



Universidad Nacional  
de La Matanza



# A rational mechanics course where everything is made with Python code

Bettachini, Víctor A.; Real, Mariano A.; Palazzo, Edgardo

New Media Pedagogy 23



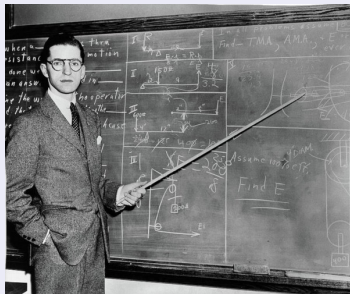
# Valorizar el tiempo de docentes y alumnos

Licklider (1957): 85 % de “pensar” es lo mundano (calcular, dibujar, etc.)



# Valorizar el tiempo de docentes y alumnos

Licklider (1957): 85 % de “pensar” es lo mundano (calcular, dibujar, etc.)



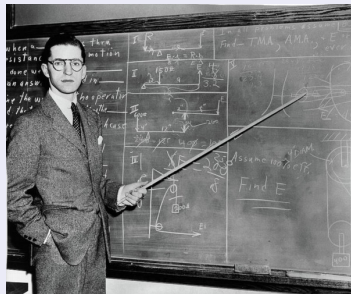
Aula y práctica: transcripción y reiteración

- Memoria  $\xrightarrow{\text{profesor}}$  pizarrón/presentación



# Valorizar el tiempo de docentes y alumnos

Licklider (1957): 85 % de “pensar” es lo mundano (calcular, dibujar, etc.)



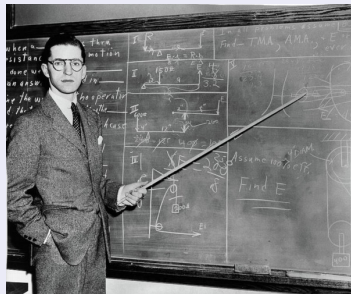
Aula y práctica: transcripción y reiteración

- Memoria  $\xrightarrow{\text{profesor}}$  pizarrón/presentación
- Pizarrón/presentación  $\xrightarrow{\text{alumno}}$  cuaderno



# Valorizar el tiempo de docentes y alumnos

Licklider (1957): 85 % de “pensar” es lo mundano (calcular, dibujar, etc.)



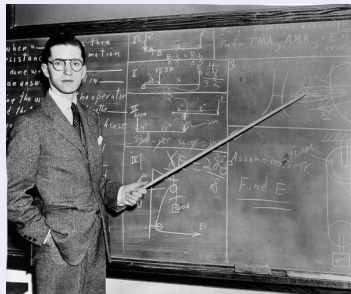
Aula y práctica: transcripción y reiteración

- Memoria  $\xrightarrow{\text{profesor}}$  pizarrón/presentación
- Pizarrón/presentación  $\xrightarrow{\text{alumno}}$  cuaderno
- Práctica: **reiterar** diagramas, cálculos, etc.



# Valorizar el tiempo de docentes y alumnos

Licklider (1957): 85 % de “pensar” es lo mundano (calcular, dibujar, etc.)



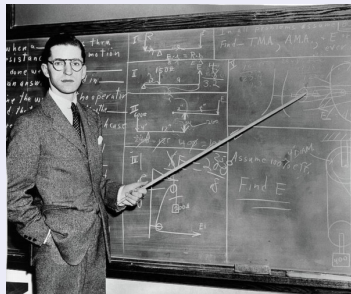
## Aula y práctica: transcripción y reiteración

- Memoria  $\xrightarrow{\text{profesor}}$  pizarrón/presentación
- Pizarrón/presentación  $\xrightarrow{\text{alumno}}$  cuaderno
- Práctica: **reiterar** diagramas, cálculos, etc.
- Aburrimiento  $\Rightarrow \downarrow$  concentración



# Valorizar el tiempo de docentes y alumnos

Licklider (1957): 85 % de “pensar” es lo mundano (calcular, dibujar, etc.)



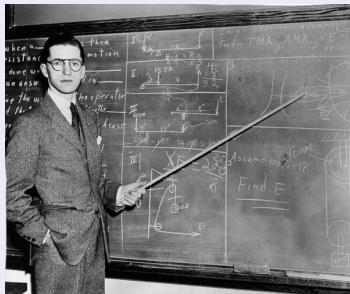
Aula y práctica: transcripción y reiteración

- Memoria  $\xrightarrow{\text{profesor}}$  pizarrón/presentación
- Pizarrón/presentación  $\xrightarrow{\text{alumno}}$  cuaderno
- Práctica: **reiterar** diagramas, cálculos, etc.
- Aburrimiento  $\Rightarrow \downarrow$  concentración



## Valorizar el tiempo de docentes y alumnos

Licklider (1957): 85 % de “pensar” es lo mundano (calcular, dibujar, etc.)



