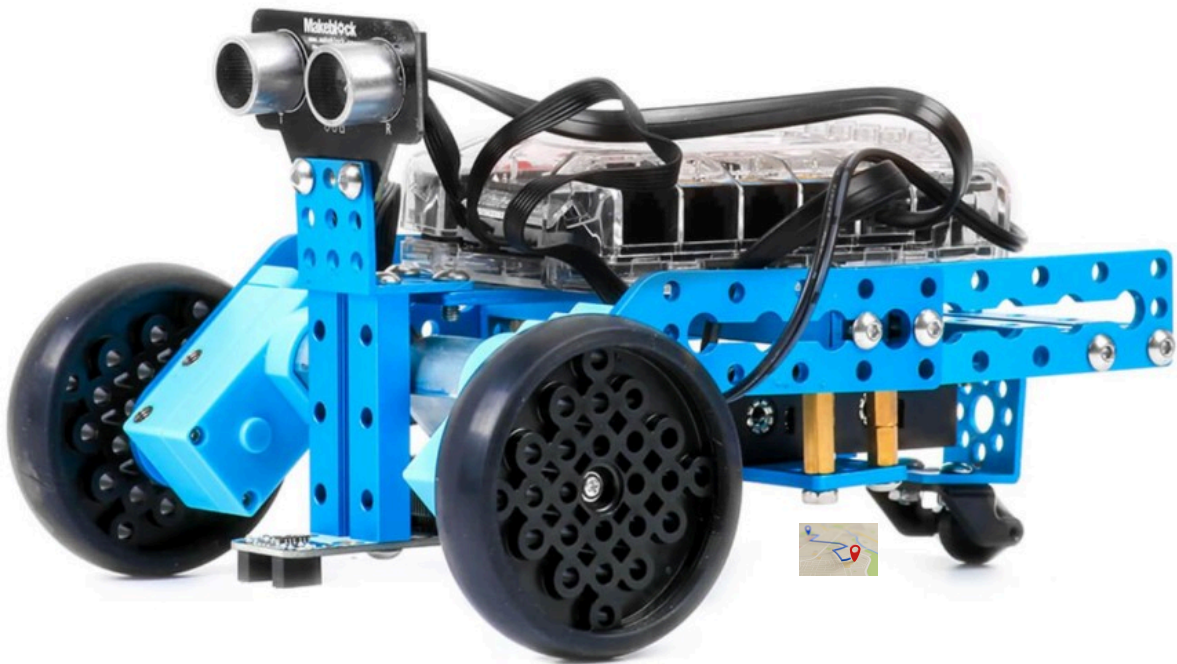


1ª PARTE

SISTEMAS INTELIXENTES

1ª Parte: Percepción, actuación e representación



GRUPO _____ MBOT RANGER _____

Nomes	Faltas	Observacións

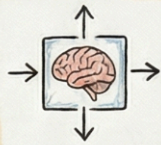
Táboa de exercicios

(marcar cales estan feitos)

Práctica 1 · Como mBlock lee un sensor: o son da aula.....	4
Exercicio 1 · Un personaxe di o ruído da aula.....	4
Exercicio 2 · O personaxe interpreta o ruído.....	6
Exercicio 3 · Observar o sensor de ultrasons.....	7
Exercicio 4 · O ángulo de detección do sensor.....	8
Exercicio 5 · O mBot reacciona coas luces.....	9
Exercicio 6 · Combinar os dous sensores.....	10
Borrador Wiki do grupo · Bloque Percepción.....	11
Práctica 2 · Sensores de liña e sensores de luz.....	12
Exercicio 1 · O obxecto interpreta a posición na liña.....	12
Exercicio 2 · Observar os sensores de luz.....	14
Exercicio 3 · O mBot reacciona á luz coas luces RGB.....	15
Exercicio 4 · Combinar luz e liña.....	16
Borrador Wiki do grupo · Peche do bloque Percepción.....	17
Práctica 3 · O mBot actúa: o mando de control.....	18
Exercicio 1 · Explorar os bloques de movemento.....	18
Exercicio 2 · O mando de control.....	19
Exercicio 3 · O mBot comunica o seu estado.....	20
Exercicio 4 · Control de roda independente.....	21
Borrador Wiki do grupo · Bloque Actuación.....	21
Práctica 4 · O bucle Percepción → Decisión → Actuación.....	23
Exercicio 1 · Bucle básico: obstáculo → parar.....	23
Exercicio 2 · Bucle con seguimento de liña.....	24
Exercicio 3 · Bucle completo: liña + obstáculo + sinalización.....	25
Exercicio 4 · Engade o sensor de son ao bucle.....	26
Borrador Wiki do grupo · Bloque Actuación — peche.....	27

Práctica 5 · Representación I: o gráfico do eclipse.....	29
Exercicio 1 · Configurar o gráfico e ver os primeiros datos.....	29
Exercicio 2 · O experimento do eclipse.....	30
Exercicio 3 · Mellora o experimento.....	32
Exercicio 4 · Deseña o teu propio experimento.....	33
Borrador Wiki do grupo · Bloque Representación.....	34
Práctica 6 · Representación II: sismógrafo e LiDAR.....	35
Experimento A · O sismógrafo do mBot.....	35
Exercicio A1 · Configurar o sismógrafo.....	36
Exercicio A2 · Rexistrar tremores simulados.....	37
Exercicio A3 · Detector de tremores con alarma.....	38
Experimento B · O escáner LiDAR do mBot.....	39
Exercicio B1 · Perfil estático: obstáculos a diferentes alturas.....	39
Exercicio B2 · LiDAR + xiroscopio: perfil con inclinación.....	40
Exercicio B3 · Experto: deseño dun sistema de navegación segura.....	41
Comparación: sismógrafo vs. LiDAR.....	42
Borrador Wiki do grupo · Bloque Representación — peche.....	42
Práctica 7 · Representación III: barras, tarta e táboa.....	44
Experimento A · Mapa de luz da aula (gráfico de barras).....	44
Exercicio A1 · Medir a luz en cada zona da aula.....	44
Experimento B · Diagnóstico de seguimento de liña (gráfica de tarta).....	46
Exercicio B1 · Configurar o sistema de contadores.....	47
Exercicio B2 · Medir e comparar escenarios.....	48
Experimento C · A caixa negra do mBot (táboa).....	49
Exercicio C1 · Configurar a caixa negra.....	49
Exercicio C2 · Probar a caixa negra na pista de prácticas.....	50
Exercicio D · Experto: combinar os tres tipos de gráfico.....	51

COMO FUNCIONA UN SISTEMA INTELIXENTE? EXPLORANDO A IA NA NOSA SOCIEDADE.



Que é un Axente Intelixente?

É un sistema situado nun entorno (real ou virtual) capaz de percibir datos, procesalos e seleccionar accións para acadar obxectivos específicos de forma autónoma.

1. Percepción: Os 'Sentidos' da IA

Cámaras Micrófonos GPS

Sensores Lidar (láser para medir distancias)

O sistema capta información mediante sensores como cámaras, micrófonos, GPS ou sensores Lidar.

2. Representación: Crear un Mapa

Mapa 3D Modelos Datos Estructurados

A IA organiza a información percibida en modelos ou mapas de datos estruturados para que sexa procesable.

3. Razoamento: Tomar Decisións

Algoritmos de Lóxica Redes Neuronais

Mediante algoritmos de lóxica ou redes neuronais, o sistema analiza a información e decide a acción máis adecuado (ex: suxerir un produto).

6. IA Colectiva: Traballo en Rede

Nube Datos Compartidos Experiencias

Diferentes sistemas comparten datos e experiencias a través da nube para mellorar o rendemento de todo o grupo (ex: coches que avisan doutros de accidentes).

O Ciclo de Intelixencia

5. Aprendizaxe: Mellora Continua

Datos Etiquetados (Supervisado) Patróns (Non Supervisado) Recompensa e Castigo (Reforzo)

Redes Neuronais

O sistema usa redes neuronais para aprender de forma supervisada, non supervisada ou por reforzo.

4. Actuación: Pasar á Acción

Execución Real/Dixital Brezo Robótico Asistente Virtual

É a execución de decisión no mundo real ou dixital, como o movemento dun brazo robótico ou a resposta dun asistente virtual.

A IA no día a día: O Coche Autónomo

Utiliza sensores de imaxe e Lidar para ver obstáculos

Razona a ruta segundo o tráfico

Controla os freos e o motor para conducir.

A IA no día a día: O Aspirador Intelixente

Crea un "mapa mental" de casa para non repetir zonas

Detecta obstáculos en tempo real

Aprende as rutinas dos habitantes para limpar cando non hai ninguén.

Compromiso Ético



Privacidade e Transparencia

É crucial protexer os datos persoais e que as decisións da IA sesan comprensibles (explicabilidade) e non parezan arbitrarias.



Nesgos e Responsabilidade

Debemos evitar que os algoritmos herden prexuízos sociais (nesgos) e definir claramente quen responde polas accións dun sistema autónomo.

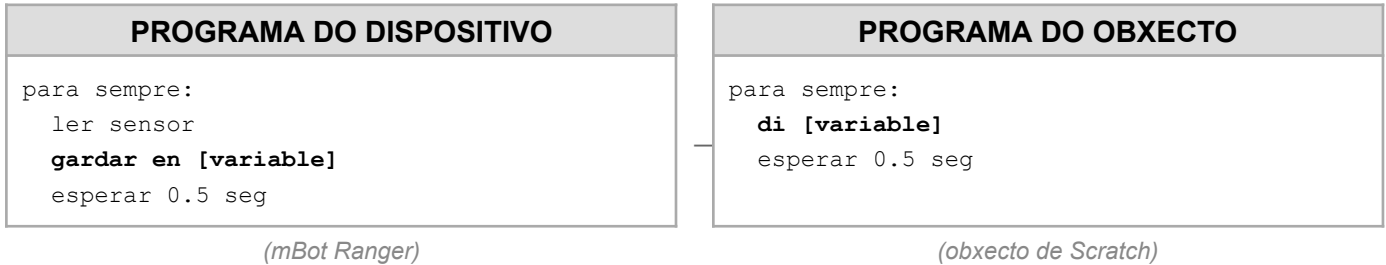
Práctica 1 · Como mBlock lee un sensor: o son da aula

O que aprendemos nesta práctica

En mBlock, o programa divídese en dous mundos que traballan **ao mesmo tempo**:

- **O dispositivo** (o mBot Ranger): le os sensores e controla motores, luces e son.
- **O obxecto** (un obxecto de Scratch): mostra información na pantalla do ordenador.


O problema é que **o dispositivo non ten bloques de «di»**. Para que o obxecto poida mostrar o que le o sensor, os dous programas precisan compartir a información a través dunha **variable común**.



A variable actúa como **ponte**: o dispositivo escribe nela o valor do sensor, e o obxecto léa e móstraa. Nesta práctica imos usar o **sensor de son** — é interno e doado de probar: só tes que falar ou aplaudir.

Antes de empezar

- O mBot está conectado en mBlock (indica «Conectado» na esquina superior dereita)
- Tes a extensión do Ranger activada (búscaa en «Extensiones»)
- Ves o obxecto por defecto na área de obxectos (o personaxe que xa vén con mBlock)

 Se o mBot non conecta ou a extensión non aparece, avisa antes de continuar. Non pases ao seguinte paso ata que funcione.


Exercicio 1 · Un personaxe di o ruído da aula

Obxectivo: facer que o obxecto mostre na pantalla o valor do sensor de son.

Paso 1 — Crea a variable

Vai á categoría «Variables» e crea unha variable nova. Ponlle o nome:

son_aula

 Asegúrate de que a variable sexa «para todos os obxectos» (non só para este obxecto). Así o dispositivo e o obxecto comparten o mesmo valor.

Paso 2 — Programa o dispositivo

Selecciona o **dispositivo** (mBot Ranger) na lista da esquerda. O fondo vólvese azul escuro cando está seleccionado.

Escribe este programa:

```
ao premer a bandeira verde
para sempre:
  pon [son_aula] a «volumen del sensor de sonido integrado»
  espera 0.5 segundos
```

Paso 3 — Programa o obxecto

Selecciona o **obxecto** (o personaxe) na lista de obxectos. O fondo vólvese branco.

Escribe este programa:

```
ao premer a bandeira verde
para sempre:
  di [son_aula] por 0.5 segundos
```

Paso 4 — Proba

Preme a bandeira verde. O obxecto debe mostrar un número que cambia cando fas ruído.

Situación	Valor que mostra o obxecto
Silencio total (ninguén fala nin se move)	
Ruído normal da aula (conversación baixa)	
Aplausos fortes preto do mBot	

¿O número cambia en tempo real cando fas ruído? ¿Si ou non?

¿Que pasaría se esqueceras seleccionar o dispositivo no Paso 2 e programaras o obxecto en lugar del?
(*Pensa antes de responder*)

◆ **Exercicio 1 completado — o obxecto mostra o valor do sensor**

SELO

Exercicio 2 · O personaxe interpreta o ruído

Obxectivo: en lugar dun número, o obxecto di unha frase segundo o nivel de ruído.

