

# **Evaluación Final de Programación I FGI104**

## **Trabajo Final. Orientaciones**



**Facilitadora:** Dr.C. Zoila Esther Morales Tabares

## ÍNDICE

Prólogo .....	2
Descripción de los proyectos .....	2
Proyecto Gestión de equipos electrónicos y mobiliario de HardSoft .....	2
Ensambladora de Carros.....	3
Organización de los equipos de proyecto .....	6
Consejos para el trabajo final .....	7
Herramienta para el diseño de algoritmos .....	7
Entorno de Desarrollo Integrado (IDE) sugeridos .....	7
Rúbrica de evaluación del trabajo final.....	10

## Prólogo

Estimado participante de Programación I,

Hemos llegado al final del curso. Este ejercicio de confirmación de saberes te permitirá retomar los aspectos claves trabajados en la asignatura.

Se espera que al concluir adquieras las **competencias requeridas** para desarrollar aplicaciones informáticas y dar respuestas a necesidades presentes en tu campo laboral.

- **Competencia específica 1.** Domina los fundamentos teóricos y técnicos en el área de Ingeniería en Software, tomando en cuenta las diferentes herramientas tecnológicas, a fin aportar soluciones en la automatización de los procesos.
- **Competencia específica 2.** Maneja las técnicas de programación, los elementos de análisis y diseño como herramientas fundamentales para el desarrollo de proyectos computacionales.

Recuerda que la honestidad académica es vital en el proceso evaluativo y cualquier acción que atente contra el código de ética de la Universidad, implica sanciones.

## Descripción de los proyectos

### Proyecto Gestión de equipos electrónicos y mobiliario de HardSoft

#### Equipo #1, Equipo #2, Equipo #3 y Equipo #4

Se necesita mantener actualizados los medios electrónicos y el mobiliario de la compañía HardSoft a través de una aplicación informática.

De los medios se conoce el tipo (equipos electrónicos y mobiliario), el número de inventario, el costo unitario y el estado en que se encuentra (Funcionando, Defectuoso y Roto).

En el caso de los equipos electrónicos se conoce además el tipo (nevera, TV, PC, abanico, aire acondicionado, laptop, caja de agua), la marca (Sony, Samsung y LG), el proveedor usual y otros posibles proveedores (defina mínimo tres) y los años de garantía.

En el caso de los mobiliarios se conoce el tipo (silla, sofá, butaca, mesa, escritorio, otros), el proveedor, el tiempo estimado de explotación y el local donde está ubicado (laboratorio, aula, departamento, Vicerrectoría).

El costo de un equipo se calcula multiplicando el costo unitario por los años de garantía y el costo de un mueble es el costo unitario por el tiempo estimado de explotación más un plus. El plus está determinado por el local donde se encuentra ubicado el mueble, siendo \$250 para cuando es un aula, \$320 para un laboratorio, \$450 para un departamento y \$500 para una oficina.

La aplicación debe permitir:

- a) Conocer la cantidad de equipos electrónicos de un mismo tipo desglosado por marcas.
- b) Conocer la cantidad total de medios (equipos y mobiliario) en estado “Roto” y desglosados por tipo.
- c) Conocer el equipo de más y menos años de garantía.
- d) Conocer los principales proveedores de equipamiento y mobiliario.
- e) Conocer el costo total por equipo y por muebles.
- f) Conocer el costo promedio por tipo de equipo y por tipo de mueble.

Elabore una aplicación que permita darle solución al problema anterior, la misma debe contar con un menú que permita entrar los datos y procesar toda la información solicitada.

## **Ensambladora de Carros**

### **Equipo #5, Equipo #6, Equipo #7 y Equipo #8**

En una planta ensambladora de carros Marca Mercedes Benz, se necesita llevar un control estricto de todos los motores y carros que se producen diariamente.

La planta debe llevar el control de la cantidad de motores y carros producidos, ficha técnica de cada uno de ellos, así como el plan de producción anual en ambos casos.

En la planta se ensamblan 3 tipos de motores, de altas revoluciones, de fuerza y de trabajo.

Para todos los casos sin importar su tipo la fábrica graba un código en el motor de 12 caracteres único para cada motor y que garantiza la autenticidad del producto. Este código debe quedar registrado en la ficha técnica de cada motor conjuntamente con la fecha de salida del área de ensamblaje, el nombre del especialista que realizó la certificación y cuántas veces ha regresado a el área de ensamblaje por defectos.

