

# Tecnología de la Programación 2 - Curso 2020/2021

Convocatoria ordinaria (09/06/2021) - Duración: 2 horas

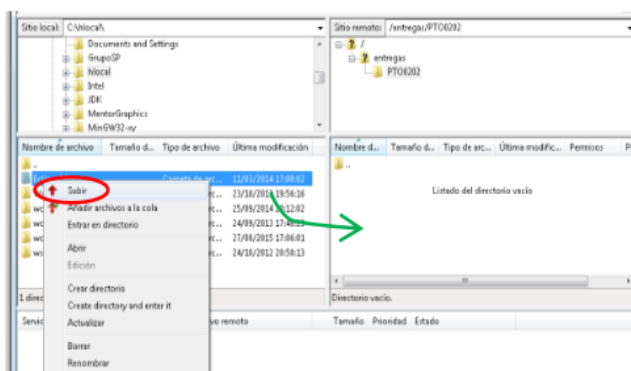
Puntuación máxima del examen: 8 puntos

Puntuación mínima en el examen para poder aprobar la asignatura: 4 puntos

## INSTRUCCIONES:

1. Descarga tu Práctica 2 del Campus Virtual, ponla a funcionar en Eclipse y comprueba que funciona correctamente.
2. Crea un fichero de texto **cambiosExamen.txt** en la raíz de tu proyecto (dentro de src). En este fichero deberás incluir tu nombre completo y el nombre de todos los ficheros .java que modifiques. Además puedes incluir otros comentarios relevantes sobre tu solución.
3. El código entregado debe compilar , y no debe romper la encapsulación de las clases (acceso a atributos privados y protegidos desde clases externas, utilización de atributos públicos, etc.). Si el código no compila o rompe encapsulación de clases, tendrá la calificación de 0 puntos en la pregunta correspondiente.
4. En la corrección se valorará el funcionamiento, la claridad del código, el uso conveniente de los medios proporcionados por la Programación Orientada a Objetos
5. **No mezclar las soluciones de las preguntas.** Hay que resolver cada pregunta partiendo de la versión original de la Práctica 2 que has entregado para el examen. No se corregirá una solución a la Pregunta 1 entregada con la solución de la Pregunta 2, ni viceversa.

## INSTRUCCIONES DE ENTREGA:



Para entregar la solución del examen, crea un fichero **NombreApellidos.zip** (usar sólo ZIP, no RAR, no 7z). En él debes incluir todo el proyecto una vez limpiado de archivos intermedios (es decir, sin el directorio bin/, que contiene los .class). Haz doble clic en el icono del escritorio denominado: **“EXAMENES en LABs entregas”**, y dentro de la ventana que aparece, doble clic en **“ALUMNOS entrega de prácticas y exámenes”**. Se abre otra ventana en la que debes seleccionar el archivo zip en el panel inferior

izquierdo y arrastrarlo al panel inferior derecho (o utiliza el botón derecho del ratón y la opción Subir), como se indica en la figura. Antes de abandonar el laboratorio debes pasar por el puesto del profesor para asegurarte de que lo que se ve en el puesto del profesor es lo que has entregado (comprobando el tamaño del archivo) y firmar en la hoja de entregas (usar tu bolígrafo para evitar el contacto).

## Pregunta 1 [3 Puntos]

Añadir a la práctica una ley nueva en la que todos los cuerpos giran en torno al cuerpo de mayor masa, la llamaremos **MovingAroundBodyGreaterMass**. Sea  $\overline{P_m}(a,b)$  la posición del cuerpo de mayor masa y sea  $\overline{P_i}(x_i,y_i)$  la posición de cada uno de los otros cuerpos, entonces la fuerza que aplica el cuerpo de mayor masa sobre cada uno de los otros es:

$$f_{m,i} = G * m_m * m_i * (1 - C) / |\overline{P_m} - \overline{P_i}|^2, \text{ siendo:}$$

- G la constante gravitacional con valor por defecto 6.67E-11 y
- C el factor de rotación, con valores de C en el intervalo [0,1) y valor por defecto 0.25.

Para calcular la dirección de esta fuerza, convertimos  $f_{m,i}$  en  $\overline{F_{m,i}}$  como sigue: sea  $\overline{d_{mi}}$  la dirección de un vector perpendicular a  $\overline{P_i P_m}$ , entonces:  $\overline{F_{m,i}} = \overline{d_{mi}} * f_{m,i}$ .

Dado un vector  $\overline{u}$  de coordenadas (x,y), un vector perpendicular a  $\overline{u}$  es  $\overline{v}$  de coordenadas (-y,x).

El formato Json para la ley sería:

```
{
  "type": "mabgm",
  "data": {
    "G": 6.67e10-11,
    "C": 0.25,
  }
}
```

La estructura JSON que debe devolver el builder para esta ley de fuerza es la siguiente:

```
{
  "type": "mabgm",
  "data": {
    "G": "the gravitational constant (a number)",
    "C": "the rotation factor (a number)"
  },
  "desc": "Moving around the body with greater mass"
}
```

Podéis añadir si lo necesitáis a la clase Vector2D un método para obtener un vector perpendicular a otro dado.

El cuadro de diálogo tiene que admitir poder elegir otros valores para G y C.

Para probarlo crea un fichero de pruebas en el que los cuerpos tengan inicialmente v[0.0,0.0].



## Pregunta 2 [5 puntos]

Añade una nueva tabla a la interfaz gráfica para mostrar la distancia total acumulada recorrida por cada cuerpo. La distancia que recorre un cuerpo en un paso de simulación es `p.distanceTo(q)` donde `p` es su posición anterior (antes de moverse) y `q` es su posición actual (después de moverse). La tabla tiene que incluir un botón “Reset Distance” (que aparece justo por encima de la tabla) para resetear las distancias acumuladas de todos los cuerpos a 0. Es decir empezamos a acumular distancias a partir de ese momento.

The screenshot shows a "Physics Simulator" window with three main sections: "Bodies", "Accumulated Distances", and "Viewer".

**Bodies Table:**

Id	Mass	Position	Velocity	Force
b1	2.6288493596...	[-3.475053465988183E10,4.6...	[1506.5473701111835,1466...	[1.1848486427211562E26,1.029...
b2	3.0E28	[-9.851748058093779E9,4.57...	[859.2114747970417,13571...	[6.373522960058252E25,-6.993...
b3	3.0E28	[9.861300709437077E9,-4.57...	[-773.3152955304787,-135...	[-4.859581935723204E25,7.140...
b4	3.0E28	[3.475146170315324E10,-4.6...	[-1498.2368169334857,-14...	[-1.3362427451546606E26,-1.1...

**Accumulated Distances Table:**

Reset Distances

Body	Accumulated Distance
b1	5.460331387154309E8
b2	4.574029953542122E9
b3	4.573117521097445E9
b4	5.454813043929282E8

**Viewer:**

h: toggle help, v: toggle vectors, +: zoom-in, -: zoom-out, =: fit  
Scaling ratio: 4.666666666666667E8

The viewer shows four bodies (b1, b2, b3, b4) as blue dots with red and green velocity vectors. b1 is at the bottom left, b2 is at the top left, b3 is at the top right, and b4 is at the bottom right.

**Status Bar:**

Time: 330000.0 | Bodies: 4 | Laws: Newton's Universal Gravitation with G=6.67E-11