Baseándote nos valores que anotaches no Exercicio 1, decide os **dous limiares** que vas usar:

Se o valor é menor de: _____ → di "Silencio"	Por que escolliches ese número?
Se o valor está entre ese e: _____ → di "Ruído normal"	Por que escolliches ese número?
Se o valor é maior do segundo limiar → di "MOITO RUÍDO!"	(non fai falta decidir nada)

Completa o pseudocódigo antes de programar

```
PROGRAMA DO obxecto:  
para sempre:  
  se [son_aula] < [limiar 1] entón  
    di "_____" por 0.5 seg  
  se non, se [son_aula] < [limiar 2] entón  
    di "_____" por 0.5 seg  
  se non  
    di "_____" por 0.5 seg
```



O programa do DISPOSITIVO non cambia. Só modificas o programa do obxecto.

Responde despois de probalo

¿Tiveches que axustar os limiares respecto ao que escribiras na táboa? ¿Cantas veces?

Este tipo de axuste chámase **calibración**. ¿Coñeces algún sistema real que tamén se teña que calibrar?

◆ Exercicio 2 completado — o obxecto di frases segundo o ruído

SELO

Exercicio 3 · Observar o sensor de ultrasóns


Obxectivo: usar a mesma arquitectura co sensor de ultrasóns.

O sensor de ultrasóns **debe estar conectado no puerto 10**. Comproba que está ben enchufado antes de continuar.

Que tes que cambiar

Só precisas modificar o programa do dispositivo. O programa do obxecto queda igual.

```
PROGRAMA DO DISPOSITIVO (versión ultrasóns):  
ao premer a bandeira verde  
para sempre:  
    pon [son_aula] a «distancia del sensor de ultrasóns puerto10»  
    espera 0.5 segundos
```

 Podes reutilizar a mesma variable «son_aula» ou crear unha nova chamada «distancia». Se creas unha nova, non esquezas cambiala tamén no obxecto.

Táboa de observación

Situación	Valor (cm)
Sen obstáculo diante (sensor libre)	
A man a ~30 cm	
A man a ~10 cm	
A man moi preto (~2 cm)	
Obstáculo colocado a un lado (fóra do ángulo)	

Responde

¿Cal é o valor cando non hai nada diante? ¿Que significa ese número?

¿O sensor detecta obxectos que están moi a un lado? ¿Por que cres que é así?

◆ Exercicio 3 completado — o obxecto mostra a distancia

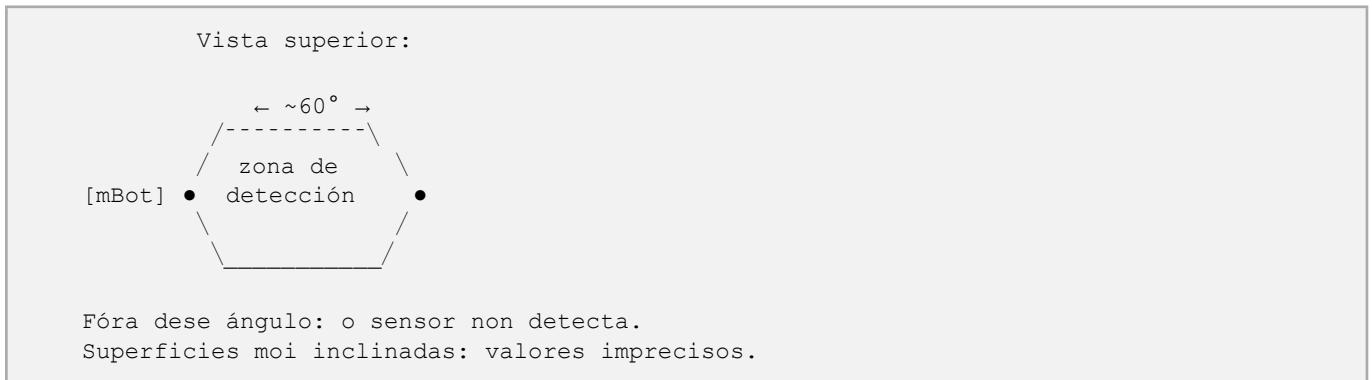
SELO

Exercicio 4 · O ángulo de detección do sensor

Obxectivo: comprobar experimentalmente ata onde chega o sensor.

Pon o mBot fixo sobre a mesa. Pon un obxecto (un estoxo) a 30 cm diante. Mirando o valor no obxecto, despraza o obxecto cara a un lado moi lentamente ata que o sensor deixe de detectalo.

O sensor de ultrasóns só detecta nunha dirección:



Debuxo — representa o que observas

[Debuxo: posición do mBot, traxectoria do obxecto e punto onde deixa de detectarse]

Responde

¿A que ángulo aproximado (en graos) deixa o sensor de detectar o obxecto?

¿Coincide co diagrama de arriba? ¿Que diferenzas atopas?

¿Que problema pode ter un robot que só teña UN sensor de ultrasóns para detectar o seu entorno?

◆ **Exercicio 4 completado — ángulo de detección comprobado**

SELO

Exercicio 5 · O mBot reacciona coas luces

Obxectivo: usar os sensores para controlar o anel de luces RGB do mBot.

Para controlar as luces **non precisas o obxecto nin a variable**. O programa de luces vai todo no dispositivo.

Parte A — ultrasons + luces

```
PROGRAMA DO DISPOSITIVO:  
para sempre:  
  se «distancia del sensor de ultrasons puerto10» < [ ]  
  entón  
    todas las luces encendidas en [cor] durante [ ] seg  
  se non  
    todas las luces encendidas en [cor] durante [ ] seg
```

Escolle ti a distancia limiar e as cores. Anota aquí o que escolliches:

Limiar (cm):	Cor cando está preto:	Cor cando está lonxe:

Parte B — Son + luces

```
PROGRAMA DO DISPOSITIVO:  
para sempre:  
  se «volumen del sensor de sonido integrado» > [ ]  
  entón  
    todas las luces encendidas en [cor] durante [ ] seg  
  se non  
    todas las luces encendidas en [cor] durante [ ] seg
```

Usa os limiares que anotaches no Exercicio 2. Anota aquí o que escolliches:

Limiar de volume:	Cor con ruído:	Cor en silencio:

Responde

¿Tiveches que axustar os limiares respecto ao que escribiras antes? ¿Cantas veces?

¿Que ten que ver axustar un limiar coa forma en que os sistemas intelixentes reais se calibran?

◆ Exercicio 5 completado — luces reaccionan aos sensores

SELO

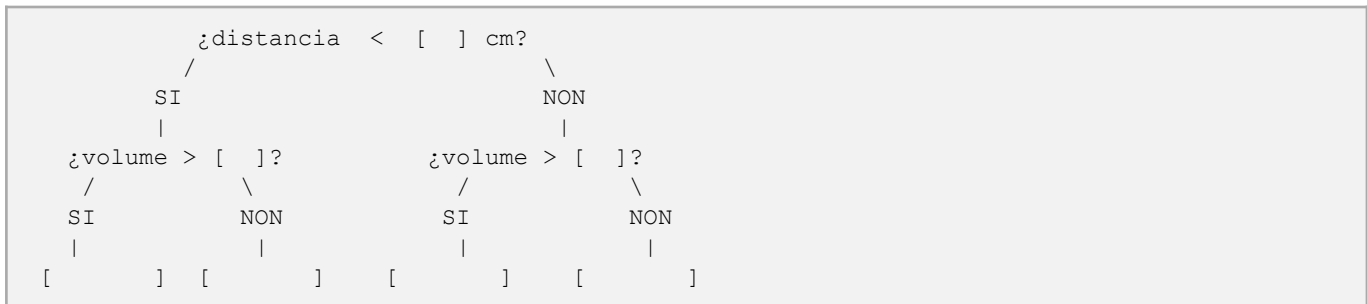
Exercicio 6 · Combinar os dous sensores

Obxectivo: combinar ultrasons e son nunha soa resposta. *Só cando tes os cinco selos anteriores.*

Define as catro situacións e as catro respostas antes de programar:

Nº	Distancia	Volume	Resposta do mBot (lucos)
1	Lonxe	silencio	
2	Lonxe	ruído	
3	Preto	silencio	
4	Preto	ruído	

Completa a árbore de decisión antes de programar



Responde

¿Este programa é un **sistema baseado en regras** ou **baseado en datos**? ¿Por que?

¿En que se parece á forma en que o Tesla decide cando frear?

◆ **Exercicio 6 completado — combinación dos dous sensores**

SELO

Borrador Wiki do grupo · Bloque Percepción

Abre o Wiki do teu grupo en Moodle. Hoxe engades a información da sección **PERCEPCIÓN** do voso sistema intelixente.

- ✿ *Para cada sensor do voso sistema: ¿como se chama? ¿que detecta? ¿que tipo de dato produce (número, imaxe, son, 0 ou 1)?*
- ✿ *¿En que condicións pode fallar ou dar valores incorrectos?*
- ✿ *Compara un sensor do voso sistema co sensor de ultrasons ou de son do mBot: ¿en que se parecen? ¿en que se diferencian?*
- ✿ *Indica a fonte de onde sacaches a información (URL, nome do artigo...).*

◆ Borrador Wiki · Percepción actualizado

SELO

Pregunta de peche · Sen mirar os apuntes

¿Por que en mBlock necesitamos unha variable para que o obxecto mostre o valor dun sensor? Explicao con tus propias palabras.

Práctica 2 · Sensores de liña e sensores de luz

O que aprendemos nesta práctica

O mBot Ranger ten dous tipos de sensores na parte inferior:

Sensor	Detecta	Tipo de dato	Bloque en mBlock
Liña E e D	Liña negra / fondo branco	Booleano (si/non)	¿sensor siguelíneas puerto9 detecta [E/D] negro?
Luz E (integrado1) Luz D (integrado2)	Intensidade de luz ambiente	Analóxico (0-100)	intensidad del sensor de luz integrado1 / integrado2

Os sensores de **liña** son **booleanos**: din só **si** (detecta negro) ou **non** (non detecta negro). Pódense usar directamente dentro dun bloque `se...entón` sen necesidade de variable.

Os sensores de **luz** son **analóxicos**: dan un número de 0 a 100. Como na Práctica 1, precisamos **variable + obxecto** para mostralos.



Comproba que o sensor de liñas está conectado no puerto 9 antes de empezar.

Exercicio 1 · O obxecto interpreta a posición na liña

Obxectivo: combinar os dous sensores de liña para que o obxecto diga onde está o mBot.


O mBot colocárase sobre un papel con liña negra sobre fondo branco. Segundo o que detecten os dous sensores, o obxecto mostrará unha mensaxe diferente:

Sensor E	Sensor D	O obxecto di...
negro	negro	"Estás na liña negra"
branco	branco	"Estás en fondo branco"
negro	branco	"Estáste saíndo pola dereita"
branco	negro	"Estáste saíndo pola esquerda"

Paso 1 — Crea as variables

Crea **dúas variables** (categoría «Variables»):

```
liña_esquerda
liña_dereita
```


 As dúas variables deben ser «para todos os obxectos», igual que na Práctica 1.

Paso 2 — Programa o dispositivo

Selecciona o **dispositivo** e escribe este programa:

```

ao premer a bandeira verde
para sempre:
  se ¿sensor siguelíneas puerto9 detecta izquierda negro? entón
    pon [liña_esquerda] a 1
  se non
    pon [liña_esquerda] a 0
  se ¿sensor siguelíneas puerto9 detecta derecha negro? entón
    pon [liña_dereita] a 1
  se non
    pon [liña_dereita] a 0
  espera 0.1 segundos
    
```

 Usamos os valores 1 e 0 para que o obxecto poida comparar os dous sensores facilmente. 1 = detecta negro, 0 = non detecta negro.

Paso 3 — Programa o obxecto

Selecciona o **obxecto** e escribe este programa usando o operador `e` da categoría «Operadores»:

```

ao premer a bandeira verde
para sempre:
  se [liña_esquerda] = 1 E [liña_dereita] = 1 entón
    di "Estás na liña negra" por 0.2 seg
  se non, se [liña_esquerda] = 0 E [liña_dereita] = 0 entón
    di "Estás en fondo branco" por 0.2 seg
  se non, se [liña_esquerda] = 1 E [liña_dereita] = 0 entón
    di "Estáste saíndo pola dereita" por 0.2 seg
  se non
    di "Estáste saíndo pola esquerda" por 0.2 seg
    
```

Paso 4 — Proba coa liña negra

Pon o mBot nas catro posicións e comproba que o obxecto di a mensaxe correcta:

Posición do mBot	O obxecto di...	¿Correcto?
Os dous sensores sobre a liña negra		
Os dous sensores sobre o fondo branco		
Sensor E na liña, sensor D no branco		
Sensor E no branco, sensor D na liña		

Responde

¿Houbo algunha posición que non funcionou como esperabas? ¿Cal e por que cres que foi?

