

Aportaciones para la comprensión de la física a través de programas en Python

Juan Pablo Martínez¹, Josefina Huespe¹, Raúl Pérez¹; Rodolfo Dematte²

Resumen

En el contexto de la mirada estratégica de esta facultad junto al Consejo Federal de Decanos de Ingeniería (CONFEDI) se presenta un trabajo de aportes a la formación por competencias desde la cátedra de Física I para la carrera de Ingeniería en Sistemas. La herramienta pedagógica que se emplea es un programa diseñado en Python por el equipo pedagógico de este trabajo. Si bien se presenta aquí un ejemplo teórico para tiro oblicuo, cabe mencionar que cada tema de la asignatura (Física I) tiene un desarrollo en código con las mismas características a modo de cierre de cada unidad de aprendizaje. Los estudiantes, pueden manipular el programa, encontrar la coherencia entre los cálculos realizados y los valores que el programa les permite ingresar, para luego interpretar a través de la interfase gráfica amigable dichos resultados. De esta forma, acreditamos que los usuarios tengan una aproximación a la programación desde la especificidad de su carrera y que vinculen estas destrezas con la asignatura que plantea la propuesta, enriqueciendo el proceso de aprendizaje desde el gabinete, pudiendo emplear esta metodología para concluir conceptos desde la teoría o desde la práctica. La retroalimentación entre el estudiante y la herramienta es fundamental, el primero necesita proporcionar datos, valores, cálculos, ecuaciones, etc. al programa para que pueda calcular y posteriormente graficar los resultados y en caso de no ser coherentes con el concepto físico estudiado, poder identificar las inconsistencias en el procedimiento de resolución llevado a cabo, propiciando entonces la discusión entre pares y con los docentes.

Palabras claves: enseñanza de la Física- Formación por competencias- Python

¹ Grupo de Investigación Educativa en Matemática para Ingeniería, Universidad Tecnológica Nacional, Facultad Regional Mendoza. E-mail: jpmartinezmanno@gmail.com.

² IENAR, CENIIT, Universidad Nacional de La Rioja.

Introducción

En este trabajo, se considera al alumno co- responsable de su propio proceso de aprendizaje. En función a sus saberes culturales (el empleo de vídeos juegos, redes sociales y por supuesto la aproximación a los lenguajes de programación), se considera al estudiante como protagonista de su camino de aprendizaje. Pensando en que luego, convertido en graduado podrá ser capaz de juzgar críticamente con cuáles elementos de ese camino de aprendizaje quedarán para siempre en su memoria emotiva, para repetirlas, repartirlas en la sociedad, y replicarlas en su realidad.

Esta propuesta, espera aportar una experiencia significativa a los estudiantes de ingeniería en sistemas basadas en la aproximación a la integración entre los conceptos estudiados en física y la comprensión de los mismos a través un programa escrito de manera sencilla (ver Materiales y Métodos).

El aprendizaje será activo y estará siempre centrado en el estudiante, que trabajará de modo autónomo y en equipo. En esta propuesta, los procesos de enseñanza y aprendizaje tienen un carácter descentralizado del aula tradicional, y se desplazan a diferentes espacios tales como virtuales e interdisciplinarios.

Se espera también a partir de este trabajo colaborar con la formación por competencias del ingeniero tecnológico, capaz de desempeñarse en grupos disciplinares y multidisciplinares, para resolver problemas y/o crear conocimientos originales vinculados a fenómenos que involucran desde sistemas a escalas subatómicas hasta los sistemas que conciernen al Universo en gran escala.

Las siguientes aportaciones corresponden entonces a prácticas de aprendizaje que representan no solamente una de las unidades del programa de Física I (Unidad N°3- Movimiento de proyectiles) pero que es extensible para todas las unidades del programa. Cabe destacar lo novedoso de la propuesta porque en la actualidad no existe la incorporación de este tipo de prácticas de aprendizaje en Física I en nuestra Facultad.

En el contexto de los objetivos de esta práctica son:

- Desarrollar la capacidad de abstracción y formalización de los procesos físicos. Aplicar los conceptos básicos del cálculo diferencial e integral a los fenómenos de la Física.
- Desarrollar la capacidad de la lectura comprensiva. Interpretar y describir fenómenos físicos de la mecánica Newtoniana.
- Desarrollar la lógica del pensamiento científico.
- Incentivar la duda científica y la formulación de hipótesis.
- Ejercitar la utilización de lenguajes de programación y la familiarización con la programación
- Desarrollar la estructura de conexión entre lo estudiado en una asignatura y lo que representa en la carrera que estudia el alumno.
- Utilización de las TIC como elemento de apoyo para el desarrollo del conocimiento y su implementación en las clases de gabinete.

