

# **Introducción a la Estructura de los Computadores**

HISTORIAL DE REVISIONES			
NÚMERO	FECHA	MODIFICACIONES	NOMBRE
0.6.0	2021 September 3		C

## Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Arquitectura de una máquina</b>	<b>1</b>
<b>3. Perspectiva HW</b>	<b>1</b>
<b>4. Lenguajes de Programación: Niveles</b>	<b>3</b>
<b>5. Interface Software/Hardware</b>	<b>5</b>
<b>6. Temario</b>	<b>6</b>
6.1. Web Estructura de Ordenadores . . . . .	6
6.2. Bibliografía Basica . . . . .	7
6.3. Bibliografía Complementaria . . . . .	8
<b>7. Profesorado</b>	<b>8</b>
<b>8. Grado Informática</b>	<b>8</b>
<b>9. Calendario</b>	<b>9</b>
<b>10. Grupos de Prácticas</b>	<b>11</b>
<b>11. Metodología</b>	<b>11</b>
11.1. Distribución de créditos . . . . .	13
11.2. Distribución de créditos de las Prácticas . . . . .	13
<b>12. Evaluación</b>	<b>13</b>
<b>13. Ejercicios mediante resolución de problemas</b>	<b>14</b>
<b>14. Prácticas</b>	<b>15</b>
14.1. Memorias . . . . .	15
14.2. Estaciones de Trabajo: 32 y 64 bits . . . . .	15
<b>15. Proyectos</b>	<b>15</b>
<b>16. Recursos Informáticos</b>	<b>16</b>
16.1. UPNA . . . . .	16
16.2. Programación . . . . .	16

---

## 1. Introducción

- El objetivo de la asignatura Estructura de Computadores (240306) del Grado de **Ingeniería Informática** de la Universidad Pública de Navarra es ser un curso introductorio universitario a la arquitectura de los computadores, estudiando sus componentes básicos (procesador, memoria y módulo de entrada/salida) así como la programación de bajo nivel en lenguaje ensamblador x86 mediante la utilización de herramientas de desarrollo software como el compilador, depurador, etc.

## 2. Arquitectura de una máquina

- Arquitectura: Organización, Estructura: Qué, Cómo, Implementación (tecnología)

`./images/introduccion/vehicle_arch.jpg`

Figura 1: Estructura del Automóvil

`./images/introduccion/Personal_computer,_exploded_6.png`

Figura 2: Personal Computer

`./images/introduccion/motherboard_parts.jpg`

Figura 3: Placa Base

## 3. Perspectiva HW

- 4 Módulos Básicos: CPU-MEMORIA-CONTROLADORES Entrada/Salida [Periféricos]-BUSES

`./images/introduccion/cpui7.jpeg`

Figura 4: CPU Intel Core i7 4ª Generación

`./images/introduccion/RAMtypes2.jpg`

Figura 5: Memoria DRAM

`./images/introduccion/Seagate.jpg`

Figura 6: Periféricos: Disco Duro Seagate

`./images/introduccion/BusOnMotherboard.gif`

Figura 7: Bus de la Placa Base

---

`./images/introduccion/busribbon.jpeg`

Figura 8: Bus Cableado

## 4. Lenguajes de Programación: Niveles

[./images/introduccion/lenguajemaquina.gif](#)

