



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



FIME

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA

Lab. Controladores y Microcontroladores

Docente: Ing. Héctor Hugo Flores Moreno

Proyecto: Pez Comeplastico

Documentación

Nombre	Matrícula	Carrera
Jaime Miguel Gómez Casas	1976345	ITS
Rodolfo aarón Gallegos Maldonado	1981050	ITS

Frecuencia: Jueves Horas: N5-N6

Grupo: 409 Salón: 9103

Índice

¿Qué hace el proyecto?	3
¿Qué problema resuelve el proyecto?	3
Propósito	3
Contexto	4
Alcance	4
Funcionamiento Interno	4
Estructura de carpetas.	5
Tecnologías usadas	6
Comunicación entre módulos	7
Decisiones técnicas	8
Diagrama de Flujo	9
Diagrama de Secuencia	10
¿Cómo se usa o contribuye alguien más?	11
Cómo poner en marcha el proyecto	12
Requisitos previos	12
Instalación de dependencias	12
Ejecutar el entorno local	13
Ejecución de pruebas o validaciones	15
Contribuir	15
Resumen técnico del sistema	15
Guía paso a paso de instalación	17
FAQ – Preguntas Frecuentes	18
Anexos	20

¿Qué hace el proyecto?

El proyecto “Pez come plástico” es un sistema mecatrónico simulado en Proteus, controlado por un Arduino UNO (ATmega328P), que imita el movimiento natural de un pez real utilizando servomotores para accionar sus aletas, cola y boca.

El circuito permite que el pez nade de forma coordinada y, mediante un botón de control, pueda activarse o detenerse. Cuando está en funcionamiento, los servomotores reproducen un movimiento fluido:

- Las aletas se mueven en direcciones opuestas, generando impulso.
- La cola oscila para simular la propulsión.
- La boca se abre y cierra, representando la acción de “comer plástico” u objetos flotantes en el agua.

¿Qué problema resuelve el proyecto?

El proyecto busca concientizar sobre la contaminación por plásticos en los océanos y demostrar una solución robótica inspirada en la naturaleza.

En el contexto educativo y de investigación, este pez simulado puede representar:

- Cómo un robot bioinspirado puede detectar o recolectar desechos flotantes.
- Una herramienta de enseñanza para comprender control de servomotores, programación en Arduino y simulación en Proteus.
- Un modelo conceptual para futuros robots acuáticos que ayuden a limpiar cuerpos de agua contaminados con plásticos.

Propósito

El propósito del proyecto “Pez come plástico” es diseñar y simular un prototipo robótico bioinspirado que represente el comportamiento de un pez capaz de moverse de forma autónoma y realizar movimientos similares a la acción de “comer” plástico u otros desechos presentes en el agua. Este proyecto busca unir la tecnología y la conciencia ambiental, mostrando cómo la robótica puede contribuir al cuidado de los ecosistemas acuáticos.

Contexto

El proyecto se desarrolla como una propuesta educativa y de sensibilización ambiental, utilizando la plataforma Arduino y el entorno de simulación Proteus. Surge ante la creciente problemática mundial de la contaminación por plásticos en ríos, lagos y océanos, que afecta gravemente la vida marina y el equilibrio ecológico. A través de la simulación, se pretende demostrar de manera didáctica cómo un sistema electrónico puede imitar organismos naturales y utilizarse como base para el desarrollo de futuros robots recolectores de residuos.

Alcance

Comprende el diseño electrónico, la programación del control de movimiento y la simulación completa del pez dentro de Proteus. Incluye la implementación de cuatro servomotores (dos para las aletas, uno para la cola y uno para la boca), un botón de control que permite activar o detener el movimiento, y una fuente externa de alimentación para garantizar la estabilidad del sistema. El proyecto no abarca aún el diseño físico o acuático real del pez, pero sienta las bases conceptuales y técnicas para su futura materialización en prototipos funcionales o educativos.

Funcionamiento Interno

Internamente, el proyecto “Pez come plástico” se basa en un sistema electrónico controlado por un microcontrolador ATmega328P, integrado en la placa Arduino UNO, que coordina los movimientos del pez mediante señales PWM (modulación por ancho de pulso) enviadas a cuatro servomotores. Cada uno de estos servos representa una parte del cuerpo del pez:

- **Aleta izquierda**
- **Aleta derecha**
- **Cola**
- **Boca**

El comportamiento general del pez se controla a través de un botón pulsador, conectado a un pin digital de entrada del Arduino. Este botón funciona como un interruptor de encendido y apagado (toggle): cada vez que se presiona, cambia el estado del pez entre “activo” (en movimiento) y “reposo” (quieto).

Cuando el pez está activo, el programa ejecuta una secuencia de movimientos cíclicos programados en el código:

- Las aletas se mueven en sentidos opuestos, generando un efecto visual de impulso hacia adelante.
- La cola oscila de un lado a otro, imitando el movimiento de propulsión característico de los peces reales.
- La boca se abre y cierra suavemente, simulando la acción de “comer” o recolectar objetos flotantes.

Todo este comportamiento se logra mediante la función `Servo.write()` de la biblioteca `Servo.h`, que permite controlar con precisión el ángulo de cada servo. Las secuencias se repiten continuamente mientras el estado del pez sea activo, y se detienen automáticamente cuando el botón se vuelve a presionar.

Para evitar sobrecargas eléctricas, los servos se alimentan desde una fuente externa de 5 V, mientras que el Arduino únicamente envía las señales de control. Ambas fuentes (Arduino y servos) comparten una tierra común (GND) para asegurar la correcta referencia de voltaje.