## Aula y práctica: transcripción y reiteración

- Memoria  $\xrightarrow{\text{profesor}}$  pizarrón/presentación
- Pizarrón/presentación  $\xrightarrow{\text{alumno}}$  cuaderno
- Práctica: **reiterar** diagramas, cálculos, etc.
- Aburrimiento  $\Rightarrow \downarrow$  concentración

- Ingenio  $\xrightarrow{\text{profesor}}$  código en repositorio

```

[+] Code + Test

#-----#
# Funciones
#-----#
1. 1) Igualdad (sympy) (sympy core relational equality)
    Si un lado derecho explicita la igualdad correcta del sistema en función de coordenadas y velocidades generalizadas y el tiempo.
2. 2) Igualdad (sympy) core relational Equality
    Si un lado derecho explicita la igualdad parcial del sistema en función de coordenadas y velocidades generalizadas y el tiempo
    coordenadas generalizadas. Idealizado (sympy core symbol logic)
    Para lo que quiere obtenerse la ecuación en Euler-Lagrange.
#-----#
# Retorno
#-----#
    Igualdad (sympy core relational equality)
    Ecuación de Euler-Lagrange homogénea para la coordenada generalizada
    (sympy)
    Lagrangiana = (T - V, sympy) (sympy)
    Lagrangiana (T - V) a como se define respecto al tiempo con la función diff se declara y como símbolo
    return sympy.diff(Lagrangiana, coordenadas generalizadas)
    - Lagrangiana, diff(coordenadas generalizadas, diff(t)): diff(t)
    )
    simplify()

[+] theta_R = molarLagrange(1, V, theta)
theta_R

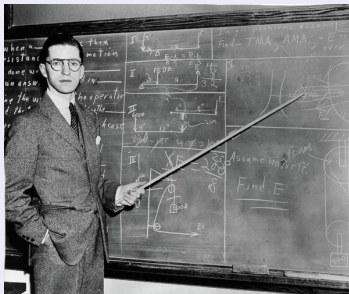
- /m/s/s [205P]
- /m/s/s [205P]

```



## Valorizar el tiempo de docentes y alumnos

Licklider (1957): 85 % de “pensar” es lo mundano (calcular, dibujar, etc.)



## Aula y práctica: transcripción y reiteración

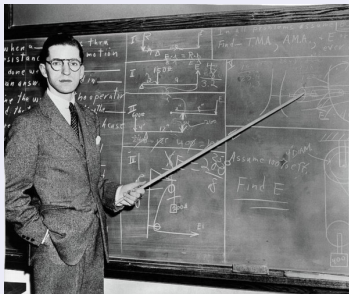
- Memoria  $\xrightarrow{\text{profesor}}$  pizarrón/presentación
- Pizarrón/presentación  $\xrightarrow{\text{alumno}}$  cuaderno
- Práctica: **reiterar** diagramas, cálculos, etc.
- Aburrimiento  $\Rightarrow \downarrow$  concentración

[illegible]

- Ingenio  $\xrightarrow{\text{profesor}}$  código en repositorio
- Repositorio del curso  $\xrightarrow{\text{alumno}}$  propio

## Valorizar el tiempo de docentes y alumnos

Licklider (1957): 85 % de “pensar” es lo mundano (calcular, dibujar, etc.)



## Aula y práctica: transcripción y reiteración

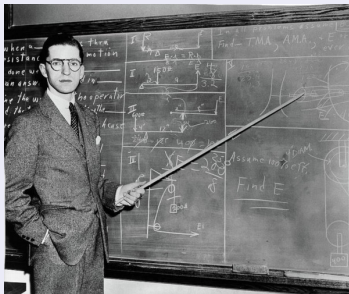
- Memoria  $\xrightarrow{\text{profesor}}$  pizarrón/presentación
- Pizarrón/presentación  $\xrightarrow{\text{alumno}}$  cuaderno
- Práctica: **reiterar** diagramas, cálculos, etc.
- Aburrimiento  $\Rightarrow \downarrow$  concentración

[illegible]

- Ingenio  $\xrightarrow{\text{profesor}}$  código en repositorio
- Repositorio del curso  $\xrightarrow{\text{alumno}}$  propio
- Poner en práctica: ~~no~~ utilizar código

## Valorizar el tiempo de docentes y alumnos

Licklider (1957): 85 % de “pensar” es lo mundano (calcular, dibujar, etc.)



