

Ciencia y Luz

75
ANIVERSARIO
Universidad Veracruzana
1944-2019

Universidad Veracruzana
Ciencia UV

LA PROGRAMACIÓN DE COMPUTADORAS ES UN ARTE... **MARCIAL**

Autor: Wulfrano Arturo Luna Ramírez*
Edición: Dirección de Comunicación
de la Ciencia, UV
Correo: dcc@uv.mx

Nada es imposible para una mente dispuesta a (re)programarse

UNA MIRADA DISTINTA DE LA ENSEÑANZA DE LA PROGRAMACIÓN

Indispensable en el mundo actual, pero con mayor énfasis en el caso de las estudiantes de tecnologías de la información y áreas afines, la enseñanza y aprendizaje de la programación de computadoras representa todo un reto para docentes y alumnos. Presenta altos índices de reprobación, y frecuentemente está seriada con otras, lo que la hace crucial para los currículos de los informáticos y tecnólogos de la información.

Si la programación de computadoras es un arte, como apunta Donald Knuth en su influyente libro *El arte de la programación*, la ciberactivista y escritora "St Jude" (Jude Milhon), postulaba el "hacking" como arte marcial: la necesidad de defenderse en un mundo de intromisiones múltiples, siguiendo los métodos de defensa del Aiki-Do (<https://www.wired.com/1995/02/st-jude/>).

Además, al puro estilo del aprendizaje de un arte marcial, la programación de computadoras es una actividad que requiere una sólida formación teórica de los principios básicos y una práctica constante de ellos, a fin de conseguir el nivel mínimo de competencia requerido por el ejercicio de la profesión.

Por su parte, Dorothy Denning por igual, menciona las similitudes entre enseñar karate y computación (<http://cpsr.org/prevsite/cpsr/privacy/crime/denning.hackers.html>). Adicionalmente, Emily Bache propone un manual para un Dojo de programación (<https://leanpub.com/codingdojohandbook>), si bien su enfoque es la industria no en la enseñanza-aprendizaje como aquí se postula.

La idea primordial se sintetiza en concebir la solución de problemas mediante programación como un combate frente a un oponente (el problema a resolver), para el que las técnicas de defensa y ataque son las herramientas y fundamentos teóricos de la programación (la estrategia de solución, su diseño, codificación y pruebas).

APRENDER LA TÉCNICA DE LA DEFENSA Y EL ATAQUE

Tomemos por caso el Karate-Do o "camino de la mano vacía" (no utiliza armas para el combate), cuya práctica incluye "kata" y "kumite". En el pensamiento japonés, la "kata" es una manera de hacer bien las cosas, un método. En Karate-Do refiere a "un combate imaginario" expresado como secuencias prediseñadas de movimientos. Su práctica pule la técnica por repetición, hasta lograr la mayor pericia posible.

Una metáfora imperfecta: la repetición no de los problemas (un equívoco y un desperdicio de tiempo), sino de la técnica para obtener las soluciones y las buenas prácticas para lograrlas.

El kumite o combate significa "entrelazar, cruzar, unir las manos". Es la aplicación práctica de la técnica en un enfrentamiento real contra un oponente. El kumite como resolución de un problema, de manera competitiva: bajo el binomio programación-pruebas y el análisis de algoritmos. O de manera cooperativa: bajo el principio de modularidad. Bajo esta óptica, habría distintos tipos de combate:

1. De estudio y aplicación de las técnicas de las katas, bajo ciertas reglas acordadas por el docente. Aplicado en la comparación de técnicas de programación, la modularización de programas y el

desarrollo de pruebas.

2. De contacto pleno: aplicación libre de las técnicas, restringida sólo por la óptima solución al problema planteado. Aplicación: diseño, programación, pruebas y refactorización.

EN BUSCA DEL... "DO"

Las artes marciales tradicionales japonesas incluyen en su nombre el concepto de "Do" o camino. Esto representa lineamientos ideológicos y propósitos que abarcan la formación técnica y ética del practicante.

El "Dojo" o "espacio para el Do", es el lugar donde se realiza el entrenamiento. En programación comprende el espacio físico destinado a la práctica (un laboratorio o el área de trabajo personal) y las herramientas de cómputo, es decir el "Ambiente de Desarrollo", que incluye:

- Herramienta de edición: para crear los diagramas de diseño.
- Herramientas de edición de código: para transcribir los programas desarrollados.
- Compilador o intérprete: para producir el archivo que la computadora ejecuta finalmente.
- Manejador de versiones: para organizar archivos y administrar los cambios en ellos.

Finalmente, El "Dojo-kum" en las artes marciales es un código de ética que trasciende al dojo, aplicable en la vida cotidiana de todo practicante. Asimilándolo a los códigos de ética de los "hackers" y profesionales de la informática y la computación, consideremos los siguientes puntos:

1. Esforzarse siempre y actuar con auto disciplina. Un programador competente velaría por el aprendizaje continuo y la actualización constante.
2. Incentivar la creatividad. Tras entender un programa, el programador debe diseñar la solución; esto pasa por imaginarla, detallarla y luego codificarla. Programar es crear, y ello implica motivar por distintos medios el pensamiento creativo.
3. Incentivar el pensamiento crítico. Análogo al punto anterior, todo buen programador es crítico de las soluciones provistas, sobre todo por las que él mismo obtiene.
4. Nunca plagiar el trabajo de otros. Emplear fragmentos o programas completos desarrollados por otros, sin permiso y sin citar la fuente, es un delito y entorpece la mente: usar código en vez de desarrollarlo provee una solución, pero a costa de no aprender, lo que es inaceptable.
5. Procurar la reutilización de código. A diferencia del plagio, es lo que ha llevado al software a escala industrial. Los bloques constitutivos de todo sistema son partes que, una vez programadas, deben aplicarse según se necesite, y no perder tiempo haciendo todo otra vez.
6. Buscar la optimización del código y refactorizar siempre. No basta tener una buena solución, hay que buscar la mejor. De ello nacen la prevención y detección de errores, la depuración y la reescritura de código.
7. Guardar estricta etiqueta de programador. Esto es una manera de seguir buenas prácticas de ingeniería de software, políticas de desarrollo y metodologías.

* INVESTIGADOR ADSCRITO A TECNOLOGÍAS DE LA INFORMACIÓN,
UAM-CUAJIMALPA
CORREO: wulfranoarturo@gmail.com

