

Proyecto Final

Entrega: **8 de Enero de 2021** (fecha examen)

- ✓ Todo el material a entregar debe estar en GitHub
- ✓ Presentación y demostración presencial

Próxima semana: colgare en ALUD requerimientos y evaluación

Agenda


Hoy

1. Repaso breve semana 1
2. Continuación de Linux y aclaración ejercicios para hoy
3. Actividad de clase

Mañana

1. Corrección ejercicios
2. Introducción a **Python**
3. Actividad de clase con Python

Repaso Semana 1

SISTEMAS EMBEBIDOS 
(EMBEDED SYSTEMS)



Deusto

Facultad de Ingeniería
Ingeniaritza Fakultatea

Grado Dual en Industria Digital

Campus Vitoria

Curso 2020-2021

Que sabemos hasta ahora?

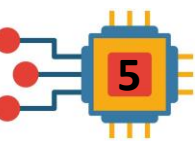
- **Que es y para que sirve un sistema embebido?**

DEFINICION

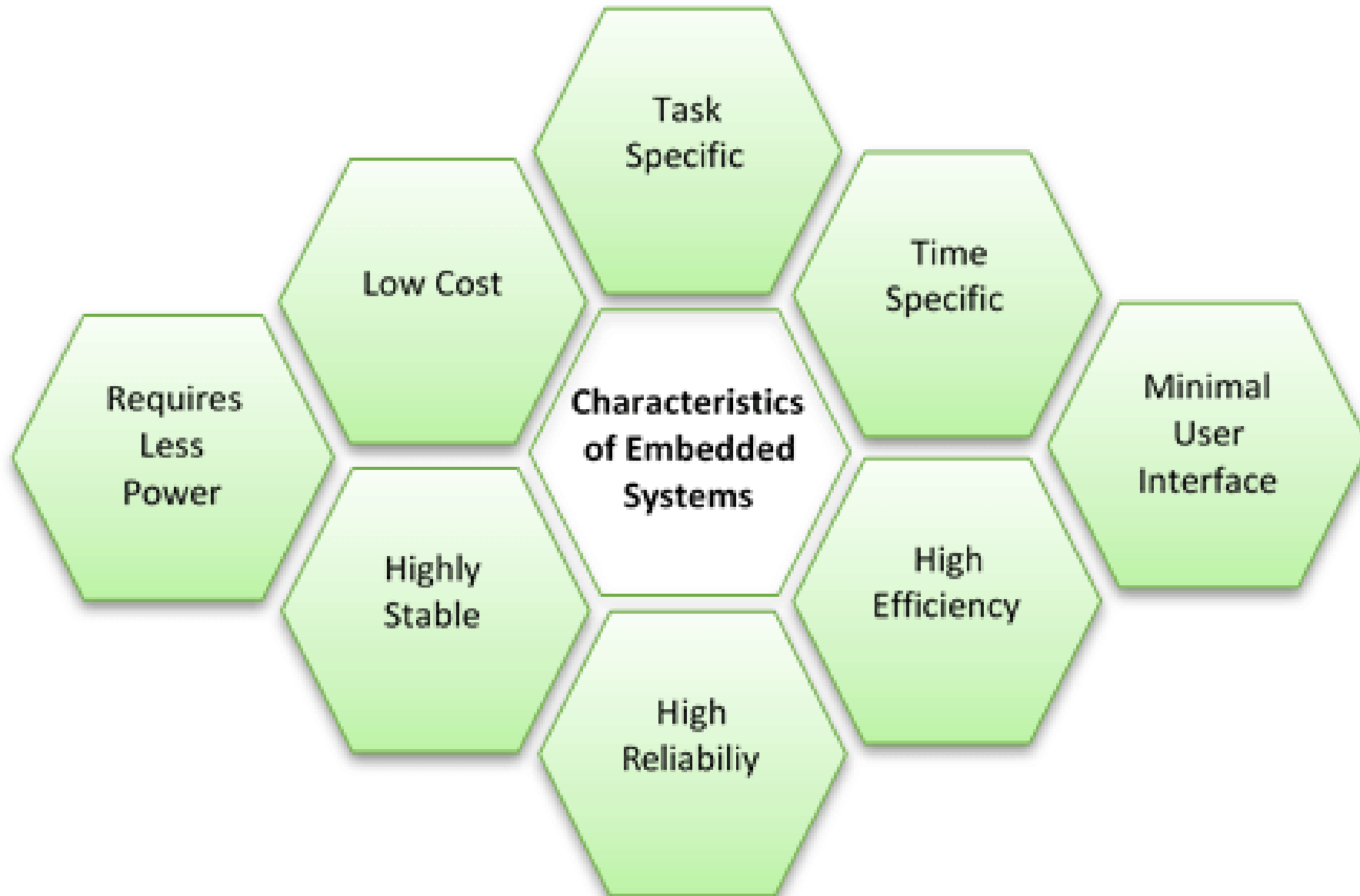
Los sistemas embebidos son sistemas electrónicos con una funcionalidad muy particular/concreta, y no de propósito general.