Esta formación asume la mediación del docente y la institución, basada en la interacción dialógica, horizontal, la relación entre pares, coherentemente con la idea de este proyecto en que se supone que cada estudiante construye y reconstruye su propio conocimiento. En otras palabras, interviene activamente para aprender, ello le permite transformar sus conocimientos y experiencias para desarrollar permanentemente sus competencias como profesional.

Hasta ahora en la mayoría de los casos se espera que el estudiante realice, por sí mismo, el trabajo interdisciplinario; situación que provoca inquietudes y contradicciones por cuanto se requiere la mediación del formador para efectuar las conexiones interdisciplinarias y concretarlas en espacios reales. (Ávalos, 2001) (Diker, 2003), (Vaillant, 2009).

Materiales y Métodos

En el presente trabajo se buscó mostrar el aporte que genera la utilización de los lenguajes de programación (en este caso se utilizó Python por la variedad de librerías y módulos orientados a las ciencias básica, pero se puede extender a

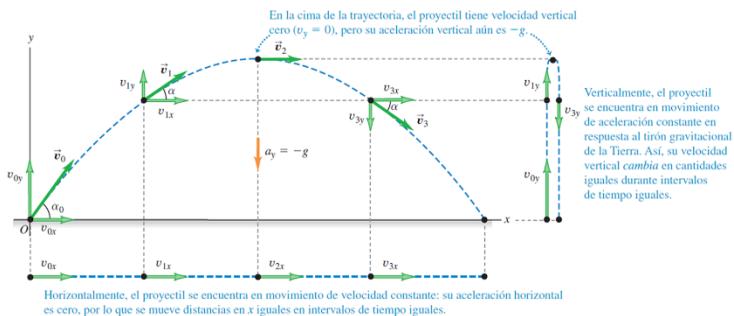
otros lenguajes) como herramientas pedagógicas en la enseñanza de la Física. La metodología consistió en tomar un problema típico, de los de las guías disponibles en la cátedra y problema tipo de los libros Física I acerca de la unidad N° 3: Movimiento de proyectiles.

Brevemente se recordarán algunas nociones muy básicas sobre el movimiento de proyectiles

Podemos analizar el movimiento de un proyectil como una combinación de movimiento horizontal con velocidad constante y movimiento vertical con aceleración constante.

Figura N°1

Movimiento de proyectiles



Nota: Corresponde a la figura 3.17 del capítulo N°3, Sears Zemansky Ed 13a. Vol I.

A partir de la premisa anterior el movimiento entonces puede ser estudiado de la siguiente manera:

En la dirección horizontal (x) el movimiento es con velocidad constante

$$a_x = 0 \quad (1)$$

$$v_x = v_{0x} + a_x \cdot t \quad (2)$$

$$v_x = v_{0x} \quad (3)$$

$$v_x = v_0 \cos\alpha_0 \quad (4)$$

$$x = x_0 + v_{0x}t + \frac{1}{2}a_x t^2 \quad (5)$$

$$x = x_0 + v_{0x}t \quad (6)$$

$$x = x_0 + (v_0 \cos\alpha_0)t \quad (7)$$

En la dirección vertical (y) el movimiento es con aceleración constante y en ningún caso se considera la resistencia del aire

$$a_y = -g \quad (8)$$

El signo negativo obedece a la elección de los sentidos de los vectores

$$v_y = v_{0y} - g \cdot t \quad (9)$$

$$v_y = (v_0 \operatorname{sen}\alpha_0) - g \cdot t \quad (10)$$

$$y = y_0 + v_{0y}t - \frac{1}{2}gt^2 \quad (11)$$

$$y = y_0 + (v_0 \operatorname{sen}\alpha_0)t - \frac{1}{2}gt^2 \quad (12)$$

$$v_y^2 = v_{0y}^2 - 2g(y - y_0) \quad (13)$$

Además, en la teoría se concluyen las siguientes ecuaciones complementarias a las expresadas anteriormente

