



Education in the Knowledge Society

E-ISSN: 2444-8729

fma@usal.es

Universidad de Salamanca

España

Ferrer Rojas, Antoni

VirPLC: una metodología para el desarrollo de capacidades, habilidades y autoestima  
mediante la estimulación de la lógica con una herramienta sencilla, funcional y  
dinámica

Education in the Knowledge Society, vol. 18, núm. 2, 2017, pp. 59-69

Universidad de Salamanca

Salamanca, España

Disponible en: <http://www.redalyc.org/articulo.oa?id=535554766004>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

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

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

# VirPLC: una metodología para el desarrollo de capacidades, habilidades y autoestima mediante la estimulación de la lógica con una herramienta sencilla, funcional y dinámica

## VirPLC: a Methodology to Developing Capacities, Skills and Self-esteem by the Logical Stimulus with a Simple, Functional and Dynamic Tool

Antoni Ferrer Rojás

Instituto de Palamós, Girona, España, [aferrer8@xtec.cat](mailto:aferrer8@xtec.cat)

### Resumen

La proliferación de entornos de programación como Logo, Minecraft, Code o Scratch es consecuencia de la efectividad de un lenguaje gráfico para la introducción del alumnado en la programación, pero algunos de los sistemas más novedosos, como App Inventor, resultan lentos en la interacción fundamental: "haz" / "prueba". Propongo una alternativa enfocada a alumnado a partir de los 15 años. Se llama VirPLC y se orienta a algo tan funcional como el control de sistemas mediante dos pantallas: una de *software*, para programar, y una de *hardware* animado, para simular. VirPLC no pretende convertir al usuario en experto en automatización, sino facilitar un primer contacto entre el alumno y la lógica, mediante propuestas de problemas en sistemas de control prácticos, cercanos y reales como: mando de grúa, concurso de TV, alarma, puerta de supermercado, puerta de garaje, semáforo, ascensor... El/la alumno/a plantea tanto la operatividad lógica, como los requisitos de *hardware* (entradas y salidas). El *software* se trabaja de forma evolutiva y reiterada "haz" / "prueba" y consolídalo para mejorarlo. Además, permite la depuración de cada proyecto o "reto" en distintos niveles de perfeccionamiento, solidez, funcionalidad, seguridad y versatilidad. VirPLC es una alternativa que motiva al alumnado inquieto y luchador, para aplicar, tras unas nociones de Álgebra de Boole y antes de a la programación escrita, con objetos, eventos y clases. VirPLC es *freeware*, trabaja bajo Windows (desde XP hasta W10) y se puede descargar desde la WEB del autor. Al instalarlo, añade una carpeta con más de 30 ejemplos a menudo deliberadamente incompletos, junto con un minicurso con propuestas prácticas y variadas.

### Palabras Clave

Innovación; metodología; lógica; funcionalidad; programación; gráfica; PLC

Recepción: 25-04-2017

Revisión: 10-05-2017

### Abstract

The proliferation of programming environments such as Logo, Minecraft, Code or Scratch is a consequence of the effectiveness of a graphic language for the introduction of students in programming but, some of the newer systems, such as App Inventor, are slow in the basic interaction: "make it" / "test it". I propose an alternative for students from the age of 15. It is called VirPLC and it is oriented to something as functional as the systems control, through two screens: one with software to program and, another one with animated hardware to simulate it. VirPLC does not pretend to turn the user into an expert in automation, but to facilitate a first contact between the student and the logic world, by posing problems in practical, near and real control systems such as: crane control; TV competition; alarm; supermarket door; garage door; traffic light; lift... The student raises both the logical operativity, as well as the hardware requirements (inputs and outputs). The software works in an evolutionary and repeated way: "make it" / "test it" and consolidate to improve it. It allows the evolution in levels of greater complexity, where it is debugged until acquiring a solid "product", functional, safe, versatile and installable. VirPLC step by step "hooks" students who pursue challenges and offers an alternative to apply after some Boolean Algebra notions, and before written programming, with objects, events and classes. VirPLC is freeware, it works under Windows (from XP to W10) and can be downloaded from the author's WEB. At install time, it adds a folder with more than 30 examples often deliberately incomplete, along with a mini-course with several proposed practices.