Para los motores de alta se registra además máximo de revoluciones x minuto que alcanza y consumo, (km x litros) que realiza.

En el caso de los motores de fuerza se incluye solo la cantidad de caballos de fuerza que tiene el motor que siempre es un número entero entre 80 y 4000.

Por último, para los motores de trabajo se controla si es artesanal (mandado hacer por pedido) o no.

El precio de costo de los motores varia en dependencia del tipo que sea y esté dado por las siguientes formulas:

#### *Motores de Alta*

$$\text{Costo} = (\text{máximas revoluciones} \times 1.5 + \text{km} \times \text{litros}) - 100 * \text{veces que regreso a ensamblaje}$$

#### *Motores de fuerza*

$$\text{Costo} = (\text{caballos de fuerza}) \times (5 - 100 \times \text{veces que regreso a ensamblaje})$$

#### *Motores de Trabajo*

$$\text{Costo} = 1000 - (100 \times \text{veces que regreso a ensamblaje})$$

*(aumenta su costo 10 veces si es artesanal)*

Para el ensamblaje de cada carro debe existir en la planta el motor adecuado para el mismo. Los carros pueden ser formula1, ómnibus, sports y de lujo, siendo el motor adecuado para cada caso los siguientes:

formula1 – motor de alta

ómnibus – motor de fuerza

sports – motor de trabajo

lujo – motor de trabajo (artesanal)

Para cada carro en su ficha además de recogerse el motor que tiene incorporado, se precisa la cantidad de plazas, que para el caso de los formula siempre es 1, y los sports y de lujo varían entre 2 y 4. En cambio en el caso de los ómnibus esta en dependencia de los caballos de fuerza del motor suponiendo que por cada 10 es capaz de cargar 1 persona. Además de esta información la ficha registra también la velocidad del vehículo y la fecha de salida de la planta.

En el caso de los formula1 se incluye en la ficha el peso de la carrocería, si es un ómnibus cantidad de puertas, para los Mercedes sport cantidad de velocidades de la caja y si es de cambio universal o no y por último si es de lujo costo de la tapicería.

Para cada caso el precio de venta varía también y está dado por las siguientes formulas:

#### *Formula1*

$$\text{Precio de venta} = \text{velocidad} \times 5 + 1/\text{peso carrocería} + \text{costo del motor}$$

#### *Ómnibus*

$$\text{Precio de venta} = (\text{cantidad de puertas} \times 1.5 + \text{costo del motor}) \times 3$$

## Sport

*Precio de venta = cantidad de velocidades  $\times$  2 + costo del motor (+1000 si es de cambio universal)*

## De lujo

*Precio de venta = (costo de la tapiceria + costo del motor)  $\times$  10*

Una vez ensamblado el carro el motor que se le incorpora deja de estar disponible para ensamblar otros carros, a no ser que el carro no pase la etapa de prueba del motor lo que provoca que sea desamblado (desarmado) y llevado el motor a la planta de ensamblaje, quedando el motor disponible para otro carro. Si el carro era de lujo el motor deja de ser artesanal.

Implemente un sistema en C++ que realice el control deseado en la fábrica y permita:

- a) Mostrar el listado de los motores por tipo disponibles.
- b) Mostrar los carros de mayor velocidad de la planta (más de 150 km x h)
- c) Mostrar Datos del ómnibus de mayor capacidad.
- d) Mostrar la ficha técnica de todos los carros incluyendo por cada uno la ficha técnica de su motor.
- e) Dar de baja un carro que no pase la prueba.
- f) Mostrar un listado con los carros que tengan motores reenzamblados y en cuanto disminuyo su precio de venta (solo formula1, sport, omnibus).
- g) Porcentaje de cumplimiento del plan de motores y carros de la planta.
- h) Ganancia total de la ensambladora una vez vendido los carros dividido por cada tipo.

Elabore una aplicación que permita darle solución al problema anterior, la misma debe contar con un menú que permita entrar los datos y procesar toda la información.

