# Crear una máquina virtual en VIRTUAL BOX

RED – NAT

Instalar Windows XP

Crear una carpeta compartida entornos_desarrollo para tener acceso a la información del anfitrión

# GCC compilador y enlazador LINEA COMANDOS

Se instala en la carpeta minGw. El ejecutable está en c:\mingw\bin

1º) edit programa.c
dir programa.*
type programa.c → código fuente (programa.c)

**compilar**
gcc –c programa.c

2º) dir programa.*
uno (Enter)
type programa.o → código objeto (programa.o)

**Sólo enlazar**
gcc programa.o -o programa

3º) dir programa.*
type programa.exe → código ejecutable (programa.exe)
**uno (Enter)**

**Para compilar y enlazar**
gcc programa.c –o programa

4º) mirar carpeta c:\mingw\bin en entorno windows
¡diferentes iconos! ¡ejecutar!

---

uno.c

```
#include <stdio.h>
int main ()
{
printf (“Hola mundo”);
system (“pause”);
return 0;
}
```

---

dos.c

```
#include <stdio.h>
#include <conio.h>
int main ()
{
int metros, millas;
printf (“por favor introduzca distancia en metros: “);
scanf (“%i”, &metros);
millas = metros * 1850;
printf (“\n La distancia en millas es: %i”, millas);
```

}

## Python intérprete entorno línea de comandos

Se instala en c:\python27

uno.py

```
print “Hola mundo”
```

dos.py

```
def poncoma( n ):
   "Regresa n como cadena con comas."
   s = str(n)
   pos = len(s)
   while pos > 3:
      pos = pos - 3
      s = s[:pos] +  ',' + s[pos:]
   return s

for n in range(60):
   print '%3d %30s' % ( n, poncoma( 2**n ) )
```

para ejecutar/intérpretar

```
uno.py
dos.py
```

## Python intérprete GUI

```
>>print “hola”
```

```
>>for n in range(60): print n, 2**n
```

## Java lenguaje virtual línea de comandos

- instalar el JDK (J2SE) distribuido por SUN Microsystems para disponer de:
  - el compilador (javac)
  - JVM (java) están accesibles

En el directorio C:\Archivos de programa\Java\jdk1.8.XX tenemos recién instalado la máquina virtual java. Por supuesto, los numeritos del final pueden cambiar si tenemos otra versión.

Con un editor de textos: HolaMundo.java

```
class HolaMundo {
    public static void main ( String [] args) {
        System.out.println("Hola a todos");
    }
}
```

**compilar**

```
javac HolaMundo.java → java.class (compilado)
```

**interpretar y ejecutar**

```
Java HolaMundo
```

## Instalar DEV C++

Se instala en c:\Dev-Cpp

## ENTORNO GRÁFICO con GUI

## Ejemplo de pasos programación

## c

## ejemplo paradigma procedimental

ortoedro.cpp

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

int main ()
{
   int base, altura, profundo;
   printf ("\n Por favor, introduzca la base de ortoedro: ");
   scanf ("%i", &base);
   printf ("\n Por favor introduzca la altura del ortoedro: ");
   scanf ("%i", &altura);
   printf ("\n Por favor introduzca la altura del ortoedro: ");
   scanf ("%i", &altura);
    printf ("\n\n El volumen del ortoedro es: %i", (base*altura*profundo);
   printf ("\n");
   system ("pause");
   return (0);
}
```

## C++

## Paradigma orientado a objetos

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

class orto
{
    int alt, anc, pro;
public:
     orto (int h, int a, int p)
     {    alt=h;
          anc =a;
          pro =p;}

     int volumen (void)
     { return (alt*anc*pro);}


     int altura (void)
     {cout << "Altura? ";
     cin>>alt;
     return (alt);}


     int ancho (void)
     {cout << "Ancho? ";
     cin >> anc;
     return (anc);}


     int profundo (void)
     {cout<< "Profundidad? ";
     cin>> pro;
     return (pro);}
};

int main()
{
   int a, b, c;
   cout << "VOLUMEN DEL ORTOEDRO::::: (introducir datos) \n";
   orto dos(a,b,c);
   a = dos.altura();
   b = dos.ancho();
   c = dos.profundo();
   cout<<"\n El volumen es: " << dos.volumen() << "\n\n";
   system ("pause");
   return 0;
}
```

**Modificar los programas para que obtengan también el área (2*ancho*profundo+2*prof*alto+2*alto*ancho )**