



Boletín de Ciencias de la Tierra

ISSN: 0120-3630

Universidad Nacional de Colombia

Suarez-Burgoa, Ludger O.

Nota corta: publicación de programas computacionales aplicados a las Ciencias de la Tierra

Boletín de Ciencias de la Tierra, núm. 45, 2019, Enero-Junio, pp. 22-25

Universidad Nacional de Colombia

DOI: <https://doi.org/10.15446/rbct.n45.71742>

Disponible en: <https://www.redalyc.org/articulo.oa?id=169559150003>

- ▶ [Cómo citar el artículo](#)
- ▶ [Número completo](#)
- ▶ [Más información del artículo](#)
- ▶ [Página de la revista en redalyc.org](#)

UNAM 

Sistema de Información Científica Redalyc

Red de Revistas Científicas de América Latina y el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Short note: publish computational programs applied to Earth Sciences

Ludger O. Suarez-Burgoa

Departamento de Ingeniería Civil, Universidad Nacional de Colombia, Medellín, Colombia. losuarezb@unal.edu.co

Received: April 16th, 2018. Received in revised form: October 11th, 2018. Accepted: November 25th, 2018

Abstract

In the era of the big data, the art of creating an applied computational program to develop a calculation procedure in Earth Sciences has become one skill more among the abilities of the scientist. This short technical note comments on the minimum rules the author must be present to divulge computational codes applied to the Earth Sciences. Finally, the author encourages every geoscientist to become a code developer.

Keywords: applied computational program; computer science, earth sciences.

Nota corta: publicación de programas computacionales aplicados a las Ciencias de la Tierra

Resumen

En la era de los grandes datos, el arte de crear un programa computacional aplicado para desarrollar un procedimiento de cálculo en las Ciencias de la Tierra se ha convertido en una habilidad más para el científico. La presente nota técnica corta comenta acerca de las mínimas reglas que se debe tomar presente para publicar código computacional aplicado a las Ciencias de la Tierra. Finalmente, el autor incentiva a todo geocientífico para que se convierta en un productor de códigos aplicados.

Palabras clave: programa computacional aplicado; ciencias de la computación; ciencias de la tierra.

El Boletín de Ciencias de la Tierra (BCT) fue creada con el fin de difundir la investigación de las *Ciencias de la Tierra* (CT) inicialmente en Colombia. Hoy en día este boletín ha trascendido y publica sobre CT más allá de su inicial frontera. Este fenómeno es parecido a la trascendencia de las *Ciencias de la Computación* (CC) hacia las CT, que es lo que aquí se escribe brevemente. Las CC están ahora inmersas en toda ciencia, en toda disciplina, en la academia superior, en la industria, en la educación escolar y en la familia. Ha trascendido de forma notoria en los últimos cinco años a similar tasa a la que predijo Gordon Moore para el caso de la tecnología electrónica.

Por ejemplo, hace no más de cinco años en los pasillos de las universidades de Colombia (tomando como referencia la UNAL), los estudiantes de cada disciplina estaban abocados a sus temas particulares de su disciplina y solo en las carreras relacionadas a los sistemas computacionales se tenía la

preocupación del desarrollo de una herramienta computacional profesional y comercialmente competitiva.

El proceso de desarrollo de un programa (bajo el concepto de la década de los noventas del siglo pasado y el primer decenio de siglo XXI) estaba inicialmente reservado sólo a los profesionales que hayan cursado y tenido experiencia relacionados a las CC; aquí el fin ha sido crear un negocio entorno al software. Ahora, casi finalizando el segundo decenio de este siglo, se ve una proporción de docentes y estudiantes (no relacionados a las CC) con la satisfacción de haber codificado un programa computacional aplicado, destinado a resolver un problema dentro de su amplio campo de la disciplina que la domina. No es un programa industrial ni comercial, es una aplicación particular.