## Aula y práctica: transcripción y reiteración

- Memoria  $\xrightarrow{\text{profesor}}$  pizarrón/presentación
- Pizarrón/presentación  $\xrightarrow{\text{alumno}}$  cuaderno
- Práctica: **reiterar** diagramas, cálculos, etc.
- Aburrimiento  $\Rightarrow \downarrow$  concentración

[illegible]

- Ingenio  $\xrightarrow{\text{profesor}}$  código en repositorio
- Repositorio del curso  $\xrightarrow{\text{alumno}}$  propio
- Poner en práctica: ~~no~~ utilizar código

# Estudiantes de ingeniería deben aprovecharse del código



# Estudiantes de ingeniería deben aprovecharse del código

- Usan calculadora pues **aprendieron** aritmética en la primaria.



# Estudiantes de ingeniería deben aprovecharse del código

- Usan calculadora pues **aprendieron** aritmética en la primaria.
- Usarán álgebra computacional pues **aprobaron** álgebra y análisis.

```
[14]: sistemaEcuaciones = [  
    x_EL,  
    phi_EL,  
]  
variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas  
variablesDespeje_sol = sym.nonlinsolve(sistemaEcuaciones, variablesDespeje ).args[0]  
  
[15]: x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0]) # [m s-2]  
phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1]) # [m s-2]  
x_pp, phi_pp  
  
[15]: 
$$\ddot{x} = \frac{-\ell g m_2 \sin(\phi) + \frac{\ell m_2 (\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2) \sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)}, \quad \ddot{\phi} = -\frac{(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2) \sin(\phi)}{\ell (m_1 + m_2 \sin^2(\phi))}$$

```



# Estudiantes de ingeniería deben aprovecharse del código

- Usan calculadora pues **aprendieron** aritmética en la primaria.
- Usarán álgebra computacional pues **aprobaron** álgebra y análisis.
  - ▶ Enfocarse en nuevas habilidades, no en cálculos automatizables.

```
[14]: sistemaEcuaciones = [  
      x_EL,  
      phi_EL,  
    ]  
variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas  
variablesDespeje_sol = sym.nonlinsolve(sistemaEcuaciones, variablesDespeje ).args[0]
```

```
[15]: x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0]) # [m s-2]  
phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1]) # [m s-2]  
x_pp, phi_pp
```

```
[15]:
```

$$\ddot{x} = \frac{-\ell g m_2 \sin(\phi) + \frac{\ell m_2 (\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2) \sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)}, \quad \ddot{\phi} = -\frac{(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2) \sin(\phi)}{\ell (m_1 + m_2 \sin^2(\phi))}$$

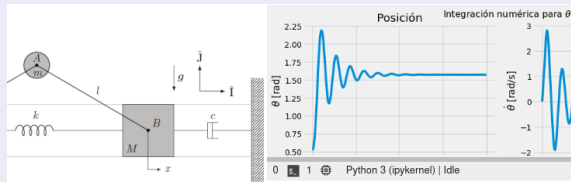


# Estudiantes de ingeniería deben aprovecharse del código

- Usan calculadora pues **aprendieron** aritmética en la primaria.
- Usarán álgebra computacional pues **aprobaron** álgebra y análisis.
  - ▶ Enfocarse en nuevas habilidades, no en cálculos automatizables.
  - ▶ Con cálculo numérico resolverán lo imposible en pizarrón/papel.

```
[14]: sistemaEcuaciones = [  
    x_EL,  
    phi_EL,  
]  
variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas  
variablesDespeje_sol = sym.nonlinsolve(sistemaEcuaciones, variablesDespeje ).args[0]  
  
[15]: x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0]) # [m s-2]  
phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1]) # [m s-2]  
x_pp, phi_pp  
  
[15]: 
$$\ddot{x} = \frac{-\ell g m_2 \sin(\phi) + \frac{\ell m_2 (\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2) \sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)}, \quad \ddot{\phi} = -\frac{(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2) \sin(\phi)}{\ell (m_1 + m_2 \sin^2(\phi))}$$