¿En que parte do **Tesla Autopilot** cres que se usa un sistema parecido ao dos sensores de liña?

◆ **Eercicio 1 completado — o obxecto interpreta a posición na liña**

SELO

Exercicio 2 · Observar os sensores de luz

Obxectivo: ler os dous sensores de luz analóxicos e entender que miden.

Os sensores de luz están na parte **superior** do mBot, un á esquerda (integrado1) e outro á dereita (integrado2). Son analóxicos: dan un número de **0 (escuridade total)** a **100 (luz moi intensa)**.

Crea as variables e o programa

Crea dúas variables novas:

```
luz_esquerda  
luz_dereita
```

Programa o **dispositivo** para que lea os dous sensores e garde os valores nas variables. Programa o **obxecto** para que mostre os dous valores á vez.

🔴 No obxecto podes usar dous bloques «di» encadeados, ou crear unha soa mensaxe unindo os dous valores con «unir» (categoría Operadores).

Táboa de observación — anota os dous valores á vez

Situación	Luz E (integrado1)	Luz D (integrado2)
Luz normal da aula (sen tapar nada)		
Tapar con man o sensor esquerdo		
Tapar con man o sensor dereito		
Tapar os dous sensores		
Foco de luz directo sobre os dous		
Foco de luz só sobre o esquerdo		

Responde

¿Os dous sensores dan sempre o mesmo valor na mesma situación? ¿Por que pode haber diferenzas?

¿Para que cres que pode servir ter dous sensores de luz en lugar de só un?

◆ **Exercicio 2 completado — dous sensores de luz observados**

SELO

Exercicio 3 · O mBot reacciona á luz coas luces RGB

Obxectivo: usar os sensores de luz para controlar o anel de luces.

Programa o **dispositivo** para que o anel de luces cambie segundo a luz que detecta. Decide ti os limiares baseándote nos valores que anotaches no Exercicio 2:

```
PROGRAMA DO DISPOSITIVO:  
para sempre:  
  se «intensidad del sensor de luz integrado1» < [limiar] entón  
    todas las luces encendidas en [cor escuridade]  
  se non  
    todas las luces encendidas en [cor luz]
```

Anota aquí o que escolliches:

Limiar de luz:	Cor en escuridade:	Cor con luz:

Responde

¿Que diferenza hai entre usar o sensor de luz esquerdo ou o dereito para este programa? ¿Importa cal escollas?

◆ **Exercicio 3 completado — luces reaccionan á luz ambiente**

SELO

Exercicio 4 · Combinar luz e liña

Obxectivo: combinar os dous tipos de sensor nunha soa resposta. *Só cando tes os tres selos anteriores.*

O mBot ten que reaccionar de forma diferente segundo a luz E a posición na liña. Define ti as combinacións:

Posición na liña	Nivel de luz	Resposta (luzes + obxecto)
<i>Na liña negra</i>	<i>Con luz normal</i>	
<i>Na liña negra</i>	<i>En escuridade</i>	
<i>En fondo branco</i>	<i>Con luz normal</i>	
<i>En fondo branco</i>	<i>En escuridade</i>	
<i>Saíndo pola dereita</i>	<i>Con luz normal</i>	
<i>Saíndo pola dereita</i>	<i>En escuridade</i>	
<i>Saíndo pola esquerda</i>	<i>Con luz normal</i>	
<i>Saíndo pola esquerda</i>	<i>En escuridade</i>	

Responde

¿Este programa é máis parecido a un sistema baseado en regras ou a un sistema baseado en datos? Xustifica a túa resposta.

¿Que pasaría se o robot tivese que tomar estas decisións sen que ningunha persoa programase as regras? ¿Como o faría?

◆ Exercicio 4 completado — combinación luz e liña

SELO

Borrador Wiki do grupo · Peche do bloque Percepción

Esta é a última práctica do bloque de Percepción. Abre o Wiki do teu grupo en Moodle e completa **toda a sección PERCEPCIÓN** co voso sistema intelixente. Revisade o que escribistes nas prácticas anteriores e engadide o que falta.

✿ *Listade TODOS os sensores do voso sistema (mínimo 3). Para cada un: nome, que detecta, tipo de dato que produce.*

✿ *¿Que sensor é o máis importante para o funcionamento do sistema? ¿Por que?*

✿ *¿En que condicións pode fallar a percepción do sistema? Poñede un exemplo concreto.*

✿ *Compare os sensores do voso sistema cos sensores do mBot que traballastes nestas dúas prácticas: ¿que teñen en común? ¿que diferenzas hai?*

✿ *Indicade as fontes de toda a información (URL, nome do artigo, data de consulta).*



Esta sección do Wiki xa non se modificará máis. Revisade que está completa e ben escrita antes de pechar.

◆ Wiki · Sección Percepción completa e pechada

SELO

Pregunta de peche · Sen mirar os apuntes

O sensor de liña é **booleano** e o sensor de luz é **analóxico**. ¿Que vantaxes ten cada tipo segundo o que se quere detectar? Pon un exemplo de situación onde sería mellor un booleano e outro onde sería mellor un analóxico.

Práctica 3 · O mBot actúa: o mando de control

O que aprendemos nesta práctica

Nas prácticas anteriores o mBot **percibía** o entorno. Agora vai **actuar** sobre el. Un actuador é calquera dispositivo que o sistema usa para modificar o mundo físico:

Actuador do mBot	Que modifica	Equivalente no Tesla
Motores (2 rodas)	Desprazamento e xiro no espazo	Motor eléctrico principal
Anel de luces RGB	Sinal visual de estado ou alerta	Luces exteriores e pantalla LCD
Buzzer (son)	Sinal sonora de aviso	Altofalante e claxon

Os bloques de movemento teñen sempre dous parámetros: a **potencia** (en %, canto forza o motor) e o **tempo** (en segundos, canto dura a acción). Excepción: `adelante a potencia [n] % e para`, que non teñen tempo — execútanse ata que chegue outra orde.

Nesta práctica conectaremos o **teclado do ordenador** co robot: cada tecla enviará unha orde diferente. Así aprenderemos o concepto de **actuación controlada por eventos**: o sistema espera, e cando ocorre algo (premes unha tecla) executa a acción correspondente.

Exercicio 1 · Explorar os bloques de movemento

Obxectivo: probar cada bloque de movemento e rexistrar o que fai.

Programa no **dispositivo** un bloque de movemento diferente para cada tecla. Proba cada un e completa a táboa:

Bloque	Que fai o mBot	A potencia 20% vs 80%: diferenza?
<code>avanza a potencia [n] % durante [1] seg</code>		
<code>retrocede a potencia [n] % durante [1] seg</code>		
<code>gira a la izquierda a potencia [n] % durante [1] seg</code>		
<code>gira a la derecha a potencia [n] % durante [1] seg</code>		
<code>adelante a potencia [n] % (sen tempo)</code>		
<code>para</code>		

Responde

¿Que diferenza hai entre «avanza durante [n] seg» e «adelante a potencia [n] %» sen tempo?

¿Que pasa se non programas o bloque «para» despois de «adelante»? Proba e describe o que ocorre.


◆ **Exercicio 1 completado — bloques de movemento explorados**



Exercicio 2 · O mando de control

Obxectivo: programar un mando completo con teclado para controlar o mBot.

Usando o evento **cuando tecla [X] pulsada** (categoría Eventos), programa o dispositivo para que responda a cada tecla. Antes de programar, decide que potencia e tempo vas usar para cada movemento:

 O evento «cuando tecla pulsada» vai **NO DISPOSITIVO**, igual que os bloques de movemento. Non vai no obxecto.

Tecla	Orde ao mBot	Potencia / Cor / Tempo escolleitos
Frecha arriba ↑	avanza a potencia [] % durante [] seg	
Frecha abaixo ↓	retrocede a potencia [] % durante [] seg	
Frecha esquerda ←	gira a la izquierda a potencia [] % durante [] seg	
Frecha dereita →	gira a la derecha a potencia [] % durante [] seg	
Tecla A	todas las luces encendidas en [cor] durante [] seg	
Tecla S	para	
Tecla D	[decides ti]	
Tecla F	[decides ti]	

As teclas D e F son libres: decides ti que fan. Algunhas ideas: *tocar unha nota, acender un LED concreto, cambiar a cor das luces, soar durante 1 segundo...*

Proba o mando e responde

¿Tiveches que axustar a potencia ou o tempo despois de probalo? ¿Por que?

¿Que pasa se premes dúas teclas á vez? ¿O mBot executa as dúas ordes ou só unha?

¿En que se parece este sistema de control por teclado ao **panel táctil do Tesla**? ¿En que se diferencia?

◆ **Exercicio 2 completado** — mando de control funcionando

SELO

Exercicio 3 · O mBot comunica o seu estado

Obxectivo: engadir ao mando un sistema de sinalización: o obxecto informa do que está facendo o mBot.

Ata agora o obxecto non facía nada. Agora vai **mostrar o estado do robot** en tempo real: cada vez que premes unha tecla, o obxecto di que acción está executando o mBot.


Para isto precisas unha variable compartida, igual que nas prácticas anteriores:

```
Variable nova: estado_robot

DISPOSITIVO - cando tecla frecha arriba pulsada:
  pon [estado_robot] a "Avanzando"
  avanza a potencia [n] % durante [n] seg

DISPOSITIVO - cando tecla S pulsada:
  pon [estado_robot] a "Parado"
  para

obxecto - para sempre:
  di [estado_robot]
```

 Copia este patrón para TODAS as teclas do teu mando. Cada tecla pon un valor diferente en [estado_robot] antes de executar a acción.

Responde

¿Por que é importante que o sistema informe do seu estado, non só que actúe?

No **Tesla Autopilot**, o panel de instrumentos mostra en todo momento o que está facendo o sistema. ¿Que actuadores e que información de estado aparecen? (pensa no que viches na presentación)

◆ **Exercicio 3 completado** — obxecto informa do estado do mBot

SELO

Exercicio 4 · Control de rueda independente

Obxectivo: usar o bloque avanzado rueda izquierda a potencia [n] %, rueda derecha a potencia [n] % para crear movementos precisos. *Só cando tes os tres selos anteriores.*

Este bloque permite controlar as dúas rodas de forma **independente**. Se as dúas van á mesma potencia, o robot vai recto. Se van a potencias diferentes, xira. Se unha vai positiva e a outra negativa, xira sobre si mesmo.

Experimenta e completa a táboa

Roda esq.	Roda der.	Que fai?	Onde se usa isto?
+50%	+50%		
+50%	+20%		
+50%	-50%		
-50%	+50%		
0%	+50%		

Reto final

Programa o mando para que a tecla B faga que o mBot **xire sobre si mesmo 180°** (dando a volta completa no sitio) e despois pare. Determina ti a potencia e o tempo necesarios probando ata conseguilo.

Potencia usada: _____ Tempo necesario para 180°: _____ Intentos que precisaches: _____

¿Como calcularías o tempo necesario para xirar exactamente 90°?

◆ Exercicio 4 completado — control independente de rodas



Borrador Wiki do grupo · Bloque Actuación

Abre o Wiki do teu grupo en Moodle e engade a sección **ACTUACIÓN** do voso sistema intelixente.

✿ *Listade todos os actuadores do voso sistema (mínimo 2). Para cada un: nome, que modifica no mundo real, tipo de sinal que recibe (eléctrico, pneumático, dixital...).*

✿ *¿Cal é o actuador máis importante? ¿Por que?*

✿ *Completade a táboa de actuadores do mBot con equivalentes do voso sistema:*

Actuador	Que fai no mundo real	Equivalente no teu sistema
Motores	Movemento físico do robot	
Anel de luces RGB	Cor e patrón de luz visible	
Buzzer (son)	Son a frecuencia determinada	

✿ *¿Hai algún actuador no voso sistema que o mBot non teña? Descríbeo.*

✿ *Indicade as fontes (URL, nome do artigo, data de consulta).*

◆ Wiki · Sección Actuación completada

SELO

Pregunta de peche · Sen mirar os apuntes

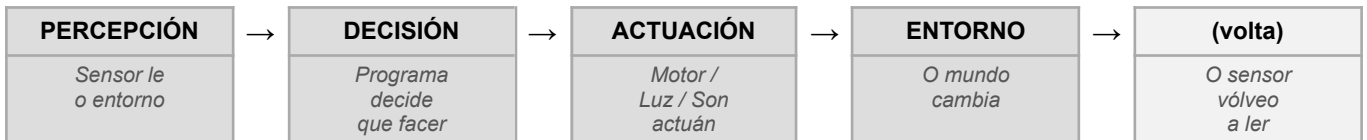
¿Cal é a diferenza entre un actuador que funciona **con tempo fixo** (como «avanza durante 1 seg») e un que funciona **sen tempo** (como «adelante a potencia 50%»)? ¿En que situacións é mellor cada un?