Si está leyendo esta nota corta es muy probable que sea un *geocientífico* que ha alcanzado el punto tal de tener un *programa computacional aplicado* (PCA) de su autoría. Pero posiblemente su PCA resuelve un problema trivial dentro de

How to cite: Suarez-Burgoa, L.O., Nota corta: Publicación de programas computacionales aplicados a las ciencias de la tierra. Boletín de Ciencias de la Tierra, 45, pp. 22-25, Agosto 2018 - Enero 2019.

su campo y/o posiblemente es un programa que no abarca más del diez por ciento de lo que un programa comercial profesional lo hace; pero al final de cuenta es su obra de arte, es la transmisión de su pensamiento plasmado en un código que sí funciona. Sin embargo, dentro de este su código puede estar plasmado una mejor alternativa de solución computacional o algorítmica que merece ser publicada para beneficio de su disciplina. Sin embargo, puede ser una información efímera y desordenada o una recolección de códigos tomados de otros autores. Por tanto, es necesario ser selectivos y analizar qué se puede publicar.

Es bueno ahora enmarcar algunas reglas que siempre existieron en las CC para publicar sobre desarrollos computacionales aplicados; y se hará para el caso de las CT.

Si uno va entregar un PCA a la comunidad que no es parte de las CC, se aconseja hacer la documentación tal que contenga los siguientes aspectos (adaptado de [1], página VIII en contenidos).

1. Lo que resuelve o soluciona el PCA y su aplicación: describa la aplicabilidad y alcance del algoritmo.
2. Antecedentes teóricos y formulismo de la solución: describa brevemente la teoría y las ecuaciones relevantes.
3. Lista formal de variables de entrada y salida: especifique detalladamente todas las variables de entrada y salida.
4. Los algoritmos que componen el PCA: detalle cada algoritmo implementado en pseudocódigo o en el lenguaje de programación escogido, explique y comente sobre los aspectos especiales del algoritmo y muestre con algunas corridas los algoritmos que ya se usan con preferencia para solucionar cada problema; compare con otras posibles soluciones de otros autores. También detalle lo innovador de sus nuevos algoritmos y compare su eficiencia computacional. Asimismo, referencie y haga un análisis de errores y eficiencias de los mismos.
5. La estructura del PCA: describa el módulo, las clases, los objetos, los métodos y las funciones que lo componen. También liste las librerías de las que el PCA depende y su compatibilidad en cuanto a licencias; aspecto importante, ya que todo el PCA debe usar librerías con licencias compatibles.
6. Las pruebas, los ejemplos y la validación: presente los resultados de las pruebas realizadas; presente ejemplos sobre el uso del algoritmo con el fin de facilitar el correcto uso del mismo. Tanto las pruebas, ejemplos como la validación deben tener un enlace en Internet de libre descarga donde se pueda descargar los datos de entrada para la réplica por parte del lector.
Las pruebas, los ejemplos de preferencia, deben ser tomadas de artículos con datos reales. Para la validación, esta condición es obligatoria; no se puede validar con ejemplos hipotéticos sino entraríamos en un mundo de fantasía numérica.
En lo que respecta la publicación del código computacional se debe proporcionar al lector los siguientes.
1. El código abierto: coloque su código en un repositorio de desarrollos computacionales (e.g. GitHub, SourceForge, Bitbucket, entre otros) y publique la ruta de acceso a ese repositorio.
2. El lenguaje de programación: cada lenguaje de programación tiene su norma de estricto cumplimiento en cuanto a la forma de escribir un código (por ejemplo, si usa el lenguaje C debe usar ANSI-C, si escribe en lenguaje Python use las recomendaciones PEP8). Los comentarios dentro del código es lo más importante para la publicación del mismo y deben ser escritos obligatoriamente en inglés aun cuando su interfaz de usuario este pensado a servir en otro idioma.
3. Autores: todo desarrollo es hecho por personas, por tanto, en cada archivo que define el código computacional se debe colocar el autor o los autores y sus respectivas afiliaciones. No puede ponerse el nombre de un grupo de trabajo o una empresa, el grupo o la empresa como tal no son personas. Un programa es un desarrollo artístico, es una forma de expresión humana, es como una canción, una obra literaria o una pintura.
4. Poseedores: los poseedores de los derechos patrimoniales sí pueden pertenecer a un grupo o empresa, un grupo de desarrollo, una universidad, etc.; se debe colocar los poseedores de los derechos del programa.
5. Licencia: es muy importante elegir el tipo de licencia del programa, si no se coloca explícitamente el tipo de licencia su PCA se convertirá automáticamente en un programa de libre dominio; al ser de dominio público tanto los autores como los poseedores de derecho podrían ser ignorados sin ninguna responsabilidad por parte del que lo omitió.
6. Versión: la versión del PCA: normalmente será un número menor a la unidad (e.g. 0.1.4) por ser un desarrollo nuevo; pero la versión tiene mucho que ver con la estabilidad del programa y eso solo se observa si muchos usuarios la están usando y si han reportado *gusanos* que han sido ya resueltos.
7. Descargo de responsabilidad: publique los términos y límites de responsabilidad de los creadores del PCA; los descargos de responsabilidad es un texto donde se define explícitamente que los desarrolladores del PCA no se hacen responsables del uso que se le pueda dar al PCA y a veces puede evitar acciones legales innecesarias.
8. Documentación: si no se tiene una documentación que explique detalladamente lo que cada variable es, lo que cada método ejecuta o la estructura de los objetos, entonces solo los autores entienden su propio programa y nadie más; por tanto, la documentación es esencial y obligatoria.
9. Metadatos del PCA: en forma de metadatos se debe colocar lo más sintético posible lo descrito en los ítems de arriba: nombre, versión, enlace permanente (repositorio), licencia, sistema de administración del programa, lenguaje de programación, requisitos para su ejecución o compilación con lista de librerías dependientes, requisitos de plataforma (i.e. sistema operativo y arquitectura del hardware), enlace a la documentación, correo electrónico de los autores.
Es importante resaltar que lo que se quiere promover como publicación es la forma cómo se resuelve un problema no el aspecto estético del PCA; por tanto, el PCA no

