

# Programación Básica

Alberto Benavides

Ago - Dic 2018

### **3. Estructura básica de un programa**

## Observaciones clase anterior

```
// Genera enteros al azar entre 100 y 1000 (incluidos)  
rand() % 901 + 100;
```

```
// Genera enteros al azar entre 900 y 1800 (incluidos)  
rand() % 901 + 900;
```

# Crear un proyecto en Visual Studio 2017

1. Abrir VS2017
2. Archivo > Nuevo > Proyecto
3. Elegir Visual C++ (columna izquierda)
4. Elegir Proyecto vacío
5. Especificar nombre y ubicación
6. Aceptar

# Crear archivos dentro de un proyecto de VS2017

1. Abrir Explorador de soluciones
  - Pestaña derecha
  - Ver > Explorador de soluciones
2. Clic derecho en el nombre de la solución (negritas)
3. Agregar > Nuevo elemento...
4. Archivo `.cpp`
5. Especificar nombre
  - Sin espacios
  - Sin caracteres especiales: [áèïñ?~...]
6. Agregar

# Ejemplos minimalistas

```
#include <iostream>
int main(){
    std::cout << "Hola mundo.";
    return 0;
}
```

```
#include <iostream>
using namespace std;

int main(){
    cout << "Hola mundo.";
    return 0;
}
```

```
getchar(); // Pausa la ejecución a la espera de <Enter>
```

- ¿Cuándo va el punto y coma al final?

# Argumentos de `main` \*

```
#include <iostream>
using namespace std;

int main(int argc, char* argv){
    // Los argv son cadenas de texto (char[])
    for (int i = 0; i < argc; i++){
        cout << argv[i] << endl;
    }

    return 0;
}
```

# Ejecutar códigos C++ desde consola (opcional)

- MinGW
  - [Instalación](#)
  - [Uso](#)

```
rem Generar un ejecutable de un código C++  
g++ archivo.cpp -o ejecutable.exe
```

```
rem Correr el ejecutable  
ejecutable  
ejecutable.exe
```

```
rem Correr el ejecutable con argumentos (argvs)  
ejecutable argumento1 argumento2 ...
```




# Ejecutar un programa en VS2017

## 1. Guardar ! ! !

- ¿Ya guardaste?
- Puedes usar `Ctrl + S` para guardar

## 2. ¿Errores? Corrígelos.

## 3. Depurar:

- Depurar > Iniciar depuración
- Clic en 
- Pulsar `F5`

## Entrada y salida de flujos: `iostream`

```
cout << "Muestra en la consola (output)" << endl;

int n; // Se declara un entero llamado n
cin >> n; // Guarda la entrada de consola en n

char s[10]; // Se declara un arreglo de 10 caracteres
cin >> s; // Guarda la entrada de consola en s

// Muestra n y s en consola
cout << n << s << endl;
```

- ¿Qué pasa si en *n* se almacena una letra?
- ¿Qué pasa si en *s* se almacenan más de 10 caracteres?
- ¿Qué pasa si...?

# Ejemplo

Hacer un programa que:

- Pregunta el nombre, la edad de una persona.
- Calcular un número de la suerte aleatorio desde 0 hasta la edad de la persona
- Mostrar en consola

```
Hola [nombre], de [edad] años.  
Tu número de la suerte es [número de la suerte].
```

¿Qué onda con esto? \*

```
echo Alberto 31 | programa
```

## Ubicación del código C++

El `archivo.cpp` del proyecto **Ejemplo** está en:

```
[...]/Ejemplo/Ejemplo/archivo.cpp
```

# ! Tarea !

+2 en primer examen parcial

- Crear un programa en C++ llamado [últimos 4 números de tu matrícula].cpp , por ejemplo 3079.cpp .
- El programa pedirá (**en este orden**): Nombre ( char[20] ), apellido ( char[20] ), edad ( int ) y teléfono( char[20] ).
- Mostrar la información con el siguiente formato:

```
Nombre: [apellido], [nombre].  
---  
Edad: [edad] años.  
Teléfono: +52[teléfono]
```

- Enviar el archivo cpp comprimido en zip o rar al correo [jose.benavidesvz@uanl.edu.mx](mailto:jose.benavidesvz@uanl.edu.mx) con asunto:

*PB Tarea 2 [Matrícula]*

# Fuentes

- <https://www.tutorialspoint.com/cplusplus/index.htm>
- <http://www.cplusplus.com/doc/tutorial/>