Práctica 4 · O bucle Percepción → Decisión → Actuación

O que aprendemos nesta práctica

Nas prácticas anteriores aprendemos por separado a **percibir** (prácticas 1 e 2) e a **actuar** (práctica 3). Agora imos unir os dous nunha soa arquitectura: o **bucle Percepción → Decisión → Actuación**.

Este bucle repítese continuamente mentres o sistema está en funcionamento:



Este é o patrón fundamental de calquera axente intelixente. O mBot fai exactamente o mesmo que o Tesla Autopilot, o robot Da Vinci ou o dron DJI Agras — **a diferenza está na velocidade, a precisión e o número de sensores e actuadores**, non na estrutura do bucle.

Material desta práctica: pista de prácticas (folio A3 con traxectoria negra sobre fondo branco). Colócaa baixo o mBot antes de empezar.

Exercicio 1 · Bucle básico: obstáculo → parar

Obxectivo: o mBot avanza pola pista de prácticas e para só cando detecta un obstáculo.

Este é o bucle máis sinxelo posible. O sensor de ultrasóns **percibe**, o programa **decide** se hai obstáculo ou non, e os motores **actúan**.

Deseña o bucle antes de programar

Completa a táboa co que vai facer o teu programa:


PERCEPCIÓN	DECISIÓN (se...entón)	ACTUACIÓN
Distancia < 15 cm	Hai obstáculo	
Distancia ≥ 15 cm	Non hai obstáculo	

Pseudocódigo — complétao

```

ao premer a bandeira verde
para sempre:
    se «distancia del sensor de ultrasóns puerto10» < 15 entón
        pon [estado_robot] a "_____ "
        _____
        todas las luces acesas en [___]
    se non
        pon [estado_robot] a "_____ "
    
```

todas las luces acesas en [____]

 Lembra: o programa do obxecto só le a variable [estado_robot] e usa di []. Non cambia nada respecto á Práctica 3.

Proba na pista de prácticas

Coloca un obstáculo (un libro, unha cartafol) na traxectoria. Activa o programa. Comproba que o mBot para antes de chegar ao obstáculo e avanza de novo cando o retiras.

¿A que distancia real para o mBot? Mídea: _____ cm ¿Coincide co limiar de 15 cm? ¿Por que pode haber diferenza?

¿Que pasa se o obstáculo está moi inclinado respecto ao sensor? Proba e describe:

◆ **Exercicio 1 completado — bucle básico con obstáculo funcionando**

SELO


Exercicio 2 · Bucle con seguimento de liña

Obxectivo: o mBot segue a traxectoria negra da pista de prácticas usando os sensores de liña.

Recupera o que aprendestes na Práctica 2 sobre os sensores de liña. Agora **a reacción non é só di unha frase — os motores responden**. O mBot debe manterse sobre a liña negra corrixindo a súa traxectoria continuamente.

Lóxica do seguimento de liña

Sensor E	Sensor D	Situación	Corrección (decide ti)
negro	negro	Centrado na liña	
branco	branco	Fóra da liña completamente	
negro	branco	Saíndo pola dereita	
branco	negro	Saíndo pola esquerda	

 Para as correccións usa o bloque «rueda esquerda a potencia [n] %, rueda derecha a potencia [n] %» que aprendestes no Exercicio 4 da Práctica 3. Permite correccións máis suaves que xirar bruscamente.

Pseudocódigo — compléatoo coas túas correccións

```
para sempre:
  se ¿detecta E negro? E ¿detecta D negro? entón
    rueda izq [ ]%, rueda der [ ]% ← avanza centrado
  se non, se ¿detecta E negro? E NON ¿detecta D negro? entón
    rueda izq [ ]%, rueda der [ ]% ← corrección dereita
  se non, se NON ¿detecta E negro? E ¿detecta D negro? entón
    rueda izq [ ]%, rueda der [ ]% ← corrección esquerda
  se non
    _____ ← fóra da liña
```

Proba na pista de prácticas e responde

¿O mBot segue a traxectoria de forma suave ou vai dando bandazos? ¿Que parámetro axustarías para mellorar?

¿En que se parece este sistema ao **Lane Keeping Assist** do Tesla (o sistema que mantén o coche no carril)?

◆ **Exercicio 2 completado — mBot segue a traxectoria da pista**

SELO

Exercicio 3 · Bucle completo: liña + obstáculo + sinalización

Obxectivo: integrar nun só programa o seguimento de liña, a detección de obstáculos e a sinalización ao obxecto.


Este é o programa máis completo que imos facer no bloque de Percepción e Actuación. O mBot **segue a liña**, pero se detecta un obstáculo a menos de 15 cm **para e avisa**. O obxecto informa do estado en todo momento.

Deseña o programa completo antes de programar

Completa este diagrama de fluxo:

```
para sempre:
  se distancia < 15 cm entón
    → ACTUACIÓN: _____
    → LUZ: _____
    → ESTADO: _____
  se non:
    se E=negro E D=negro entón
      → ACTUACIÓN: _____
      → LUZ: _____
    se non, se E=negro E D=branco entón
      → ACTUACIÓN: _____
    se non, se E=branco E D=negro entón
      → ACTUACIÓN: _____
```

```
se non
  → ACTUACIÓN: _____
  → LUZ: _____
pon [estado_robot] a _____
```

 O sensor de ultrasons ten prioridade sobre os sensores de liña: se hai obstáculo, o mBot para aínda que estea sobre a liña negra.

Proba na pista de prácticas

Activa o programa. Deixa que o mBot siga a liña e coloca un obstáculo no seu camiño. Comproba que para e avisa. Retira o obstáculo e comproba que retoma a traxectoria.

¿O mBot retoma correctamente a traxectoria tras esquivar o obstáculo? ¿Que melloras lle farías?

Este programa ten **dous sensores e tres actuadores** traballando á vez. ¿Cantos sensores e actuadores ten o Tesla Autopilot segundo o que vistes na presentación?

◆ **Exercicio 3 completado — bucle completo funcionando na pista**

SELO

Exercicio 4 · Engade o sensor de son ao bucle

Obxectivo: ampliar o programa do Exercicio 3 cunha nova percepción: o son. *Só cando tes os tres selos anteriores.*

Engade ao programa do Exercicio 3 esta nova regra: se o **sensor de son detecta aplausos** (volume > limiar que decidas), o mBot **para completamente** e agarda ata que haxa silencio antes de retomar a traxectoria. É como un **interruptor de emerxencia por voz**.

Decide os parámetros antes de programar

Limiar de volume para detectar aplausos:	
Limiar de volume para detectar silencio:	
Cor das luces cando para por aplausos:	
Mensaxe do obxecto cando para por aplausos:	

Esquema do bloque novo que debes engadir

(ANTES de calquera outra comprobación, dentro do «para sempre»:)

```
se «volumen del sensor de sonido integrado» > [limiar aplausos] entón
pon [estado_robot] a "_____"
todas las luces acesas en [cor]
para
espera hasta que «volumen del sensor de sonido integrado» < [limiar silencio]
pon [estado_robot] a "Retomando traxectoria"
```

Responde tras probalo

¿O mBot retoma a traxectoria correctamente despois do silencio? ¿Que problema pode ter?

Este sistema ten agora **tres sensores** (ultrasons, liña E/D, son) e **tres actuadores** (motores, luces, obxecto). ¿Como chamarías a este tipo de sistema segundo o que aprendestes nas unidades anteriores?

◆ **Exercicio 4 completado — interruptor de emerxencia por son**

SELO

Borrador Wiki do grupo · Bloque Actuación — peche

Abre o Wiki do teu grupo en Moodle. Revisade a sección **ACTUACIÓN** e engadide o que falta tras esta práctica.

❄ *Descríbide o bucle $P \rightarrow D \rightarrow A$ do voso sistema: que percibe, como decide e como actúa. Usade un exemplo concreto.*

❄ *¿Hai algún sensor que teña prioridade sobre os outros no voso sistema? ¿Cal e por que?*

❄ *¿Que pasa no voso sistema cando falla un sensor? ¿Ten algún mecanismo de seguridade?*

❄ *Comparade o bucle $P \rightarrow D \rightarrow A$ do voso sistema co do mBot que programastes hoxe. ¿Que semellanzas e diferenzas atopades?*

❄ *Fontes (URL, nome do artigo, data de consulta).*

◆ **Wiki · Sección Actuación pechada**

SELO

Pregunta de peche · Sen mirar os apuntes

Describe o bucle $P \rightarrow D \rightarrow A$ do programa do Exercicio 3 usando as túas propias palabras. **Non uses os nomes dos bloques de mBlock** — explica como lle explicarías a alguén que nunca programou.


Práctica 5 · Representación I: o gráfico do eclipse

O que aprendemos nesta práctica

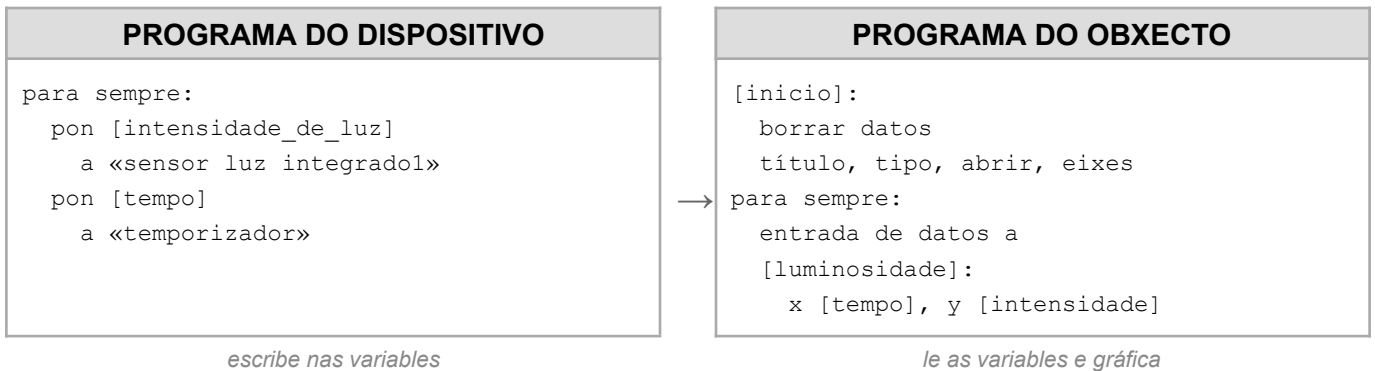
Ata agora o mBot **percibía e actuaba**, pero os datos desaparecían en canto o programa paraba. Un sistema intelixente real non só recolle datos: tamén os **almacena e representa** para poder tomar mellores decisións no futuro. Iso é a **representación**.

O Tesla Autopilot non só usa os datos dos sensores no momento — gárdaos, analízaos e actualiza o seu modelo do mundo. O mapa interno, o historial de velocidades, as alertas pasadas... todo iso é representación. **Sen representación non hai aprendizaxe**.

Nesta práctica usaremos a extensión **Data Chart** para visualizar en tempo real os datos do sensor de luz. Primeiro hai que activala:

 Vai a Extensiones en mBlock → pestana «Data Science» → busca «Data Chart» → Añadir. Verás que aparecen bloques verdes novos no obxecto.

A arquitectura é a mesma que nas prácticas anteriores: **dúas variables compartidas** (`intensidade_de_luz` e `tempo`) e **dous programas en paralelo** — o dispositivo escribe nas variables, o obxecto le os valores e gráficaos.




Exercicio 1 · Configurar o gráfico e ver os primeiros datos

Obxectivo: activar Data Chart, crear as variables e ver os datos do sensor de luz en tempo real.

Paso 1 — Crea as dúas variables


intensidade_de_luz	(escribela o dispositivo, léa o obxecto)
tempo	(escribela o dispositivo, léa o obxecto)

 As dúas variables deben ser «para todos os obxectos». É o mesmo patrón que nas prácticas anteriores.

Paso 2 — Programa o dispositivo

Selecciona o dispositivo e escribe:

```
ao premer a bandeira verde
reiniciar temporizador
para sempre:
  pon [intensidade_de_luz] a «intensidad del sensor de luz integrado1»
  pon [tempo] a «temporizador»
  espera 0.1 segundos
```

 O bloque «reiniciar temporizador» está na categoría Sensores. Úsao ao inicio para que o eixo X comece sempre desde 0.

Paso 3 — Programa o obxecto

Selecciona o obxecto e escribe:

```
ao premer a bandeira verde:
  borrar datos
  dar título ao gráfico "Intensidade lumínica durante un eclipse"
  cambia tipo de gráfico a gráfico de liñas
  abre ventana de gráfico de datos
  dar nome aos eixes: x [tempo] , y [luminosidade]
para sempre:
  entrada de datos a [luminosidade]: x [tempo], y [intensidade_de_luz]
```

Paso 4 — Comproba que funciona

Preme a bandeira verde. Debe abrirse a ventá do gráfico. Tapa e destapa o sensor de luz coa man. ¿A liña do gráfico sobe e baixa?