Figura 9: Lenguaje Máquina

### ■ Pascal

```
program Hello_world;
begin
    writeln('hello world')
end.
```

### ■ L. Máquina-Binario Intel x86

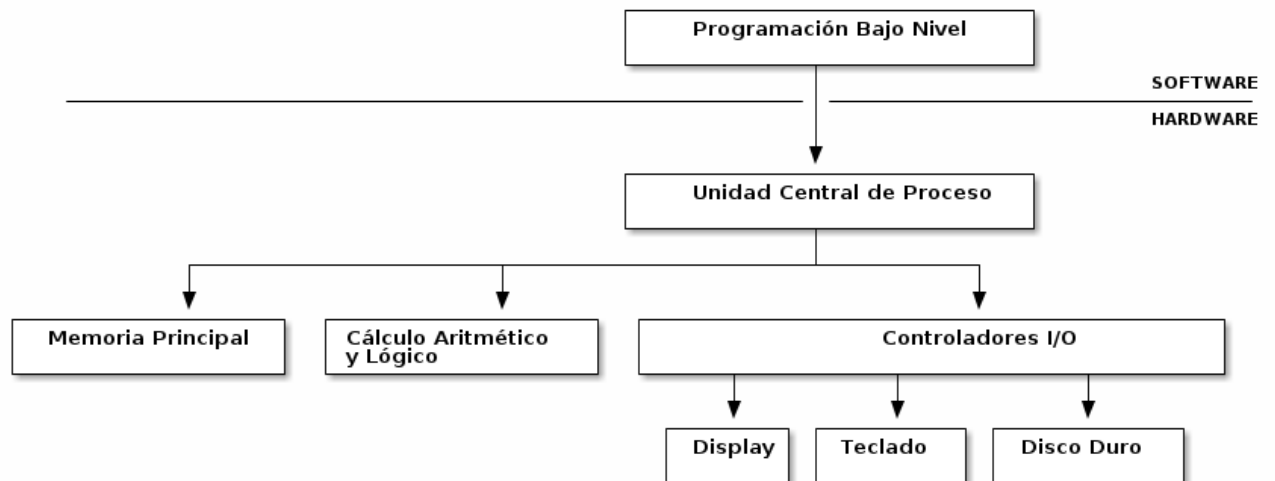
```
# [2] begin
00000000004001a0 <PASCALMAIN>:
 4001a0:      01010101
 4001a1:      01001000 10001001 11100101
 4001a4:      01001000 10000011 11101100 00010000
 4001a8:      01001000 10001001 01011101 11111000
 4001ac:      11101000 10110111 00111110 00000001 00000000
# [3] writeln('hello world')
 4001b1:      11101000 11001010 10010011 00000001 00000000
 4001b6:      01001000 10001001 10100011
 4001b9:      01001000 10001001 11011110
 4001bc:      01001000 10111010 11000000 11110110 01100001 00000000 00000000
 4001c3:      00000000 00000000 00000000
 4001c6:      10111111 00000000 00000000 00000000 00000000
 4001cb:      11101000 01111000 10010110 00000001 00000000
 4001d0:      11101000 00010011 00111101 00000001 00000000
 4001d5:      01001000 10001001 11011111
 4001d8:      11101000 01110011 10010101 00000001 00000000
 4001dd:      11101000 00000110 00111101 00000001 00000000
 4001e2:      11101000 10011001 01000010 00000001 00000000
 4001e7:      01001000 10001011 01011101 11111000
 4001eb:      10101001
 4001ec:      10010011
```

## ■ L. Máquina-Hexadecimal vs Lenguaje Ensamblador ASM x86

```
# [2] begin
00000000004001a0 <PASCALMAIN>:
 4001a0:    55                push    %rbp
 4001a1:    48 89 e5          mov     %rsp,%rbp
 4001a4:    48 83 ec 10       sub     $0x10,%rsp
 4001a8:    48 89 5d f8       mov     %rbx,-0x8(%rbp)
 4001ac:    e8 b7 3e 01 00    callq  414068 <FPC_INITIALIZEUNITS>
# [3] writeln('hello world')
 4001b1:    e8 ca 93 01 00    callq  419580 <fpc_get_output>
 4001b6:    48 89 c3          mov     %rax,%rbx
 4001b9:    48 89 de          mov     %rbx,%rsi
 4001bc:    48 ba c0 f6 61 00 movabs  $0x61f6c0,%rdx
 4001c3:    00 00 00
 4001c6:    bf 00 00 00 00    mov     $0x0,%edi
 4001cb:    e8 78 96 01 00    callq  419848 <FPC_WRITE_TEXT_SHORTSTR>
 4001d0:    e8 13 3d 01 00    callq  413ee8 <FPC_IOCHECK>
 4001d5:    48 89 df          mov     %rbx,%rdi
 4001d8:    e8 73 95 01 00    callq  419750 <fpc_writeln_end>
 4001dd:    e8 06 3d 01 00    callq  413ee8 <FPC_IOCHECK>
 4001e2:    e8 99 42 01 00    callq  414480 <FPC_DO_EXIT>
 4001e7:    48 8b 5d f8       mov     -0x8(%rbp),%rbx
 4001eb:    c9               leaveq  %rbx
 4001ec:    c3               retq
```

