



# Introducción al Pensamiento Computacional -

## Semana 9



© Todos los derechos reservados Universidad Rafael Landívar URL.

---

### DESEMPEÑOS ESPERADOS

---

- ≡ Desempeños esperados

---

### DESARROLLO DE CONOCIMIENTOS

---

- ≡ Desarrollo de conocimientos

---

### APLICANDO LO APRENDIDO

---

- ≡ Actividad 1

- ≡ Actividad 2

- ≡ Actividad 3

- ≡ Actividad 4

## **DE LA TEORÍA A LA PRÁCTICA Y REFLEXIÓN**

---

### **☰ Recursos**

#### **CRITERIOS DE EVALUACIÓN**

---

### **☰ Rúbrica de evaluación**

### **☰ Diario de experiencias de laboratorio**

#### **FUENTES DE REFERENCIAS**

---

### **☰ Referencias**

#### **CRÉDITOS**

---

### **☰ Créditos**

# Desempeños esperados

---



## El estudiante:

1

Comprende la importancia de separación de código.

2

Analiza cómo el contexto puede controlar el acceso a la información.

3

Aplica módulos que puedan:

- Integrar un programa complejo.
- Facilitar la reutilización.

- Proveer interfaces para ocultar información.

4

Explora mejores prácticas para organizar las soluciones.

## Desarrollo de conocimientos

---



## Desarrollo de conocimientos

---

“Quienes diseñan los porta-aviones comprenden cómo construir sistemas modulares. Cuando el sanitario de un porta-aviones deja de funcionar, no empieza a

disparar misiles. Esto es porque el sistema sanitario y el de misiles están separados ¡Y POR MUCHO!", Tanebaum (2010).

En el módulo 3 estudiamos la descomposición, como un método para resolver problemas, mediante su detalle en piezas más pequeñas que pueden ser resueltas individualmente. Mientras lo hacemos, vamos buscando patrones.

Hasta el momento, ya debe estar familiarizado con los bloques que constituyen el código, como instrucciones (enunciados y expresiones), condicionales, ciclos y funciones.

Para programas muy pequeños, con algunas docenas de código, estos bloques son suficientes. Sin embargo, mientras los programas incrementan su tamaño, se requieren mejores maneras para organizar el código.

## Métodos en C#



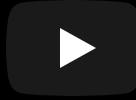
### Métodos: Guía de C#

Un método es un bloque de código que contiene una serie de instrucciones. Un programa hace que se ejecuten las instrucciones al llamar al método y especificando los argumentos de método necesarios. En C#, todas las instrucciones ejecutadas se realizan en el contexto de un método.

[MÁS INFORMACIÓN MICROSOFT >](#)

## Ámbitos de las variables en C#

 [YOUTUBE](#)



### AMBITO DE VARIABLES EN CSHARP | C# .NET

AMBITO DE VARIABLES EN CSHARP | C# .NET  
<https://c-sharp-desde-cero.blogspot.com/2020/04/accesibilidad-y-ambitos-en-csharp-net.html>  
Esta es la lista de todo el ...

[VER EN YOUTUBE >](#)

 [YOUTUBE](#)



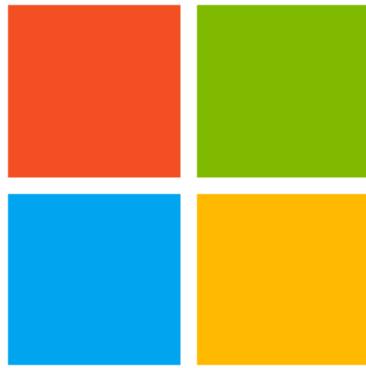
## Ámbito de Variables dentro de Funciones y Sobrecarga de Funciones (Métodos) | Curso C# **30**

Conoce el ámbito (alcance) de variables dentro de funciones y aprende a sobrecargar funciones: comprende el concepto de sobrecarga de métodos en C#.csharp #...

**VER EN YOUTUBE >**

## Forms en C#



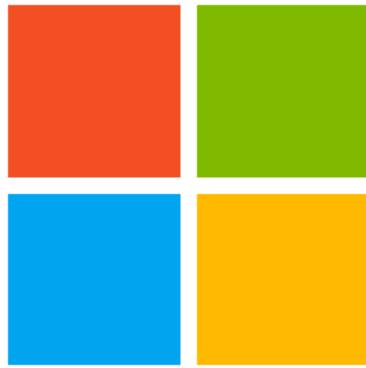


## Create a Windows Forms app with C# - Visual Studio (Windows)

Applies to: Visual Studio Visual Studio for Mac Visual Studio Code In this short introduction to the Visual Studio integrated development environment (IDE), you'll create a simple C# application that has a Windows-based user interface (UI). First, you'll create a C# application project.