¿Que valor mostra o sensor con luz normal da aula? _____ ¿E coa man tapándoo? _____

¿O eixo X avanza co tempo aínda que non movas a man? ¿Que significa iso?

◆ **Exercicio 1 completado — gráfico en tempo real funcionando**

SELO

Exercicio 2 · O experimento do eclipse

Obxectivo: simular un eclipse e analizar a forma da curva resultante.

Un **eclipse solar** ocorre cando a Lúa pasa entre o Sol e a Terra, bloqueando a luz solar. Imos simulalo: a lanterna é o Sol, o sensor de luz é a Terra, e un obxecto grande (un libro, un cartafol) é a Lúa.

Configura o experimento

1. Coloca o mBot sobre a mesa co sensor de luz (parte superior) cara arriba.
2. Pon a lanterna (ou o flash do móbil) a uns 20 cm por riba do sensor, apuntando cara abaixo. Mantena fixa durante todo o experimento.
3. Comproba que o sensor rexistra un valor alto e estable antes de empezar.
4. Preme a bandeira verde para iniciar a gravación do gráfico.
5. Pasa o obxecto grande (a «Lúa») lentamente por diante da lanterna, tapando o sensor durante 2-3 segundos, e logo retírao.
6. Para o programa e observa a curva.

Antes de facer o experimento: predición

¿Que forma esperas que teña a curva? Completa a táboa e despois debúxaa:

Momento do experimento	Valor esperado do sensor (alto/baixo/?)
Antes de pasar o obxecto (lanterna libre)	
O obxecto comeza a tapar a lanterna	
O obxecto tapa completamente a lanterna	
O obxecto comeza a descubrir a lanterna	
Despois de pasar o obxecto (lanterna libre)	

Debuxo da forma predicha da curva (antes de facer o experimento):

[Debuxo da curva predicha: eixo X = tempo, eixo Y = intensidade de luz]

Despois do experimento: observación

Fai unha captura da pantalla co gráfico real ou debúxao aquí:

[Captura ou debuxo do gráfico real obtido no experimento]

Compara e responde

¿A curva real coincide coa que predixeches? ¿Que diferenzas hai?

¿Por que a curva non baixa bruscamente a cero cando tapas o sensor, senón que fai unha curva suave?

Os astrónomos usan este tipo de gráfico (chamado **curva de luz**) para detectar planetas fóra do sistema solar. Cando un planeta pasa por diante da súa estrela, a luz da estrela baixa lixeiramente. ¿En que se parece ao que fixeches ti?

◆ **Exercicio 2 completado — eclipse simulado e curva analizada**

SELO

Exercicio 3 · Mellora o experimento

Obxectivo: axustar a frecuencia de mostreo e comparar dous experimentos no mesmo gráfico.

No Exercicio 2 o dispositivo tomaba unha mostra cada 0,1 segundos. Imos experimentar con frecuencias diferentes e engadir un segundo sensor para comparar.

Parte A — Frecuencia de mostreo

Cambia o **tempo de espera** no programa do dispositivo a 0,5 segundos e repite o experimento. Logo proba con 0,01 segundos.

Espera (seg)	Como queda a curva?	Mellor ou peor? Por que?
0,5 seg (lento)		
0,1 seg (estándar)		
0,01 seg (rápido)		

Este concepto chámase **frecuencia de mostreo** (en inglés, *sampling rate*). ¿Por que non sempre é mellor tomar mostras máis rápido?

Parte B — Comparar dous sensores no mesmo gráfico

Engade o sensor **integrado2** (luz dereita) ao mesmo gráfico como un segundo grupo de datos. Necesitarás:

```
Variable nova:  intensidade_de_luz_2
```

```
DISPOSITIVO — engade ao para sempre:
```

```
    pon [intensidade_de_luz_2] a «intensidad del sensor de luz integrado2»
```

OBXECTO – engade ao para sempre:
entrada de datos a [luminosidade_2]: x [tempo], y [intensidade_de_luz_2]

Fai o experimento tapando só un dos sensores. ¿Que diferenza ves no gráfico entre os dous sensores?

◆ **Exercicio 3 completado — frecuencia de mostreo e dous sensores**



Exercicio 4 · Diseña o teu propio experimento

Obxectivo: aplicar o que aprendestes para rexistrar e analizar un fenómeno diferente. *Só cando tes os tres selos anteriores.*

Escollede un fenómeno que poidades medir co sensor de son ou co de ultrasóns e deseñade o voso propio experimento. Antes de programar, completade esta ficha de deseño:

Fenómeno que queredes medir:	
Sensor que usaredes:	
Título do gráfico:	
Nome do eixo X:	
Nome do eixo Y:	
Nome do grupo de datos:	
Frecuencia de mostreo (espera en seg):	
Forma que esperades na curva:	

Debuxo da curva predicha antes do experimento:

[Debuxo da curva predicha]

Captura ou debuxo da curva real obtida:

[Captura ou debuxo do gráfico real]

Responde

¿A curva real coincidiu coa predicha? ¿Que aprendestes do experimento?

◆ **Exercicio 4 completado — experimento propio deseñado e executado**



Borrador Wiki do grupo · Bloque Representación

Abre o Wiki do teu grupo en Moodle e engade a sección **REPRESENTACIÓN** do voso sistema intelixente.

- ✿ ¿Que datos almacena o voso sistema? Poñede exemplos concretos (non só «datos dos sensores»).
- ✿ ¿Como representa internamente o mundo o voso sistema? ¿Ten mapas, modelos, historial...?
- ✿ ¿Para que usa eses datos almacenados? ¿Mellora co tempo grazas a eles?
- ✿ Compare coa práctica do eclipse: ¿que ten en común o gráfico de luz co que fai o voso sistema cos seus datos?
- ✿ Fontes (URL, nome do artigo, data de consulta).

◆ **Wiki · Sección Representación completada**



Pregunta de peche · Sen mirar os apuntes

¿Por que dixemos que **sen representación non hai aprendizaxe**? Explícao usando como exemplo o que fixeches hoxe co gráfico do eclipse e o que sabes do Tesla Autopilot.

Práctica 6 · Representación II: sismógrafo e LiDAR

O que aprendemos nesta práctica

Na Práctica 5 usamos o gráfico de liñas para representar un único valor ao longo do tempo. Hoxe imos máis aló: **dous sensores ao mesmo tempo**, dous eixes Y independentes, e un novo tipo de sensor que aínda non usamos — o **xiroscopio**.

O xiroscopio mide a **inclinación en graos** en tres eixos: X (inclinación cara adiante/atrás), Y (inclinación lateral) e Z (rotación sobre si mesmo). Os valores van de **-180° a +180°**. É o mesmo sensor que usan os smartphones para detectar a orientación da pantalla, os drones para manterse nivelados e os barcos para medir o cabeceo.

Bloque	Que mide	Rango	Exemplo de uso real
ángulo eje x integrado	Inclinación cara adiante / atrás	-180° a +180°	<i>Cabeceo dun barco ou avión</i>
ángulo eje y integrado	Inclinación lateral esq./der.	-180° a +180°	<i>Balance dun barco</i>
ángulo eje z integrado	Rotación sobre si mesmo	-180° a +180°	<i>Compás, orientación dun drone</i>

Nesta práctica imos facer dous experimentos con sensores diferentes, pero coa mesma arquitectura que xa coñeces: **variable no dispositivo** → **obxecto le e gráfica**. O novo nesta práctica é o tipo de gráfico: a **gráfica de dobre eixo** (dous eixes Y independentes no mesmo gráfico).

Experimento A · O sismógrafo do mBot

Conexión co mundo real: a rede de sismógrafos do Teide

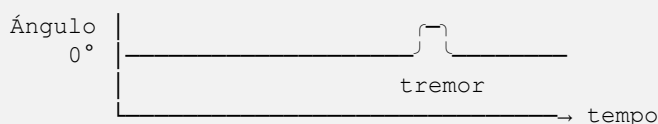
O Instituto Volcanolóxico de Canarias (INVOLCAN) ten máis de 30 sismógrafos distribuídos ao redor do Teide. Rexistran continuamente a actividade sísmica do volcán en tempo real. Cando se detecta un tremor, o gráfico mostra exactamente a mesma forma de onda que vas ver hoxe: valores estables, un pico brusco, e un amortecemento gradual. No futuro, estes sistemas non só rexistrarán — aprenderán a distinguir tremores volcánicos de ruído ordinario usando aprendizaxe automática. Isto é o que veremos no seguinte bloque da unidade.

Que é un sismógrafo e como funciona

Un **sismógrafo** é un instrumento que rexistra as vibracións do chan — tanto terremotos como tremores volcánicos ou explosións. O sensor principal é un **acelerómetro** ou **xeófono** que detecta cambios na aceleración en varias direccións.

O xiroscopio do mBot funciona de forma semellante: detecta cambios na orientación e aceleración angular. Cando golpeamos a mesa, o xiroscopio rexistra o impacto como un cambio brusco nos ángulos X e Y, seguido dun amortecemento gradual — exactamente como nun sismograma real.

Sismograma real durante un tremor:



Antes do tremor: liña plana (sen vibración)
Durante o tremor: pico brusco
Despois: amortecemento gradual ata volver á calma

Exercicio A1 · Configurar o sismógrafo

Obxectivo: rexistrar os eixos X e Y do xiroscopio nunha gráfica de dobre eixo.

Paso 1 — Crea as variables

```
xiro_x      (ángulo eje x del xiroscopio integrado)  
xiro_y      (ángulo eje y del xiroscopio integrado)
```

Paso 2 — Programa o dispositivo

```
ao premer a bandeira verde  
para sempre:  
  pon [xiro_x] a «ángulo eje x del xiroscopio integrado»  
  pon [xiro_y] a «ángulo eje y del xiroscopio integrado»  
  espera 0.05 segundos
```

A frecuencia de mostreo é 0,05 seg (20 mostrás por segundo) para capturar ben os picos bruscos dos impactos.

Paso 3 — Programa o obxecto con gráfica de dobre eixo

```
ao premer a bandeira verde:  
  borrar datos  
  dar título ao gráfico "Sismógrafo do mBot"  
  cambia tipo de gráfico a gráfica de dobre eje  
  abre ventana de gráfico de datos  
  dar nome aos eixos: x [tempo] , y [eixe X xiroscopio (graos)]  
  nombre del eje Y de la derecha: [eixe Y xiroscopio (graos)]  
  asignar grupo de datos [sismo_x] a eje Y de la izquierda  
  asignar grupo de datos [sismo_y] a eje Y de la derecha  
para sempre:  
  entrada de datos a [sismo_x]: x [tempo], y [xiro_x]  
  entrada de datos a [sismo_y]: x [tempo], y [xiro_y]
```

A gráfica de dobre eixo permite ver dous grupos de datos con escalas diferentes. O eixo Y esquerdo é para sismo_x e o dereito para sismo_y.

Paso 4 — Proba inicial

Pon o mBot sobre a mesa completamente quieto. Preme a bandeira. Observa o gráfico durante 5 segundos sen tocar nada.

¿Os dous eixos están a 0° cando o mBot está quieto? ¿Ou hai un valor de fondo? Anota:

Valor de xiro_x en repouso:	Valor de xiro_y en repouso:

◆ **Exercicio A1 completado — sismógrafo de dobre eixo configurado**

SELO

Exercicio A2 · Rexistrar tremores simulados

Obxectivo: simular diferentes tipos de tremor e analizar a forma dos sismogramas.

Realiza cada tipo de tremor mentres o gráfico está gravando. Anota o que observas:

Tipo de tremor	Como afecta ao eixo X?	Como afecta ao eixo Y?
Golpe suave na mesa desde diante		
Golpe suave na mesa desde un lado		
Golpe forte na mesa desde diante		
Mover o mBot cara adiante e atrás rapidamente		
Vibración continua (rascar a mesa)		

Captura o mellor sismograma e pégaos aquí:

[Captura do gráfico co sismograma máis interesante]

Responde

¿Como distinguirías no sismograma un golpe **cara adiante** dun golpe **lateral**? ¿Que eixo varía en cada caso?

Os sismógrafos do Teide tamén usan dous ou tres eixos. ¿Por que cres que é importante rexistrar varios eixos á vez en lugar de só un?

◆ **Exercicio A2 completado — tremores rexistrados e analizados**

SELO

Exercicio A3 · Detector de tremores con alarma

Obxectivo: engadir ao sismógrafo un sistema de alarma que se active cando o tremor supera un limiar.

Os sistemas reais de alerta temperá por terremoto non só rexistran — **actúan** cando a magnitude supera un valor crítico. Imos facer o mesmo co mBot.

