

Aprobado por la reunión del Grupo Temático "Cómputo General" en su reunión del 24/09/2019

Respuesta a crítica del profesor 1

1. Comentario sobre eliminación de seriación con 1112027 "Introducción al Cálculo": *"Sin embargo, es justo esta UEA en donde se presenta por primera vez a los alumnos de CBI el concepto de función (Tema 1 de su contenido sintético) el cual resulta relevante para la presentación del Tema 4 (Programación modular) de la UEA Programación Estructurada..."*

Respuesta: El concepto de función mencionado por el honorable profesor, en el contexto de la UEA "Programación Estructurada" puede ser fácilmente explicado en cinco minutos. Es decir, este punto tiene carácter secundario para justificar la seriación actual.

Objetivos de la UEA 1112027 Introducción al cálculo:

"Aplicar los conceptos de límite y continuidad para obtener y analizar la gráfica de una función real de una variable real"

"Obtener la ecuación de la recta tangente a una curva en un punto usando la definición de derivada"

Tal como se presentan en los objetivos anteriores, se propone analizar su continuidad y conocer su gráfica, en este sentido la UEA Programación Estructurada no contempla el "ploteo" o "graficación" de la función para ver su gráfica o profundizar en el conocimiento de su continuidad o derivabilidad.

En el comentario del "profesor 1" se presupone que en cursos de niveles previos (medio superior o secundaria) no se da a conocer el concepto de dominio-contradominio, como relación entre conjuntos o de variable real. Por tanto, no se debe presuponer que un alumno no pueda comprender la asignación de un valor contra otro, y que pueda hacerse uso de una computadora para ello.

2. Comentario del profesor 1 sobre Python: *"En la justificación se dice que "el número de aplicaciones implementadas en Python creció en las últimas décadas de una manera significativa". Esto mismo se puede decir de decenas de otros lenguajes de programación y no por eso se justifica que sean adecuados para el propósito de la UEA Programación Estructurada."*

Respuesta: Desgraciadamente para su punto de vista el honorable profesor 1 no cita ninguna fuente actualizada y veraz. Pero un simple análisis de la información existente sobre la dinámica de popularidad de diferentes lenguajes en <https://www.tiobe.com/tiobe-index/> demuestra que no existen "decenas de otros lenguajes" con esta propiedad.

Lo que se sugiere con la modificación no es que Python sea la panacea de solución (o único que sea el adecuado) en lenguajes de programación. Por el contrario retomando la afirmación del profesor 1, habiendo decenas de otros lenguajes de programación resulta inaceptable que el programa actual de PE solo contemple lenguaje C sin apertura a ningún otro, dejando a los egresados con “corta visión” ante un mercado laboral, evolutivo y competitivo.

3. Otro comentario del profesor 1 sobre Python: *“es inconcebible que se proponga dar un paso enorme hacia atrás”*

Respuesta: Por el contrario, esta modificación pretende dar un paso hacia adelante. El argumento sobre cambio climático e ineficiencia de Python no contempla el proceso enseñanza aprendizaje centrado en el alumno, como persona, ingeniero, y egresado para incluirse al campo laboral, en donde empresas líderes en apoyo a la mejora climática utilizan lenguajes (incluido Python) para crear sus soluciones.

Objetivos de la DCBI UAM -A:

“Actuar con conciencia de los efectos de las obras de ingeniería en el medio que lo rodea.”

“Considerar en el análisis y solución de problemas, factores técnicos, ambientales, sociales y económicos.”

Se debe considerar no solo el tiempo efectivo de ejecución del programa, a esto se deben considerar factores como el tiempo que un alumno (o profesional) mantiene encendida su computadora diseñando e implementando su programa que es mayor, dado el nivel de abstracción de C.

Bajo el concepto energético, sería más conveniente lenguaje ensamblador, o implementaciones por hardware, sin embargo, el tiempo de desarrollo, facilidad de mantenimiento, cometer errores, detectar errores, costos de mantenimiento, etc. Es que han llevado a la humanidad a crear lenguajes de mayor nivel de abstracción (con herramientas de depuración mas amigables, multiplataforma) y más aún, en particular Python de manera más natural que lenguajes como C, facilita el uso de múltiples núcleos, nuevas arquitecturas de procesadores, con APIs como Tensorflow.

4. Comentario del profesor 1 sobre propósitos de la UEA Programación Estructurada: *“... hay que recordar que esta UEA tiene dos propósitos: uno general de formar ingenieros y otro específico de formar ingenieros en computación”*

Respuesta: Esta UEA pertenece al Tronco General. Los objetivos generales de la UEA no corresponden a los propósitos mencionados por el profesor.

5. Comentario del profesor 1 sobre ineficiencia de Python: *“ En un mundo que se enfrenta ya a las consecuencias del cambio climático y en el que las universidades tienen la enorme responsabilidad de proveer soluciones de ingeniería que disminuyan el consumo de energía, ...”*

Respuesta: Compartimos completamente la preocupación del honorable profesor sobre el medioambiente. Sin embargo, para los ejecutables de programas didácticos escritos en Python las pérdidas de energía serán casi invisibles. Por otro lado, cabe mencionar que el honorable profesor olvidó que el problema de conservación del medioambiente depende de varios factores. Es muy probable que el ahorro en el consumo de energía por el tiempo gastado por los desarrolladores de programas, tanto los alumnos como los profesionistas, será mucho más significativo que las pérdidas de energía por ineficiencia de los ejecutables.

Además, al respecto, podemos referir a una opinión del experto en el área: Para aquellos quien busca dialectos de Python más rápidos, pueden usar, por ejemplo, Cython o Julia. En <https://hackernoon.com/why-is-python-so-slow-e5074b6fe55b> el autor, Anthony Shaw, indica que Cython acelera programas en Python en 84 veces. En tanto que Julia, se ejecuta inclusive sobre GPUs.