## 5. Interface Software/Hardware

- Interacción del Programador con la Máquina: Programación de bajo nivel. Integración en el Kernel del Sistema Operativo.



- Programación de bajo nivel
  - Módulo fuente: Lenguaje C ó Lenguaje ensamblador
  - Módulo objeto: Lenguaje máquina → Compilación, Ensamblaje y Enlazado del Módulo Fuente.
- Dos tipos de Programación de bajo nivel en lenguaje máquina
  - Bare metal → Programación de sistemas empotrados → Raspberry Pi (Ej: control de una video cámara)
  - Sistema Operativo → Programación de Sistemas (Programación de Drivers, etc)



## 6. Temario

### 6.1. Web Estructura de Ordenadores

#### Temario

- 1 - Introducción
- 2 - Arquitectura de Von Neumann
  - 2.1 CPU
  - 2.2 Memoria
  - 2.3 Entrada / Salida
- 3 - Representación de datos
  - 3.1 Bit, Byte y Palabra
  - 3.2 Caracteres, enteros y reales
- 4 - Aritmética y lógica
  - 4.1 Operaciones aritméticas y lógicas sobre enteros en binario
  - 4.2 Redondeo y propagación de error en números reales
- 5 - Representación de instrucciones
  - 5.1 Lenguaje máquina, lenguaje ensamblador y lenguajes de alto nivel
  - 5.2 Formato de instrucción
  - 5.3 Tipos de instrucción y modos de direccionamiento
- 6 - Programación en lenguaje ensamblador de construcciones básicas de los lenguajes de alto nivel ←
  - 6.1 Sentencias de asignación
  - 6.2 Sentencias condicionales
  - 6.3 Bucles
  - 6.4 Llamadas y retorno de función o subrutina
- 7 - Arquitectura y organización de la CPU
  - 7.1 Conjunto de instrucciones
  - 7.2 Arquitecturas CISC, RISC y VLIW
  - 7.3 Fases de ejecución de una instrucción
  - 7.4 Camino de datos
- 8 - Sistema de entrada / salida
  - 8.1 Sincronización por encuesta
  - 8.2 Sincronización por interrupción
  - 8.3 Vector de interrupciones
  - 8.4 Acceso directo a memoria DMA
  - 8.5 Programación en lenguaje ensamblador de rutinas de entrada/salida
- 9 - Organización de la memoria
  - 9.1 Jerarquía de memoria
  - 9.2 Latencia y ancho de banda
  - 9.3 Memoria cache
  - 9.4 Memoria virtual

## 6.2. Bibliografía Basica

- **book info:** por autores → william stalling, etc

`./images/introduccion/william_stalling7_portada.jpg`

Figura 10: William Stalling Book

Organización y arquitectura de computadores .William Stallings  
Edición 7, reimpresa Pearson Prentice Hall  
ISBN 8489660824, 9788489660823 . 2006

Computer Organization and Architecture: Designing for Performance.  
William Stallings  
9ª Ed Upper Saddle River (NJ) : Prentice Hall, [2013]  
ISBN 0-273-76919-7 . 2012

- **COA 7ª**
- **Pearson**
- **11º Ed 2019**

`./images/introduccion/programming_from_the_ground_up.jpg`

Figura 11: Programming Assembly

Programming from the Ground Up  
by Jonathan Bartlett Edited by Dominick Bruno, Jr.  
Copyright © 2003 by Jonathan Bartlett  
ISBN 0-9752838-4-7  
Published by Bartlett Publishing in Broken Arrow, Oklahom