Decide o limiar de alarma baseándote no que observaches no Exercicio A2:

```
ENGADE ao programa do DISPOSITIVO, dentro do «para sempre»:
```

```
se valor absoluto de [xiro_x] > [limiar]
OU valor absoluto de [xiro_y] > [limiar]
entón
  todas las luces encendidas en [cor alarma] durante 0.3 seg
  suena con frecuencia [Hz] durante 0.2 seg
```

🔴 O valor absoluto úsase porque os ángulos poden ser positivos ou negativos segundo a dirección do tremor. Podes calculalo con «valor absoluto de []» na categoría Operadores.

Limiar que escolliches: _____° Cor de alarma: _____ Frecuencia do son: _____ Hz

¿O sistema de alarma actívase con movementos normais da aula (alguén camiñando, pechando a porta)?
¿Como afina o limiar para evitar falsas alarmas?

◆ **Exercicio A3 completado — sismógrafo con alarma funcionando**

SELO

Experimento B · O escáner LiDAR do mBot

Conexión co mundo real: LiDAR na exploración lunar

LiDAR significa «Light Detection and Ranging» (Detección e Medición por Luz). Emite pulsos láser e mide o tempo que tardan en rebotar nun obstáculo — exactamente o mesmo principio que o sensor de ultrasons do mBot, pero con luz láser en lugar de son. O rover VIPER da NASA, reactivado en 2025 e previsto para alunizar no polo sur da Lúa en 2027 a bordo dun módulo de Blue Origin, usará LiDAR para cartografiar o terreo antes de avanzar: xera un perfil tridimensional do chan que lle permite detectar cráteres, pedras e pendentes perigosas. O mBot fará hoxe algo moi parecido: rexistrará a distancia a obstáculos ao longo do tempo para construír un «perfil» do seu entorno.

Que é LiDAR e en que se parece ao noso sensor

O sensor de **ultrasons** do mBot e o **LiDAR** dos rovers lunares usan o mesmo principio físico:

Principio de funcionamento:

```
Emisor → [pulso] → obstáculo  
← [eco] ←
```

$Distancia = (velocidade\ do\ sinal \times tempo\ de\ ida\ e\ volta) \div 2$

ultrasons mBot: velocidade do son $\approx 340\ m/s$

LiDAR real: velocidade da luz $\approx 300.000\ km/s$

O LiDAR é millóns de veces máis rápido e preciso,
pero o principio é o mesmo.

Mentres o rover avanza, o LiDAR vai rexistrando distancias continuamente. O resultado é un **perfil do terreo**: un gráfico que mostra a distancia ao chan en cada punto do percorrido. Se hai un cráter, a distancia aumenta bruscamente. Se hai unha pedra, diminúe.

Exercicio B1 · Perfil estático: obstáculos a diferentes alturas

Obxectivo: construír un perfil de distancias usando o sensor de ultrasons apuntando cara a unha «paisaxe» de obstáculos.

O mBot quedará **fixo na mesa**. En lugar de mover o robot, imos mover os obstáculos por diante del, como se o sensor estivese a voar por riba dun terreo. Usaremos o **gráfico de liñas** (non dobre eixo) e o temporizador no eixo X.

Configura o experimento

1. Crea a variable `distancia_obstaculo` no dispositivo.
2. O dispositivo le «distancia del sensor de ultrasons puerto10» cada 0,1 seg.
3. O obxecto configura o gráfico con título «Perfil LiDAR do mBot», tipo «gráfico de liñas», eixos «tempo» e «distancia (cm)».
4. O obxecto rexistra «entrada de datos a [lidar]: x [tempo], y [distancia_obstaculo]».

Protocolo de medición

Pon o mBot fixo apuntando cara a un espazo baleiro de polo menos 60 cm. Coloca obstáculos de **diferentes alturas** a diferentes distancias do sensor e pásaos lentamente por diante del. Suxestión: usa libros apilados, caixas ou estoxos de diferentes tamaños.

Predicción: debuxa a forma que esperas antes de facer o experimento

[Debuxo do perfil predicho: que forma terá a curva de distancias?]

Captura o gráfico real e pégoa aquí:

[Captura do gráfico LiDAR obtido]

Responde

¿Como aparece un obstáculo alto no gráfico? ¿E un obstáculo baixo? ¿E un espazo baleiro?

O rover VIPER usará LiDAR para decidir se pode avanzar con seguridade. ¿Que valor de distancia indicaría un **cráter perigoso**? ¿E unha **pedra pequena** que pode esquivar?

◆ Exercicio B1 completado — perfil LiDAR construído e analizado


SELO

Exercicio B2 · LiDAR + xiroscopio: perfil con inclinación

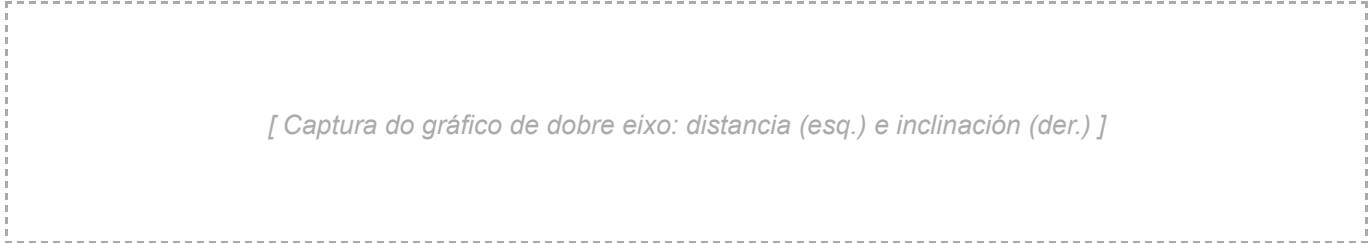
Obxectivo: combinar o sensor de distancia co xiroscopio nun gráfico de dobre eixo.

Os rovers lunares non só miden a distancia ao chan — tamén miden a **inclinación** para non volcar. Imos facer o mesmo: rexistraremos distancia e inclinación á vez no mesmo gráfico de dobre eixo.

Engade **xiro_x** como segundo grupo de datos no eixo Y dereito do gráfico. Mentres pasan os obstáculos por diante do sensor, inclina tamén o mBot lixeiramente cara adiante e cara atrás.

 Necesitarás engadir no programa do dispositivo a lectura de xiro_x, e no obxecto un segundo «entrada de datos» para o grupo [inclinación] no eixo dereito.

Captura o gráfico de dobre eixo con distancia e inclinación:



¿Que conclusión sacarías se nun momento do gráfico ves que a distancia **augmenta moito** (posible cráter) E ao mesmo tempo a inclinación **augmenta cara adiante**? ¿Avanzarías ou pararías o rover?

◆ **Exercicio B2 completado — LiDAR e xiroscopio combinados**



Exercicio B3 · Experto: deseño dun sistema de navegación segura

Obxectivo: usar os datos de LiDAR e xiroscopio para tomar decisións de movemento. *Só cando tes os selos anteriores.*

Deseña un programa que combine percepción (distancia + inclinación) con actuación (motores + luces) para que o mBot **decida automaticamente** se pode avanzar con seguridade, igual que faría un rover lunar real.

Define as regras do teu sistema de navegación:

Condición	Interpretación	Acción do rover
distancia > [] cm E inclinación < []°	<i>Terreo libre e nivelado</i>	
distancia < [] cm	<i>Obstáculo próximo</i>	
inclinación > []°	<i>Pendente perigosa</i>	
distancia > [] cm E inclinación > []°	<i>Posible cráter</i>	

¿Este sistema é capaz de aprender de erros anteriores? ¿Por que si ou por que non?

◆ **Exercicio B3 completado — sistema de navegación segura deseñado**

Comparación: sismógrafo vs. LiDAR

Reflexiona sobre os dous experimentos e completa a táboa:

Experimento	Forma esperada da curva	Forma real obtida
Sismógrafo — repouso	<i>Liña plana a 0°</i>	
Sismógrafo — golpe suave	<i>Pico pequeno + amortecemento</i>	
Sismógrafo — golpe forte	<i>Pico grande + amortecemento lento</i>	
LiDAR — espazo baleiro	<i>Valor alto e estable</i>	
LiDAR — obstáculo próximo	<i>Valor baixo brusco</i>	
LiDAR — obstáculo que se afasta	<i>Valor que sube gradualmente</i>	

Pregunta de síntese: os dous experimentos usaron representación de datos para tomar decisións (alarma de tremor, navegación segura). ¿En que se parecen á forma en que o Tesla Autopilot usa os seus datos de sensores para decidir?

Borrador Wiki do grupo · Bloque Representación — peche

Abre o Wiki do teu grupo en Moodle e **completa e pecha** a sección REPRESENTACIÓN do voso sistema intelixente.

- ✿ ¿O voso sistema usa algún tipo de gráfico ou representación visual dos datos? Descríbeo.
- ✿ ¿Con que frecuencia rexistra datos o voso sistema? ¿Cada segundo, en tempo real, só cando hai un evento?
- ✿ Comparade o experimento do sismógrafo ou do LiDAR co voso sistema: ¿que teñen en común?
- ✿ ¿Que pasaría se o voso sistema perdese os datos acumulados? ¿Seguiría funcionando igual?
- ✿ Revisade todo o que escribistes na sección Representación nas prácticas 5 e 6. Corrixide o que sexa necesario e aseguraosvos de que está completa antes de pechar.
- ✿ Fontes (URL, nome do artigo, data de consulta).



Esta é a última práctica do bloque de Representación. A sección do Wiki xa non se modificará.

SELO

◆ Wiki · Sección Representación completa e pechada

Pregunta de peche · Sen mirar os apuntes

Un sismógrafo do Teide rexistra un tremor. Un rover lunar rexistra un obstáculo. Un coche autónomo rexistra un peón na estrada. Os tres son sistemas moi diferentes, pero os tres fan o mesmo. **¿Que fan exactamente? Exprésao coas túas propias palabras usando os conceptos que aprendestes nestas dúas prácticas.**

Práctica 7 · Representación III: barras, tarta e táboa

O que aprendemos nesta práctica

Nas prácticas anteriores usamos o gráfico de liñas para representar datos **continuos no tempo**. Hoxe imos usar tres tipos de gráfico diferentes, cada un adecuado para un tipo diferente de información:

Tipo de gráfico	Cando usalo	Exemplo desta práctica
Gráfico de barras	Comparar categorías ou lugares distintos (non series temporais)	<i>Luz en diferentes zonas da aula</i>
Gráfica de tarta	Mostrar proporcións ou porcentaxes dun total	<i>% do tempo en cada estado da liña</i>
Tabla	Rexistrar eventos individuais con detalles (log)	<i>Cada obstáculo detectado polo mBot</i>

Unha diferenza importante respecto ás prácticas anteriores: nos gráficos de barras e tarta, os datos **non se envían en tempo real** — rexístranse primeiro e represéntanse despois. Na táboa si se pode facer en tempo real, como un **log de eventos**.

Experimento A · Mapa de luz da aula (gráfico de barras)

Conexión co mundo real: sistemas de xestión enerxética de edificios

Os edificios intelixentes modernos usan sensores de luz en cada zona para axustar automaticamente a iluminación artificial. Se unha zona ten moita luz natural (preto da xanela), as luces apáganse. Se está escura (corredor interior), acéndense ao máximo. O sistema da UE de eficiencia enerxética EN 15232 esixe este tipo de control adaptativo en edificios de oficinas e centros educativos. Hoxe imos facer un mapa de luz da nosa aula usando o mBot como sensor móbil, exactamente como faría un sistema de xestión enerxética profesional.

Que é un gráfico de barras e cando se usa

Un **gráfico de barras** compara valores entre **categorías distintas** — neste caso, as diferentes zonas da aula. A diferenza do gráfico de liñas, as barras non implican continuidade: cada barra é independente e representa un lugar diferente, non un momento diferente.

En Data Chart, o gráfico de barras funciona de forma diferente ao de liñas: en lugar de enviar datos continuamente, **enviamos un só valor por categoría**. O eixo X son as categorías (zonas) e o eixo Y é o valor medido (intensidade de luz).


Exercicio A1 · Medir a luz en cada zona da aula

Obxectivo: rexistrar a intensidade de luz en 5 zonas da aula e representala nun gráfico de barras.

Paso 1 — Protocolo de medición

Cada grupo levará o mBot ás 5 zonas indicadas. En cada zona: pon o mBot quieto durante **10 segundos**, anota os valores que mostra o obxecto (integrado1 e integrado2) e calcula a media dos dous. Completa a táboa:

Zona	Luz E (integrado1)	Luz D (integrado2)	Media dos dous
Zona 1 — Xanela principal			
Zona 2 — Centro da aula			
Zona 3 — Corredor interior			
Zona 4 — Esquina sen fiestras			
Zona 5 — Baixo o proxector			


 As medicións fanse co programa da Práctica 2 aínda activo (sensor de luz → variable → obxecto di o valor). Non precisas programar nada novo para a medición.

