

## OBJETIVO PRINCIPAL

Construir un sistema lanzador de pelotas de pin pon y realizar el análisis de un gol de campo bajo condiciones de viento en 3D

## ABSTRACT

El proyecto se enfoca en el desarrollo de un lanzador que permite estudiar el movimiento parabólico de proyectiles y hacer una simulación de su trayecto. Tiene como objetivo principal el proporcionar una herramienta que nos permita explorar y comprender los principios detrás de los movimientos parabólicos, al igual que evaluar la influencia de variables como el ángulo de lanzamiento, la velocidad inicial y la masa del proyectil.

El lanzador ofrece la capacidad de controlar y ajustar múltiples parámetros para llevar a cabo experimentos repetibles. Este dispositivo es capaz de lanzar proyectiles con precisión, lo que facilita la recopilación de datos y el análisis de resultados.

Adicionalmente, como parte de este proyecto, se lleva a cabo una simulación detallada del tiro parabólico utilizando el programa GeoGebra. El simulador consiste en un conjunto de ecuaciones paramétricas y deslizadores que permiten ajustar las condiciones variables del viento y sus ángulos. De igual manera, se modeló un poste de anotación por medio de vectores fijos para así visualizar el trazado del proyectil.

## METODOLOGÍA

Para ambas partes se requirió de un manejo y desarrollo de ecuaciones cinemáticas, al igual que un entendimiento de la Ley de Hooke y factores externos como velocidad de viento.

$$y = y_0 + \vec{v}_0 \sin(\theta)t + \frac{1}{2}at^2$$

$$\Sigma \vec{F}_x = -mg \sin(\theta) + k\Delta x - \mu_k \vec{N} = m\vec{a}_x$$

$$\int_0^{\Delta x} \frac{k\Delta x}{m} - g(\sin(\theta) + \mu_k \cos(\theta)) dx = \int_0^{\vec{v}_f} v dv$$

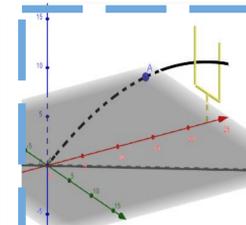
$$v_f = \sqrt{\frac{k}{m} \Delta x^2 - 2g(\sin(\theta) + \mu_k \cos(\theta)) \Delta x}$$

Figura 1. Algunas de las ecuaciones utilizadas para el desarrollo del problema.

## DESARROLLO DE SIMULADOR 3D

Se tuvo que ajustar las ecuaciones a los tres planos cartesianos y por medio de vectores, "dibujar" un poste hipotético según medidas reales sobre un entorno real. De igual forma, se introdujeron nuevos parámetros, como velocidad y ángulo de viento y un ángulo de patada para compensar dicho factor externo.

```
from sympy import symbols, cos, sin, solve
from math import radians
# recibe v0, u, cte, c
# devuelve a, b, t
v0 = float(input("Velocidad inicial de la patada (m/s-1) [0 <=] \n"))
u = float(input("Velocidad del viento (m/s-1) [u <=] \n"))
cte = float(input("Constante de arrastre del viento? [0 <=] \n"))
c = float(input("Ángulo de dirección de viento [0-360] \n"))
a, b, t = symbols('a b t', real=True)
eq1 = -22.5 + v0 * cos(radians(a)) * sin(radians(b)) + t + cte * u * t * sin(radians(c))
eq2 = 0 + v0 * cos(radians(a)) * cos(radians(b)) + t + cte * u * t * cos(radians(c))
eq3 = -5.97 + v0 * sin(radians(a)) * t - 0.5 * 9.81 * t ** 2
solution = solve((eq1, eq2, eq3), (a, b, t), (1,1,1))
a_res, b_res, t_res = solution
print("\nEl ángulo de disparo a = %s°. \nEl ángulo de compensación lateral b = %s°. \nLa res, b_res, t_res" % (a_res, b_res))
print("El tiempo de la trayectoria es de %s segundos." % (t_res))
```



$$Pos = \begin{cases} x(t) = v_0 \cos(\theta) \sin(\phi) t + cte[\vec{u} \sin(\gamma)] \\ y(t) = v_0 \cos(\theta) \cos(\phi) t + cte[\vec{u} \cos(\gamma)] \\ z(t) = y_0 + v_0 \sin(\theta) t - \frac{gt^2}{2} \end{cases}$$

Figura 3. Ecuaciones gobernantes y visualización de simulado

## DESARROLLO DEL PROTOTIPO FÍSICO

El desarrollo del prototipo físico comenzó con el diseño del lanzador utilizando Tinkercad, una herramienta de modelado 3D. Una vez completado el diseño, procedimos a imprimir el lanzador en 3D para obtener un prototipo físico.