necesariamente debe tener una interfaz gráfica de usuario. Bajo el mismo término, un PCA publicable no es un archivo ejecutable .exe encriptado, donde el lector no pueda tener acceso al código fuente.

En este aspecto ¿será bienvenida una publicación de un PCA que hace de todo, que sea profesional? –Posiblemente ¡no! porque al hacer de todo no puede ser descrito en un artículo, al ser profesional tiene grandes posibilidades de ser comercial, y al ser comercial no estaría disponible el código para entenderlo, estudiarlo, modificarlo y mejorarlo. Este espacio no sería el lugar para publicitar sobre el PCA sino es el espacio para promocionar el estudio y comprensión de los algoritmos que resuelven uno o varios problemas aplicados a las CT.

También se resalta aquí sobre la publicación de aquellos PCA a las CT que solucionan un problema en particular (no la colección de problemas que hacen todo un proceso). Bajo este término se puede caer en programas triviales que hacen un cálculo tan sencillo como resolver una ecuación algebraica empírica aplicado a las CT. Para limitar este aspecto, el PCA útil de difundir en una publicación tiene que ser uno tal que requiera necesariamente una carga computacional que no pueda ser reemplazada por un cálculo manual o una solución gráfica como un ábaco.

El PCA además debe enseñar al geocientífico cómo calcular, de tal forma que cuando se publique su código abierto se transforme en una herramienta más de las múltiples herramientas computacionales que pueda tener el usuario para lograr la interacción con el grupo de investigadores de otras ciencias que necesitarán que su resultado sea proporcionado bajo los requisitos actuales enmarcados en las CC. También tome en cuenta que al mencionar un *problema particular* no se refiere a que tenga una única función computacional, ya que hoy en día un problema en específico se resuelve con una serie de funciones computacionales; más aún, uno o más módulos, conteniendo en cada uno de ellos: clases, objetos y métodos.

¿Ya existe un PCA similar desde hace años? --Sí, es muy probable que su desarrollo computacional que resuelve un problema ya exista desde hace años; sin embargo, no lo resuelve de la manera que ahora se hizo, no bajo el paradigma computacional que ahora empleó o no en el lenguaje de programación que hizo (los más importante es que sea su pensamiento, no la copia de otro código u otro algoritmo).

Pero además, es muy probable que lo que se hizo hace años esté bajo un código totalmente cerrado por las restricciones de la mentalidad comercial; es decir, no es a lo mínimo un código abierto.