```



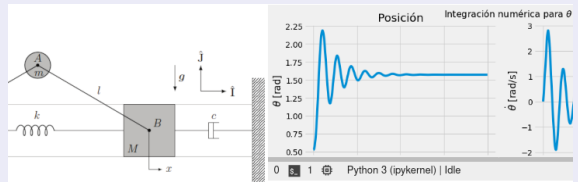


# Estudiantes de ingeniería deben aprovecharse del código

- Usan calculadora pues **aprendieron** aritmética en la primaria.
- Usarán álgebra computacional pues **aprobaron** álgebra y análisis.
  - ▶ Enfocarse en nuevas habilidades, no en cálculos automatizables.
  - ▶ Con cálculo numérico resolverán lo imposible en pizarrón/papel.

```
[14]: sistemaEcuaciones = [  
    x_EL,  
    phi_EL,  
]  
variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas  
variablesDespeje_sol = sym.nonlinsolve(sistemaEcuaciones, variablesDespeje ).args[0]  
  
[15]: x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0]) # [m s-2]  
phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1]) # [m s-2]  
x_pp, phi_pp  
  
[15]: 
$$\ddot{x} = \frac{-\ell g m_2 \sin(\phi) + \frac{\ell m_2 (\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2) \sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)}, \quad \ddot{\phi} = -\frac{(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2) \sin(\phi)}{\ell (m_1 + m_2 \sin^2(\phi))}$$

```

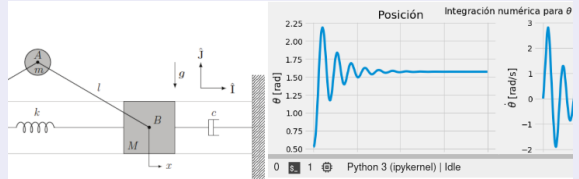


# Estudiantes de ingeniería deben aprovecharse del código

- Usan calculadora pues **aprendieron** aritmética en la primaria.
- Usarán álgebra computacional pues **aprobaron** álgebra y análisis.
  - ▶ Enfocarse en nuevas habilidades, no en cálculos automatizables.
  - ▶ Con cálculo numérico resolverán lo imposible en pizarrón/papel.

```
[14]: sistemaEcuaciones = [  
    x_EL,  
    phi_EL,  
]  
variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas  
variablesDespeje_sol = sym.nonlinsolve(sistemaEcuaciones, variablesDespeje ).args[0]  
  
[15]: x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0]) # [m s-2]  
phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1]) # [m s-2]  
x_pp, phi_pp  
  
[15]: 
$$\ddot{x} = \frac{-\ell g m_2 \sin(\phi) + \frac{\ell m_2 (\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2) \sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)}, \quad \ddot{\phi} = -\frac{(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2) \sin(\phi)}{\ell (m_1 + m_2 \sin^2(\phi))}$$

```



Papert (1980) “El aprendizaje sucede cuando el alumno toma las riendas”

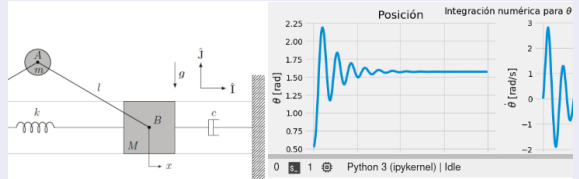
- Cierta problema es resuelto por un código provisto por el docente.

# Estudiantes de ingeniería deben aprovecharse del código

- Usan calculadora pues **aprendieron** aritmética en la primaria.
- Usarán álgebra computacional pues **aprobaron** álgebra y análisis.
  - ▶ Enfocarse en nuevas habilidades, no en cálculos automatizables.
  - ▶ Con cálculo numérico resolverán lo imposible en pizarrón/papel.

```
[14]: sistemaEcuaciones = [  
    x_EL,  
    phi_EL,  
]  
variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas  
variablesDespeje_sol = sym.nonlinsolve(sistemaEcuaciones, variablesDespeje ).args[0]  
  
[15]: x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0]) # [m s-2]  
phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1]) # [m s-2]  
x_pp, phi_pp  
  
[15]: 
$$\ddot{x} = \frac{-\ell m_2 \sin(\phi) + \frac{\ell m_2 (\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2) \sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)}, \quad \ddot{\phi} = -\frac{(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2) \sin(\phi)}{\ell (m_1 + m_2 \sin^2(\phi))}$$

```



Papert (1980) “El aprendizaje sucede cuando el alumno toma las riendas”

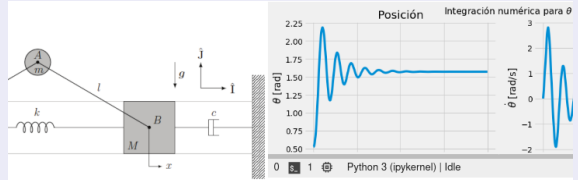
- Cierta problema es resuelto por un código provisto por el docente.
- El alumno realiza modificaciones para resolver nuevas problemáticas.