Paso 2 — Configurar o gráfico de barras no obxecto

Escrebe un programa novo no obxecto para representar os datos recollidos:

```
ao premer a bandeira verde:
  borrar datos
  dar título ao gráfico "Mapa de luz da aula"
  cambia tipo de gráfico a gráfico de barras
  abre ventana de gráfico de datos
  dar nome aos eixes: x [zona] , y [intensidade media]

  entrada de datos a [luz_aula]: x "Zona 1 - Xanela", y [valor medido]
  entrada de datos a [luz_aula]: x "Zona 2 - Centro", y [valor medido]
  entrada de datos a [luz_aula]: x "Zona 3 - Corredor", y [valor medido]
  entrada de datos a [luz_aula]: x "Zona 4 - Esquina", y [valor medido]
  entrada de datos a [luz_aula]: x "Zona 5 - Proxector", y [valor medido]
```

 Substitúe [valor medido] polo número real que anotaches na táboa. Neste gráfico os datos introdúcense manualmente, non en tempo real.

Captura o gráfico de barras e pégo aquí:

[Captura do gráfico de barras — mapa de luz da aula]

Responde

¿Que zona ten máis luz? ¿E menos? ¿Coincide co que esperabas antes de medir?

Un sistema de xestión enerxética usaría estes datos para **decidir automaticamente** que luces acender e a que intensidade. ¿Como programarías esa decisión usando o que sabes do bloque de Razoamento?

Reflexión ética: este sistema necesita sensores en toda a aula para funcionar. ¿Que implicacións ten instalar sensores permanentes nun espazo onde hai persoas? ¿É diferente se os datos son anónimos?

♦ **Exercicio A1 completado** — mapa de luz da aula representado

SELO

Experimento B · Diagnóstico de seguimento de liña (gráfica de tarta)

Conexión co mundo real: sistemas de monitorización de condutas e fisioterapia

Os sistemas de monitorización postural en fisioterapia usan acelerómetros para clasificar os movementos dunha persoa en categorías (correcto, leve desviación, desviación moderada, incorrecto) e mostran a distribución en gráficas de tarta. Os fisioterapeutas usan estes datos para ver en que tipo de postura pasa máis tempo o paciente. De forma moi semellante, os sistemas de control de calidade industrial clasifican produtos en categorías (conforme, defecto leve, defecto grave, rexeitado) e mostran a proporción de cada categoría. Hoxe imos clasificar os estados do mBot sobre a pista de prácticas e ver en que estado pasa máis tempo.

Como funciona a gráfica de tarta con contadores

A gráfica de tarta non recibe datos continuamente — recibe **un valor por categoría**, igual que as barras. A diferenza é que os valores representan **proporcións** dun total. Para iso precisamos **contadores**: variables que se incrementan cada vez que o robot está nun estado determinado.

Lóxica dos contadores (no DISPOSITIVO):

```
para sempre:
  se E=negro E D=negro entón
    pon [conta_liña] a [conta_liña] + 1
  se non, se E=negro E D=branco entón
    pon [conta_dereita] a [conta_dereita] + 1
  se non, se E=branco E D=negro entón
    pon [conta_esquerda] a [conta_esquerda] + 1
  se non
    pon [conta_fora] a [conta_fora] + 1
  espera 0.1 segundos
```

Despois de 30 segundos de medición, enviamos os 4 contadores ao gráfico de tarta. A tarta mostrará a **proporción de tempo** que o mBot pasou en cada estado.

Exercicio B1 · Configurar o sistema de contadores

Obxectivo: crear as 4 variables contadoras e o programa do dispositivo.

Crea as 4 variables contadoras

```
conta_liña      conta_dereita  
conta_esquerda  conta_fora
```

Programa do dispositivo

Copia o pseudocódigo de arriba ao dispositivo. Engade ao inicio:

```
ao premer a bandeira verde:  
  pon [conta_liña]      a 0  
  pon [conta_dereita]   a 0  
  pon [conta_esquerda] a 0  
  pon [conta_fora]     a 0  
  (a continuación o bucle de contadores)
```



O reinicio a 0 é fundamental: sen el, os contadores acumularían valores de execucións anteriores.

Programa do obxecto — gráfica de tarta

```
cando tecla [espazo] pulsada:  
  borrar datos  
  dar título ao gráfico "Diagnóstico de seguimento de liña"  
  cambia tipo de gráfico a gráfica de tarta  
  abre ventana de gráfico de datos  
  entrada de datos a [estados]: x "Na liña",      y [conta_liña]  
  entrada de datos a [estados]: x "Saíndo D",     y [conta_dereita]  
  entrada de datos a [estados]: x "Saíndo E",     y [conta_esquerda]  
  entrada de datos a [estados]: x "Fóra",        y [conta_fora]
```



A gráfica actualízase ao premer a barra espaciadora, non en tempo real. Así podes medir durante 30 segundos e despois ver o resultado de golpe.

◆ **Exercicio B1 completado — sistema de contadores configurado**

SELO

Exercicio B2 · Medir e comparar escenarios

Obxectivo: usar a gráfica de tarta para comparar dous escenarios de colocación do mBot na pista.

Fai a medición (30 segundos) en cada un dos dous escenarios. Para o programa, preme espazo para ver a tarta e fai a captura.

Escenario 1 — mBot perfectamente centrado na liña

[Captura da gráfica de tarta — Escenario 1]

Estado	Condición	Contador	% do tempo total
Na liña negra	E=negro E D=negro		
Saíndo pola dereita	E=negro E D=branco		
Saíndo pola esquerda	E=branco E D=negro		
Fóra da liña	E=branco E D=branco		

Escenario 2 — mBot lixeiramente desprazado cara á dereita

[Captura da gráfica de tarta — Escenario 2]

¿Como cambia a distribución entre os dous escenarios? ¿Que estado aumenta no Escenario 2?

Se este fose un sistema de control de calidade industrial, ¿que porcentaxe de tempo «fóra da liña» considerarías aceptable antes de parar a produción?

◆ Exercicio B2 completado — dous escenarios comparados con gráfica de tarta

SELO

Experimento C · A caixa negra do mBot (táboa)

Conexión co mundo real: caixas negras de vehículos e avións

Os avións levan un Flight Data Recorder (caixa negra) que rexistra cada evento relevante: cambios de altitude, velocidade, alertas, accións dos pilotos... Os coches modernos teñen un Event Data Recorder (EDR) que garda os últimos segundos antes dun accidente: velocidade, freada, posición do volante. Os coches autónomos como o Tesla rexistran miles de eventos por segundo. En caso de accidente, estes rexistros permiten reconstruír exactamente o que pasou. Hoxe imos crear a caixa negra do mBot: cada vez que detecta un obstáculo, rexistrará na táboa o momento, a distancia e a acción que tomou.

Como funciona a táboa en Data Chart

A táboa é o tipo de gráfico máis diferente dos outros: non dibuja curvas nin sectores — mostra **filas de datos**, como unha folla de cálculo. Cada vez que chega un novo dato, engádese unha nova fila. É ideal para rexistrar **eventos discretos**: cousas que ocorren nun momento concreto e que queres gardar con todos os seus detalles.

En Data Chart, a táboa usa o mesmo bloque entrada de datos a [grupo]: x [valor1], y [valor2], pero agora o eixo X pode ser texto (o nome do evento) e o eixo Y o valor numérico (a distancia, o tempo...).

Exercicio C1 · Configurar a caixa negra

Obxectivo: rexistrar na táboa cada vez que o mBot detecta un obstáculo a menos de 20 cm.

Variables necesarias


distancia_obstaculo	(xa a tes da Práctica 4)
tempo	(xa a tes da Práctica 5)
num_evento	(variable contadora de eventos)

Programa do dispositivo

```
ao premer a bandeira verde:
  pon [num_evento] a 0
  reiniciar temporizador
para sempre:
  pon [distancia_obstaculo] a «distancia ultrasonidos puerto10»
  pon [tempo] a «temporizador»
  se [distancia_obstaculo] < 20 entón
    pon [num_evento] a [num_evento] + 1
    pon [estado_robot] a "OBSTÁCULO"
    todas las luces encendidas en [cor alerta]
    para
  se non
    pon [estado_robot] a "Libre"
    adelante a potencia 40 %
    espera 0.1 segundos
```

Programa do obxecto — táboa

```
ao premer a bandeira verde:
  borrar datos
  dar título ao gráfico "Caixa negra do mBot"
  cambia tipo de gráfico a tabla
  abre ventana de gráfico de datos
  dar nome aos eixes: x [evento nº] , y [distancia (cm)]
para sempre:
  se [estado_robot] = "OBSTÁCULO" entón
    entrada de datos a [log]:
      x unir "Evento " con [num_evento],
      y [distancia_obstaculo]
    espera hasta que [estado_robot] ≠ "OBSTÁCULO"
```

 O bloque «espera hasta que» evita que o mesmo obstáculo se rexistre centos de veces mentres o mBot está parado diante del. Só rexistra un evento por obstáculo.

◆ **Exercicio C1 completado — caixa negra configurada**

SELO

Exercicio C2 · Probar a caixa negra na pista de prácticas

Obxectivo: xerar un log de obstáculos real e analízalo.

Activa o programa. Coloca e retira 5 obstáculos diferentes diante do mBot a diferentes distancias. Despois de cada obstáculo, retírao para que o mBot poida continuar.

Captura a táboa co log de eventos e péga aquí:

[Captura da táboa — log de eventos da caixa negra]

Responde

¿Todos os eventos teñen a mesma distancia? ¿Por que pode haber diferenzas entre obstáculos que colocaches á mesma distancia?

Imaxina que este log é o rexistro dun accidente dun coche autónomo. ¿Que información engadirías á táboa para que fose máis útil para investigar o que pasou?

Reflexión ética: os coches Tesla envían automaticamente os datos do EDR á empresa en caso de accidente. ¿Quen debería ter acceso a eses datos: só a empresa, tamén as autoridades, tamén as aseguradoras? ¿Que opinas?

◆ Exercicio C2 completado — log de eventos analizado


SELO

Exercicio D · Experto: combinar os tres tipos de gráfico

Obxectivo: nun só programa, usar os tres tipos de gráfico vistos hoxe para dar unha visión completa do sistema. *Só cando tes os selos dos tres experimentos anteriores.*

Deseña un programa que, durante 60 segundos de funcionamento do mBot na pista, **rexistre ao mesmo tempo:**

Gráfico	Que mostra	Cando se actualiza
Táboa (log)	Cada obstáculo detectado con tempo e distancia	<i>En tempo real, cada evento</i>
Tarta (estados)	Proporción de tempo en cada estado da liña	<i>Ao premer espazo (cada 30 seg)</i>
Barras (luz)	Intensidade de luz nas 5 zonas da aula	<i>Ao inicio (datos manuais)</i>

 Data Chart só pode ter un gráfico aberto á vez. Para ver os tres, terás que pechalos e abrilos alternadamente usando «cierra ventana de gráfico de datos» e «abre ventana de gráfico de datos».

Capturas dos tres gráficos activos ao mesmo tempo (abre cada un e captura):

[Captura táboa / tarta / barras — sistema completo de representación]

¿Que conclusión sobre o comportamento do mBot sacarías mirando os tres gráficos xuntos que non poderías sacar mirando só un?

◆ Exercicio D completado — tres tipos de gráfico combinados

SELO



Pregunta de peche · Sen mirar os apuntes

Tes tres tipos de gráfico: **barras, tarta e táboa**. Para cada un destes sistemas reais, di cal usarías e por que: (a) o rexistro dun accidente de tráfico, (b) o consumo de auga en diferentes barrios dunha cidade, (c) a proporción de coches eléctricos, diésel e gasolina nun aparcamento.

2° PARTE

SISTEMAS INTELIXENTES

2ª Parte: Razoamento e aprendizaxe



GRUPO _____ MBOT RANGER _____

Nomes	Faltas	Observacións

Práctica 8 · Razoamento: árbores de decisión e semáforo intelixente

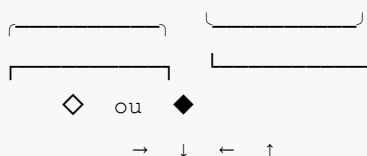
O que aprendemos nesta práctica

Nas prácticas anteriores o mBot usaba regras sinxelas: `se distancia < 20 → para`. Hoxe imos **formalizar** esas regras usando a ferramenta máis importante dos sistemas baseados en regras: a **árbore de decisión**.

Unha árbore de decisión é unha estrutura de **preguntas encadeadas** onde cada resposta leva a unha nova pregunta ou a unha acción final. Cada punto de decisión chámase **nodo**, cada posible resposta é unha **rama**, e cada acción final é unha **folia**.

Nos diagramas de fluxo, representamos as árbores de decisión con símbolos estándar. Antes de programar calquera exercicio nesta práctica, **debes debuxar o diagrama primeiro**.

