



OpenWebinars certifica que  
**Juan Miguel Carrasco Negrillo**

Ha superado con éxito  
**Programador en C++**

Duración de la ruta

**40 horas**

Fecha de expedición

**11 julio 2020**

A handwritten signature in black ink, appearing to read 'Manuel Agudo', positioned over a large, light gray watermark of the OpenWebinars logo.

CEO de OpenWebinars

**Manuel Agudo**

# Contenido

## Programador en C++

### 1. Curso de Introducción a la programación con Pseudocódigo

#### 1. Introducción a la programación

Presentación

Resolución de problemas

Análisis del problema

Diseño de algoritmos

#### 2. Entorno de trabajo: PSeInt

Instalación de PseInt en Windows

Instalación de PseInt en GNU/Linux

#### 3. Pseudocódigo: Introducción

Estructura del algoritmo

Tipos de datos simples

Variables

Operadores y expresiones

Asignación de variables

Entrada y salida de información

Otras instrucciones

Funciones matemáticas

Funciones de cadenas de texto

Nuestro primer pseudocódigo completo

Ejecución paso a paso

Secuenciales: Ejercicio III

Secuenciales: Ejercicio X

Secuenciales: Ejercicio XV

Secuenciales: Ejercicio XVII

Secuenciales: Ejercicio XVIII

Más ejercicios de secuenciales

#### **4. Pseudocódigo: Estructuras alternativas**

Estructuras alternativas: Si

Estructuras alternativas: Segun

Alternativas: Ejercicio II

Alternativas: Ejercicio V

Alternativas: Ejercicio IX

Alternativas: Ejercicio XIII

Más ejercicios de alternativas

#### **5. Pseudocódigo: Estructuras repetitivas**

Estructuras repetitivas: Mientras

Estructuras repetitivas: Repetir-Hasta Que

Estructuras repetitivas: Para

Uso específico de variables: Contadores, acumuladores e indicadores

Repetitivas: Ejercicio I

Repetitivas: Ejercicio III

Repetitivas: Ejercicios VII y X

Repetitivas: Ejercicio XI

Más ejercicios de Repetitivas

Caracteres: Ejercicio I

Caracteres: Ejercicio II

Caracteres: Ejercicio VI

Más ejercicios de caracteres

## **6. Pseudocódigo: Arreglos**

Estructuras de datos: Arreglos (array)

Arreglos unidimensionales: Vectores

Arreglos multidimensionales: Tablas

Arreglos: Ejercicio III

Arreglos: Ejercicio IV

Arreglos: Ejercicio V

Arreglos: Ejercicio VIII

Arreglos: Ejercicio X

Más ejercicios de arreglos

## **7. Pseudocódigo: Programación estructurada**

Programación estructurada

Funciones y procedimientos

Funciones recursivas

Funciones: Ejercicio III

Funciones: Ejercicio V

Funciones: Ejercicio VII

Funciones: Ejercicio XI

Funciones: Ejercicio II

Más ejercicios de funciones

## **8. Lenguajes de Programación**

Introducción a los lenguajes de programación

Programas traductores

Compilación y ejecución de un lenguaje compilado: C++

Compilación e interpretación de un programa Java

Ejecución de programas interpretados con Python

## **2. Curso de SOLID y patrones de diseño**

### **1. Introducción al curso**

Presentación

## **2. SOLID**

¿Qué es SOLID?

S: Single Responsibility Principle

O: Open Close Principle

L: Liskov Substitution Principle

I: Interface Segregation Principle

D: Dependency Inversion Principle

## **3. Principios de diseño de software**

La navaja de Ockham y los principios YAGNI, KISS y DRY

Ley de Demeter

MOSCOW

## **4. Patrones de diseño**

¿Qué son los patrones de diseño?

¿Qué objetivos tienen los patrones de diseño?

## **5. Patrones creacionales**

Abstract Factory

Builder

Factory Method

Prototype

Singleton

## **6. Patrones estructurales**

Adapter

Bridge

Composite

Decorator

Facade

Flyweight

Proxy

## **7. Patrones de comportamiento**



Chain of Responsibility

Command

Interpreter

Iterator

Mediator

Memento

Observer

State

Strategy

Template Method

Visitor

Conclusiones sobre los patrones de diseño

## **8. Antipatrones**

¿Qué son los antipatrones de diseño?

Antipatrones metodológicos

Code Smells

## **3. Curso de C++: Introducción**

### **1. Introducción a la programación**

Presentación

Análisis de problemas y diseño de algoritmos

Introducción a los lenguajes de programación

### **2. Introducción a C++**

Introducción a C++

Instalación del IDE Zinjai

Estructura del programa

Compilación y ejecución de programas

### **3. Tipos de datos básicos**

Datos y tipos de datos

Literales y constantes

Variables y operadores de asignación

Tipos de datos numéricos y operadores aritméticos

Introducción a las cadenas de caracteres

Entrada y salida estándar

Pasando de pseudocódigo a C++

#### **4. Ejercicios**

Ejercicios estructura secuencial

Ejercicio 3

Ejercicio 10

Ejercicio 17

Ejercicio 18

#### **4. Curso de Git.**

##### **1. Introducción**

Presentación

Conceptos básicos

##### **2. Herramientas de trabajo**

Instalación y configuración

Servidor GIT básico e introducción a GitLab

##### **3. Organización del código fuente y otros elementos del proyecto**

Creación de un repositorio

Aprobando cambios (Parte I)

Aprobando cambios (Parte II)

Trabajo remoto

Deshacer cambios

Reparando conflictos de unión con merge

##### **4. Flujos de trabajo**

Experimentando con ramas (Parte I)

Experimentando con ramas (Parte II)

Experimentando con ramas (Parte III)

Encontrando errores

Git Blame

Siendo "quisquilloso" con Git

Reescribir la historia de un proyecto con Rebase

Etiquetas

Moverse adelante y atrás entre estados de un repositorio git

Git Stash

## **5. Repositorios Corporativos**

Ventajas de usar un servicio en la nube: GitHub y GitLab

Flujo DevOps

## **5. Curso de Programación orientada a objetos con C++**

### **1. Introducción**

Presentación

### **2. Memoria**

Stack vs. Heap

Punteros

### **3. Estructura de datos**

Struct

Union

Enum Class

### **4. Clases**

Introducción

Instancias y memorias

Atributos

Public y private

Constructores, destructores y funciones

Struct vs. clase

### **5. Herencia**

Introducción



Vtables

Override, protected y friend

Polimorfismo

## **6. Nuestro primer videojuego**

Nuestro primer videojuego

## **6. Curso de Programación genérica y concurrente con C++**

### **1. Introducción**

Presentación

### **2. Macros**

Introducción

Casos de uso

Consejos y trucos

### **3. Templates**

Introducción

Casos de uso

Ejemplos prácticos

Macros vs. templates

### **4. Threading**

Introducción

Librería thread STL

Control de flujo

Casos de uso

### **5. Procesos**

Introducción

Comunicación entre procesos

Thread vs. procesos

## **7. Curso de Patrones de diseño e interfaces en C++**

### **1. Introducción**

Presentación



## **2. STL**

Introducción

Contenedores secuenciales

Contenedores asociativos

Iteradores

Smart pointers

Librerías

## **3. Interfaces**

Clase abstracta

Usos de las interfaces

## **4. Patrones de diseño**

Introducción

Patrones creacionales

Patrones estructurales

Patrones de comportamiento

## **5. Organización de código**

Resumen. Creando un videojuego