## Organización de los equipos de proyecto



<b>Equipo #1</b>	1) Marcos Aquino Nolasco 2) Wilber Beltr 3) Eliezel Collado Viola
<b>Equipo # 2</b>	4) Yudelka Corcino de Reyes 5) Dianny Crismeily Del Orden Santana 6) Jeremy Domínguez 7) Jennifer García Rosario
<b>Equipo #3</b>	8) Guillermo Hernández Cortorreal 9) Franklin Jimenez 10) Yohangel Ariel Jiménez Rodríguez
<b>Equipo #4</b>	11) Edrid Mejías Rijo 12) Ezequiel Mena Minaya 13) Evelin Masiel Muñoz Ceri 14) Jefany Pascual Romero
<b>Equipo #5</b>	15) Edwin Alfonso Polanco Tejada 16) Junnior Pérez Gómez 17) Leudy Yaritza Puello Reynoso 18) Genesis Milagros Quezada Taveras
<b>Equipo #6</b>	19) Deward Francisco Reyes Sosa 20) Alex Rodríguez Espinal 21) Rafael Salado Luciano 22) Erick Sencion Restituyo
<b>Equipo #7</b>	23) Aarón Sánchez Taveras 24) Shantal Soriano 25) Lola Iris Toribio ruiz 26) Divacny Tomas Contreras Ramírez
<b>Equipo #8</b>	27) Pedro José Torres Almonte 28) Samuel Alfred Ureña Hilario 29) Dahianna Vasquez Ceballos 30) Juan Francisco Lara Mejia

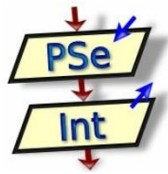
## Consejos para el trabajo final

Estudiar las **guías de contenidos** de cada Unidad puestas a su disposición, en la que aparecen ejemplos resueltos que están en la bibliografía.

**Nota.** Si considera que tiene las competencias para presentar el Proyecto Final de forma Visual, puede hacerlo como iniciativa del equipo.

## Herramienta para el diseño de algoritmos

### PSEINT



URL: <https://www.rollapp.com/app/pseint>

## Entorno de Desarrollo Integrado (IDE) sugeridos

❖ **Visual Studio:** <https://visualstudio.microsoft.com/es/>






### Visual Studio

Windows | Versión 17.1

El mejor IDE completo para desarrolladores de .NET y C++ en Windows. Completamente equipado con una buena matriz de herramientas y características para elevar y mejorar todas las etapas del desarrollo de software.

Más información >

Descargar Visual Studio



### Visual Studio para Mac


Mac | Versión 8.10

Un IDE completo nativo de macOS para desarrolladores de .NET. Incluye soporte técnico de primerísima categoría para el desarrollo web, en la nube y de juegos, además de excelentes herramientas para crear aplicaciones móviles multiplataforma.

Más información >

Más información sobre [activando su licencia](#)

Descargar Visual Studio para Mac



### Visual Studio Code

Windows | Versión 1.64

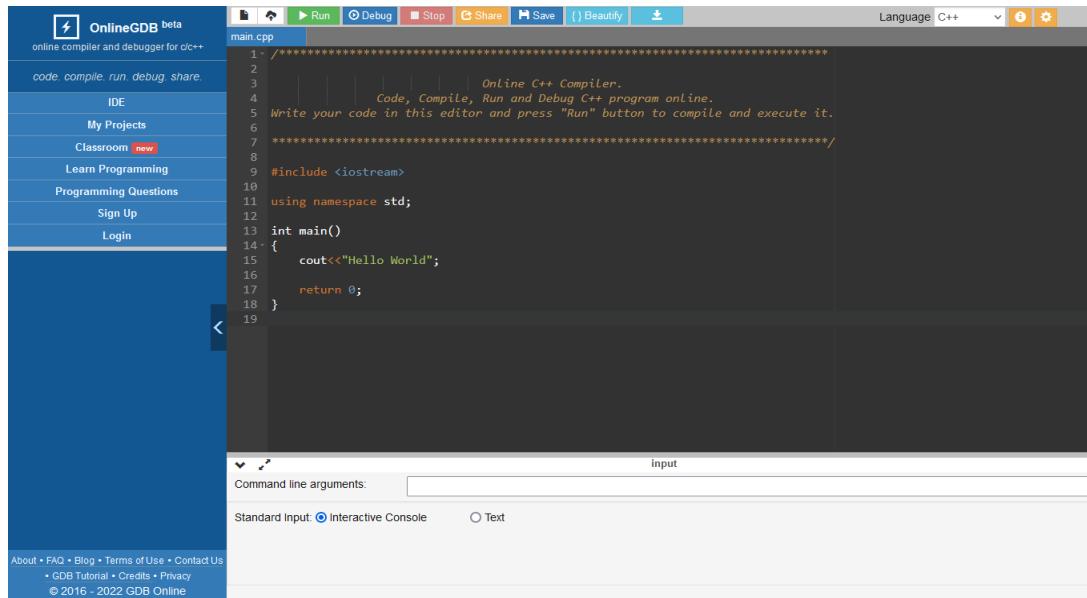
Editor de código fuente independiente que se ejecuta en Windows, macOS y Linux. La elección principal para desarrolladores web y JavaScript, con extensiones para admitir casi cualquier lenguaje de programación.