Si hacemos una analogía con una pintura artística de un caballo, pregúntese ¿Cuántos caballos han sido pintados por tantos artistas o amadores? Simplemente ponga en la Internet las palabras clave *famous artistic paintings of horses* y verá que hay al menos 22 pinturas de caballos hechos por artistas famosos y ahora piense cuántos más no famosos cuadros de caballos existe en la expresión de las humanidades. No espere a verificar si su cuadro de caballo es el más innovador o si nadie se ha puesto a pensar (más que usted) a pintar por primera vez un caballo.

Lo que se quiere aquí es difundir nuevas expresiones de pensamiento computacional aplicados a las CT pero que estén regidas por la documentación necesaria y con el mínimo de condiciones para que sea útil a los lectores. Al tener expresiones de pensamiento de diversas formas de solucionar computacionalmente problemas en CT, con el tiempo, se podrá cubrir gran parte de los requerimientos de cálculo de estas ciencias. Con esto es muy alentador que a corto plazo se tenga una colección de módulos (de licencia libre) que se puedan acoplar y adaptar fácilmente para formar una solución más extensa y compleja. Posiblemente su solución será igual de competente (o hasta mejor) que el programa profesional completo y sofisticado que le cuesta a su institución anualmente por la compra de su licencia limitada a pocos usuarios.

Una última pregunta que se puede hacer el geocientífico es ¿Qué estoy programando: *software*, códigos, rutinas, programas o qué? –Se puede decir que lo que un geocientífico puede lograr con mucho esfuerzo bajo las recomendaciones de arriba es hacer un Software Aplicado o Software de Aplicación (pero no un Software como tal); según la definición presentada en el Glosario.

¿Quiénes serían los interesados en estas publicaciones? – Sería todo geocientífico que tiene conocimiento para programar o que desea aprender a programar y crear herramientas computacionales para las CT. Los PCA que se publicarían no serían en primera instancia productos de aplicación industrial para el usuario inexperto con necesariamente una interfaz gráfica; esto porque la interfaz gráfica restringe al usuario a un tipo de análisis y lo que se quiere es más bien abrir hacia la creatividad.

El desarrollo de PCA bajo el contexto mencionado arriba ya se está haciendo a nivel mundial, eso porque hoy en día la tendencia es hacer PCA con capacidades intercambiables, flexibles, fáciles de acoplar a otras implementaciones de PCA en distintos intérpretes y multiplataforma.

Una revista internacional que publica PCA bajo estos términos es SoftwareX. Debido a que en el BCT somos relativamente nuevos en la promoción de este tipo de artículos, recomiendo que visite y explore el artículo de la referencia [2], también puede consultar artículos de PCA que ya se han publicado en el BCT y otras revistas latinoamericanas en temas relacionados a las CC; por ejemplo, refiérase a [3, 4].

Que los aportes de los autores latinoamericanos en el desarrollo de PCA para las CC no se quede atrás. Se espera que cada vez más se vea publicaciones de PCA en el campo de las CC y el Boletín de Ciencias de la Tierra es una alternativa.

Glosario de términos

Software. Como software se define al conjunto de programas computacionales, procedimientos, documentación asociada y datos pertinentes para la operación de un sistema computacional [6].

Software de aplicación. Es aquel conjunto de programas computacionales diseñado para ayudar a los usuarios a llevar

a cabo tareas específicas o encargarse de ciertos tipos de problemas, distinto de aquel software que controla el computador [6].

Programa computacional. Es una unidad sintáctica que usa un particular lenguaje de programación y que está compuesto por declaraciones, sentencias condicionales e instrucciones necesarias para lograr una o varias tareas o la solución de uno o más problemas, adaptado de [5].

Programa computacional aplicado. Es una versión menos sofisticada y elaborada de un software de aplicación basado en la definición de programa computacional. Lo que convierte un programa computacional aplicado en un software de aplicación son los siguientes insumos adicionales: programas que ayudan a ordenar y unificar un conjunto de programas, interfaz gráfica y documentación asociada (i.e. a parte de los manuales de usuario, páginas web con ayudas tipo Wiki, ejemplos de descarga, ayuda al usuario).