Estos sistemas tipicamente requieren un riguroso rendimiento, potencia, coste, espacio, y restricciones de tiempo real.

Los sistemas embebidos estan dominando, cada vez más, casi todos los aspectos de nuestras vidas.



Sistema Embebido



Que sabemos hasta ahora?

- Que es y para que sirve un sistema embebido?
- **Que aplicaciones (+industriales) tiene?**

Ejemplo Escenario A

Tema 4

Tema 3b

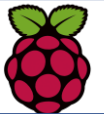
Protocolo de
comunicación Industrial



Servicio Web



Sistema
Embebido



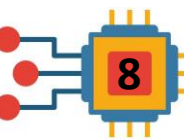
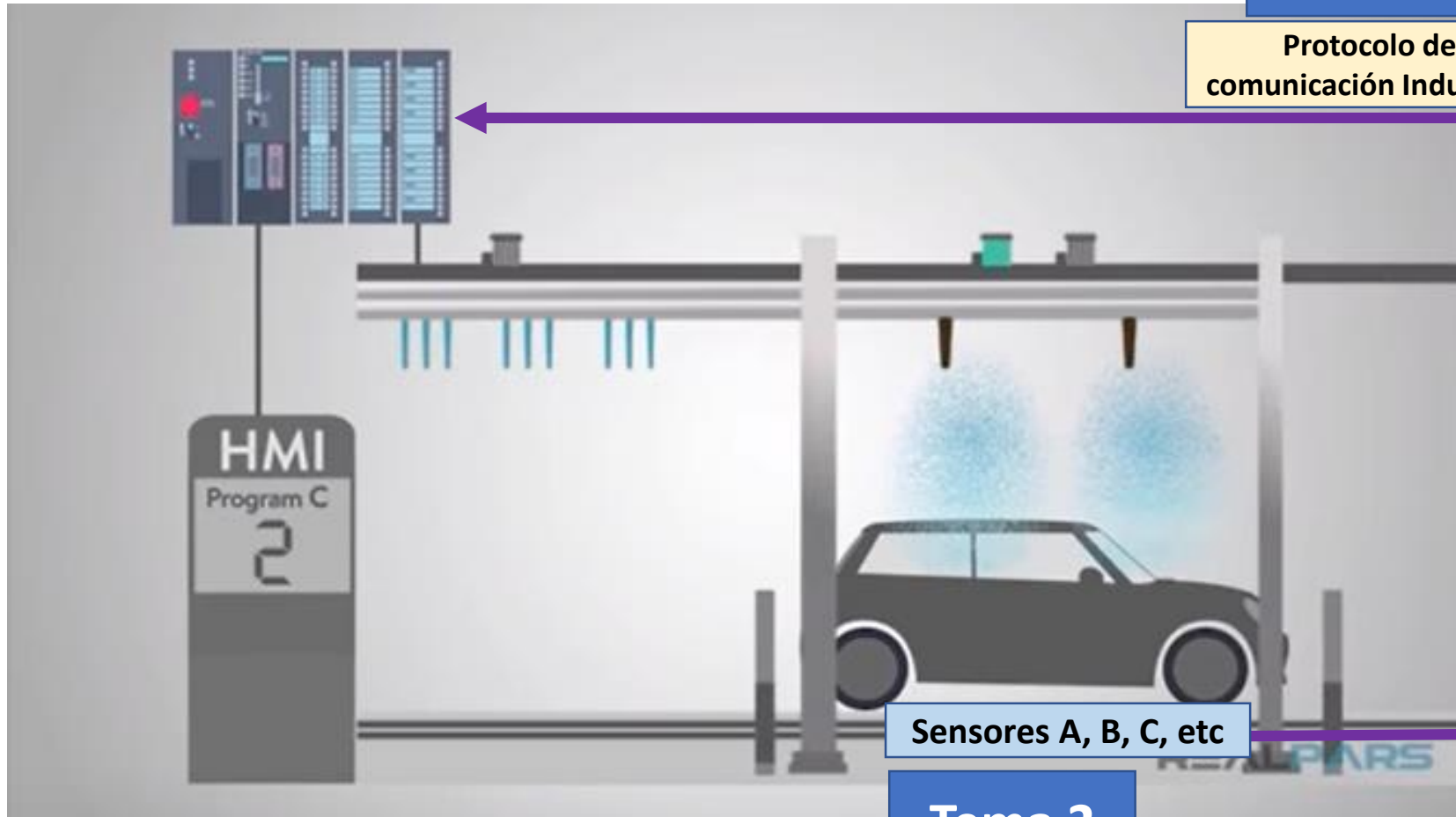
Tema 1