# Estudiantes de ingeniería deben aprovecharse del código

- Usan calculadora pues **aprendieron** aritmética en la primaria.
- Usarán álgebra computacional pues **aprobaron** álgebra y análisis.
  - ▶ Enfocarse en nuevas habilidades, no en cálculos automatizables.
  - ▶ Con cálculo numérico resolverán lo imposible en pizarrón/papel.

```
[14]: sistemaEcuaciones = [  
    x_EL,  
    phi_EL,  
]  
variablesDespeje = [x.diff(t,2), phi.diff(t,2)] # despejar aceleraciones generalizadas  
variablesDespeje_sol = sym.nonlinsolve(sistemaEcuaciones, variablesDespeje).args[0]  
  
[15]: x_pp = sym.Eq(variablesDespeje[0], variablesDespeje_sol.args[0]) # [m s-2]  
phi_pp = sym.Eq(variablesDespeje[1], variablesDespeje_sol.args[1]) # [m s-2]  
x_pp, phi_pp  
  
[15]: 
$$\ddot{x} = \frac{-\ell g m_2 \sin(\phi) + \frac{\ell m_2 (\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2) \sin(\phi)}{m_1 + m_2 \sin^2(\phi)}}{\ell m_2 \cos(\phi)}, \quad \ddot{\phi} = -\frac{(\ell m_2 \cos(\phi) \dot{\phi}^2 + g m_1 + g m_2) \sin(\phi)}{\ell (m_1 + m_2 \sin^2(\phi))}$$

```



Papert (1980) “El aprendizaje sucede cuando el alumno toma las riendas”

- Cierta problema es resuelto por un código provisto por el docente.
- El alumno realiza modificaciones para resolver nuevas problemáticas.
- Paulatinamente se torna autónomo reutilizando el propio código.

Todo el material es editable en línea



# Todo el material es editable en línea

## Cuaderno programable en línea: texto + ecuaciones + código

The screenshot displays a Jupyter Notebook titled "PÉNDULO ENHEBRADO SOLVED.IPYNB". The left sidebar contains a table of contents with four items:

- Enunciado
- 1. Construya el Lagrangiano y la(s) función(es) de ligadura
- 2. Calcule las ecuaciones de Euler-Lagrange
- 3. Obtenga una expresión para la tensión que ejerce la barra
- 4. Grafique la tensión en los primeros diez segundos de la dinámica.

The main content area shows the text for item 3: "3. Obtenga una expresión para la tensión que ejerce la barra". Below this, the equation  $Q_d = \lambda_1 \frac{\partial f_1}{\partial d} = \lambda_1$  is displayed. The text continues: "Por tanto hay que resolver el sistema con las 3 ecuaciones de Euler-Lagrange y la única de ligadura para determinar  $\lambda_1$ . Esta última hay que resolverla para su caso homogéneo y expresar su derivada segunda para que esté en el mismo orden que las de Euler-Lagrange, a fin de cuentas estamos resolviendo sistemas diferenciales de 2.º orden."

The notebook shows the following code cells:

```
[14]: f_1
```

```
[14]: f_1 = -l + d
```

Determinamos también  $\ddot{\theta}_1$  y  $\ddot{\theta}_2$  pues serán necesarias para los cálculos numéricos posteriores.

```
[15]: sistema = [theta1_EL.expand(),
               theta2_EL.expand(),
               d_EL.expand(),
               sym.Eq(f_1.rhs.diff(t,2), 0), # esto es igual a d punto punto = 0
               ]
variables = [theta1.diff(t,2), theta2.diff(t,2), lambda_1]
variables_sol = sym.nonlinsolve(sistema, variables).args[0]
```

```
[16]: lambda_1_sol = sym.Eq(lambda_1, variables_sol.args[2])
      lambda_1_sol.simplify()
```