- Programming from the Ground Up by Jonathan Bartlett. Programación en Lenguaje Ensamblador AT&T para la arquitectura x86.
  - **PGU book online**
  - **PGU book home**
    - **pdf**
- Documentación de los Manuales: La mayoría de las aplicaciones y herramientas software de la fundación GNU y Linux disponen de Manuales bien on-line o bien localmente en la propia máquina. Manual del compilador gcc: `$man gcc`



### **importante**

Es una habilidad a adquirir por el profesional informático el acceder y utilizar dichos manuales así como disponer de hojas de referencia de acceso rápido durante la sesiones de trabajo.

---

### 6.3. Bibliografía Complementaria

- David A. Patterson, John L. Hennessy. Computer Organization and Design. The Hardware / Software Interface. RISC-V Edition. Morgan Kaufmann Series in Computer Architecture. 2017.
  - Libro Standard de la mayoría de las Universidades.
- Randal E. Bryant, David R. O'Hallaron. Computer Systems: A Programmer's Perspective. Addison-Wesley. 2nd Edition. 2010.
- Paul Carter. PC Assembly Language. Acceso libre. 2006.
- K.N.King. C programming, a Modern Approach W.W. Norton 2ªEd. 2008.
- Jeff Duntemann. Assembly Language Step-by-Step: Programming with Linux. Wiley Ed. 3rd Edition. 2009.
- Kip R. Irvine. Assembly Language for x86 Processors. Pearson. 6th Edition. 2014.

## 7. Profesorado

- Cándido Aramburu Mayoz.
  - Doctor Ingeniero Telecomunicación. Profesor Titular de Universidad.
  - Edificio los Tejos, planta 2ª. Despacho 2028.
  - Tutorías semestre Otoño : Viernes de 8:00 a 14:00
  - correo electrónico interno: a través del servidor Miaulario.
  - **PDI Profesorado**
- Andrés Garde Gurpegi
  - Técnico Superior de la Dirección General de Informática y Telecomunicaciones del Gobierno de Navarra.
  - Profesor Asociado (Prácticas de Laboratorio)
  - Edificio de Los Tejos, planta 2ª, Sala de Asociados del departamento de INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE COMUNICACIÓN.
  - andres.gardeATunavarra.es
  - **PDI Profesorado**

## 8. Grado Informática

- **Grado Informatica**
  - 200 - Escuela Técnica Superior de Ingeniería Industrial, Informática y de Telecomunicación
  - 240 - Graduado o Graduada en Ingeniería Informática por la Universidad Pública de Navarra

## 9. Calendario

### ■ Académico

### ■ [calendario-aulas-semanal](https://academicos.unavarra.es/pds/consultaPublica/look%5Bconpub%5DInicioPubHora?entradaPublica=true&idiomaPais=es.ES)

<https://academicos.unavarra.es/pds/consultaPublica/look%5Bconpub%5DInicioPubHora?entradaPublica=true&idiomaPais=es.ES> ↩

- Teoría: Grupo1 GG Viernes 15:00 A318 ; Grupo2 GG Viernes 17:00 A207 ; Grupo91 GG Lunes 19:00 A135

### ■ [Calendario Prácticas Serv Info](#)

### ■ <https://www.unavarra.es/sites/estudios/calendarios.html>

- Teoría

	Lunes	Martes	Miércoles	Jueves	Viernes
15:00					T1-A318
17:00					T2-A207
19:00	T91-A135	P91			

### ■ Prácticas

A310 P3	A310 P2	A306 P91	P-ARQ P1
15:00-17:00	17:00-19:00	19:00-21:00	19:00-21:00
30/9	30/9		
14/10	14/10	5/10	5/10
28/10	28/10	19/10	19/10
		2/11	2/11
11/11	11/11	16/11	16/11
25/11	25/11	30/11	30/11