Protocolo de
comunicación IoT

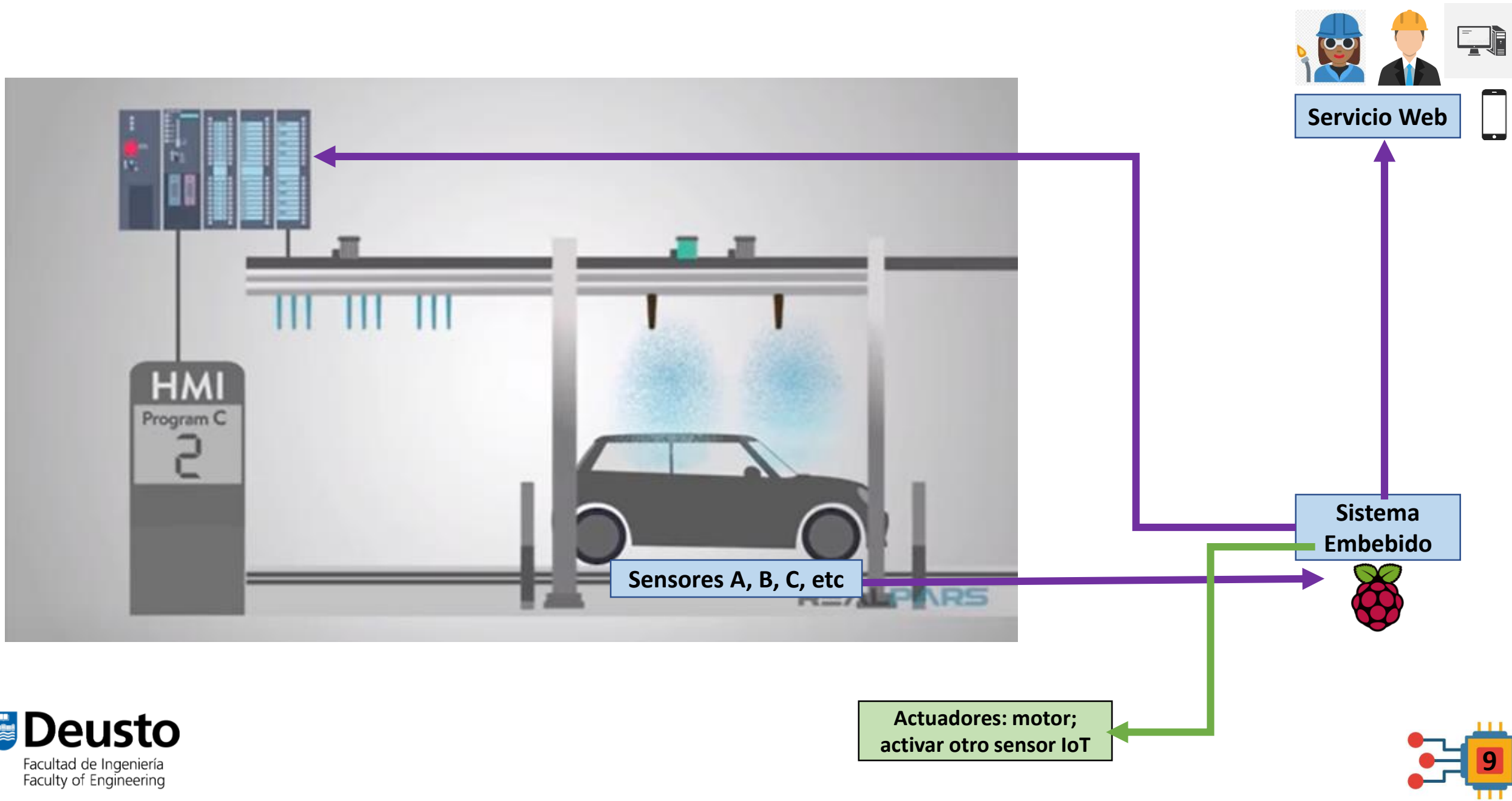
Tema 3a

Sensores A, B, C, etc

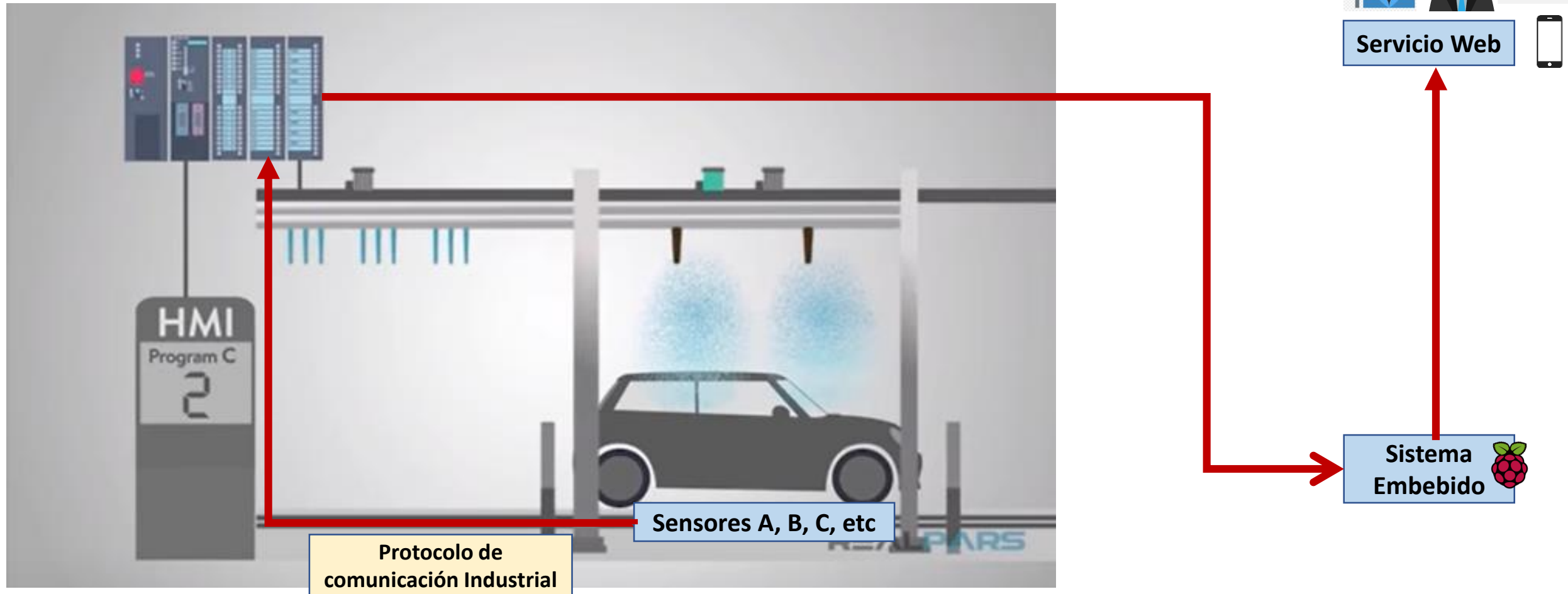
Tema 2



Ejemplo Escenario B



Ejemplo Escenario C



Ejemplo D