Símbolos estándar dos diagramas de fluxo:



Óvalo (elipse): INICIO ou FIN do programa

Rectángulo: ACCIÓN ou instrución

Rombo: DECISIÓN (pregunta si/non)

Frecha: Dirección do fluxo

Os sistemas ADAS: a IA que xa vai no teu coche (obrigatorio desde 2022)

Desde xulio de 2022, o Regulamento Europeo 2019/2144 esixe que todos os coches novos vendidos na UE incorporen sistemas de asistencia á condución (ADAS). Non é opcional: se o teu familiar mercou un coche novo nos últimos anos, xa leva IA a bordo. Todos estes sistemas funcionan mediante árbores de decisión con regras explícitas programadas, exactamente como as que imos programar hoxe co mBot.

Sistema	Que fai	Sensor	Regra básica
AEB(Freo emerxencia autónomo)	Frea automaticamente ante un obstáculo	Radar + cámara	SE distancia < X E velocidade > Y → frear a fondo
ISA(Asistencia velocidade intelixente)	Detecta o límite e avisa ou limita	Cámara + GPS	SE velocidade > límite_mapa → avisar / reducir
LKA(Asistencia de carril)	Mantén o coche no carril	Cámara	SE desviación > X → corrixir dirección
DDW(Alerta somnolencia)	Detecta se o condutor adormece	Cámara interior	SE parpadeado < Y por minuto → alertar

¿Notaches algunha vez que o coche frease só, che avisase de que te saías do carril ou che alertase de que estabas canso? Iso é IA baseada en regras traballando en tempo real. **O teu mBot vai facer hoxe exactamente o mesmo, con menos sensores pero o mesmo principio.**

Exercicio 1 · O teu primeiro diagrama de fluxo

Obxectivo: aprender a debuxar diagramas de fluxo representando regras que xa programaches.

O programa do Exercicio 1 da Práctica 4 (ultrasonidos → para ou avanza) é a árbore de decisión máis sinxela que fixeches. **Debúxa como diagrama de fluxo** usando os símbolos da lenda:

[Debuxo do diagrama de fluxo: sensor ultrasonidos → para ou avanza + luces]

Responde

¿Cantos nodos de decisión ten o teu diagrama? _____ ¿Cantas follas (accións finais)? _____

Escribe en linguaxe natural (sen código) a regra do AEB do coche: «*Se a distancia ao obxecto diante é menor de X metros E a velocidade é maior de Y km/h, entón...*» ¿Cantos nodos de decisión necesitaría ese diagrama?




◆ Exercicio 1 completado — primeiro diagrama de fluxo debuxado

SELO

Exercicio 2 · O semáforo intelixente — diagrama guiado

Obxectivo: completar un diagrama de fluxo parcialmente feito e programar o semáforo básico.

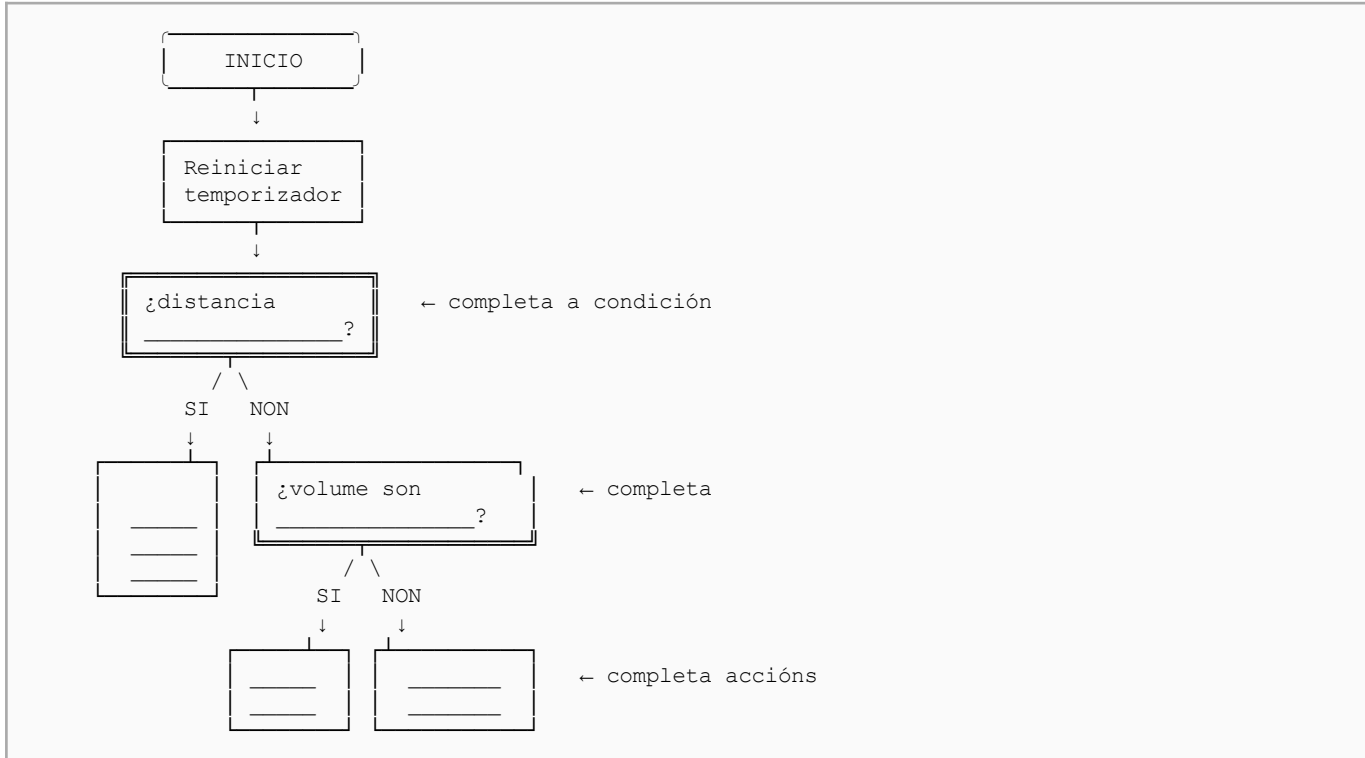
O mBot vai funcionar como un **semáforo intelixente** con dous sensores: **ultrasonidos** (detecta coches/obstáculos) e **son** (detecta peóns que apaluden para cruzar). O anel RGB e o buzzer son os actuadores.

Estado semáforo	Cor anel	Son buzzer	Motor
 VERMELLO (obstáculo próximo)	Vermello	Silencio	Para
 ÁMBAR (zona de precaución)	Amarelo	Silencio	Avanza lento (20%)
 VERDE (libre)	Verde	Silencio	Avanza normal (60%)

PASO PEONIL (peóns aplaudiron)	Azul parpadeante	Bip cada 0.5 seg	Para
---------------------------------------	------------------	------------------	------

Completa o diagrama de fluxo parcialmente feito

Os nodos de decisión e os rectángulos baleiros (marcados con ← *completa*) son os que tes que encher. Os óvalos de INICIO e FIN xa están postos.



📌 O diagrama ten dous nodos de decisión encadeados: primeiro comproba os ultrasonidos (obstáculo), despois comproba o son (peóns). Se non hai ningún dos dous, o semáforo está en verde.

Agora programa o semáforo no mBot

Usa o diagrama que completaches como guía. O pseudocódigo debe seguir exactamente a estrutura do diagrama:

```

DISPOSITIVO - para sempre:
  se «distancia ultrasonidos» < [ ] cm entón
    → VERMELLO: anel vermello + para
  se non, se «volume son» > [ ] entón
    → PASO PEONIL: anel azul + bip + para durante [ ] seg
  se non
    → VERDE ou ÁMBAR segundo distancia
    (decide ti o limiar para ámbar)
  
```

Limiars que escolliches:

Distancia VERMELLO < ___ cm	Distancia ÁMBAR < ___ cm	Volume son PEÓNS > ___
-----------------------------	--------------------------	------------------------

--	--	--

Responde

¿En que se parece o semáforo que programaches ao sistema **AEB do coche**? ¿En que se diferencia?

◆ Exercicio 2 completado — diagrama e semáforo básico funcionando



Exercicio 3 · Engadir prioridade de regras

Obxectivo: resolver un conflito de regras engadindo prioridade ao sistema.

O semáforo do Exercicio 2 ten un **problema**: ¿que pasa se hai un obstáculo próximo E un peón aplaudindo ao mesmo tempo? As dúas regras actívanse, pero o programa só pode facer unha cousa. Iso é un **conflito de regras**.

Os sistemas reais resolven isto con **prioridade**: algunhas regras teñen máis peso que outras. Nos semáforos reais, os vehículos de emerxencia teñen prioridade absoluta. No noso semáforo, temos que decidir:

¿**Que ten prioridade: o obstáculo (VERMELLO) ou o peón (PASO PEONIL)**? Discúteo co teu grupo antes de responder.

A nosa decisión de prioridade: _____

Razón: _____

Debuxa o diagrama de fluxo completo con prioridade

Agora tes que debuxar o diagrama **desde cero** (sen guía), incorporando a prioridade que decidistes. Usa todos os símbolos correctamente.

[Diagrama de fluxo completo con sistema de prioridade]

Adapta o programa ao novo diagrama

Modifica o código para que a prioridade funcione. Proba os dous casos simultáneos e comproba que o sistema responde correctamente.

¿O resultado é o que esperabas? ¿Houbo algún comportamento inesperado?

◆ **Eercicio 3 completado — conflito de regras resolto con prioridade**

SELO

Eercicio 4 · Semáforo con ciclo temporal

Obxectivo: engadir o factor tempo ao semáforo para que funcione en ciclos como un real.

Os semáforos reais non só reaccionan a sensores — tamén funcionan en **ciclos temporais**. O estado mínimo de verde dura polo menos 15 segundos (para que os peóns poidan cruzar). O ámbar sempre dura exactamente 3 segundos. Imos engadir estes tempos mínimos ao noso semáforo.

Debuxa o diagrama de fluxo do ciclo completo

O ciclo ten tres fases: VERMELLO (mentres hai obstáculo), ÁMBAR (2 seg fixos de transición), VERDE (mínimo 5 seg). Debúxao antes de programar.

[Diagrama do ciclo completo: VERMELLO → ÁMBAR (3 seg) → VERDE (≥15 seg)]



Usa o bloque «repite hasta que» para o estado VERDE: repite o comportamento verde ata que pase un obstáculo E leven polo menos 5 segundos. O «temporizador» do Ranger podes usalo para medir o tempo.

¿En que se parece este ciclo ao sistema **ISA** do coche (Intelligent Speed Assistance)? ¿Tamén ten estados e transicións entre eles?

◆ **Eercicio 4 completado — ciclo temporal funcionando**

SELO

Exercicio 5 · Experto: semáforo adaptativo

Obxectivo: deseñar e programar un semáforo que aprenda do tráfico. *Só cando tes os catro selos anteriores.*

Os semáforos intelixentes reais (sistema SCOOT usado en Madrid, Barcelona e A Coruña) axustan a duración do verde segundo o tráfico acumulado. Imos simular isto: o noso semáforo vai **contar cantos obstáculos pasaron** e axustar a duración do verde en consecuencia — máis tráfico, máis tempo de verde.

Deseña o sistema de adaptación:

Nivel de tráfico	Condición (obstáculos)	Duración VERDE
Baixo	<i>< 3 obstáculos por minuto</i>	
Medio	<i>3-6 obstáculos por minuto</i>	
Alto	<i>> 6 obstáculos por minuto</i>	

Debuxa o diagrama completo antes de programar

[Diagrama do semáforo adaptativo con tres niveis de tráfico]

¿Este sistema é realmente **intelixente** ou só segue regras fixas? ¿Que habería que cambiar para que **aprendese** de verdade do tráfico en lugar de seguir regras predefinidas?

◆ **Exercicio 5 completado — semáforo adaptativo deseñado e programado**



Borrador Wiki do grupo · Bloque Razoamento

Abre o Wiki do teu grupo en Moodle e engade a sección **RAZOAMENTO** do voso sistema intelixente.

- ✿ *¿Como decide o voso sistema que facer? ¿Usa regras explícitas, modelos aprendidos, ou unha combinación?*
- ✿ *Debuxade (ou describide) a árbore de decisión principal do voso sistema. ¿Cantos nodos ten?*
- ✿ *¿Que pasa cando dúas regras do voso sistema entran en conflito? ¿Ten sistema de prioridade?*
- ✿ *Comparade o razoamento do voso sistema co semáforo intelixente e co AEB do coche: ¿en que se parece? ¿en que se diferencia?*
- ✿ *Fontes (URL, nome do artigo, data de consulta).*

◆ Wiki · Sección Razoamento iniciada

SELO

Pregunta de peche · Sen mirar os apuntes

O sistema AEB dun coche frena automaticamente. O semáforo que programaches para cando hai peóns. Un médico decide a quen operar primeiro. Os tres toman decisións. **¿Que diferenza hai entre decidir con regras fixas programadas e decidir con xuízo propio? ¿Pode unha máquina ter xuízo propio?**