$$v_{0x} = v_0 \cos\alpha_0 \quad (14)$$

$$v_{0y} = v_0 \operatorname{sen}\alpha_0 \quad (15)$$

la distancia r del proyectil al origen (la magnitud del vector \vec{r} de posición) está dada por

$$r = \sqrt{x^2 + y^2} \quad (16)$$

La rapidez del proyectil (la magnitud de su velocidad) en cualquier instante es

$$v = \sqrt{v_x^2 + v_y^2} \quad (17)$$

La *dirección* de la velocidad, en términos del ángulo α que forma con el eje $+x$

$$\tan \alpha = \frac{v_y}{v_x} \quad (18)$$

Es posible también encontrar la ecuación de la trayectoria en términos de x e y ,

$$y = (\tan \alpha_0)x - \frac{g}{2v_0^2 \cos^2 \alpha_0} x^2 \quad (19)$$

Además, el alcance máximo horizontal puede ser expresado como

$$R = \frac{v_0^2 \sin 2\theta_0}{g} \quad (20)$$

$$\text{Si } \theta_0 = 45^\circ \text{ entonces,} \quad (21)$$

$$R_{\text{máx}} = \frac{v_0^2}{g} \quad (22)$$

Por ello, para la resolución del problema propuesto, los estudiantes a partir del programa escrito en Python pueden optar para resolver el caso propuesto utilizando las técnicas de cálculo y contrastar mediante el soporte gráfico que ha sido incorporado a través de la librería Matplotlib.

Esto les permite a los alumnos visualizar la física del problema pudiendo modificar los datos de entrada, viendo cómo esto afecta la trayectoria del proyectil y por otro lado comparar los tiempos, dificultades y resultados respecto a los cálculos que ellos mismos realicen en las actividades de gabinete.

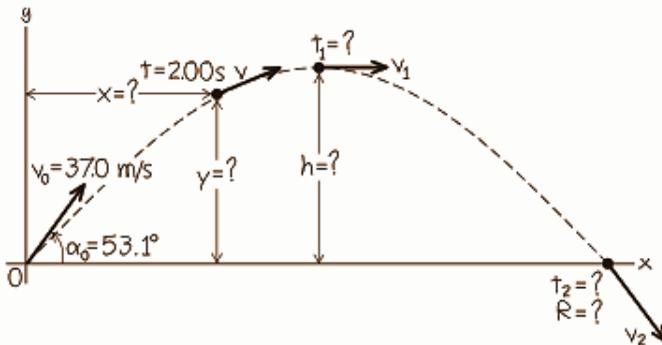
Propuesta de resolución del 3.7 “Altura y alcance de un proyectil” (Sears et al, 2013).

El problema seleccionado corresponde al *Ejemplo 3.7 - Altura y alcance de un proyectil I*: Es un caso de tiro oblicuo en el que se debe calcular la altura máxima y el alcance de un proyectil:

Un bateador golpea una pelota de béisbol de modo que esta sale del bate a una rapidez $v_0 = 37,0 \text{ m/s}$ con un ángulo $\alpha_0 = 53,1^\circ$.

- Calcule la posición de la pelota y su velocidad (magnitud y dirección) cuando $t = 2,00 \text{ s}$.
- Determine cuándo la pelota alcanza el punto más alto de su vuelo y su altura h en ese punto.
- Obtenga el alcance horizontal R , es decir, la distancia horizontal desde el punto de partida hasta donde la pelota cae al suelo.

Figura N° 2



Nota: Corresponde a la figura 3.23 del capítulo 3. Ejemplo 3.7

El programa escrito para la resolución de este problema utiliza las librerías:

- *Numpy*: manejo de vectores, funciones matemáticas y métodos numéricos.
- *Scipy*: utilización de constantes físicas.
- *Matplotlib*: obtención de gráficas en 2D.
- *Ipywidgets*: interacción en tiempo real con el soporte gráfico.