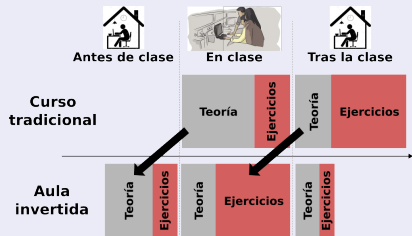
```
[16]: m * (2*a*cos(theta_1 - theta_2)*theta_1**2 + g*cos(2*theta_1 - theta_2) + g*cos(theta_2) + 2*d*theta_2**2 - 2*d)
```

# Trabajo sincrónico y asincrónico sobre el código

## Teoría y ejercicios resueltos en línea en cuadernos programables

- Consultas **asincrónicas** en línea (24/7) **públicas** hacia otros alumnos.

### Aula invertida



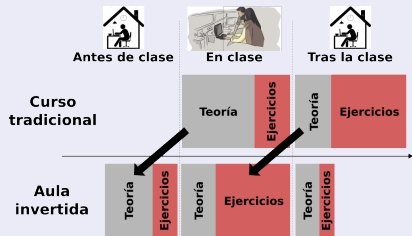
Sincrónico	Teoría	Ejercicios
Antes	Leer y aplicar	Iniciarles
Durante	Aclarar dudas	Terminarles
Luego	Consultas adicionales	Correcciones del docente

# Trabajo sincrónico y asincrónico sobre el código

## Teoría y ejercicios resueltos en línea en cuadernos programables

- Consultas **asincrónicas** en línea (24/7) **públicas** hacia otros alumnos.
- Trabajo **colaborativo remota** en cuadernos multi-usuario.

## Aula invertida



Sincrónico	Teoría	Ejercicios
Antes	Leer y aplicar	Iniciarles
Durante	Aclarar dudas	Terminarles
Luego	Consultas adicionales	Correcciones del docente

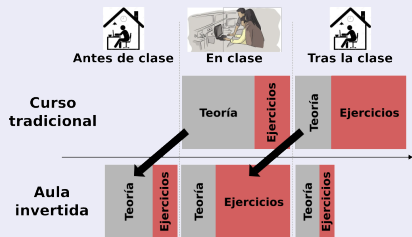


# Trabajo sincrónico y asincrónico sobre el código

## Teoría y ejercicios resueltos en línea en cuadernos programables

- Consultas **asincrónicas** en línea (24/7) **públicas** hacia otros alumnos.
- Trabajo **colaborativo remota** en cuadernos multi-usuario.
- Al finalizar ejercicios, asistencia docente **sincrónica individual**

## Aula invertida



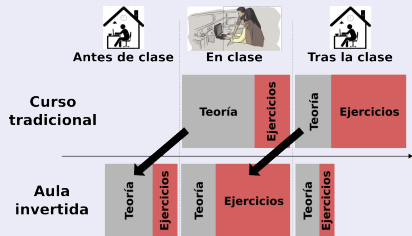
Sincrónico	Teoría	Ejercicios
Antes	Leer y aplicar	Iniciarles
Durante	Aclarar dudas	Terminarles
Luego	Consultas adicionales	Correcciones del docente

# Trabajo sincrónico y asincrónico sobre el código

## Teoría y ejercicios resueltos en línea en cuadernos programables

- Consultas **asincrónicas** en línea (24/7) **públicas** hacia otros alumnos.
- Trabajo **colaborativo remota** en cuadernos multi-usuario.
- Al finalizar ejercicios, asistencia docente **sincrónica individual**
- Entrega **obligatoria** para su corrección **semanal**.


## Aula invertida



Sincrónico	Teoría	Ejercicios
Antes	Leer y aplicar	Iniciarles
Durante	Aclarar dudas	Terminarles
Luego	Consultas adicionales	Correcciones del docente

# Asistencia docente y corrección asincrónica

## Google Colaboratory: comentando y editando el ejercicio del alumno

 07 No conservativas | ej4 ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda [Se editó por última vez: 3 de junio](#)

