al menos un 50% de sus componentes desarrollados de manera correcta.	Define al menos el 50% de los TAD a utilizarse, incluyendo, al menos un 25% de sus componentes desarrollados de manera correcta.	Define menos del 25% de los TAD a utilizarse, incluyendo, menos de un 25% de sus componentes desarrollados de manera correcta.
	Diseño de diagramas de clase	Modela todos los conceptos y relaciones necesarias para dar solución al problema, incluyendo en éstas la información detallada necesaria para modelar adecuadamente la solución, cumpliendo con la notación propia del lenguaje en que está soportado.	Modela todos los conceptos y relaciones necesarias para dar solución al problema, incluyendo en éstas la información detallada necesaria para modelar adecuadamente la solución, aunque no cumpliendo completamente con la notación propia del lenguaje en que está soportado.	Se modela al menos el 50% de los conceptos y relaciones que dan solución al problema, así como también el 50% de los detalles de cada uno de ellos.	Se modela al menos el 50% de los conceptos y relaciones que dan solución al problema aunque se describa menos del 50% de los detalles de cada uno de ellos.	El modelo entregado no cumple suficientemente con ninguna de los condiciones planteadas.
	Diseño de Casos de Pruebas Unitarias	Diseña para cada operación de cada una de las estructuras de datos utilizadas al menos tres casos de prueba para verificar su correcto funcionamiento (uno estándar, uno que pruebe casos límite y otro interesante) cumpliendo con el formato donde se especifica la clase, el método a probar, el escenario, las entradas y la salida esperada	Diseña para cada operación de cada una de las estructuras de datos utilizadas al menos dos casos de prueba para verificar su correcto funcionamiento (uno estándar, uno que pruebe casos límite y otro interesante) cumpliendo con el formato donde se especifica la clase, el método a probar, el escenario, las entradas y la salida esperada.	Diseña para cada operación de cada una de las estructuras de datos utilizadas al menos un caso de prueba para verificar su correcto funcionamiento (uno estándar y otro interesante) en un formato que permita implementar las pruebas sin ambigüedad.	Diseña al menos un caso de prueba para al menos el 40% de las operaciones de cada una de las estructuras de datos utilizadas.	No diseña casos de prueba o lo hace para menos del 40% de las operaciones de cada una de las estructuras de datos utilizadas.
Implementación	Estructuras de Datos	Todas las estructuras de datos utilizadas funcionan correctamente. Incluyen pruebas unitarias automáticas, con casos interesantes para verificarlas.	Al menos el 70% de las estructuras de datos utilizadas funcionan correctamente. Incluyen pruebas unitarias automáticas, con casos interesantes para verificarlas.	Al menos el 50% de las estructuras de datos utilizadas funcionan correctamente. Incluyen pruebas unitarias automáticas para verificarlas.	Al menos el 25% de las estructuras de datos utilizadas funcionan correctamente. Incluyen pruebas unitarias automáticas para verificarlas.	Menos del 25% de las estructuras de datos utilizadas funcionan correctamente.
	Programa completo	El programa ofrece una solución a todas las necesidades planteadas en el enunciado.	El programa ofrece una solución a al menos un 70% de las necesidades planteadas en el enunciado.	El programa ofrece una solución a al menos un 50% de las necesidades planteadas en el enunciado.	El programa ofrece una solución a al menos un 25% de las necesidades planteadas en el enunciado.	El programa ofrece una solución a menos del 25% de las necesidades planteadas en el enunciado.
	Implementación de las Pruebas Unitarias	Implementa correctamente los casos de prueba diseñados	Implementa el 80% los casos de prueba diseñados	Implementa los casos de prueba diseñados con algunos errores	Implementa algunos de los casos de prueba	No implementa casos de prueba
	Inglés en codificación	Los identificadores y los comentarios en el código fuente están definidos en su totalidad en inglés	Los identificadores y los comentarios en el código fuente están definidos al menos en un 70% en inglés	Los identificadores y los comentarios en el código fuente están definidos al menos en un 50% en inglés	Los identificadores y los comentarios en el código fuente están definidos al menos en un 25% en inglés	Los identificadores y los comentarios en el código fuente están definidos en inglés en menos del 25%
	Uso de un sistema de control de versiones [1]	El proyecto ha sido desarrollado utilizando un sistema de control de versiones y se evidencia las continuas contribuciones (de todos los archivos del desarrollo -análisis, diseño, implementación y reporte del método de la ingeniería-) de los colaboradores (que deben ser todos los integrantes del grupo) desde el inicio del mismo hasta la fecha de entrega	El proyecto ha sido desarrollado utilizando un sistema de control de versiones y se evidencia las continuas contribuciones (de todos los archivos del diseño y la implementación) de los colaboradores (que deben ser todos los integrantes del grupo) desde el inicio del mismo hasta la fecha de entrega	El proyecto ha sido desarrollado utilizando un sistema de control de versiones y se evidencia las continuas contribuciones (de todos los archivos del diseño y la implementación) de los colaboradores (que deben ser al menos dos de los integrantes del grupo) desde el inicio del mismo hasta la fecha de entrega	El proyecto ha sido desarrollado utilizando un sistema de control de versiones y se evidencia las continuas contribuciones (de todos los archivos de la implementación) de los colaboradores (que debe ser al menos uno de los integrantes del grupo) desde el inicio del mismo hasta la fecha de entrega	No hay evidencia de que el proyecto fue desarrollado utilizando un sistema de control de versiones [2]

[1] Git
GitHub

[2] Si se evidencia que el proyecto no fue trabajado en Git durante su desarrollo (sino subido al final) entonces el trabajo tiene 0 en este criterio.