Aparcamiento: entrada autorizada por código

Un teclado y una pantalla acompañan la validación de acceso antes de mover la barrera del aparcamiento.

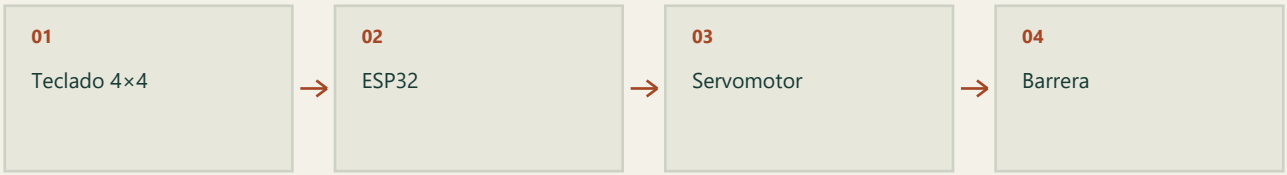
RESPONSABLE

Daniel Serrano Carrasco

ÁREA / CURSO

DPIOT / 2026-2027

EL RECORRIDO DE LA DECISIÓN



PREGUNTA DE DISEÑO

¿Qué distingue un intento de una autorización?

CRITERIO DE DECISIÓN

La barrera se gobierna por el resultado de validar el código recibido.

ENSAYO PROPUESTO

Comparar dos intentos y comprobar que un rechazo no produce apertura.

Memoria de planteamiento. Las conexiones y los importes son propuestas de trabajo; la construcción y sus resultados requieren comprobación en el aula.

Mapa del documento

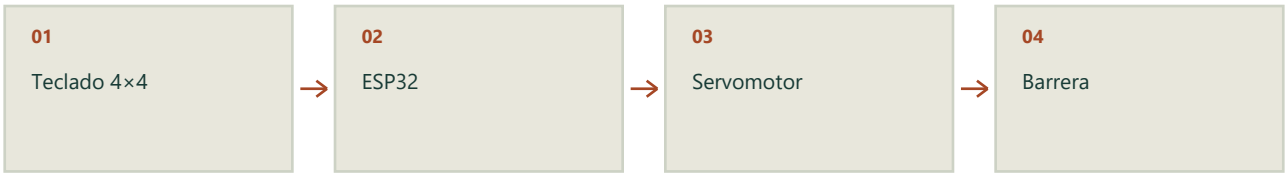
El texto sigue la decisión del sistema antes de detallar su construcción. Las pruebas incluidas son criterios de trabajo: no se presentan como ensayos ya superados.

01 / Finalidad

Hacer que la barrera permanezca bloqueada hasta recibir por teclado una clave que el programa considere autorizada.

02 / Cómo se conectan las funciones

El flujo parte del código de teclado. El ESP32 toma la decisión y actualiza tanto la información de pantalla como la posición del servo.



Salida complementaria: Visualización Pantalla LCD Muestra el resultado

03 / Recorrido de una decisión

1. Introducir contraseña
2. Comprobar contraseña
3. Mostrar acceso permitido o denegado
4. Abrir barrera si es correcta
5. Cerrar barrera y esperar

04 / Recursos y estimación de coste

Elementos previstos

- 1 ESP32.
- 1 teclado matricial 4x4.
- 1 pantalla LCD 16x2 I2C.
- 1 servomotor.
- 1 sensor infrarrojo de proximidad.
- Protoboard, cables y material para la maqueta.

Estimación inicial de recursos

Cantidad	Recurso previsto	Estimación inicial
1	ESP32	8 €
1	Teclado 4x4	5 €
1	LCD 16x2 I2C	6 €
1	Servomotor	6 €
1	Sensor infrarrojo	3 €
1	Cableado y maqueta	12 €

Total estimado: 40 €

05 / Propuesta y límites del montaje

Cada pulsación construye la contraseña. El controlador la compara y comunica la aceptación o el rechazo en la LCD. Solo en el primer caso el servomotor levanta la barrera.

Incluido en esta versión

- Introducción de contraseña mediante teclado.
- Comprobación de la contraseña con el ESP32.
- Apertura y cierre de la barrera mediante servomotor.
- Comunicar la autorización o el rechazo mediante mensajes en la LCD.