+ Código + Texto

Conectar ▾ Editando ↕

```
[ ] # Energía potencial
m1_V = - (m1* g* (- N.y)).dot(m1_r)
# pot_k1 = unMedio* ( -k1* ((l10 + x1)* (sym.cos(theta) - sym.sin(theta)) )**2 ) # mal
pot_k1 = unMedio* k1* (l10 + x1)**2 # Lo escribí yo
# pot_k2 = unMedio* -k2* (l20 + x)**2
pot_k2 = unMedio* k2* (l20 + x)**2
V = sym.Eq(sym.Symbol('V'), m1_V + pot_k1 + pot_k2 ) #agrega el potencial elastico k en la ecuacion
V
```


$$V = gm_1(-l_{10} - x_1)\sin(\theta) + \frac{k_1(l_{10} + x_1)^2}{2} + \frac{k_2(l_{20} + x)^2}{2}$$

▼ Lagrangiano

```
[ ] L = sym.Eq(sym.Symbol('\mathcal{L}'), (T.rhs - V.rhs))
L
```

$$\mathcal{L} = -gm_1(-l_{10} - x_1)\sin(\theta) - \frac{k_1(l_{10} + x_1)^2}{2} - \frac{k_2(l_{20} + x)^2}{2} + \frac{(m_0 + m_1)(2\cos(\theta)\dot{x}_1 + \dot{x}^2 + \dot{x}_1^2)}{2}$$

ECUACIONES DE EULER

 Victor Alexis Bettachini [Resolver](#) ⋮  
31 de may. de 2021  
(editado el 31 de may. de 2021)

- El estiramiento del resorte de k\_1 es colineal con x1. No tienen sentido pensar en proyecciones (si es lo que hiciste, que realmente no entiendo).  
- ¿Porque negativos los k?

# Seguimiento individualizado

## Registro del cumplimiento con entregas semanales

Q Buscar

MR

Calificaciones

MR

Vencimiento el 28 sept

Exportar a Excel

	g06e03	g06e04	g06e05	g05e01a	g05e01c	g05e02	g05e03
Buscar alumnos	28 sept	28 sept	28 sept	14 sept	14 sept	14 sept	14 sept
Promedio de clase							
...	Visto		Visto	Devuelto	Entregado	Entregado	Entregado
...				Devuelto	Entregado	Entregado	Entregado
...				Entregado	Entregado	Entregado	Entregado
...	Visto	Visto	Visto	Entregado	Entregado	Entregado	Entregado
...				Entregado	Entregado		Entregado
...	Visto	Visto	Visto	Entregado	Entregado	Visto	Entregado

# Resumiendo

## Curso centrado en código

- Teoría: texto + ecuaciones + código ejecutable en cuadernos digitales.

## Modalidad de aula invertida

# Resumiendo

## Curso centrado en código

- Teoría: texto + ecuaciones + código ejecutable en cuadernos digitales.
- Reforzados con videos propios y bibliografía.

## Modalidad de aula invertida

# Resumiendo

## Curso centrado en código

- Teoría: texto + ecuaciones + código ejecutable en cuadernos digitales.
- Reforzados con videos propios y bibliografía.
- Práctica: reutilización del código del docente.

## Modalidad de aula invertida

# Resumiendo

## Curso centrado en código

- Teoría: texto + ecuaciones + código ejecutable en cuadernos digitales.
- Reforzados con videos propios y bibliografía.
- Práctica: reutilización del código del docente.
- Ejecución en línea:

## Modalidad de aula invertida



## Curso centrado en código

- Teoría: texto + ecuaciones + código ejecutable en cuadernos digitales.
- Reforzados con videos propios y bibliografía.
- Práctica: reutilización del código del docente.
- Ejecución en línea:
  - ▶ Colaboración y corrección remota.

## Modalidad de aula invertida

# Resumiendo

## Curso centrado en código

- Teoría: texto + ecuaciones + código ejecutable en cuadernos digitales.
- Reforzados con videos propios y bibliografía.
- Práctica: reutilización del código del docente.
- Ejecución en línea:
  - ▶ Colaboración y corrección remota.
  - ▶ No requiere computadoras en el campus, ni que sean poderosas.

## Modalidad de aula invertida

# Resumiendo

## Curso centrado en código

- Teoría: texto + ecuaciones + código ejecutable en cuadernos digitales.
- Reforzados con videos propios y bibliografía.
- Práctica: reutilización del código del docente.
- Ejecución en línea:
  - ▶ Colaboración y corrección remota.
  - ▶ No requiere computadoras en el campus, ni que sean poderosas.
  - ▶ Registro fechado del trabajo del alumno.