- P3: Jueves a las 15:00 en A310 → Grupos de teoría G1 y G2
- P2: Jueves a las 17:00 en A310 → Grupo de teoría G2
- P1: Martes a las 19:00 en P-ARQ → Grupo de teoría G1
- P91: Martes a las 19:00 → Grupo de teoría Euskera
- Prof Andrés Garde: 3 Grupos de Castellano P1,P2,P3
- Prof Cándido Aramburu: Grupo Euskera: Aula A306
- Aula P-ARQ: Aula Arquitectura, segunda planta del edificio Pinos. Aula con 30 puestos y sistema de video proyección. Software igual que el resto del aulario.
- Semanas del curso :
  - 15 semanas: Duración del curso 2 de Septiembre al 17 de Diciembre
  - Período de Evaluación:
    - ordinaria: del 17 al 22 de diciembre y del 7 al 12 de enero 2022
    - recuperación: del 17 de enero al 25 de enero 2022
- Fiestas

12 de octubre (Fiesta Nacional de España).  
01 de noviembre (Festividad de Todos los Santos).  
03 de diciembre (San Francisco Javier, Día de Navarra).  
06 de diciembre (Día de la Constitución).  
08 de diciembre (Inmaculada Concepción).

## 10. Grupos de Prácticas

### ■ P1,P2,P3

P1	P2	P3
Aguirre Eguizabal, Endika Manuel	Espinoza Zarate, Danny Alexander	Da Cunha Moreira, Diogo ↔
Aldanondo Jimenez, Roberto	Ezziani , Yasin	de la Nava Ortego, Asier
Aldaz Burgoa, Miguel Pablo	Flamarique Arellano, Aritz	Dealbert Gonçalves, ↔
Aller Errea, Yeray Alba	Gallo Ansa, Borja	Diez de Rada Beltran, ↔
Alvarez Garcia, David	Garde Gurpegui, Andrés	Hijarrubia Bezares, Odei
Andreu Mangado, Alvaro	Gomez Jauregui, Samuel	Lopez Goñi, Juan
Aranguren Ortiz, Javier	Itoiz Garcia, Raúl	Muñoz Lahoz, David
Arriazu Muñoz, Jon	Jimenez Navarcorena, Oscar	Peñas Escribano, Lucas
Arribillaga Ciriza, Ekaitz Mikelats	Kamp Ortega, Axel	Perez de Iriarte Les, ↔
Artazcoz Velaz, Javier	Larralde Baldanta, Ibai	Resano Aisa, Jaime
Arteaga Mayor, Diego Adrián	Leon Iso, Oriol	Reviriego Reinaldo, ↔
Ayala Lauroba, Dennis	Lopez de Vicente, Miguel	Salvoch Vilches, Mikel
Azcoiti Peña, Luis	Mancho Gochi, Daniel	Sanchez Monreal, Julen
Azcona Recari, Guillermo Felipe	Marin Munarriz, Adriana	Santos Garzon, Jaider ↔
Barandiaran Lasheras, Nerea Alexánder	Monclus Radigales, Pablo	Sarango Quizhpe, ↔
Barriuso Cervera, Pablo	Morala Martinez, Andrea	Sola Delgado, Iván
Basterra Arteaga, Paula	Morales Zufia, Martín	Sotomayor Ramos, Eliezer
Bastida Vidaurre, Andrés	Muro Marco, Javier	Tarragona Rebollo, Odei
Blasco Di Rosa, Sol	Navascues Martinez, Ivan	Uranga Roldan, Aitor
Bodziony , Karolina Maria	Pernaut Murolas, Javier	Urdirroz Arroyo, Iker
Bouzarhoun el Asry, Wassim Sergio	Rodriguez Meneses, Julen	Valencia Aldunate, ↔
Cardenas Curicho, Solange Samanta	Ruiz Cabodevilla, Adrian	Varela Martinez, Iñigo
Castilla Jaurrieta, Pedro	Ruiz Oyarzun, Carlos	Verde Perez, Pablo
Ciordia Barcos, Antonino Sergio	Sagaseta de Ilurdoz Sanchez, Miguel	Vergara Dominguez, ↔
Cojocarú , Mihail Volodimir	Sanchez Albardiaz, Juan	Vons Vons, Igor ↔
Couceiro Eizaguirre, Javier Iker	Zubiri Remirez, Fermín	Zabalegui Murugarren, ↔
Razvant Sofrone, Dragos		