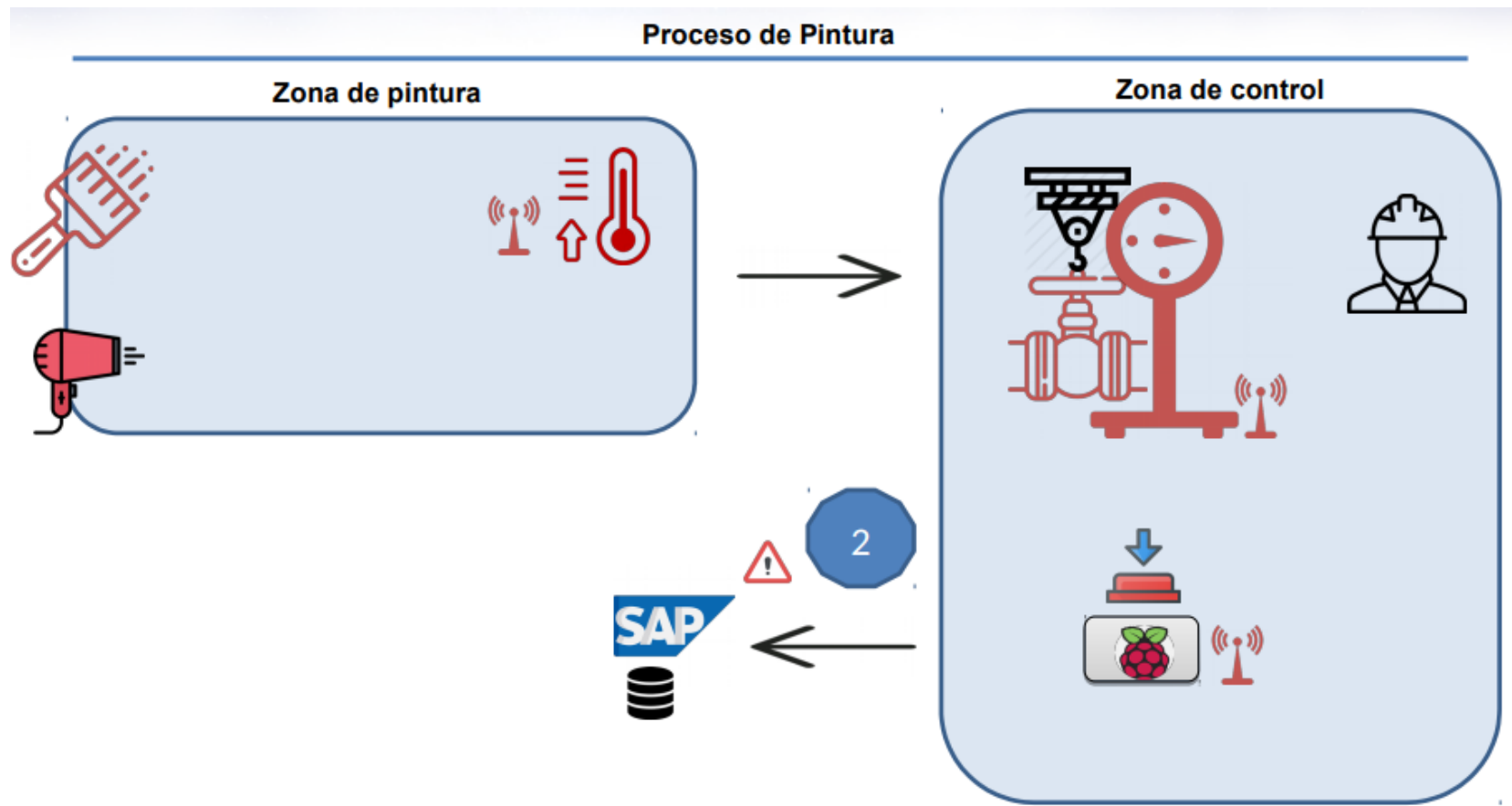
Una empresa líder internacional en componentes de fundición de acero inoxidable y alta aleación, y que realiza válvulas de alto valor tecnológico para las aplicaciones e industrias más exigentes.

El proceso de fabricación de válvulas de alto valor tecnológico consiste en la recepción de las materias primas, montaje, pintura, banco de Pruebas, Embalaje y Distribución



