

# Protocolo ODD - Modelo de Caminos Emergentes con Walkers y Buildings (modificado)

Arango Isabella, Salcedo laura

April 2025

## 1. Propósito y patrones

**Propósito:** Explorar cómo los caminos emergen en un entorno según la popularidad de los parches, influenciada por los movimientos repetidos de agentes hacia objetivos. Se busca observar cómo la interacción entre los movimientos y la colocación de edificios genera rutas estables sin haberlas predefinido para poder construir corredores ecológicos en esa zona.

**Patrones observados esperados:**

- Formación espontánea de caminos (parches grises) entre los edificios más visitados.
- Mayor popularidad en zonas intermedias entre edificios frecuentados.
- Regeneración de césped en zonas poco transitadas.
- Evitación de la línea intermedia (carretera)
- Regeneración y reducción de la comida en los edificios.

## 2. Entidades, variables de estado y escalas

**Entidades:**

- Walkers: agentes móviles que se dirigen hacia un objetivo.
- Buildings: creados por el usuario, funcionan como objetivos fijos.
- Patches: celdas del entorno con color y nivel de popularidad.

**Variables de estado:**

- Walkers:
  - **goal:** patch objetivo del walker. Si hay dos o más edificios bajo cierta probabilidad elige uno al azar o elige un patch aleatorio.
  - **attempts:** Intentos del walker por cruzar la línea intermedia.

- **Patches:**
  - **popularity:** aumenta al ser visitado, decrece si no es visitado.
- **Globals:**
  - **mouse-clicked?:** controla creación/eliminación de edificios.
  - **repulsion-probability:** controla la probabilidad de que el walker decida no pasar la linea intermedia.
- **buildings:**
  - **...:** Disminuye cantidad de recursos(comida) al ser visitado, aumenta si no es visitado.
  - **repulsion-probability:** controla la probabilidad de que el walker decida no pasar la linea intermedia.

#### **Escalas:**

- **Espacial:** rejilla de patches (NetLogo).
- **Temporal:** cada **tick** representa una unidad de tiempo arbitraria.

### **3. Resumen de procesos y calendario**

go ejecuta en cada **tick**:

1. **check-building-placement:** responde al clic del usuario (coloca o elimina edificios).
2. **move-walkers:** mueve walkers hacia su objetivo.
3. **decay-popularity:** reduce la popularidad de parches no visitados.
4. **recolor-patches:** actualiza color según popularidad.

### **4. Conceptos de diseño**

- **Emergencia:** las rutas aparecen sin estar predefinidas, a partir del uso repetido.
- **Adaptación:** no aplica.
- **Objetivos:** los walkers buscan llegar a su **goal**.
- **Aprendizaje:** no se implementa.
- **Predicción:** no aplicable.
- **Percepción:** los walkers perciben parches grises cercanos.

- **Interacción:** indirecta, compartiendo caminos.
- **Estocasticidad:** selección aleatoria de objetivos.
- **Colectivos:** no existen.
- **Observaciones:** se observa la formación y persistencia de caminos.
- **Racionalidad:** comportamiento programado para alcanzar metas.

## 5. Inicialización

- El entorno se limpia y se pinta de verde, a excepcion de la carretera que tiene un color negro.
- Se crean `walker-count` walkers en posiciones aleatorias.
- Cada walker recibe un `goal` inicial aleatorio.

## 6. Datos de entrada

- No hay datos externos.
- Los edificios son creados manualmente mediante clics del usuario.

## 7. Submodelos

- `check-building-placement`: gestiona los clics para agregar/quitar edificios.
- `move-walkers`: si está en su meta, se asigna una nueva; si no, avanza hacia ella.
- `walk-towards-goal`: incrementa la popularidad, elige mejor dirección y avanza.
- `decay-popularity`: reduce la popularidad de parches no visitados.
- `recolor-patches`: actualiza colores según `popularity`.
- `will-cross-repulsion-line?`: detecta si el agente esta cruzando.