## 11. Metodología

### ■ Teoría, Ejercicios, Prácticas y Exámenes.

Cuadro 1: metodología

Metodología - Actividad	Horas Pre-senciales	Horas no presenciales
A-1 Clases magistrales	24	
A-2 Estudio autónomo		30
A-3 Sesiones prácticas	16	

Cuadro 1: (continued)

<b>Metodología - Actividad</b>	<b>Horas Pre-senciales</b>	<b>Horas no presenciales</b>
<i>A-4 Programación / experimentación u otros trabajos en ordenador / laboratorio</i>		<i>20</i>
<i>A-5 Resolución de problemas, ejercicios y otras actividades de aplicación</i>		<i>12</i>
<i>A-6 Aprendizaje basado en problemas y/o casos</i>	<i>14</i>	
<i>A-7 Elaboración de trabajos y/o proyectos y escritura de memorias</i>		<i>11</i>
<i>A-8 Lectura de Guiones, preparación de presentaciones de trabajos, proyectos, etc. . .</i>		<i>15</i>
<i>A-9 Actividades de Evaluación</i>		<i>6</i>
<i>A-10 Tutorías</i>	<i>2</i>	
<i>Total</i>	<i>62</i>	<i>88</i>

- La asignatura se desdobla en 2 partes
  - Parte I: Temas 1-9 (conceptos básicos)
  - Parte II: Temas 1-9 (conceptos avanzados y casos prácticos)

### 11.1. Distribución de créditos

- Distribución de créditos:
  - total créditos: 6 ECTS
  - total horas: 6x25 : 150 horas
  - horas presenciales: 62 horas
    - clases : 24 horas. Durante 12 semanas (2horas/semana) de 15 semanas totales del curso
    - prácticas de laboratorio: 16 horas. Durante 8 semanas (2horas/semana)
    - problemas : 14 horas. Durante 7 semanas (2horas/semana)
    - evaluación: 6 horas. Durante 3 semanas (2horas/semana) de 15 semanas totales del curso
  - tutorías : 2 horas
  - horas no presenciales: 88 horas

### 11.2. Distribución de créditos de las Prácticas

- Cada alumno tendrá 16 horas de prácticas en sesiones de 2 horas: 12 en el laboratorio y 4 para la realización de Memorias.

## 12. Evaluación

- Ver ficha de la asignatura
  - prácticas: parcial después de las dos primeras sesiones y final de las 3 sesiones restantes. Nota media mínima de 4.
  - teoría: parcial cada 2 temas → 4 pruebas parciales. Nota media mínima de 4
  - 10 %(asistencia y actitud en clase y laboratorio)+30 %(prácticas y trabajos)+45 %(conceptos y ejercicios)+15 %(programación)

**atención**

OBLIGATORIEDAD DE LAS PRACTICAS: hay que asistir al 14/16 horas de las prácticas y entregar el 100 % de las memorias dentro del plazo establecido en miaulario.

---

**atención**

OBLIGATORIEDAD DE LAS PRUEBAS TEORICAS: Para la realización de la media es necesario haber realizado TODAS las pruebas teóricas: las de evaluación continua y las del examen final.