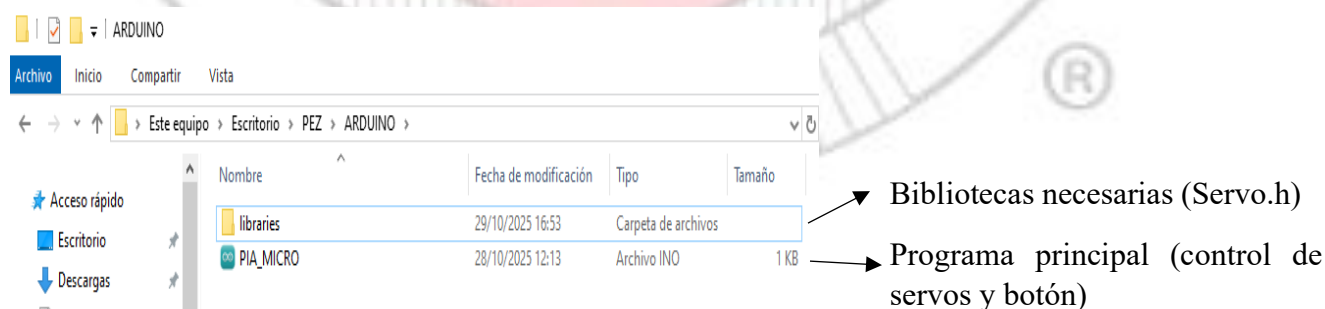
Estructura de carpetas.

El proyecto “Pez come plástico” se organiza en una estructura de carpetas que facilita la gestión, documentación y simulación del sistema. Esta estructura permite mantener separadas las partes de programación, simulación, diagramas y documentación, asegurando un trabajo ordenado y reproducible.

Carpeta Principal “Pez”



Carpeta Arduino



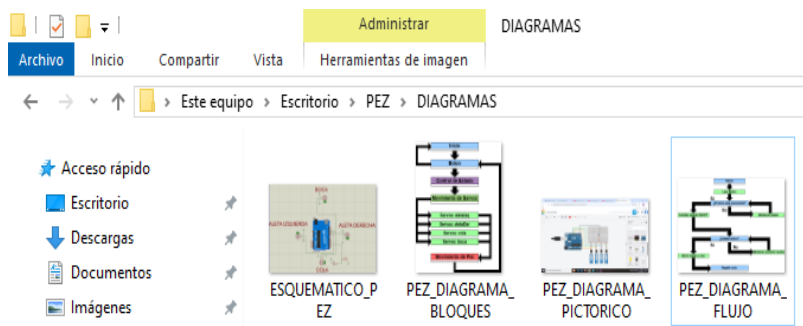
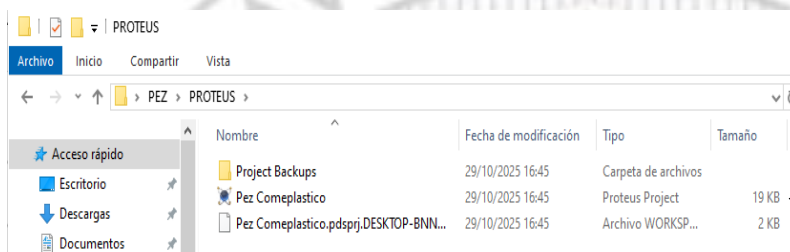


Diagrama de bloques general, representación visual del montaje, captura del diagrama esquemático y diagrama de flujo.

Carpeta “Proteus”



Proyecto principal de Proteus

Tecnologías usadas.

Las tecnologías utilizadas para la elaboración de este proyecto son las siguientes:

Tecnología	Uso principal	Descripción
Arduino IDE	Programación	Entorno de desarrollo para escribir, cargar y depurar el código (.ino) que controla los servos y el botón.
Proteus	Simulación y diseño esquemático	Permite crear el circuito electrónico del pez, simular el comportamiento de los servos y visualizar el diagrama esquemático.
Tinkercad	Diagrama pictórico y visualización didáctica	Herramienta online usada para generar un diagrama visual (pictórico) que muestra las conexiones físicas de los componentes.
Draw.io (Diagrams.net)	Diagramas de bloques y arquitectura	Utilizada para diseñar diagramas de bloques funcionales, flujo de información y arquitectura general del sistema.
Microsoft Word	Documentación técnica	Elaboración del informe del proyecto con introducción, desarrollo, resultados y conclusiones.

Tecnología	Uso principal	Descripción
Servo.h (Librería de Arduino)	Control de movimiento	Biblioteca incluida en el código para generar señales PWM que controlan los servomotores.
Fuente externa 5V / 2A (hardware real o simulada)	Alimentación	Proporciona corriente estable para los servos sin sobrecargar el Arduino.

Comunicación entre módulos

El proyecto *Pez Come Plástico* se compone de varios módulos físicos y lógicos que se comunican de manera sencilla, principalmente a través de señales eléctricas digitales y analógicas controladas por el microcontrolador ATmega328P (presente en el Arduino UNO).

Los módulos y su interacción son los siguientes:

Módulo	Función principal	Comunicación / Interfaz
Módulo de control (Arduino UNO)	Actúa como el cerebro del sistema; interpreta la señal del botón y genera pulsos PWM para los servomotores.	Comunicación digital con el botón y PWM con los servos.
Módulo sensorial (Botón)	Permite activar o desactivar el movimiento del pez con una sola pulsación.	Señal digital (HIGH/LOW) hacia el pin de entrada del Arduino.
Módulo de movimiento (Servomotores)	Tres servos controlan las aletas izquierdas, derecha, boca y la cola, generando un movimiento coordinado.	Señales PWM desde los pines 9, 10, 11 y 12 del Arduino.
Módulo de alimentación	Fuente de energía para los servos y el microcontrolador.	Distribuye 5V y GND a cada componente (con fuente externa para estabilidad).