Ejemplo D

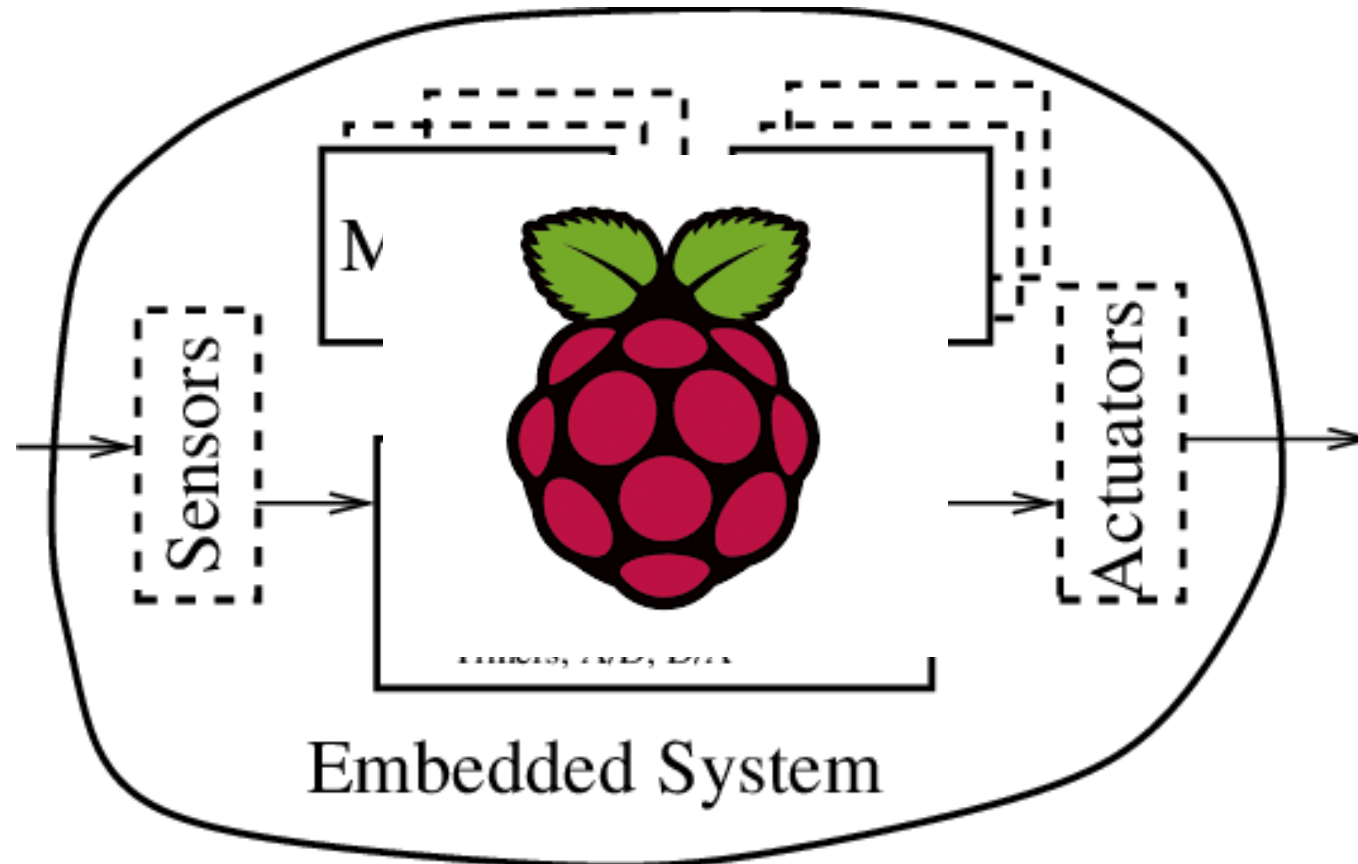
El proceso de fabricación de válvulas de alto valor tecnológico consiste en la recepción de las materias primas, montaje, pintura, banco de Pruebas, Embalaje y Distribución



Que sabemos hasta ahora?

- Que es y para que sirve un sistema embebido?
- Que aplicaciones (+industriales) tiene?
- **Elementos hardware principales de un SE (continuaremos cuando tengáis la RPi)**


Componentes de un Sistema Embebido



Que sabemos hasta ahora?

- Que es y para que sirve un sistema embebido?
- Que aplicaciones (+industriales) tiene?
- Elementos **hardware** principales de un SE (continuaremos cuando tengáis la RPi)
- **Elementos software: programación de un sistema embebido**
(Linux ->Python)

Semana 2: Linux Embebido (continuación)

SISTEMAS EMBEBIDOS 
(EMBEDED SYSTEMS)

Que es Windows Subsystem for Linux (WSL) ??

El WSL permite tener un ambiente de desarrollo en Linux, incluyendo todas las herramientas, utilidades y aplicaciones, directamente en Windows

Evita el “overhead” de usar un dual-boot o una maquina virtual

Permite usar GIT desde la ventana de comandos!



Windows Subsystem for Linux (WSL)

Done están mis archivos que creo con WSL
en Windows??