En primer lugar, se tuvo que calcular las posiciones x e y con las componentes v_x y v_y de la velocidad para el instante $t = 2,00$

s. Mediante las ecuaciones de la cinemática y manipulación algebraica se obtuvieron los valores de las magnitudes mencionadas:

$$x = v_{0x}t = 44,4 \text{ m} \quad (23)$$

$$y = v_{0y}t - \frac{1}{2}gt^2 = 39,6 \text{ m} \quad (24)$$

$$v_x = v_{0x} = 22,2 \frac{\text{m}}{\text{s}} \quad (25)$$

$$v_y = v_{0y} - gt = 10,0 \text{ m} \quad (26)$$

Luego se obtuvo los valores de v y α para el $t = 2,00 \text{ s}$:

$$v = \sqrt{v_x^2 + v_y^2} = 24,4 \frac{\text{m}}{\text{s}} \quad (27)$$

$$\alpha = \left(\frac{v_x}{v_y} \right) = 24,2^\circ \quad (28)$$

A continuación, para la parte b del problema, teniendo en cuenta que en el punto más alto de la trayectoria la velocidad $v_y = 0$, se obtuvo:

$$t = 3,02 \text{ s} \quad (29)$$

$$h = 44,6 \text{ m} \quad (30)$$

Finalmente, en la parte c del ejercicio se pide obtener el alcance del movimiento por lo que se propuso hacerlo en dos pasos. En primer lugar, se calculó el tiempo en el que $y = 0$:

$$y = 0 = v_{0y}t - \frac{1}{2}gt^2 \quad (31)$$

Dado que se obtuvo una ecuación cuadrática en el tiempo se procedió a utilizar un método eficaz para el cálculo de raíces que ofrece la función *poly1d* de *Numpy*. Se debe ingresar como argumento de la función un vector integrado por las constantes a , b y c de una ecuación cuadrática que en este caso son,

$$a = -\frac{1}{2}g \quad (32)$$

$$b = v_{0y} \quad (33)$$

$$c = 0 \quad (34)$$

Resultados

A partir de lo comentado en la sección anterior, se muestra aquí el código para llamar a las funciones mencionadas y la incorporación de las ecuaciones de movimiento de proyectiles en el programa empleado.

El código para llamar la función y obtener las raíces queda de la siguiente manera:

```
1 # Se calculan las raíces utilizando la clase Poly1d de Numpy.
2 # En primer lugar se ingresan las constantes de la ecuación cuadrática en el formato [A, B, C], por lo
3 # que el polinomio se arma así: p = np.poly1d([A, B, C]).
4 p = np.poly1d([-1/2*constants.g, v0_y, 0])
5
6 # Luego, mediante el método r se calculan las raíces del polinomio p.
7 raíces = p.r
```

El resultado es el vector *raíces* con dos componentes siendo cada una de ellas las raíces de la ecuación cuadrática planteada. La primer componente vale $t_1 = 0$ que corresponde al instante inicial del movimiento y la segunda componente es $t_2 = 6,03$ s que es el tiempo buscado para obtener el alcance de la trayectoria,

$$R = 134,1 \text{ m} \quad (35)$$

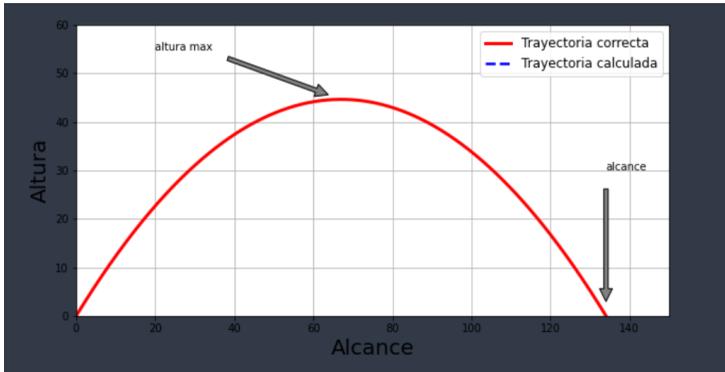
Construcción de la gráfica mediante Matplotlib

Una vez resuelto el problema mediante código se procedió a graficar la trayectoria encontrada utilizando la librería *Matplotlib* (Figura N°2). Pero no con la idea de obtener un plot estático sino buscando crear una interfaz gráfica interactiva de manera que el usuario pueda modificar en tiempo real las variables de entrada. Esto permite, en primer lugar, darle al estudiante la

opción de realizar los cálculos, obtener los resultados e ingresarlos en la interfaz para verificar si los valores calculados coinciden con los correctos (*trayectoria correcta versus trayectoria calculada*).