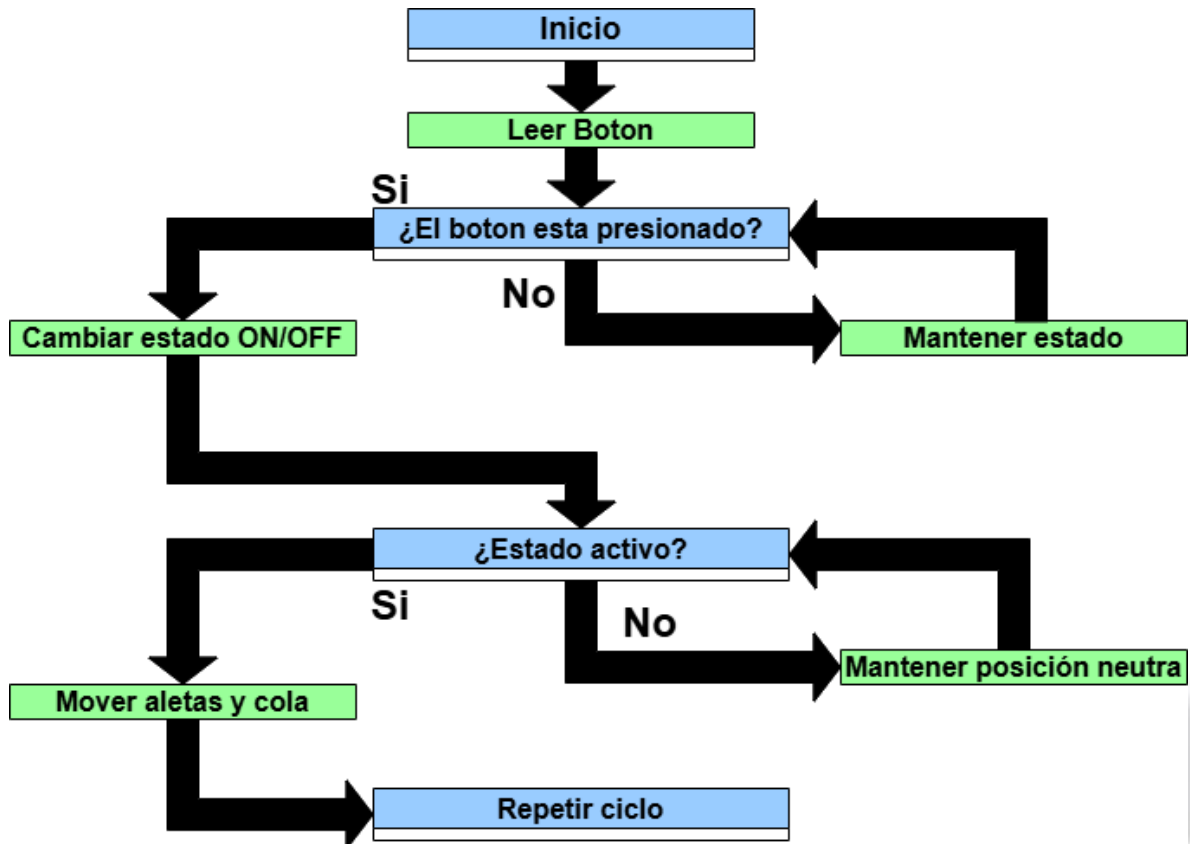
Decisiones técnicas

Durante el desarrollo del proyecto se tomaron diversas decisiones técnicas para optimizar la funcionalidad y la viabilidad de la simulación:

1. Uso de Arduino UNO (ATmega328P):
Elegido por su simplicidad, disponibilidad en Proteus y facilidad para generar señales PWM precisas.
2. Librería Servo.h:
Permite un control suave y sincronizado del movimiento de los servomotores sin necesidad de cálculos manuales de temporización.
3. Fuente de alimentación externa:
Decisión clave para evitar sobrecargar los pines de 5V del Arduino, ya que varios servos pueden requerir más corriente de la que el microcontrolador puede ofrecer.
4. Diseño modular:
Cada parte (control, sensores, actuadores) se conecta de manera independiente, lo que facilita la ampliación futura del sistema (por ejemplo, agregar sensores o comunicación inalámbrica).
5. Simulación combinada (Proteus + Tinkercad):
Proteus se usó para simular la electrónica real, mientras que Tinkercad permitió un enfoque más visual (montaje pictórico) para presentaciones educativas.
6. Botón con cambio de estado (toggle):
El sistema no depende de mantener presionado el botón, sino que cambia entre movimiento activo e inactivo con cada pulsación, mejorando la usabilidad.

Diagrama de Flujo

El diagrama de flujo es el siguiente:



Nivel 1 – Entrada

Botón de activación: Envía una señal digital al microcontrolador (Arduino UNO).

Nivel 2 – Procesamiento (Lógica de control)

Arduino UNO (ATmega328P):