Aspectos que no cubre esta versión

- Reconocimiento de matrículas.
- Pago automático.
- Control mediante aplicación móvil.

06 / Qué permite aprender

La propuesta combina adquisición de una clave, comunicación de su resultado y movimiento de una barrera, tres tareas coordinadas por el ESP32.

07 / Condiciones de funcionamiento

Comportamientos esperados

1. Leer la contraseña introducida en el teclado.
2. Verificar la coincidencia con el código definido dentro del programa.

3. Abrir la barrera si la contraseña es correcta.
4. No habilitar el movimiento de apertura tras una validación negativa.
5. Mostrar el estado del acceso en la pantalla LCD.

Condiciones de construcción

- Programación mediante Arduino IDE para ESP32.
- Alimentación de baja tensión.
- Montaje inicial en protoboard.

08 / Conexiones propuestas: pendientes de verificar

Elemento	Interfaz	Conexión propuesta	Criterio de montaje
Teclado 4x4	Digital	GPIO varios	Filas y columnas
Servomotor	PWM	GPIO 18	Control de barrera
LCD I2C	SDA / SCL	GPIO 21 / 22	Mensajes de acceso
Sensor IR	Digital	GPIO 32	Detección de paso

09 / Secuencia de desarrollo

Etapa	Trabajo previsto	Dedicación orientativa	Situación prevista
1. Anteproyecto	Documento y presupuesto	1 sesión	Planificado: sin validación física
2. Montaje	Montar teclado, LCD y barrera	2 sesiones	Planificado: sin validación física
3. Firmware	Programar contraseña y control	2 sesiones	Planificado: sin validación física
4. Pruebas	Comprobar acceso correcto e incorrecto	1 sesión	Planificado: sin validación física
5. Memoria	Fotos, código y conclusiones	1 sesión	Planificado: sin validación física

10 / Qué tendría que demostrar el prototipo

- La secuencia capturada coincide con las teclas utilizadas.
- La contraseña correcta abre la barrera.
- Una contraseña incorrecta mantiene la barrera cerrada.

- El estado presentado en la LCD corresponde a la decisión del controlador.
- Finalizado el paso, el acceso recupera su posición de cierre.

11 / Puntos que deben revisarse antes de alimentar

Punto a revisar	Qué puede ocurrir	Comprobación preventiva
Contraseña incorrecta en el programa	Acceso bloqueado	Comprobar el código antes de las pruebas.
Servomotor forzado	Daño en la barrera	Limitar los ángulos de movimiento.
Sensor mal ajustado	Cierre incorrecto	Calibrar la distancia de detección.

Validación y procedencia

Esta hoja separa lo que el diseño propone de lo que deberá demostrarse en la maqueta. Las casillas de situación no acreditan pruebas realizadas.

Comprobación	Evidencia que debe incorporarse	Situación
Conexiones y alimentación	Esquema contrastado con los componentes utilizados	Pendiente de registrar
Programa del demostrador	Versión compilada y cargada en la placa	Pendiente de registrar
Comportamiento previsto	Tabla de entradas y respuestas observadas	Pendiente de registrar
Incidencias	Descripción del fallo y de la corrección aplicada	Pendiente de registrar
Cierre de la entrega	Fotografías y conclusiones de las pruebas	Pendiente de registrar

Referencias y alcance de esta memoria

Consigna de DPIOT: presentar diez anteproyectos básicos con título, objetivo, material y funcionamiento. Esta memoria desarrolla una de esas ideas y acompaña a su página web.

Ideas de partida: resumen facilitado del vídeo Top 10 Arduino Projects 2026, del canal Simple Circuits, y colección documental recuperada durante la preparación del sitio. La redacción de esta edición se corresponde con el desarrollo revisado de ORIGINAL_DANIEL.

Los importes, tiempos y conexiones son estimaciones o asignaciones de proyecto, no presupuestos comerciales actualizados ni un cableado validado. El programa y el montaje deben contrastarse con la placa y las instrucciones del aula.

La entrega final debe incorporar el programa realmente utilizado, las evidencias del montaje y los resultados observados. Los modelos de navegador explican las respuestas previstas y no sustituyen esas pruebas.