Más información >

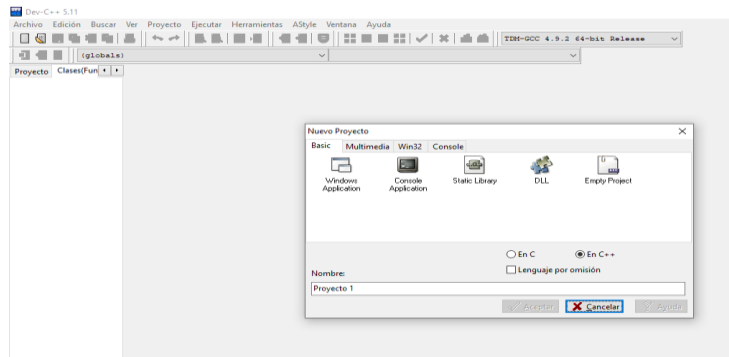
Al usar Visual Studio Code, confirma que acepta la [licencia](#) & [declaración de privacidad](#)

Descargar Visual Studio


❖ **Compilador online para C++:**  
[https://www.onlinegdb.com/online\\_c++\\_compiler](https://www.onlinegdb.com/online_c++_compiler)



❖ **Compilador Dev C++:** <https://sourceforge.net/projects/orwelldevcpp/files/latest/download>



## Rúbrica de evaluación del trabajo final



**Universidad Abierta para Adultos (UAPA)**  
**Departamento de Evaluación de los Aprendizajes**  
**Escuela de Ingeniería y Tecnología**  
**Rubrica de Evaluación – Revisión de Software**

**Asignatura:** Programación I    **Clave:** FGI-104    **Grupo:** \_\_\_\_\_

**Facilitador (a):** Zoila Esther Morales Tabares, Dr.C    **Fecha:** \_\_\_\_\_

**Participante:** \_\_\_\_\_    **Matrícula:** \_\_\_\_\_

**Trabajo final:** \_\_\_\_\_    **Valor de la evaluación:** 15 puntos

CRITERIO	EXCELENTE	BUENO	SUFICIENTE	INSUFICIENTE	CALIFIC.
<b>Desarrollo del programa</b>	La presentación del trabajo incluye todos los requerimientos especificados para la realización del programa.	La presentación del trabajo incluye al menos 85% de los requerimientos especificados para la realización del programa.	La presentación del trabajo incluye al menos 75% de los requerimientos de interfaz solicitado para la aplicación.	El diseño de interfaz no cumple con los requerimientos especificados.	3%
<b>Código</b>	El código funciona bien y de manera eficiente en el proyecto. Define: 1) Palabras reservadas o claves 2) Tipos de datos 3) Variables y declaración de variables 4) Funciones matemáticas incorporadas por el lenguaje 5) Operadores en C++ 6) Estructura del fichero de programa 7) Entrada/salida elemental	El funcionamiento del código no es completo, aunque tiene fallos de pocas relevancias.  Definido correctamente el 85% de las especificaciones: 1) Palabras reservadas o claves 2) Tipos de datos 3) Variables y declaración de variables 4) Funciones matemáticas incorporadas por el lenguaje 5) Operadores en C++	El funcionamiento del código tiene fallos importantes, pero compila.  Definido correctamente el 75% de las especificaciones: 1) Palabras reservadas o claves 2) Tipos de datos 3) Variables y declaración de variables 4) Funciones matemáticas incorporadas por el lenguaje 5) Operadores en C++	El código no funciona.  Definido correctamente el 65% o menos de las especificaciones: 1) Palabras reservadas o claves 2) Tipos de datos 3) Variables y declaración de variables 4) Funciones matemáticas	3%