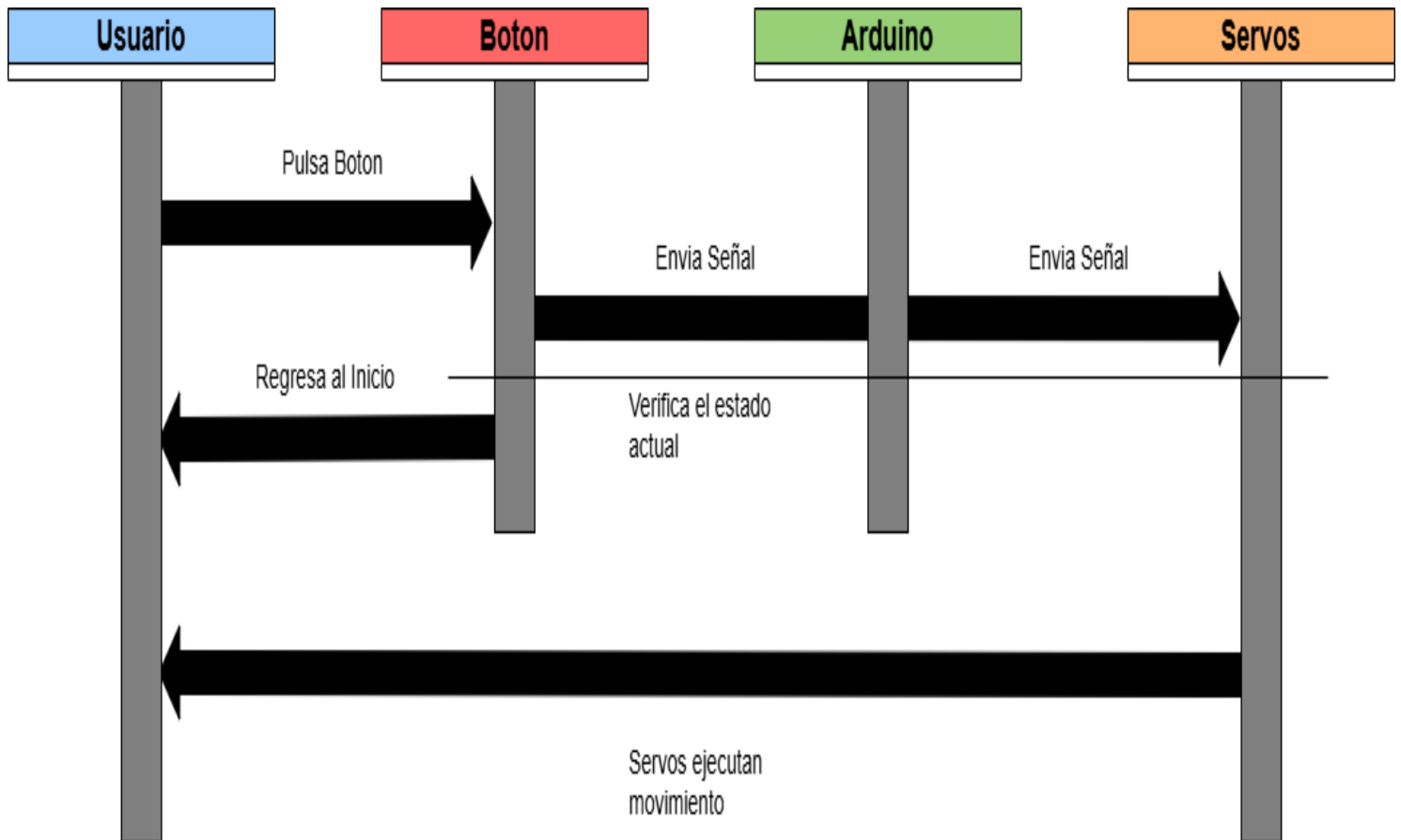
- Lee el estado del botón.
- Decide si activar o detener el movimiento.
- Genera señales PWM para los servos según las rutinas de movimiento.

Nivel 3 – Salida (Actuadores y movimiento)

- Servomotor Aleta Izquierda (Pin 9): Movimiento ascendente/descendente.
- Servomotor Aleta Derecha (Pin 10): Movimiento inverso al izquierdo.
- Servomotor Cola (Pin 11): Movimiento oscilante para propulsión
- Servomotor Boca (Pin 12): Movimiento para la boca.

Diagrama de Secuencia

El diagrama de secuencia es el siguiente:



¿Cómo se usa o contribuye alguien más?

Uso del proyecto por otras personas

Cualquier usuario puede:

- Simular el funcionamiento del pez en Proteus o Tinkercad, utilizando los archivos del proyecto.
- Cargar el código Arduino (.ino) en un microcontrolador ATmega328P o una placa Arduino UNO para replicar el comportamiento físico.
- Aprender conceptos sobre control de servomotores, señales PWM y programación de sistemas embebidos.
- Usar el código base como punto de partida para proyectos de robótica, biotecnología o educación ambiental.

Contribución técnica de otros participantes

El proyecto permite la colaboración en varios niveles:

Tipo de contribución	Ejemplo de aportes posibles
Hardware	Incorporar sensores (ultrasónicos o de color) para detectar plásticos.
Software	Mejorar el algoritmo de movimiento o agregar modos de comportamiento (nado, reposo, detección).
Diseño	Crear una carcasa 3D del pez o mejorar el aspecto visual en Tinkercad.
Simulación	Optimizar el circuito en Proteus o agregar un módulo de alimentación realista.
Documentación	Traducir, documentar o crear tutoriales para nuevos usuarios.

Herramientas para la colaboración

- GitHub o Google Drive: Repositorio donde se pueden subir versiones del código, diagramas y simulaciones.
- Draw.io: Permite editar conjuntamente los diagramas de arquitectura y flujo.
- Proteus y Tinkercad: Facilitan la simulación colaborativa entre estudiantes (cada quien puede modificar su parte).
- Arduino IDE: Los participantes pueden probar variaciones del código localmente antes de subirlo al repositorio.

Cómo poner en marcha el proyecto

El siguiente procedimiento permite instalar, ejecutar, simular y modificar el proyecto completo, incluyendo el código Arduino, los diagramas y las simulaciones en Proteus o Tinkercad.

El objetivo es que cualquier usuario pueda clonar el repositorio y replicar el comportamiento del pez sin configuración adicional.

