**Pauta de evaluación Módulo 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pauta de evaluación Módulo 1** | **1** | **2** | **3** | **4** |
| Utiliza estructuras de control condicional y expresiones lógicas para dar solución a un problema de condicionalidad |  |  |  | x |
| Utiliza estructuras de control repetitivas para dar solución a un problema de repetitividad |  |  | x |  |
| Utiliza estructuras de arreglos unidimensionales y bidimensionales para dar solución a un problema de baja complejidad |  |  | x |  |
| Representa un algoritmo utilizando diagramas de flujos para dar solución a un problema de baja complejidad |  |  |  | x |
| Estructura un algoritmo utilizando pseudocódigo para dar solución a un problema de baja complejidad |  |  | x |  |
| Utiliza tipos de datos, operadores, sentencias de control, ciclos y colecciones de Java para resolver un problema de baja complejidad |  |  | x |  |
| Utiliza clases predefinidas y bibliotecas del entorno Java para resolver un problema de baja complejidad |  |  |  | x |
| Analiza el código de programación Java utilizando las herramientas de depuración disponibles en el IDE para detectar errores y verificar su funcionamiento |  | x |  |  |
| Realiza la documentación del código fuente utilizando buenas prácticas y herramientas del entorno Java |  |  | x |  |
| Realiza codificación mantenible de acuerdo a buenas prácticas, convenciones y estilos de programación |  |  | x |  |
| Construir una aplicación que se ejecuta en la consola de comandos utilizando el lenguaje Java para resolver una problemática dada de baja complejidad |  |  | x |  |
| Codifica una clase en lenguaje Java utilizando constructores, accesadores y mutadores para dar solución a un problema de baja complejidad |  | x |  |  |
| Codifica clases en lenguaje Java utilizando colaboración y composición para dar solución a un problema de baja complejidad |  |  | x |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pauta de evaluación Módulo 1** | **1** | **2** | **3** | **4** |
| Codifica clases utilizando el lenguaje Java a partir de un Diagrama de Clases para resolver un problema de baja complejidad |  |  |  | x |
| Representa en un Diagrama de Clases el modelo de clases que da solución a un problema de baja complejidad |  |  |  | x |
| Codifica clases en lenguaje Java utilizando herencia para resolver un problema de polimorfismo de baja complejidad |  |  | x |  |
| Codifica clases en lenguaje Java utilizando interfaces para resolver un problema de polimorfismo de baja complejidad |  |  | x |  |
| Codifica clases en lenguaje Java utilizando principios de diseño para obtener componentes con alta cohesión y bajo acoplamiento para así dar solución a un problema de baja complejidad |  |  | x |  |
| Construye Casos de Prueba utilizando JUnit para verificar el correcto funcionamiento de una pieza de software |  |  | x |  |
| Construye una Suite de Pruebas utilizando JUnit para verificar el correcto funcionamiento de una pieza de software |  |  | x |  |
| Ejecuta una suite de pruebas utilizando JUnit para detectar errores de funcionamiento de una pieza de software |  |  | x |  |
| Construye una pieza de software utilizando prácticas de desarrollo de acuerdo con el método de desarrollo dirigido por pruebas (TDD) |  |  | x |  |

Me falta mucho todavía por mejorar, pero creo que voy en buen camino.

¡Saludos!