	8) Operadores de entrada/salida empleando flujos 9) Condicionales 10) Estructuras repetitivas 11) Funciones 12) Llamada de funciones 13) Uso de arreglos: estáticos y/o dinámicos 14) Uso de punteros 15) Clases 16) Objetos 17) Mensajes  - Uso correcto del compilador y/o Visual Studio	6) Estructura del fichero de programa 7) Entrada/salida elemental 8) Operadores de entrada/salida empleando flujos 9) Condicionales 10) Estructuras repetitivas 11) Funciones 12) Llamada de funciones 13) Uso de arreglos: estáticos y/o dinámicos 14) Uso de punteros 15) Clases 16) Objetos 17) Mensajes  - Uso del compilador y/o Visual Studio	6) Estructura del fichero de programa 7) Entrada/salida elemental 8) Operadores de entrada/salida empleando flujos 9) Condicionales 10) Estructuras repetitivas 11) Funciones 12) Llamada de funciones 13) Uso de arreglos: estáticos y/o dinámicos 14) Uso de punteros 15) Clases 16) Objetos 17) Mensajes  - Uso del compilador y/o Visual Studio	incorporadas por el lenguaje 5) Operadores en C++ 6) Estructura del fichero de programa 7) Entrada/salida elemental 8) Operadores de entrada/salida empleando flujos 9) Condicionales 10) Estructuras repetitivas 11) Funciones 12) Llamada de funciones 13) Uso de arreglos: estáticos y/o dinámicos 14) Uso de punteros 15) Clases 16) Objetos 17) Mensajes  - Uso del compilador y/o Visual Studio	
<b>Usabilidad</b>	El diseño del programa es agradable y fácil de utilizar, no requiere de mucho tiempo para ejecutar las funcionalidades.	El diseño del programa es agradable y fácil de utilizar, sin embargo, requiere de mucho tiempo para ejecutar las funcionalidades.	El diseño del programa no es agradable, sin embargo, es fácil de ejecutar las funcionalidades.	El diseño del programa no es agradable, y es complicado el poderlo utilizar.	3%
<b>Rol dentro del grupo</b>	- Participa activamente en la organización del equipo. - Aporta opiniones, ofrece ayuda, y coopera para que el	- Participa en la organización del equipo. - Ofrece ayuda cuando considera necesario.	- Trabaja con sus compañeros de acuerdo a su rol, pero sin tomar decisiones o hacer aportes. - Hace solo lo que le solicitan.	- Intenta hacer el trabajo asignado, pero no participa de las decisiones ni	2%

	<p>trabajo se desarrolle adecuadamente.</p> <ul style="list-style-type: none"> <li>- Comprende y asume con responsabilidad su rol en el equipo de trabajo.</li> <li>- Aporta para que otros miembros del equipo asuman el suyo.</li> </ul>	<ul style="list-style-type: none"> <li>- Casi siempre comprende y asume su rol dentro del equipo y lo desempeña efectivamente.</li> </ul>	<ul style="list-style-type: none"> <li>- Comprende el rol asignado, pero no se adhiere consistentemente a él.</li> </ul>	<p>discusiones del equipo.</p> <ul style="list-style-type: none"> <li>- No hace ningún esfuerzo para asumir su rol en el equipo.</li> </ul>	
<b>Acceso a datos</b>	La aplicación hace uso de manera eficiente de las estructuras de datos declaradas (arreglos dinámicos y/o estáticos), para el almacenamiento y manipulación de los registros.	La aplicación hace uso de manera correcta de las estructuras de datos declaradas (arreglos dinámicos y/o estáticos), para el almacenamiento y manipulación de los registros en el 85% de las funciones.	La aplicación hace uso de las estructuras de datos declaradas (arreglos dinámicos y/o estáticos), para el almacenamiento y manipulación de los registros en el 75% de las funciones.	La aplicación hace uso de las estructuras de datos declaradas (arreglos dinámicos y/o estáticos), para el almacenamiento y manipulación de los registros en el 65% o menos de las funciones.	2%
<b>Paradigma de programación</b>	Para el desarrollo de la aplicación emplea el paradigma de programación Orientado a Objetos.	Para el desarrollo de la aplicación emplea en un 85% el paradigma de programación Orientado a Objetos.	Para el desarrollo de la aplicación emplea en un 75% el paradigma de programación Orientado a Objetos.	Para el desarrollo de la aplicación no emplea en el paradigma de programación Orientado a Objetos.	2%
<b>CALIFICACIÓN TOTAL</b>					15%

¡Te deseo éxitos en esta experiencia formativa!



**Zoila E. Morales Tabares**

Directora de la Escuela de Ingeniería y  
Tecnología/Agrimensura

Teléfono: (809) 724-0266 Ext.: 430

<https://www.uapa.edu.do/>