Requisitos previos

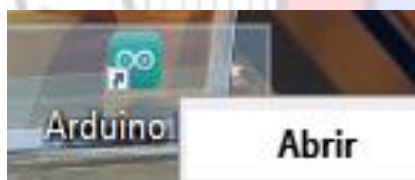
Antes de comenzar con la ejecución del proyecto, se debe de asegurar de tener instaladas las siguientes herramientas:

Herramienta	Versión recomendada	Uso
Arduino IDE	2.x o superior	Programar y cargar el código al microcontrolador.
Proteus 8 Professional	8.13 o superior	Simulación electrónica (circuito y movimiento).

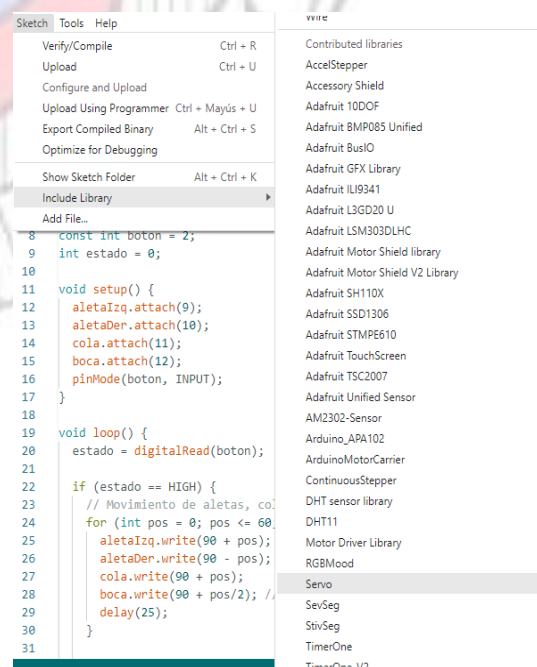
Instalación de dependencias

El proyecto utiliza únicamente librerías estándar de Arduino, por lo que no requiere dependencias externas complejas.

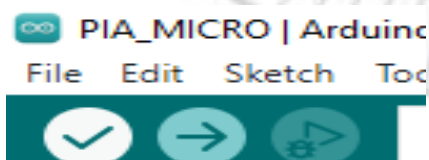
1. Abrir Arduino IDE.



2. En el menú superior, seleccionar: Programa → Incluir Librería → Servo



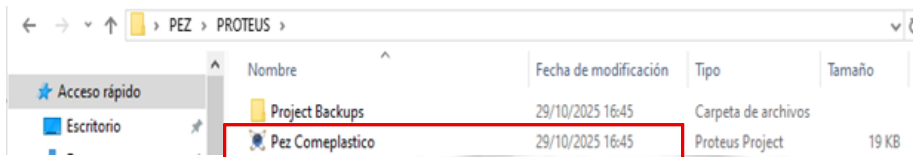
3. Verificar que el código compile correctamente con: Verificar (✓) o presionando Ctrl + R



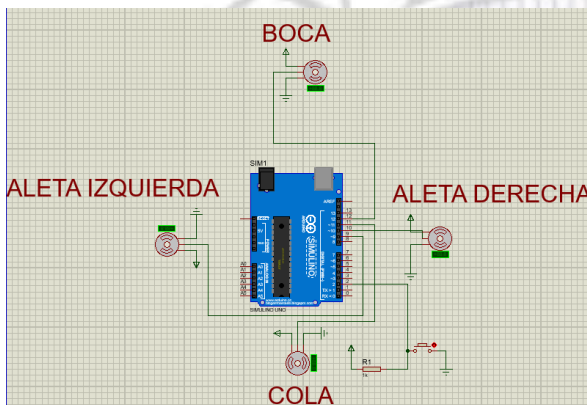
Ejecutar el entorno local

Opción A: Simulación en Proteus

1. Abrir el archivo:



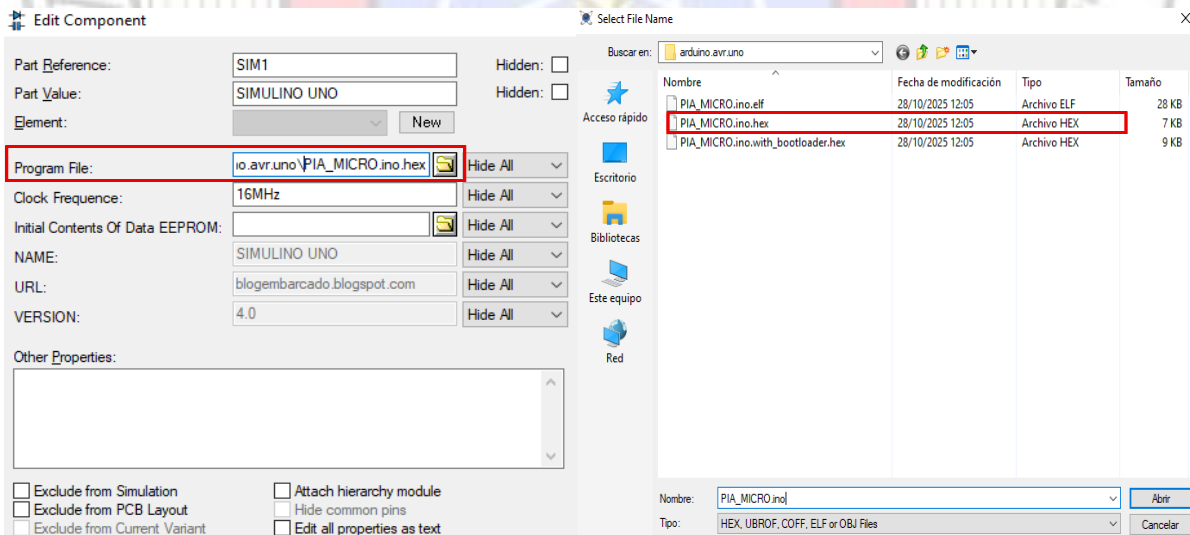
2. Verificar las conexiones de los servos y del botón según el esquemático.