[MÁS INFORMACIÓN MICROSOFT >](#)

## Gráficas en C#

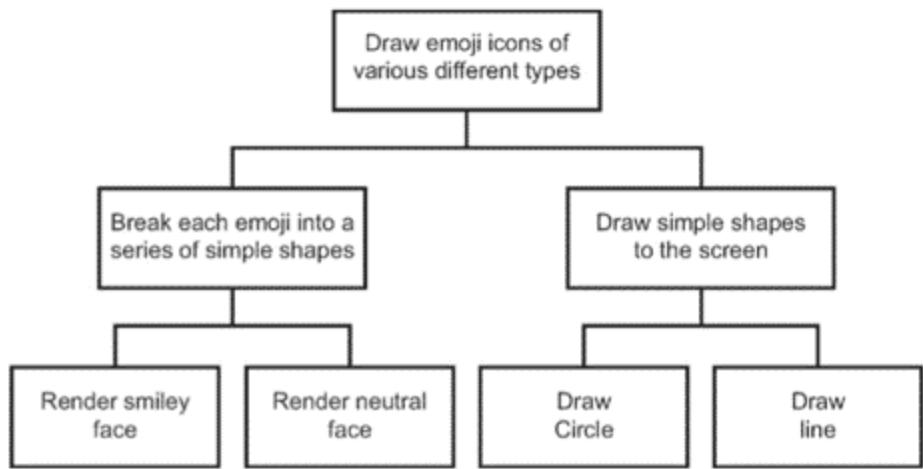


## How to: Create Graphics Objects for Drawing - Windows Forms .NET Framework

Before you can draw lines and shapes, render text, or display and manipulate images with GDI+, you need to create a Graphics object. The Graphics object represents a GDI+ drawing surface, and is the object that is used to create graphical images. There are two steps in working with graphics: Creating a Graphics object.

**[MÁS INFORMACIÓN MICROSOFT >](#)**





Recordando el caso de la carita feliz del módulo 3, la descomposición de funciones para resolver el problema podría verse de la siguiente manera. ¿Cómo quedarían los diferentes métodos en C#? Consulte con su profesor sobre la reutilización de métodos para diferentes programas.



## Actividad 1

---



Actividad 1



## Instrucciones

Marque si los siguientes enunciados son verdaderos o falsos.

Para pasar información entre funciones es mejor utilizar variables globales.

Verdadero



Falso

SUBMIT

Cuando unas funciones conocen mucho de otras funciones, pueden arriesgar su desempeño y tener efectos no deseados.

---



Verdadero



Falso

SUBMIT

Los objetos definidos dentro de una función no pueden ser referenciados fuera de la función.

---

Verdadero

Falso

SUBMIT

Una función puede estar almacenada en un archivo aparte y ser utilizada en varios programas.

Verdadero

Falso

SUBMIT

## Actividad 2

---





## Instrucciones

El siguiente programa está escrito en Python, pero le resultará sencillo comprenderlo. ¿Cuál es la salida de este programa? Considere el alcance y los valores de las variables.

```
def my_function():
    s = 'No, print this!'
    print(s)

s = 'Print this message.'
my_function()
print(s)
```

Diríjase a la plataforma para subir su actividad en el espacio correspondiente.

## Actividad 3

---



Actividad 3



Este programa está escrito en Python, pero le resultará sencillo comprenderlo. Cuando se intenta ejecutar da error. ¿A qué se debe el error?

```
def my_function():
    print(s)
    s = 'No, print this!'

s = 'Print this message.'
my_function()
```

Diríjase a la plataforma para subir su actividad en el espacio correspondiente.