---

- Asistencia a las prácticas en el laboratorio: Es **obligatorio** asistir al 87.5 % de las horas de prácticas en el laboratorio.
  - Entrega de las memorias de prácticas: Es **obligatorio** entregar el 100 % de las memorias dentro del plazo establecido en la fecha habilitada en el servidor de miaulario. No se reciben memorias de prácticas ni de tareas fuera del plazo fijado por el servidor de miaulario.
  - La obligatoriedad de la asistencia al 87.5 % de las horas de prácticas así como la entrega de memorias en el plazo y medio establecido es condición necesaria para poder superar la asignatura.
-



### **13. Ejercicios mediante resolución de problemas**

- Realización de ejercicios básicos a lo largo del capítulo y ejercicios del fin de cada capítulo en clase y fuera de clase.
- Ejercicios tipo examen.

## 14. Prácticas

### 14.1. Memorias

- Guiones :aulario virtual
- Prácticas Individuales.
- Memorias :
  - La entrega de la memoria a través del servidor de miaulario se realizará antes de la siguiente sesión de prácticas en la fecha indicada y publicada por el profesor.
  - La memoria es un documento único en formato PDF.
  - El nombre del fichero debe ser apellido1\_apellido2\_tituloquionpractica.pdf
  - Contenido de la memoria.
    - El programa descrito en pseudocódigo
    - El código fuente en lenguaje ensamblador debidamente comentados en coherencia con el programa en pseudocódigo.
  - No se valora el estilo de la memoria sino su contenido ya que interesa que sirva como apuntes para las pruebas de evaluación.

---

#### nota

- Imprescindible tomar notas dentro y fuera del laboratorio
  - Salvar todo el trabajo en un pendrive o enviarlo por correo
- 



#### atención

- Borrado automático a diario del contenido del disco duro.
- 

### 14.2. Estaciones de Trabajo: 32 y 64 bits

- Es necesario disponer de una plataforma linux de 64 bits con el sistema operativo instalado de forma nativa o virtual (VMware, Virtualbox, etc). Por ejemplo en el laboratorio está instalada la distribución Ubuntu.
- Si la estación de trabajo particular de un alumno es Ubuntu (64 bits) sobre un procesador x86-64 no hay ningún problema para la ejecución de programas con una arquitectura de x86-64, en cambio, los programas a desarrollar en la asignatura utilizan una arquitectura de 32 bits que para poder compilar y ejecutar dichos programas es necesario
  - a. Instalar las librerías **gcc-multilib** que necesita el compilador y el linker para que el procesador de 64 bits ejecute programas de 32 bits.

```
instalación en Ubuntu -> sudo apt-get install gcc-multilib
comprobación -> dpkg -l gcc-multilib
```

## 15. Proyectos

- Los proyectos deben ser propuestos por los alumnos durante la primera parte de la asignatura para su aprobación.
  - Parte II de la asignatura: Optativo
  - Utilización de la tarjeta Raspberry Pi (linux/ARM) o de la tarjeta DE1-SoC (linux/ARM) del laboratorio remoto
-

## 16. Recursos Informáticos

### 16.1. UPNA

- Servicio Informático UPNA
  - Aulas VDI
  - Escritorio Virtual
    - Sistema Operativo linux/gnu. Distribución Ubuntu.
    - Herramientas binutils de bajo nivel: Compilador gcc, Assembler as, Debugger gdb, Editor vim
    - Herramientas de diseño dispositivos digitales: quartus
- Laboratorio Remoto de Computación
  - <https://laboratoriosvirtuales.unavarra.es/> → ¿?
  - <https://laboratoriosvirtuales.unavarra.es/fpga/#/> → laboratorio remoto FPGA-Raspberry
    - Guacamole login
    - Acceso a la tarjeta DE1-SoC (ARM) → login → Consola shell linux
    - Acceso a la tarjeta Raspberry (ARM) → login → Consola shell linux
  - Empresa Colaboradora Labsland: Instancias DE1-SoC. Interfaz Web.
    - Se instalará su acceso a través de miaulario mediante el botón EDA.
    - Herramientas de diseño dispositivos digitales

### 16.2. Programación

- [ias assembler univcamp online](#)
- [gdb online](#)