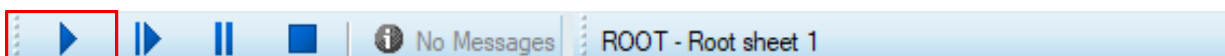
3. Asociar el archivo .hex generado por Arduino al microcontrolador en Proteus:

Hacer doble clic en el componente ATmega328P.

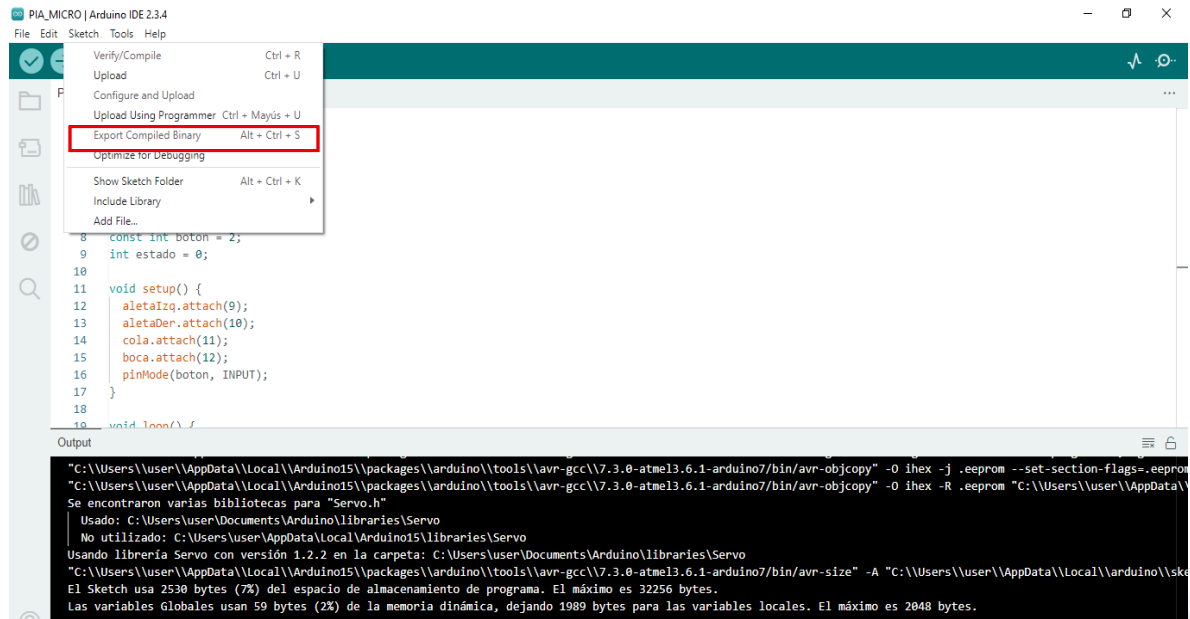
En la pestaña *Program File*, seleccionar el archivo .hex exportado desde Arduino.



4. Hacer clic en Play para iniciar la simulación.



Nota: Para obtener el archivo. hex de Arduino para cargarlo en el microcontrolador de proteus, es necesario ir a Programa → Export Compile Binary, luego iniciará la exportación y una vez finalizada se tendrá el archivo. hex en la ubicación donde se encuentra el archivo. ino del proyecto.



Opción B: Ejecución física con Arduino UNO

1. Conectar los servos y el botón a los pines:
 - Aleta izquierda → pin 9
 - Aleta derecha → pin 10
 - Cola → pin 11
 - Boca → pin 12
 - Botón → pin 2 (con resistencia pull-down o interna activada)
2. Cargar el programa pez comeplastico.ino desde la carpeta Arduino/.
3. Alimentar los servos con una fuente externa de 5V (compartiendo GND con el Arduino).
4. Presionar el botón físico para alternar entre movimiento activo y reposo.

Ejecución de pruebas o validaciones

No se utilizan pruebas automatizadas, pero se pueden hacer verificaciones manuales:

Prueba	Acción esperada
Presionar el botón una vez	Inicia el movimiento coordinado de aletas y cola.
Presionar el botón otra vez	Detiene el movimiento, dejando al pez en posición neutral.
Desconectar la fuente o reiniciar Arduino	Sistema inicia en modo inactivo.

Contribuir

Si se desea mejorar este proyecto:

1. Hacer un *fork* del repositorio.
2. Crear una nueva rama: *feature/nueva-funcion*.
3. Realizar tus cambios y súbelos:

`git add .`

`git commit -m "Agregué sensor ultrasónico"`

`git push origin feature/nueva-funcion`

4. Enviar un *Pull Request*.

Resumen técnico del sistema

El proyecto Pez Come Plástico es un sistema educativo y ambiental que simula un pez robótico capaz de moverse y “comer” residuos plásticos mediante servomotores controlados por un microcontrolador Arduino UNO (ATmega328P).

Componentes principales

Componente	Función
Arduino UNO (ATmega328P)	Controla la lógica del movimiento y recibe señales del botón pulsador.
Servomotor Aleta Izquierda	Genera movimiento oscilante hacia arriba y abajo.
Servomotor Aleta Derecha	Genera movimiento inverso a la aleta izquierda para simular nado.
Servomotor Cola	Proporciona propulsión con movimiento lateral.
Servomotor Boca	Genera movimiento para el uso de la boca.
Fuente de alimentación 5V	Suministra energía a los servos y al microcontrolador.
Pulsador (botón)	Permite alternar entre movimiento activo y reposo.