## Modalidad de aula invertida

# Resumiendo

## Curso centrado en código

- Teoría: texto + ecuaciones + código ejecutable en cuadernos digitales.
- Reforzados con videos propios y bibliografía.
- Práctica: reutilización del código del docente.
- Ejecución en línea:
  - ▶ Colaboración y corrección remota.
  - ▶ No requiere computadoras en el campus, ni que sean poderosas.
  - ▶ Registro fechado del trabajo del alumno.

## Modalidad de aula invertida

- Teoría: énfasis en la lectura autónoma por parte del alumno.

# Resumiendo

## Curso centrado en código

- Teoría: texto + ecuaciones + código ejecutable en cuadernos digitales.
- Reforzados con videos propios y bibliografía.
- Práctica: reutilización del código del docente.
- Ejecución en línea:
  - ▶ Colaboración y corrección remota.
  - ▶ No requiere computadoras en el campus, ni que sean poderosas.
  - ▶ Registro fechado del trabajo del alumno.

## Modalidad de aula invertida

- Teoría: énfasis en la lectura autónoma por parte del alumno.
- Consultas: asincrónicas y públicas.

# Resumiendo

## Curso centrado en código

- Teoría: texto + ecuaciones + código ejecutable en cuadernos digitales.
- Reforzados con videos propios y bibliografía.
- Práctica: reutilización del código del docente.
- Ejecución en línea:
  - ▶ Colaboración y corrección remota.
  - ▶ No requiere computadoras en el campus, ni que sean poderosas.
  - ▶ Registro fechado del trabajo del alumno.

## Modalidad de aula invertida

- Teoría: énfasis en la lectura autónoma por parte del alumno.
- Consultas: asincrónicas y públicas.
- Finalizar ejercicios: asistencia personalizada del docente

# Actualidad del proyecto



2023 Retro-alimentación de los alumnos mejoró:





2023 Retro-alimentación de los alumnos mejoró:

- Apuntes y código en el repositorio.



## 2023 Retro-alimentación de los alumnos mejoró:

- Apuntes y código en el repositorio.
  - Metodología ejercitación y evaluación.
- Mayor exigencia de ejercicios → mejor respuesta.



## 2023 Retro-alimentación de los alumnos mejoró:

- Apuntes y código en el repositorio.
  - Metodología ejercitación y evaluación.
- Mayor exigencia de ejercicios → mejor respuesta.



# Actualidad del proyecto

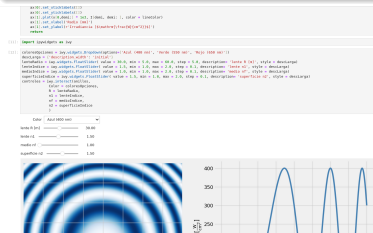
## 2023 Retro-alimentación de los alumnos mejoró:

- Apuntes y código en el repositorio.
- Metodología ejercitación y evaluación.

Mayor exigencia de ejercicios → mejor respuesta.

## 2024

- Física II empleará simulaciones provistas por nosotros.



# Actualidad del proyecto

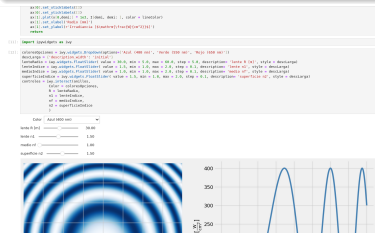
## 2023 Retro-alimentación de los alumnos mejoró:

- Apuntes y código en el repositorio.
- Metodología ejercitación y evaluación.

Mayor exigencia de ejercicios → mejor respuesta.

## 2024

- Física II empleará simulaciones provistas por nosotros.
- *Prompt engineering*: alumnos generarán código con IA.



```
lagrangiano = (T.xhs - V.yhs).expand()
t = sym.Symbol('t') # como se deriva respecto al tiempo con la función diff se declara t como simbolo
return sym.Eq(
    lagrangiano.diff(coordenadaGeneralizada)
    - lagrangiano.diff(coordenadaGeneralizada.diff(t)).diff(t)
    + 0
).simplify()
```

121

```
x1_EL = euleriLange(T, V, x1)
x1_EL
```

122

$$\frac{m^2 M \ddot{x}_1}{2} - g m_1 + g m_2 + m_1 \ddot{x}_1 + m_2 \ddot{x}_1 = 0$$

Esta es una ecuación diferencial lineal de segundo orden homogenea. De aquí se puede despejar  $\ddot{x}$

123

```
#Despejar x1PuntoPunto
x1PuntoPunto = sym.solve(x1_EL, x1.diff(t, t)).args[0]
```