Código abierto. Es un conjunto de archivos no encriptados y almacenados en formato legible para cualquier editor de texto escritos en lenguajes de programación estándares de alto nivel y de dominio público; archivos que conforman la base para crear los ejecutables de un PCA.

Código libre. Es un código abierto que tiene una licencia que posibilita a cualquier persona de leer, estudiar, entender y modificar el código sin restricciones de licencia que les impida crear a partir de ello una nueva implementación computacional bajo una licencia también del tipo libre.

Programa libre. Programa que tiene código abierto y libre.

Programa gratis. Programa que no necesariamente es ni abierto ni libre pero que no tiene costo para ejecutarlo.

Interfaz gráfica de usuario. Un entorno normalmente gráfico que ayuda a interactuar un usuario con un PCA de forma de hacer el trabajo humano intuitivo sin necesidad de entrar a detalles de programación computacional.

Referencias

- [1] Wilkinson, J.H. and Reinsch, C. Handbook for automatic computation, volume 2. Springer, Berlin, [online]. 1 edition, 1971. Available at: <https://www.springer.com/la/book/9783642869426>
- [2] Montoya-Araque, E.A. and Suarez-Burgoa, L.O., pyBIMstab: application software for 2D slope stability analysis of block-in-matrix and homogeneous materials. SoftwareX, [online]. 7, pp. 383-387, 2018. Available at: <https://www.sciencedirect.com/science/article/pii/S2352711018301638>
- [3] Suarez-Burgoa, L.O., Geonetdigitizer: código abierto para digitalizar información plasmada en mallas Wulff y Lambert. Bol. Cienc. Tierra, [online]. 37, pp. 25-34, 2015. Disponible en: <https://revistas.unal.edu.co/index.php/rbct/article/view/43392/61376>
- [4] Montoya-Araque, E.A. y Suarez-Burgoa, L.O., Programa en código abierto para el análisis bidimensional de estabilidad de taludes por el método de equilibrio límite. Rev. Fac. Cienc. [en línea]. 5(2), pp. 88-104, 2016. Disponible en: <https://revistas.unal.edu.co/index.php/rfc/article/view/59914>
- [5] ISO/IEC. International standard information technology vocabulary. Technical Report 100.64930555556, International Organization for Standardization, International Electrotechnical Commission, Geneva Switzerland, 2015.
- [6] ISO/IEC/IEEE. International standard systems and software engineering vocabulary. Technical Report 24765:2017(E), International Organization for Standardization, International Electrotechnical Commission, Institute of Electrical and Electronics Engineers, Geneva - Switzerland, Aug 2017. DOI: 10.1109/IEEESTD.2017.8016712.

L.O. Suárez-Burgoa, es graduado como Ing. Civil en el 2001 de la Universidad Mayor de San Andrés, La Paz, Bolivia, MSc. en Ingeniería Geotécnica el 2008 de la Universidad Nacional de Colombia, Medellín Colombia y PhD. en Geotecnia el 2012 de la Universidad de Brasilia, Brasil. En la actualidad es profesor asociado en dedicación exclusiva del Departamento de Ingeniería Civil de la Facultad de Minas, de la Universidad Nacional de Colombia en Medellín. Ha desarrollado actividades como consultor e investigador por más de diez años en el campo de la geotecnia civil y minera en Bolivia, Chile, Brasil y Colombia. Actualmente dirige el Semillero de Geología Matemática y Computacional del Grupo de Investigación de Geotecnia de la Facultad de Minas, habiendo desarrollado software de aplicación para la geología.
ORCID: 0000-0002-9760-0277



UNIVERSIDAD NACIONAL DE COLOMBIA

SEDE MEDELLÍN
FACULTAD DE MINAS

Área Curricular de Ingeniería
de Sistemas e Informática

Oferta de Posgrados

Especialización en Sistemas
Especialización en Mercados de Energía
Maestría en Ingeniería - Ingeniería de Sistemas
Doctorado en Ingeniería- Sistemas e Informática

Mayor información:

E-mail: acsei_med@unal.edu.co
Teléfono: (57-4) 425 5365