Para que el prototipo funcionara era necesario un resorte con una constante elástica (k) lo suficientemente pequeña para permitir la compresión manual, pero lo bastante grande para proporcionar la potencia necesaria al disparar; por lo mismo optamos por utilizar un resorte con una constante elástica de 2.77 N/m. Sin embargo, el resorte era muy pequeño, por lo que se optó por atar dos resortes en serie para obtener la longitud necesaria y así lograr un disparo efectivo.

m(kg)	Δx1 Leo	Δx2 Sofia	Δx3 Nicole	Δx4 Nani	Δx5 Genaro	Δx prom	xy	x^2	ax+b	ya-x	(ya-x)^2	Mi=0.284
0.05	0.001	0.002	0.003	0.003	0.001	0.002	0.0001	0.00004	-0.061434	0.168868	0.02851640142	0.0700701416
0.17	0.004	0.003	0.005	0.004	0.004	0.004	0.00068	0.00016	-0.060868	0.368302	0.158444832	0.1068963887
0.4	0.006	0.006	0.007	0.005	0.006	0.006	0.0024	0.00036	-0.060302	0.697736	0.4868355267	0.1997744538
0.7	0.007	0.009	0.009	0.007	0.008	0.008	0.0056	0.00064	-0.059736	1.195682	1.430100752	0.4667165128
1.2	0.01	0.015	0.017	0.013	0.018	0.016	0.01752	0.0021316	-0.057682	2.510292	6.301145207	1.845224094
2.52	0.028	0.035	0.041	0.032	0.037	0.0346	0.0263	0.0033316	-0.3002082	4.9709824	8.405242369	2.689683591

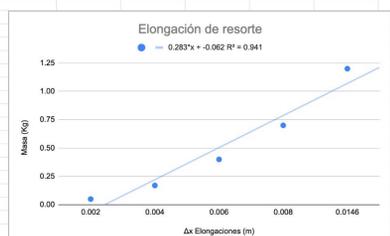


Figura 2. Calculo para sacar la constante de elongación

## RESULTADOS

Utilizando los métodos mencionados, se obtuvo una constante de elongación con valor de 827 N/m, y por ende, una velocidad de disparo máximo teórico de 2.08 ms<sup>-1</sup>. Esto, comparado con el valor experimental obtenido con Tracker de 2.51 ms<sup>-1</sup>, difiere en un 20.78%. Así mismo, el programa desarrollado en Python permite, recibiendo valores del entorno como datos de viento, conocer una aproximación numérica a cómo debiera patear el jugador el balón en términos de ángulos de disparo y ajuste, cuyos efectos son visualizables en el simulador Geogebra.

$$v_f = \sqrt{\frac{827.081 \text{ N/m}}{0.0027 \text{ kg}} (0.0038)^2 - 2(9.81)[\sin(32^\circ) + 0.07 \cos(32^\circ)](0.0038 \text{ m})}$$

$$v_f = 2.08296 \text{ ms}^{-1}$$

$$\epsilon_{\%} = \frac{|2.08296 - 2.5159|}{2.08296} (100)$$

$$\epsilon_{\%} = 20.785\%$$



Figura 4. Lanzador

## CONCLUSIONES

En resumen, se cumplió el objetivo, ya que pudimos modelar una patada de gol de campo con un prototipo hecho por nosotros. No obstante, algunas mejoras posibles podrían ser, por la parte de el lanzador el hacer la impresión con mayor calidad para que la palanca pueda atorar de mejor manera, y así conseguir un disparo más preciso. Por otro lado, el simulador podría tener la mejora en un mejor control sobre los parámetros y los datos, al igual que un manejo más amable con un usuario nuevo. Pero más importante aún, el programa, por cómo aproxima funciones trigonométricas por métodos de sustitución, llega a tener imprecisiones en la información que ofrece, ando aproximaciones en lugar de predicciones.

## REFERENCIAS

- Fernández, J. (s.f.). *Movimiento Parabólico*. FisicaLab. Recuperado el 19 de septiembre de 2023 de: <https://www.fisicalab.com/apartado/movimiento-parabolico>
- ¿Qué es la Ley de Hooke? (Artículo) | Khan Academy. (s. f.). Khan Academy. <https://es.khanacademy.org/science/physics/work-and-energy/hookes-law/a/what-is-hookes-law>
- Sympy. (n.d.). *Solveset Python documentation*.