

Semana 6

Fundamentos de Programación (PRY2201)

Pauta de evaluación sumativa

# Depurando mi proyecto en el Entorno de Desarrollo Integrado (IDE)

A continuación, se presentan los descriptores de los niveles de logro con los que serás retroalimentado/a y/o evaluado/a:

| CL | Completamente Logrado (100%) | Logro **completo** del manejo del indicador, se considera como el punto **óptimo**, de acuerdo al área evaluada, dentro del rango de competente |
| --- | --- | --- |
| L | Logrado (80%) | Logro del indicador, se observan **pequeñas dificultades o errores** para alcanzar el logro total y sobresaliendo de lo mínimo aceptable |
| ML | Medianamente Logrado (60%) | Logro **mínimo aceptable** en el manejo del indicador que permite ser considerado competente |
| LI | Logro Insuficiente (30%) | **Logro insuficiente** en el manejo del indicador, se observan **varias dificultades o errores** que no le permiten alcanzar el logro mínimo para ser considerado competente |
| NL | No Logrado (0%) | Se observan un **escaso, nulo o incorrecto** logro en el manejo del indicador |

| N° | Criterios | Completamente Logrado (100%) | **Logrado (80%)** | Medianamente Logrado (60%) | **Logro Insuficiente** **(30%)** | **No Logrado** **(0%)** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Identifica los requisitos específicos del sistema.  **(10 puntos)** | Identifica todos los requisitos específicos, documentándolos de manera clara y detallada. | Identifica la mayoría de los requisitos, documentándolos con claridad. | Identifica algunos requisitos, pero la documentación carece de detalles o claridad | Identifica muy pocos requisitos y la documentación es insuficiente. | No identifica los requisitos específicos ni presenta documentación. |
| 2 | Diseña la estructura del programa en Java.  **(10 puntos)** | Diseña la estructura del sistema en Java de forma completa y detallada, incluyendo todas las funciones y módulos requeridos. | Diseña la estructura del sistema en Java bien elaborado, cubriendo la mayoría de las funciones y módulos necesarios. | Diseña la estructura del sistema en Java a nivel básico, cubriendo algunas funciones y módulos importantes. | Diseña la estructura del sistema en Java de manera incompleta, sin cubrir las necesidades del sistema. | No diseña la estructura del sistema. |
| 3 | Implementa un menú de venta interactivo que permita al usuario realizar diversas acciones  **(10 puntos)** | Implementa un menú excepcionalmente intuitivo, incluyendo funciones adicionales relevantes o mejoras significativas. | Implementa un menú funcional, incluyendo todas las acciones y fácil de usar. | Implementa  un menú que incluye todas las acciones, pero con algunos errores menores o mejoras en la usabilidad. | Implementa un menú funcional pero la implementación es confusa o difícil de usar. | Implementa un menú que no es funcional o falta implementar más de una acción requerida. |
| 4 | Implementa lógica para la selección de asientos, considerando entradas.  **(10 puntos)** | Implementa una lógica completa para selección de asientos, validación de disponibilidad y manejo de casos especiales. | Implementa lógica adecuadamente con pequeñas omisiones o errores. | Implementa lógica básica que cubre algunos aspectos del manejo de entradas. | Implementa lógica con errores significativos o falta de funcionalidades clave. | No implementa lógica para la selección o disponibilidad de asientos. |
| 5 | Genera boletas con información relevante.  **(15 puntos)** | Genera boletas correctamente con todos los detalles requeridos. | Genera boletas con la mayoría de los detalles necesarios. | Genera boletas con información básica y algunos errores o faltantes. | Genera boletas con datos insuficientes o con información incorrecta. | No genera boletas. |
| 6 | Declara variables que almacenan información importante.  **(15 puntos)** | Declara correctamente todas las variables, inicializándolas con adiciones creativas o mejoras significativas. | Declara casi todas las variables requeridas, inicializándolas correctamente. | Declara algunas variables requeridas, inicializándolas con algunas omisiones. | Declara pocas variables, inicializándolas con algunos problemas. | No declara variables.. |
| 7 | Incorpora puntos de depuración para optimizar la implementación del programa.  **(15 puntos)** | Incorpora todos los puntos de depuración de forma excepcional, con informes detallados de errores o integraciones avanzados. | Incorpora la mayoría de los puntos de depuración bien implementados que facilitan la identificación y corrección de errores. | Incorpora la mitad de los puntos de depuración requeridos con utilidad moderada. | Incorpora algunos puntos de depuración, pero con limitada utilidad. | No incorpora puntos de depuración. |
| 8 | Optimiza el código a través de estructuras de control que mejoren la legibilidad y mantenibilidad del código.  **(15 puntos)** | Optimiza el código con innovaciones en dos o más estructuras de control, modularización excepcional y gran legibilidad. | Optimiza el código con dos estructuras de control eficientes y buena modularización. | Optimiza el código solo con una estructura de control, con alguna redundancia o falta de modularización. | Optimiza el código con un uso básico de estructuras de control. | No optimiza el código. |



Reservados todos los derechos Fundación Instituto Profesional Duoc UC. No se permite copiar, reproducir, reeditar, descargar, publicar, emitir, difundir, de forma total o parcial la presente obra, ni su incorporación a un sistema informático, ni su transmisión en cualquier forma o por cualquier medio (electrónico, mecánico, fotocopia, grabación u otros) sin autorización previa y por escrito de Fundación Instituto Profesional Duoc UC La infracción de dichos derechos puede constituir un delito contra la propiedad intelectual.