

# Propuesta de Proyecto: "Robotic Checkers Swarm" (Damas Robóticas Autónomas)

**Enlace al Repositorio (Código Fuente):** *(Aquí debéis crear un repo vacío en GitHub y pegar vuestro link, por ejemplo: <https://github.com/aalvaroonce/damatronics>)*

## 1. Definición y Complejidad

- **Complejidad Elegida: Alta.** No nos limitaremos a mover objetos estáticos mediante teletransporte. El proyecto implementa un sistema multi-agente donde 24 robots independientes coordinan sus movimientos físicos en tiempo real. Involucra navegación, comunicación entre procesos y manipulación física de objetos (acoplamiento mecánico).
- **Punto de Partida: Mundo Vacío (Empty World).** Construiremos todo desde cero: el diseño del tablero, el modelado de los robots y la programación de la lógica, sin utilizar demos prefabricadas de juegos.

## 2. Cumplimiento de Requisitos Técnicos

El proyecto integra los siguientes elementos obligatorios:

- **PROTO (Prototipo):** Crearemos un archivo `.proto` llamado `CheckerBot`. Este definirá la estructura física genérica de la ficha (chasis cilíndrico, ruedas, sensores, nodo *Connector*).
  1. *Uso:* Nos permite instanciar las 24 fichas (12 blancas y 12 negras) cambiando solo el campo de "color" y "nombre" sin duplicar código del modelo.
- **Dos Controladores Distintos:**
  1. **`game_supervisor.py` (Jerarquía Superior):** El "cerebro" del tablero. Lee los sensores táctiles (`TouchSensor`), gestiona los turnos, valida las reglas de las Damas y envía comandos de alto nivel.
  2. **`robot_driver.py` (Jerarquía Inferior):** El controlador individual de cada robot. Recibe órdenes (ej: "Ir a coordenada X,Y"), calcula la odometría, controla la velocidad de los motores y gestiona la activación del imán para coronar reinas.

### 3. Alcance del Proyecto

¿Hasta dónde queremos llegar? El objetivo final es una simulación funcional de una partida de Damas completa con las siguientes características:

1. **Interacción Humano-Máquina:** El usuario juega haciendo clic en las casillas del tablero virtual.
2. **Movimiento Autónomo:** Las fichas no se deslizan por magia; ruedan hacia su destino orientándose con brújula (Compass) y GPS.
3. **Lógica de Captura Avanzada ("El Retiro"):**
  - Cuando una ficha va a ser comida, el Supervisor ordena a la víctima retirarse autónomamente a una zona de "Cementerio" fuera del tablero para despejar el camino al atacante.
4. **Mecánica de Coronación (Reina):**
  - Implementación de coronas físicas. Cuando una ficha llega al final, una pieza "corona" se acopla magnéticamente sobre el robot usando el nodo Connector de Webots, permitiendo que ambas piezas se muevan juntas como una sola unidad física.

### 4. Arquitectura del Sistema

El flujo de información para lograr este alcance será:

1. **Input:** Clic en TouchSensor de la casilla.
2. **Procesamiento:** El Supervisor consulta la matriz de estado del juego.
3. **Comunicación:** El Supervisor usa un nodo Emitter para enviar un paquete de datos (String/Struct) al canal específico del robot.
4. **Acción:** El robot destinatario recibe el mensaje por su Receiver, decodifica la orden y ejecuta el movimiento físico (Motores).

### 5. Diagrama de Estructura Visual

El escenario constará de:

- **Zona A:** Tablero 8x8 con sensores.
- **Zona B:** Área de "Cementerio" (donde van los robots eliminados).
- **Zona C:** Almacén de Coronas (piezas pasivas esperando ser invocadas).