Arquitectura del sistema

1. Entrada: Pulsador que indica al sistema si debe activarse o detenerse.
2. Procesamiento: Arduino UNO interpreta el estado del botón y genera señales PWM para los servos.
3. Salida: Servomotores ejecutan movimientos coordinados (aletas y cola), simulando el nado y el comportamiento del pez.

Funcionamiento

- Cada vez que se presiona el botón, el sistema cambia entre modo activo (pez nada) y modo reposo (posición neutral).
- El Arduino envía señales PWM a los servos para controlar sus ángulos, generando un movimiento fluido y sincronizado.
- La simulación puede realizarse en Proteus (electrónica) o ejecutarse en hardware real con servos y fuente externa de 5V.

Decisiones técnicas clave

- Uso de librería Servo.h para control preciso de servomotores.
- Diseño modular que permite agregar sensores adicionales (como ultrasónicos) en el futuro.
- Fuente de alimentación externa para evitar sobrecarga del Arduino.
- Botón tipo toggle para alternar entre movimiento y reposo con una sola pulsación.

Propósito educativo y ambiental

El sistema permite:

- Comprender control de servomotores y señales PWM.
- Simular un comportamiento biológico en robótica.
- Fomentar la conciencia ambiental mediante una demostración interactiva de cómo un pez podría “comer” plásticos.

Guía paso a paso de instalación

Software:

- Arduino IDE 2.x o superior
- Proteus 8.13 o superior

Hardware:

- Arduino UNO (ATmega328P)
- 4 × Servomotores SG90
- 1 × Pulsador (botón)
- Fuente de alimentación 5V / 2A (recomendado)
- Cables Dupont

Clonar el repositorio

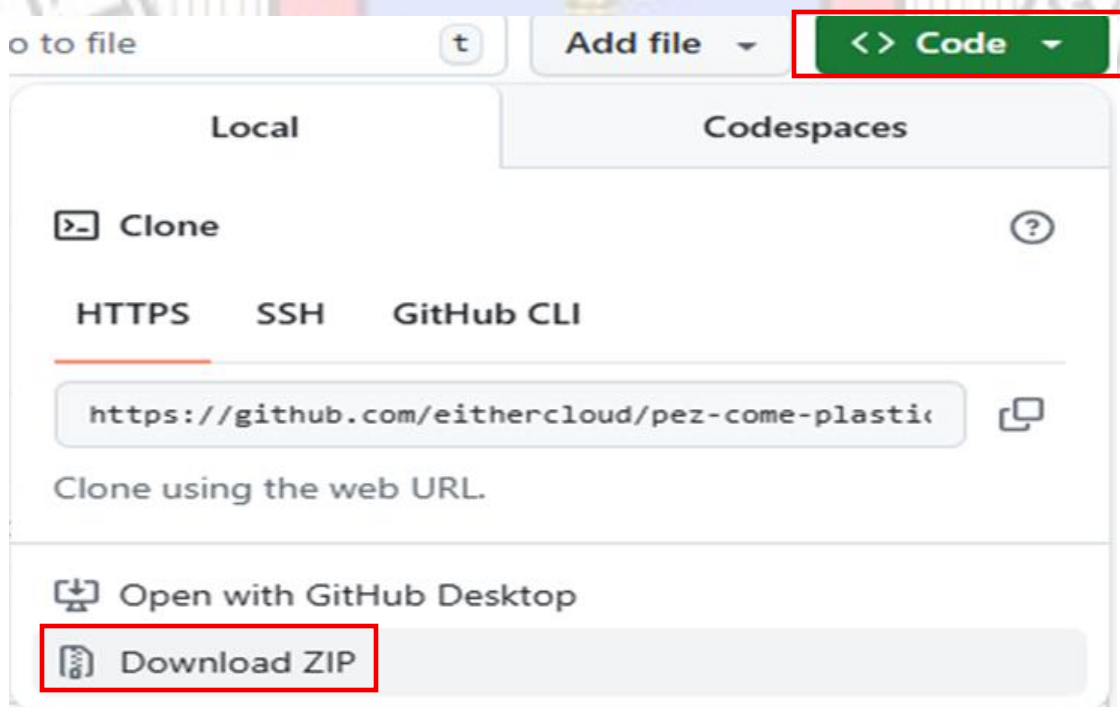
Abrir la terminal o Git Bash en la computadora y escribir:

```
git clone https://github.com/TU_USUARIO/pez-come-plastico.git  
cd pez-come-plastico
```

Descargar el repositorio

Repositorio: <https://github.com/eithercloud/pez-come-plastico>

Hacer click en code → Download ZIP para descargar el proyecto



FAQ – Preguntas Frecuentes

¿Qué hace el proyecto Pez Come Plástico?

El proyecto simula un pez robótico que mueve sus aletas y cola para “nadar” y puede interactuar con residuos plásticos flotantes. Se controla mediante un botón y un Arduino UNO.

¿Qué componentes se necesitan para replicarlo físicamente?

- 1 × Arduino UNO (ATmega328P)
 - 4 × Servomotores SG90
 - 1 × Pulsador (botón)
 - Fuente de 5V (2A recomendada)
 - Cables Dupont
- Opcional: sensores adicionales para mejorar la interacción con el plástico.

¿Se puede simular el proyecto sin tener hardware físico?

Sí, se puede simular completamente en Proteus para electrónica y Tinkercad para el diagrama pictórico. Solo se necesita el archivo .hex generado desde Arduino IDE.

¿Cómo funciona el botón?

El botón funciona como un toggle:

- Primera pulsación → activa el movimiento del pez
- Segunda pulsación → detiene el movimiento

¿Qué hace cada servomotor?

- Aleta izquierda y derecha → generan el movimiento de nado lateral.
- Cola → proporciona propulsión y dirección.
- Boca → proporciona movimiento en la parte delantera.