Figura N°3

Gráfica de la trayectoria que es solución del problema propuesto



Por otro lado, se puede obtener una visión más amplia del problema estudiado, modificando arbitrariamente los valores de entrada. Lo que permite visualizar con mayor profundidad la situación física en cuestión permitiéndole al usuario poner en valor distintas consideraciones y condiciones sin estar restringido ya solo a las que indica el ejercicio propuesto.

Para la obtención de la gráfica interactiva se definió la siguiente función,

```

8 def f(R, h):
9     x = np.arange(0, R, 0.1)
10    y = ((-4*h/np.square(R)) * np.square(x-(R/2))) + h
11    plt.figure(figsize=(10,5))
12    plt.grid(True)
13    plt.plot(xs, ys, color='r', label='Trayectoria correcta', linewidth=3.0)
14    plt.plot(x, y, 'b--', label='Trayectoria calculada', linewidth=2.5)
15    plt.ylabel('Altura', fontsize = 20)
16    plt.xlabel('Alcance', fontsize = 20)
17    plt.axis([0, 150, 0, 60])
18    plt.annotate('altura max', xy=(alcance/2, altura), xytext=(20, altura+10),
19                arrowprops=dict(facecolor='grey', shrink=0.1),
20                )
21    plt.annotate('alcance', xy=(alcance, 0), xytext=(alcance, 30),
22                arrowprops=dict(facecolor='grey', shrink=0.1),
23                )
24    plt.legend(loc=1, fontsize = 12)

```

En las líneas 9 y 10 se crean dos vectores, el vector x estará compuesto por valores que van de 0 hasta el valor ingresado de R en incrementos de 0,1 unidades. Entre las líneas 11 y 24 se introducen los comandos para realizar las gráficas, tanto la calculada como la correcta, y las configuraciones de visualización de dichas trayectorias.

La finalidad de graficar mediante una función tuvo como objetivo aplicar a la misma el método *interact* de la librería

```

25 interact(f, R=widgets.FloatSlider(min=0, max=300, step=0.1, value=0),
26          h=widgets.FloatSlider(min=0, max=100, step=0.1, value=0))

```

ipywidgets, el cual permite poder modificar y actualizar en tiempo real el gráfico obteniendo la interfaz interactiva que se buscaba,

Este método toma la función f definida anteriormente y se ocupa de actualizarla en tiempo real con los valores ingresados, los cuales se implementaron en la interfaz como las variables de entrada (R y h) por medio de la interacción con deslizadores o directamente mediante teclado,

Figura N°4

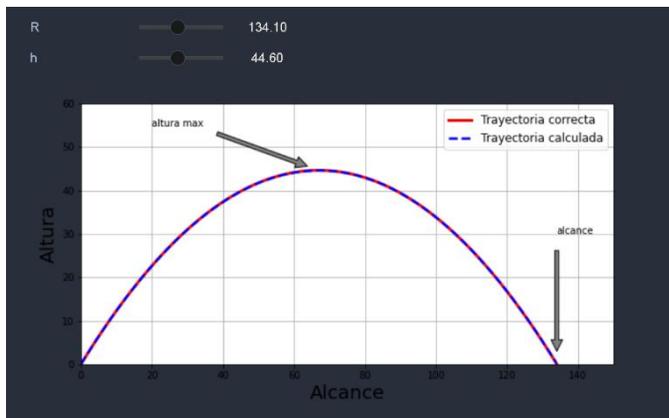
VARIABLES DE ENTRADA

R	<input type="text"/>	0.00
h	<input type="text"/>	0.00

En el siguiente ejemplo se puede observar cómo queda la gráfica para el caso en el cual el usuario ingresa los valores correctos para la *trayectoria calculada*,

Figura N°5.

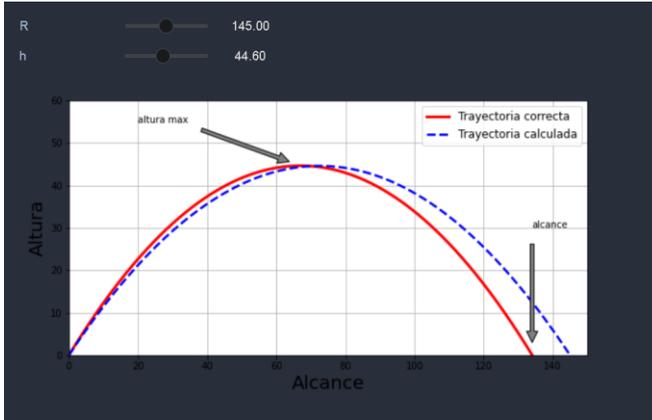
La trayectoria calculada (trazo punteado azul) es igual a la trayectoria correcta (trazo rojo).