6. Comentario del profesor 1 sobre falta de estándares.

Respuesta: No es la primera vez en la historia de computación que en las más demandas y más avanzadas áreas de la tecnología de información los estándares no alcanzan a correr con la misma velocidad como la práctica. Por ejemplo, durante las últimas décadas la situación mucho más dramática la observamos sobre los estándares en el área de Gráficas por Computadora. Pero esto no debe frenar el progreso: consideramos que sería un error obligar a que nuestros alumnos esperen la publicación de una versión final de estándares de Python.

Se habla sobre estándares de C/C++ indistintamente. En la UEA Programación Estructurada se especifica el lenguaje C (no C++). Se debe tener cuidado de ello, ya que C++ es más adhoc a la POO.

Python tiene y tendrá cambios, “lo único constante es el cambio” pero los conceptos fundamentales que es el soporte de los 3 tipos de instrucción secuencial, condicional e iterativa los tiene (como otros lenguajes de programación) por lo que más que apegarse a un estándar (que si bien es importante) se debe considerar la factibilidad de implementar algoritmos bajo el paradigma de programación estructurada, bajo este concepto software como Raptor (<https://raptor.martincarlisle.com/>) que está instalado en las salas donde se imparte Programación estructurada permite ilustrar y ejecutar programas con diagramas de flujo. Más aún, la existencia de software como Structorizer (<https://structorizer.fisch.lu/>) muestra que es posible enseñar el uso de instrucciones secuenciales, condicionales y de flujo para resolver problemas y muestra la generación automática del código fuente a diferentes lenguajes de programación lo que manifiesta la diferencia entre la programación estructurada como forma de pensamiento, y la implementación en un lenguajes específico (dígase C por ejemplo).

Lenguaje C también tiene serias deficiencias motivo por el cual no es el lenguaje preferido en muchos contextos, en el contexto de programación web los programas de CGI se dejaron de usar porque representa una gran vulnerabilidad en seguridad informática al trabajar directamente sobre el hardware y dar acceso a instrucciones privilegiadas (peligrosas) pero más relevante sería

ver el contexto académico en donde, por ejemplo, lenguaje C sobrecarga de operador * el cual confunde demasiado ya que se puede usar para multiplicar, definir apuntadores a memoria, recibir parámetros de dirección de memoria. Lenguajes como Java minimizan su uso, y Python se une a esta idea.

La argumentación del profesor parece centrarse en monopolizar y presentar a lenguaje C como único lenguaje eficiente, sin embargo debe considerarse que este lenguaje al ser de bajo nivel de abstracción se enfoca al uso eficiente del hardware y no en el proceso de enseñanza-aprendizaje y es justo donde el colectivo de profesores del grupo temático contempla la incorporación de Python (adicional a C) es decir, la propuesta dice "lenguaje C o Python".

6. Comentario del profesor 1 sobre incompatibilidad con otros programas de estudio

Respuesta: En la propuesta de actualización del programa del curso se propone abrir los grupos tanto para lenguaje C como para Python. Cómo precisamente hacerlo -- no es una cuestión del programa sinóptico sino operacional: ¿cómo equilibrar apertura de los grupos y realizar la inscripción de alumno al grupo con un lenguaje preferible o mejor compatible con el programa de estudio. Por otro lado, si los futuros ingenieros en computación, electrónica, o física, es decir los alumnos de las carreras más ligados a la programación práctica, aprenderán Python, entonces el aprendizaje autodidacto del lenguaje C sería la cuestión de un par de semanas.

Sobre la problemática que tendrían los alumnos en cursos posteriores, si bien podría presentarse esta situación, se considera que los cursos en los que podría repercutir el lenguaje impartido en Programación Estructurada más que enfocarse en enseñar sus temas basados en un lenguaje en particular, están centrados en enseñar las bases, los conceptos y los algoritmos de los distintos temas que ahí se imparten.

Por ejemplo, el curso de Algoritmos y Estructuras de Datos no debe tener como fin enseñar la implementación de los diferentes tipos de listas en un lenguaje en particular (sea C, C++, Java, etc) si no en los conceptos de Listas, sus algoritmos para inserción, recorridos, búsquedas, eliminación, etc y que el alumno pueda implementarlos en algún lenguaje que él maneje. Si bien el profesor podría hacer énfasis en mostrar cómo se implementan en un lenguaje en particular, al final la implementación es un reflejo de los conceptos y del algoritmo.

7. Comentario del profesor 1 sobre eliminación de UNIX: *"Esto podría ser acertado, pero por las razones incorrectas. Es decir, lo correcto sería reemplazar "ambiente de desarrollo UNIX" por "ambiente de desarrollo POSIX"..."*

Respuesta: Consideramos que al eliminar "UNIX" del programa de la UEA "Programación Estructurada", sus alternativas no deben ser analizadas en el marco de la discusión actual. Sin embargo pensamos que no hay ninguna alternativa. De hecho, todos conceptos incluidos en POSIX necesarios para la UEA "Programación Estructurada" se cubren completamente por el concepto de la interfaz gráfica amigable. Dicho concepto es universalmente aceptado y puede ser

explicado a los alumnos en cinco minutos. Consideramos que no es correcto sobrecargar el curso introductorio a la programación con estándares o protocolos como POSIX. La tendencia de incluir un estándar o protocolo en un programa de estudios existe en varias UEA del tronco profesional, por ejemplo en el Departamento de Electrónica, pero no es una buena idea para un curso introductorio a la programación del tronco general.

Coordinador del Grupo Temático

Dr. Gueorgi Khatchatourov