Acceder a los **archivos de WSL** desde Windows

- WSL instala un nuevo gestor de ficheros, separado del sistema de Windows NTFS C:\
- EN Linux, a los discos no se les asignan letras. Se les dan “mount points”
- El directorio raíz es \

```
\\wsl$\UbuntuVersion\home\username
```

Abre el explorador de ficheros de Windows, y escribe esta dirección

En mi caso: UbuntuVersion = **Ubuntu**
username = **laura**

Acceder a los **archivos de Windows** desde la terminal WSL

Desde la terminal de Ubuntu:

```
$cd ~
```

```
$ cd /mnt/c/Users/username
```

username es tu nombre de usuario en Windows.
En mi caso
username = **laura**

Comandos Linux gestion ficheros

Ejemplo con \$grep

```
$grep palabra archivo
```

- **palabra:** la palabra que estás buscando
- **archivo:** el archivo en el que estás buscando la palabra

Ignorar mayúsculas y minúsculas

```
$grep -i palabra archivo
```

Contar veces que aparece palabra

```
$grep -c palabra archivo
```

Comandos Linux gestion Ficheros

Ejemplo con \$find

```
find . -name "my-file"
```

Buscar archivo por **nombre** en directorio actual

```
find . -type f -name "fichero"
```

Buscar archivo por **tipo** en directorio actual

Tipos:

f – archivo normal

d – directorio o carpeta

l – enlace simbólico

c – dispositivos de caracteres

b – dispositivos de bloque

Commands linux

Instalar paquetes

```
$ sudo apt-get update
```

```
$ sudo apt-get install git
```

Shell Scripting

- El intérprete de comandos o shell es un programa que permite a los usuarios interactuar con el sistema, procesando las órdenes que se le indican.
- Además de comandos, los shells ofrecen otros elementos para mejorar su funcionalidad, tales como variables, funciones o estructuras de control

Que es un shell-script?

Un shell-script o “guión de órdenes” es un fichero de texto que contiene un conjunto de **comandos y órdenes interpretables por el shell.**

Que es un shell-script?

Ejemplos

```
#!/bin/dash
#Esto no se interpreta
echo Hola
ps w
echo "Proceso lee el script: $$"
```

Shell Scripting

- Variables

Sólo Definición	<code>VAR=""</code> <code>VAR=</code>
Definición y/o Inicialización/Modificación	<code>VAR=valor</code>
Expansión (Acceso a Valor)	<code>\$VAR</code> <code>\${VAR}</code>
Eliminación de la variable	<code>unset VAR</code>

Shell Scripting

❑ Variables

- Case-sensitive
- Es importante no incluir ningún espacio ni antes ni después del signo =.
- El valor de una variable siempre es tomado por el shell como una cadena de caracteres.
- Expansión de una variable: el uso de las llaves {}

Shell Scripting

❑ Expresion aritmetica

`$(expresión)`

- Permite evaluar las cadenas indicadas en la expresión como enteros, admitiendo gran parte de los operadores usados en el lenguaje C
- Si se usan variables en la expresión, no es necesario que vayan precedidas por el carácter \$ si ya contienen un valor entero válido (sí es necesario para los parámetros posicionales y especiales).