En el siguiente caso se muestra una trayectoria calculada con ambos valores diferentes ($R = 120 \text{ m}$ y $h = 50 \text{ m}$) al de la trayectoria correcta,

Figura N° 6

Trayectoria calculada para $R = 120\text{ m}$ y $h = 50\text{ m}$.

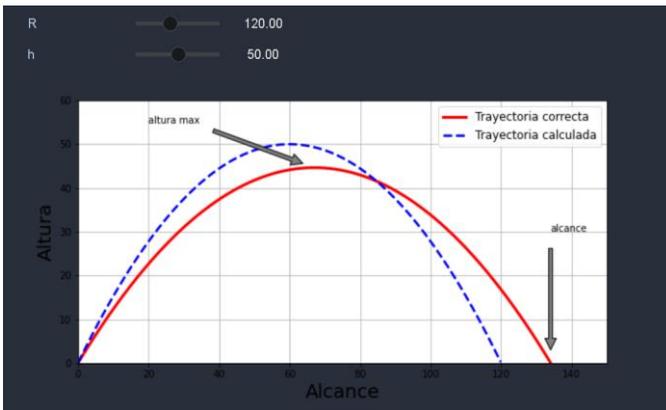


Se puede observar en este último caso la posibilidad de verificar cuando los cálculos son diferentes a los correctos y por otro lado, poner en valor la comparación de diferentes casos de estudio.

Finalmente es la siguiente figura se muestra cómo se visualiza el caso en el que una de las variables está bien calculada ($h = 44,6\text{ m}$) y la otra no ($R = 145\text{ m}$),

Figura N°7

Trayectoria calculada para $h = 44,6\text{ m}$ y $R = 145\text{ m}$.



Discusión

Se utilizó la librería Matplotlib, para trabajar simulaciones de forma más atractiva a través del manejo de espacios en entornos 3D. Entre otras librerías que se indicaron a lo largo del trabajo, porque además como se mencionó anteriormente conseguir familiarizar a los estudiantes con el uso de lenguajes de programación más aún en la carrera de Ingeniería de Sistemas no solo contribuye con la formación por competencias de un ingeniero tecnológico sino también apropia la cultura del entorno de estudio con la asignatura Física, creando un lazo que no siempre es visible para los estudiantes. Este tipo de práctica de aprendizaje se propone para Física I en principio de Ing. de Sistemas pero puede ser extendida a otras carreras según el interés y la interacción interdisciplinar que se pretenda conseguir así como las competencias que se busque generar en los estudiantes.

Estas prácticas de aprendizaje, son sin dudas del tipo significativas porque generan en los estudiantes la percepción y la interpretación del proceso constante de producción porque además ellos mismos pueden trabajar y aportar al código (que dejamos disponible de acceso libre). El alumno así es co-responsable de su propio camino de aprendizaje, en un ambiente familiar como la programación. En palabras de Bransford (2004) : *“La investigación sobre el aprendizaje muestra que es necesario que los estudiantes construyan su propia comprensión de las ideas científicas dentro del marco de sus conocimientos existentes, a fin de permitir a los estudiantes construir su propia comprensión de ideas científicas, ellos deben ser alentados a participar activamente en el tema, para que puedan tener la oportunidad de aprender a través de ese compromiso”*.

Los estudiantes se benefician de preguntas de predicción en la identificación de lo que es importante y la construcción de una estructura mental para la evaluación de los fenómenos. Este marco ayuda a los estudiantes a eliminar detalles innecesarios y especificar los componentes fundamentales que se han de memorizar (Zamarro, Molina y Núñez, 2004).

Salir de los modelos tradicionales de enseñanza es imperativo, la vinculación entre las actividades propias de las carreras de ingeniería con los recursos de simulación y programación son importantes. Los estudiantes de hoy están muy familiarizados con el uso de dispositivos electrónicos, el Internet y la cultura de los videojuegos, conseguir involucrar a los estudiantes en la física mediante el uso de modelos visuales por computador, que simulen la apariencia y el movimiento, para brindar orientación e incentivar al mismo tiempo la innovación y la formación; es realmente una experiencia pedagógica significativa.