¿Se puede agregar más funcionalidad al proyecto?

Sí, el sistema es modular: se pueden añadir sensores (ultrasónicos, de color o infrarrojos) o modificar el código para cambiar el comportamiento del pez.

¿Qué software se necesita para trabajar con el proyecto?

- Arduino IDE para cargar el código y generar el archivo .hex.

- Proteus para simulación electrónica.

¿Cómo se puede contribuir al proyecto?

1. Hacer un *fork* del repositorio.
2. Crear una rama nueva (feature/nombre-funcion).
3. Realizar tus cambios (código, diagramas o documentación).
4. Enviar un Pull Request para revisión.

¿Se necesita una fuente externa para alimentar los servos?

Sí, se recomienda una fuente de 5V y al menos 2A para evitar sobrecargar el pin 5V del Arduino.

¿El proyecto es educativo o también puede usarse como prototipo funcional?

Principalmente es educativo, pero se puede extender con sensores y mejoras para convertirse en un prototipo de demostración ambiental.

Anexos

Link del Proyecto: <https://github.com/eithercloud/pez-come-plastico>

Readme.md: <https://github.com/eithercloud/pez-come-plastico/blob/main/README.md>

Readme

README

Pez Come Plástico

Descripción del proyecto:

Pez Come Plástico es un proyecto educativo de robótica ambiental que simula un pez capaz de moverse mediante servomotores y "comer" residuos plásticos flotantes. Su objetivo es crear conciencia sobre la contaminación marina y demostrar cómo la tecnología puede ayudar a reducir los efectos del plástico en los ecosistemas acuáticos. El sistema se basa en un Arduino UNO (ATmega328P), tres servomotores que controlan las aletas y la cola, y un pulsador que activa el movimiento del pez. El comportamiento del pez se puede simular en Proteus o probar físicamente con el montaje real.

Requisitos e instalación

Requisitos de hardware

- *1 x Arduino UNO (ATmega328P)
- *4 x Servomotores SG90
- *1 x Pulsador (botón)
- *1 x Fuente de 5V (2A recomendada)
- *Cables Dupont macho-macho / macho-hembra

Requisitos de software

- *Arduino IDE 2.x o superior
- *Proteus 8.13 o superior
- *Librería Servo.h (incluida por defecto en Arduino IDE)

Instalación

- 1.- Clona el repositorio del proyecto:
git clone <https://github.com/tuusuario/pez-come-plastico.git>
- 2.- Abre la carpeta del proyecto:
cd pez-come-plastico/Código_Arduino
- 3.- Abre el archivo Pez Comeplastico.ino en Arduino IDE.
- 4.- Verifica la compilación (Ctrl + R) y genera el archivo .hex para usarlo en Proteus.

Cómo usarlo

Simulación en Proteus:

- 1.- Abre el archivo: Pez Comeplastico.pdsprj
- 2.- Asocia el archivo .hex al microcontrolador ATmega328P.
- 3.- Inicia la simulación y presiona el botón para alternar entre:

- *Modo activo: el pez se mueve simulando que nada.
- *Modo reposo: el pez se detiene.

Uso en hardware real

- 1.- Conecta los componentes así:
 - *Servo aleta izquierda → Pin 9
 - *Servo aleta derecha → Pin 10
 - *Servo cola → Pin 11
 - *Servo boca → Pin 12
 - *Pulsador → Pin 2
 - *GND común para todos los servos y botón
- 2.- Carga el código desde Arduino IDE.
- 3.- Alimenta el sistema con 5V (fuente externa o USB).
- 4.- Presiona el botón para activar el movimiento del pez.

Estructura del proyecto

```
PEZ/
|
|--- ARDUINO/
|    |--- PIA_MICRO.ino
|
|--- PROTEUS/
|    |--- Pez_Proteus.pdsprj
|
|--- DIAGRAMAS/
|    |--- ESQUEMATICO_PEZ.png
|    |--- PEZ_DIAGRAMA_BLOQUES.png
|    |--- PEZ_DIAGRAMA_PICTORICO.png
|    |--- PEZ_DIAGRAMA_ARQUITECTURA.png
|    |--- PEZ_DIAGRAMA_SECUENCIA.png
|    |--- PEZ_DIAGRAMA_FLUJO.png
|
|--- DOCUMENTACION/
|    |--- DOCUMENTACION.pdf
|    |--- DOCUMENTACION.docx
|
|--- README.md
```


Carpetas - github

eithercloud / pez-come-plastico

Q

Type / to search

+

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

pez-come-plastico

Public

Pin

Watch 0

Fork 0

Star 0

main 1 Branch 0 Tags

Go to file

Add file

<> Code

About

eithercloud

Add files via upload

6e75bf3 · now

4 Commits

ARDUINO	Proyecto Pez Come Plástico	1 hour ago
DIAGRAMAS	Add files via upload	now
PROTEUS	Proyecto Pez Come Plástico	1 hour ago
README.md	Update documentation files in README	50 minutes ago

README

Pez Come Plástico

Descripción del proyecto:

Pez Come Plástico es un proyecto educativo de robótica ambiental que simula un pez capaz de moverse mediante servomotores y "comer" residuos plásticos flotantes. Su objetivo es crear conciencia sobre la contaminación marina y demostrar cómo la tecnología puede ayudar a reducir los efectos del plástico en los ecosistemas acuáticos. El sistema se basa en un Arduino UNO (ATmega328P), tres servomotores que controlan las aletas y la cola, y un pulsador que activa el movimiento del pez. El comportamiento del pez se puede simular en Proteus o probar físicamente con el montaje real.

"Proyecto Pez Come Plástico — simulador y control Arduino"

Readme

Activity

0 stars

0 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

C++ 90.0%

C 10.0%