

# Programació Avançada

---

## Pràctica 2. Curs 2024/2025

*Centre universitari adscrit a la*



## Rosa Herrero

Consultes presencials els divendres de 10.00 a 12.00 hores al despatx 14.

Consultes virtuals a través del correu electrònic [rHerrero@tecnocampus.cat](mailto:rHerrero@tecnocampus.cat)

Si us plau, per consultes virtuals envieu no envieu tot el projecte, només aquella part que tingueu dubte i en format PDF.

# Normativa

- Cal treballar en parelles, i cal que tots dos alumnes siguin del mateix grup de classe (G101, G102, G103, G104 o G105).
- No s'admeten lliuraments per correu electrònic o demores sense cap justificació (citació judicial o assumpte mèdic). Una pràctica no entregada sense justificació compta com un 0.
- Les pràctiques s'han de lliurar a través de l'aula virtual.

# Sistema d'avaluació

Pràctica 1 6%

Pràctica 2 6%

Pràctica 3 6%

Pràctica 4 6%

Pràctica 5 6%

Per aprovar l'assignatura és necessari que l'estudiant:

- com a mínim, hagi lliurat una de les dues primeres pràctiques, i què la qualificació sigui superior o igual a 4.
- com a mínim, hagi lliurat dues de les tres darreres pràctiques, i què la qualificació de cadascuna sigui superior o igual a 4.

Si no es donen aquestes premisses l'assignatura quedarà suspesa i en aquest cas sense possibilitat de recuperar-la.

# Pràctica 2: Seqüències enllaçades

## Objectius:

- Treballar amb seqüències enllaçades (amb i sense cap).

**Llenguatge:** JAVA

**Duració:** 2 sessions = 2 setmanes

**Projecte inicial:** disposeu de la solució de la pràctica 1

Cal llegir la normativa de practiques.

**Lliurament:** 20/10/2024

## Pràctica 2: classe Jugador

```
class Jugador<E extends ItipoPieza> {  
  
    private class NodePieza{  
        public E pieza;  
        public NodePieza seguent;  
        public NodePieza(E pieza, NodePieza seguent)  
        {  
            this.pieza = pieza;  
            this.seguent = seguent;  
        }  
    }  
  
    private NodePieza piezasVivas; // seqüència enllaçada de peces amb capçalera  
    // no té sentit que el llistat de peces sigui null, com a mínim tindrà una peça
```

# Pràctica 2: classe Jugador

```
public Jugador(ArrayList<E> piezasIniciais)
```

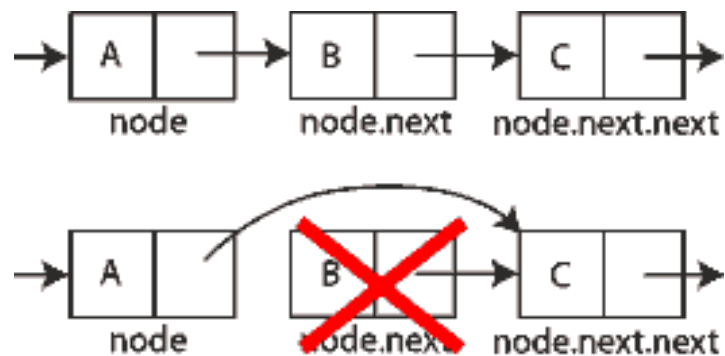
- Inicialitza tota la seqüència lineal a partir de l'arrayList

```
public ArrayList<E> getPiezasVivas()
```

- Retorna un arrayList amb tot el contingut de la seqüència lineal

```
public boolean eliminarPiezaEnPosicion(int columna, int fila) throws FijocException
```

- Elimina l'element desitjat, si no es troba retorna false, si s'elimina el rei es llança una excepció, i per qualsevol altra peça retorna true. Al bucle sempre cal mirar si el següent és el buscat:



# Pràctica 2: classe Jugador

private E buscarEnPosicion(int fila, int columna)

- Retorna un element de la seqüència enllaçada, si no es troba retorna null.

public void moverPieza(int columnaAnterior, int filaAnterior, int nuevaColumna, int nuevaFila)

- Invoquem al buscar a la posició actual (on cal trobar un element) i busquem a la nova posició (on cal trobar un null).
- El setPosition cal invocar-ho des del contingut del node.



## Pràctica 2: classe Torns

```
public class Torns <E>{  
  
    private class NodeTorn{  
        public E moviment;  
        public NodeTorn seguent;  
        public NodeTorn(E moviment, NodeTorn seguent)  
        {  
            this.moviment = moviment;  
            this.seguent = seguent;  
        }  
    }  
  
    private NodeTorn llistatTorns; // seqüència enllaçada de torns  
    // sense capçalera
```