Conclusiones

El empleo de programas de softwares y paquetes de simulación para el estudio de muchas disciplinas entre ellas la física lleva ya varios años en el sistema educativo, por caso PhET (acceso libre), Interactive Physics sólo por nombrar algunos, y ha permitido que la academia de a poco incrementa el uso del poder computacional en las asignaturas, por ello incorporar programas propios y con lenguajes conocidos por los estudiantes es un avance más en este camino de desarrollar competencias y habilidades en nuestros estudiantes pensando sobre todo en que este caso particular que se presenta una integración entre la programación con la Física I, con el objetivo de obtener resultados de aprendizaje más trascendente que los conseguidos de forma habitual.

Pero además en concordancia con el desarrollo de competencias específicas, se cita a Yasar, (2003) “estas prácticas presenta un camino para seguir, pues integra las experiencias para estimular y activar en los estudiantes de ingeniería el deseo de generar una base sólida del conocimiento, útil para su experiencia profesional”

Las ciencias computacionales son el origen de la física computacional, mediante la combinación de la física, la matemática aplicada y la informática, para solucionar problemas científicos. Se busca con estas aplicaciones fomentar el contacto alumno-profesor, la reciprocidad y la cooperación de los estudiantes, el aprendizaje activo, dar

retroalimentación inmediata, destacar la importancia de pasar tiempo en una tarea, comunicar las expectativas y respetar la diversidad de talentos y formas de aprender (Chickering y Ehmann, 1996).

El software puede constituir un valioso medio de enseñanza dentro del proceso de enseñanza-aprendizaje de la Física, soportado sobre una bien concebida estrategia de aprendizaje (Candelario-Dorta, 2018).

Los estudiantes de estos años necesitan considerablemente mucho más la incorporación de herramientas de programación, de softwares, que sus predecesores, esto no solo apunta a cómo se ha mencionado a la formación de competencias sino a generar en estos estudiantes ventajas competitivas en el mundo que les toca afrontar.

Finalmente queremos aclarar que los autores han decidido dejar el código abierto para que cualquier docente/estudiante tenga acceso al mismo, reproducirlo y emplearlo en sus clases https://github.com/jpmartinezmanno/problemasFisica_Py.git

Agradecimientos

A la Facultad Regional Mendoza, Universidad Tecnológica Nacional que nos permite desarrollar estas actividades pedagógicas libremente.

Al Departamento de Materias Básicas, por apoyar siempre este tipo de iniciativas

Al grupo de investigación IEMI que es el espacio en donde desarrollamos estas investigaciones.

Referencias

- Ávalos, B. (2001). Mejoramiento de la formación inicial de profesores: conjunción de políticas nacionales e iniciativas institucionales. OEI., 5
- Bransford, J., Brown, A. y Cocking, R. (2004). How People Learn: Brain, Mind, Experience, and School. Washington DC: National Academy Press. DOI:10.17226/9853.
- Candelario-Dorta, O. (2018) El software en el proceso de enseñanza aprendizaje de la Física EduSol, vol. 18, núm. 63, Centro Universitario de Guantánamo, Cuba. Disponible en: <https://www.redalyc.org/articulo.oa?id=475756619014>
- Chickering, A. y Ehrmann, S. (1996). Implementing the Seven Principles: Technology as Lever. American Association for Higher Education, 3-6.
- Design Simulation Technologies, Inc.(s.f.) Interactive Physics. <https://www.design-simulation.com/IP/spanish/index.php>
- Diker, G. (2003). La formación de maestros y profesores: hoja de ruta. Buenos Aires: Paidós.
- Universidad de Colorado Boulder (s.f.) PhET Interactive Simulations.: <https://phet.colorado.edu/es/>
- Vaillant, D. (2009). Formación de profesores de Educación Secundaria: realidades y discursos. Revista de Educación, 105-122.
- Yasar, O. y Landau, R. (2003). Elements of computational Science and engineering Education. Society for Industrial and Applied Mathematics, 45(4), 787-805. DOI: 10.1137/S0036144502408075
- Young, H. D. y Freedman, R. A. (2013). Física universitaria volumen 1. Décimo tercera edición- Pearson, México, 2013. ISBN: 978-607-32-2124-5
- Zamarro, J., Molina, G. y Núñez, M. (2004). Teaching Physics Modelling with Graphic Simulations Tools. HSCI.

* * *