

# ≡ ALGORITMOS Y PROGRAMACIÓN 1 (/) | EJ2 - GAMELIB

INICIO (/)

MATERIAL  
(/MATERIAL)

CALENDARIO  
(/CALENDARIO)

BIBLIOGRAFÍA  
(/BIBLIOGRAFIA)

PYTHON (/PYTHON)

ENCUESTAS  
(/ENCUESTAS)

DOCENTES  
(/DOCENTES/)

RÉGIMEN DE  
CURSADA  
(/REGIMEN)

ENTREGAS  
(/ENTREGAS/)

NOTAS (/NOTAS/)

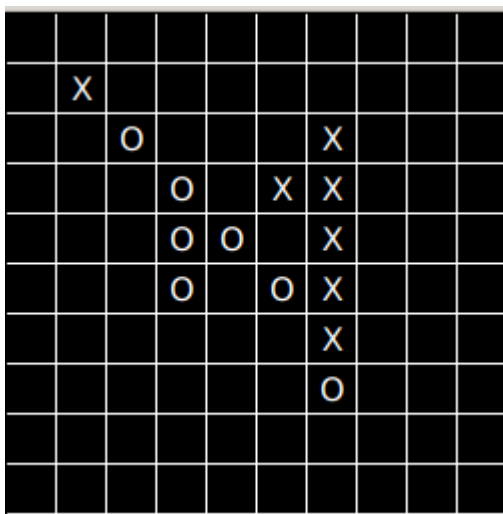
## EJ2 - Gamelib

Gamelib (<https://github.com/dessaya/python-gamelib>) es una biblioteca de funciones para implementar videojuegos simples en Python.

Utilizaremos Gamelib para implementar los próximos trabajos prácticos, así que el objetivo de este ejercicio es hacer un juego simple para tomar práctica con las funcionalidades de Gamelib.

### 5 en línea

El 5 en línea es un juego muy simple, muy parecido al Ta-Te-Ti, pero con la diferencia de que el tablero es de 10 x 10 y se debe hacer 5 en línea para ganar.



### Reglas

La grilla:

- El juego se desarrolla en una **grilla** rectangular de 10 filas y 10 columnas, formando así un total de **10 x 10 celdas**.
- Una celda puede estar vacía, o contener una **O** o una **X**.

El juego:

- El juego se juega de a 2 jugadores, **O** y **X**, por turnos.
- El primer turno es del jugador **O**, luego **X** y así sucesivamente.
- En su turno, el jugador debe ubicar una **O** o **X** (según corresponda) en una celda vacía de la grilla.

### Resolución:

- El juego termina con un ganador cuando hay 5 celdas iguales no vacías en la misma fila o en la misma columna o en la misma diagonal, o en empate cuando no hay más celdas vacías.

## Consigna

El objetivo de este ejercicio es implementar una aplicación gráfica que permita a 2 jugadores jugar al 5 en línea.

### Requerimientos mínimos:

- Dibujar la grilla
- Mostrar de quién es el turno
- Cuando se hace click en una celda vacía, ubicar una `0` o una `X` según corresponda.

**NO** es necesario que el juego detecte cuando hay 5 en línea. Lo único necesario es que la aplicación permita poner círculos y cruces en una grilla como si los jugadores utilizaran una hoja y un lápiz.

## Cómo utilizar Gamelib

Para utilizar Gamelib, el primer paso es descargar `gamelib.py` (<https://raw.githubusercontent.com/dessaya/python-gamelib/master/gamelib.py>) y ubicarlo en la misma carpeta que `5_en_linea.py`.

Luego, en `5_en_linea.py` recomendamos arrancar con el siguiente código y modificarlo a gusto:

```

import gamelib

def juego_crear():
    """Inicializar el estado del juego"""
    return '???'

def juego_actualizar(juego, x, y):
    """Actualizar el estado del juego

    x e y son las coordenadas (en pixels) donde el usuario hizo click.
    Esta función determina si esas coordenadas corresponden a una
    celda
    del tablero; en ese caso determina el nuevo estado del juego y
    lo
    devuelve.
    """
    return juego

def juego_mostrar(juego):
    """Actualizar la ventana"""
    gamelib.draw_text('5 en línea', 150, 20)

def main():
    juego = juego_crear()

    # Ajustar el tamaño de la ventana
    gamelib.resize(300, 300)

    # Mientras la ventana esté abierta:
    while gamelib.is_alive():
        # Todas las instrucciones que dibujen algo en la pantalla
        # deben ir
        # entre `draw_begin()` y `draw_end()`:
        gamelib.draw_begin()
        juego_mostrar(juego)
        gamelib.draw_end()

        # Terminamos de dibujar la ventana, ahora procesamos los
        # eventos (si el
        # usuario presionó una tecla o un botón del mouse, etc).

        # Esperamos hasta que ocurra un evento
        ev = gamelib.wait()

        if not ev:
            # El usuario cerró la ventana.
            break

        if ev.type == gamelib.EventType.KeyPress and ev.key == 'Escape':
            # El usuario presionó la tecla Escape, cerrar la aplicación.

```

```
break

if ev.type == gamelib.EventType.ButtonPress:
    # El usuario presionó un botón del mouse
    x, y = ev.x, ev.y # averiguamos la posición donde se hizo click
    juego = juego_actualizar(juego, x, y)

gamelib.init(main)
```

Este código sirve como punto de partida para implementar el juego. Prestar atención a las funciones `juego_crear`, `juego_actualizar` y `juego_mostrar` que son los lugares donde seguramente habrá que agregar código para implementar el 5 en línea.

En las funciones `juego_crear` y `juego_actualizar` tenemos que manipular el estado del juego. Esto es similar a lo que hicimos en el TP1, y no debería involucrar llamadas a funciones de Gamelib.

En la función `juego_mostrar` tenemos que utilizar las funciones de Gamelib para dibujar el tablero y mostrar de quién es el turno. ¡No es necesario dibujar nada muy sofisticado! Debería ser suficiente con utilizar las funciones `draw_text` y `draw_rectangle` / `draw_line`. Leer la referencia (<https://dessaya.github.io/python-gamelib/>) de Gamelib para saber qué funciones ofrece y cómo utilizarlas.

La función `main` se encarga del resto de la funcionalidad del juego. No debería ser necesario modificarla, a menos que quieras cambiar algo en el funcionamiento general del juego.

## Entrega

Se deberá entregar el archivo `5_en_linea.py`:

- La versión impresa debe entregarse en clase.
- La versión digital debe enviarse utilizando el formulario de entregas (/entregas).