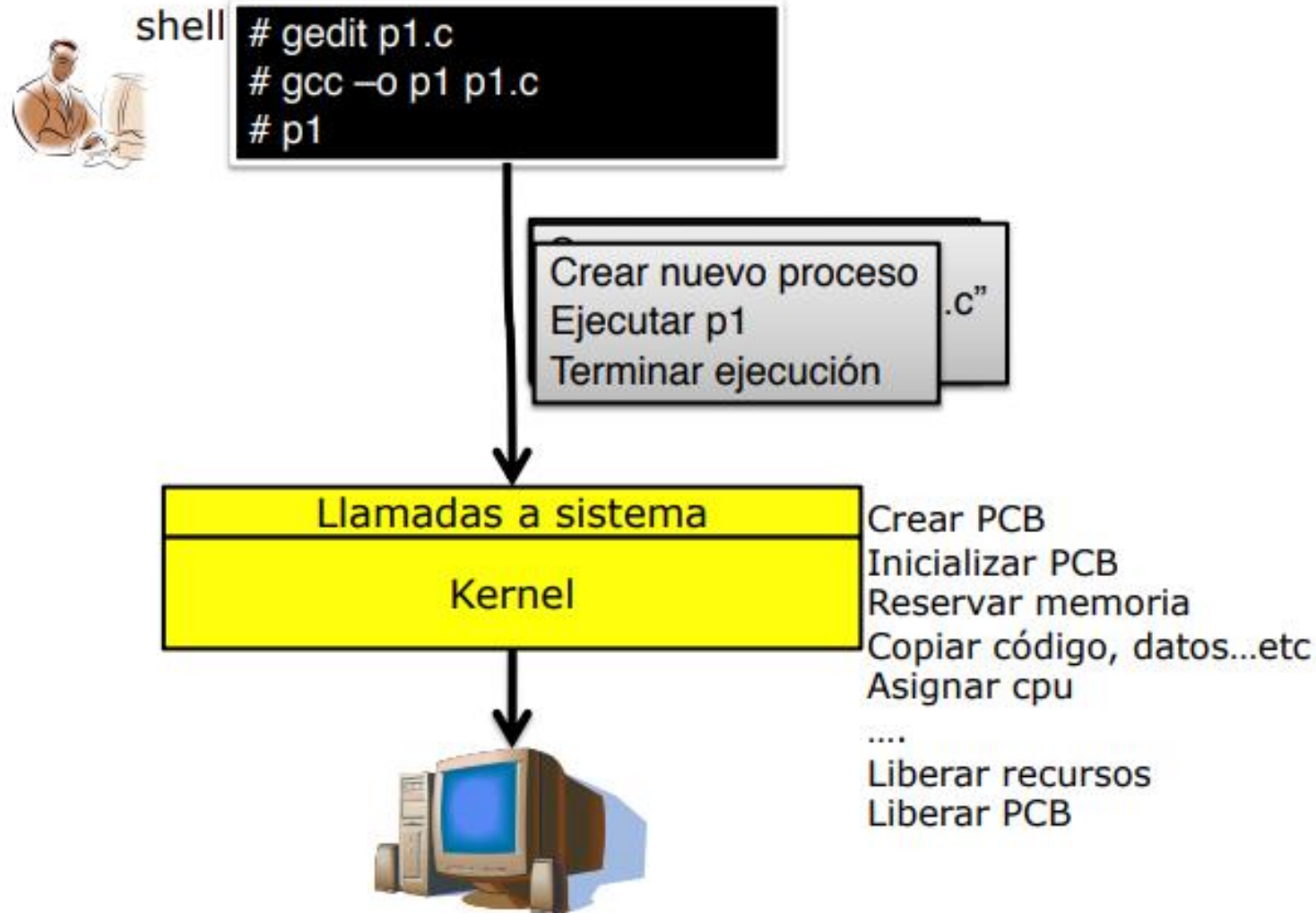
Programa VS Proceso (Linux)

- Programa: colección de instrucciones y datos almacenados en un archivo ejecutable.
- Proceso: programa es leído del disco por el *kernel* y cargado en memoria para ejecutarse. No solo hay una copia del programa – el *kernel* además añade información para manejarlo.

Gestion de procesos con Linux

- Proceso en ejecución:
 - Asignarle memoria para el código, los datos y la pila
 - Inicializar los registros de la CPU para que se empiece a ejecutar
 - Ofrecer acceso a los dispositivos (ya que necesitan acceso en modo kernel)
 - Muchas más cosas que iremos viendo
- En UNIX los procesos se identifican con un número entero denominado ID de proceso o **PID**

Gestion de procesos con Linux



PCB: *Process Control Block*
Contiene toda la información
del proceso

Gestion de procesos con Linux

- Comando \$ps
 - Lista de procesos que se están ejecutando

Campo	Significado
UID	ID del usuario propietario del proceso
PID	ID del proceso
PPID	ID del padre del proceso
C	Utilización del preprocesador de C para la administración de procesos
STIME	Hora de comienzo
TTY	terminal de control
TIME	Tiempo acumulado de CPU
CMD	Nombre del comando

Gestion de procesos con Linux

- Comando `$top`
 - Muestra información de procesador, procesos, y memoria
- Comando `$kill -9 PID`
 - Cierra el proceso PID. -9 corresponde al termino SIGKILL
 - Terminacion inmediata del proceso por el kernel de Linux

Actividad

Realizaremos alguna actividad practica sobre gestión de procesos en Linux cuando tengamos las Raspberry Pi (1-2 semanas)