## Actividad 4

---



Actividad 4



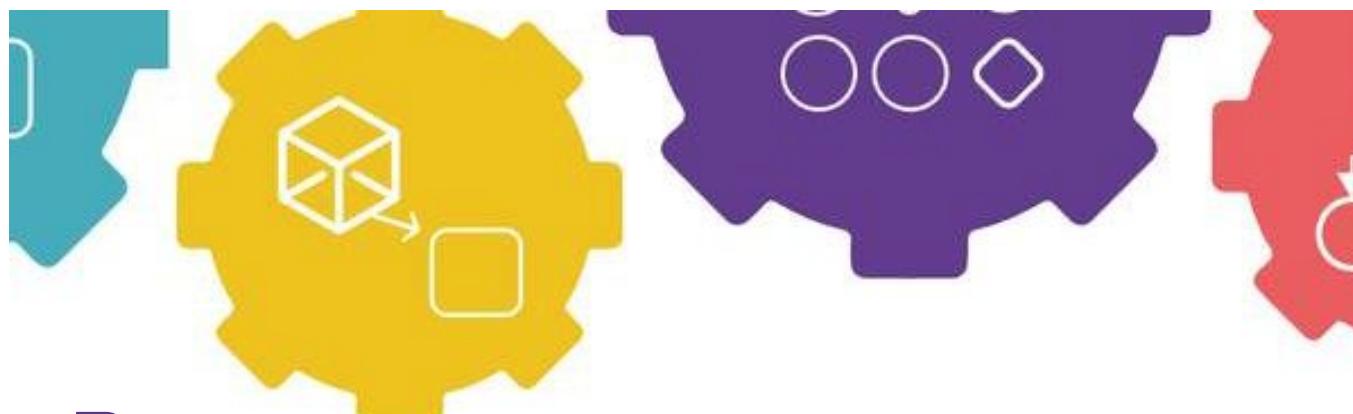
## Instrucciones

Desarrolle un programa que implemente el dibujo de una carita feliz con base a figuras geométricas. Utilice descomposición y reutilice funciones.

Diríjase a la plataforma para subir su actividad en el espacio correspondiente.

## Recursos

---



## Recursos

### Instrucciones:

Haber instalado el lenguaje de programación en las computadoras que utilizará para realizar las prácticas.



## Rúbrica de evaluación

---

Rúbrica





Descargue la siguiente rúbrica de evaluación.



**Rúbrica de Evaluación.pdf**

1635 KB



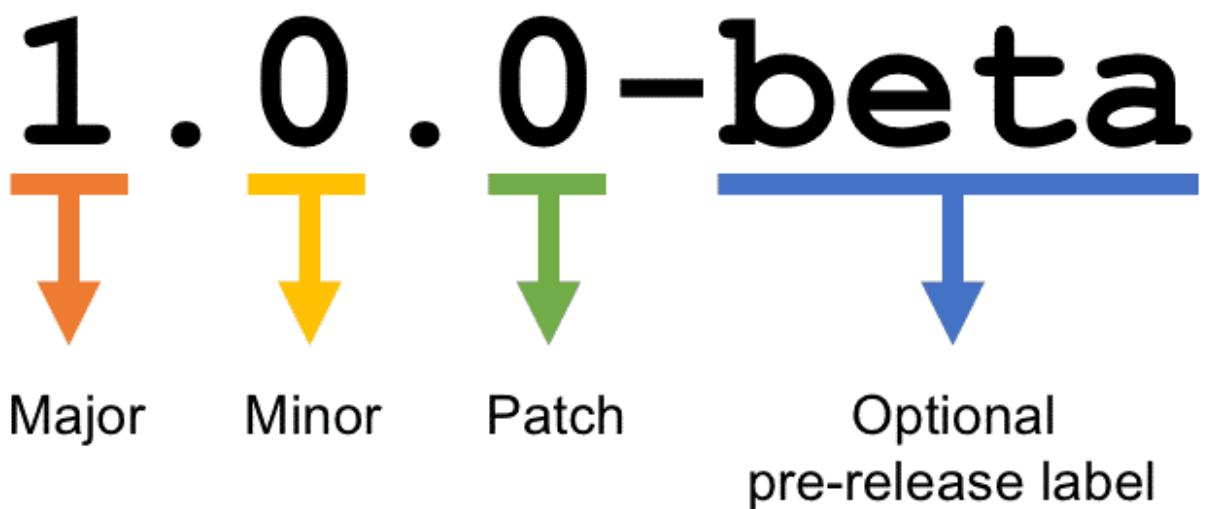
## Diario de experiencias de laboratorio

---



# Diario de experiencias del laboratorio

Cada práctica de laboratorio deberá publicarse en una carpeta en el diario de experiencias de laboratorio, bajo la versión 1.0.0.



# Referencias

---

## Referencias

Bordigón, F. e Iglesias, A. (2020). Introducción al pensamiento computacional. Universidad

Pedagógica Nacional, Educar Sociedad del Estado y el Ministerio de Educación Argentina.

Beecher, K. (2017). Computational Thinking. A beginner's guide to problem-solving and programming.

ISTE (2018). Computational Thinking. Meets Students Learning. International Society for Technology in Education.

Microsoft (2022). Documentación de C#. <https://docs.microsoft.com/es-es/dotnet/csharp/>

MIT (2022). Computational Thinking, a live online Julia/Pluto textbook. Julia: A Fresh Approach to Computing. [computationalthinking.mit.edu](http://computationalthinking.mit.edu). Massachusetts Institute of Technology

Pourbahrami & Tritty (2018). Computational Thinking: How Computer Science Is Revolutionizing Science and Engineering. ENGenious (15) 8-11.

<https://resolver.caltech.edu/CaltechCampusPubs:20181025-110029157>.

## Créditos

---