### Keywords

Innovation; logical; methodology; functionality; programming; graphic; PLC

Aceptación: 25-05-2017

Publicación: 30-06-2017

---

# 1. Introducción : crítica constructiva de la innovación por la innovación

Ante todo, para desmentir algunos tópicos y en denuncia a determinados proyectos de innovación educativa, cabe constatar la frecuencia en que se incurre en la extravagante moda de premiar proyectos de innovación simplemente por el hecho de serlo. En este proceder a menudo se suelen cometer los siguientes errores:

- Se confunde tecnología con informática, que algunos llaman TIC, otros, “Nuevas Tecnologías”, prescindiendo de la ingeniería de telecomunicaciones, omitiendo la evolución de los turborreactores, los chips de electrónica, las centrales de cogeneración... (Márquez, Garrido, & Moreno, 2006), (ORDEN 1275/2010). Se da por hecho, que hacer una tabla en una hoja de cálculo, con una *tablet*, con las capturas de peces de un municipio y su precio de mercado, es un proyecto multidisciplinar donde se adquieren competencias de matemáticas, biología, economía y tecnología... y son pocos quienes se dan cuenta de que las nuevas tecnologías son solo una herramienta más que, bien utilizada, es muy útil.
- Se dice que el profesor ya no debe enseñar, sino solamente facilitar el aprendizaje, asimilando así que todas las materias son iguales, que las posibilidades de autoaprendizaje son similares y que hay que abandonar completamente métodos anticuados, ya que explicar algo es, —hablando con clara intencionalidad peyorativa—, hacer una “clase magistral” (Luján, 2013).
- Se dice, y es un tópico, que hay profesores muy expertos en la materia, pero que no saben explicarse (Nóvoa, 2009) y nadie quiere darse cuenta de que muchos profesores simplemente no saben nada de lo que explican: basta escuchar a un biólogo explicando electromagnetismo, o a un ingeniero eléctrico hablando del plancton, algo que, en la mayoría de casos, es realmente insoportable. Quien domina un tema es porque ya le motivaba desde su infancia y lo considera más importante que nada en este mundo; por eso lo vive y lo transmite desde el hígado; y el canal de comunicación no es la boca, ni siquiera la metodología aplicada, sino los poros de la piel.
- Se supone que con la robótica se aprende de todo porque es multidisciplinar. Ciertamente es que la robótica es motivadora —y eso es importante—, y que la motivación para tomar algún camino futuro empieza a surgir a partir de los 14 años. Pero a menudo solemos inculcar a alumnos, sin bases fundamentales ni capacidad deductiva, que al construir un robot adquieren muchas capacidades. Incluso el propio profesor se autoengaña, creando unas actividades de enseñanza / aprendizaje cerradas para culminar el robot. Pero el resultado es que alumnos provenientes de ciclos formativos de grado superior, constructores quizás de robots submarinos, en la universidad

---

suspenden, en elevado porcentaje, asignaturas tan fundamentales como la física en unidades didácticas de electricidad, y ya no digamos asignaturas como teoría de circuitos o máquinas eléctricas o bases de programación, o cálculo... (Nieto, & Ramos, 2013).

En definitiva: con demasiada frecuencia se cree que innovar en educación es, *per se*, mejorar; que quien no innova es porque no sabe o no quiere; y se premia a quien experimenta con los alumnos, sin exigir ningún *feedback* de los resultados reales. Y es que quien lleva a cabo un proyecto educativo, lo justifica con aprobados masivos, que al final es lo que interesa a los gobiernos y representa la única unidad de medida del fracaso escolar (Caballero, 2011). No importa si el nivel que imparte el profesor es bajísimo, mientras el nivel exigido a los alumnos sea más bajo todavía.

Evitando cometer los errores mencionados, innovar tampoco es malo *per se*. Es como todo: una excelente estrategia complementaria, si se hace correctamente y en la justa proporción. Resulta entonces más que justificada la inclusión de nuevas metodologías de enseñanza / aprendizaje, que afortunadamente empiezan a ser ampliamente aceptadas, valoradas y, cada vez más, incorporadas en los diferentes currículums. Son metodologías generadoras del llamado cuarto bloque lingüístico, que consigue una plaza propia de la mano del lenguaje literario, el científico-matemático y el humanístico: el lenguaje digital, fundamentado en el pensamiento computacional y enseñanza de la programación. Un nuevo lenguaje que ofrece varios beneficios: “motiva a los alumnos a aprender, se adapta a los estilos de aprendizaje de cada alumno, facilita la explicación y la comprensión, desarrolla las habilidades sociales, abre una puerta a la exploración, promueve la competencia digital y la alfabetización mediática, desarrolla el pensamiento lógico y matemático, permite trabajar las inteligencias múltiples, facilita el trabajo autónomo de los alumnos y fomenta la creatividad” (Llorens, 2015).

A este efecto y desde hace más de doce años, el autor imparte un crédito variable optativo en cuarto curso de ESO, cuyo nombre es “informática de programación”, y que, en resumen, pretende el desarrollo cognitivo de un aspecto muy importante y demasiado olvidado en el desarrollo curricular tradicional: el razonamiento lógico-deductivo y el sentido común.

El propósito de este artículo es doble: primero, ofrecer al lector una herramienta eficaz; y segundo, ofrecerle también los resultados de su aplicación durante estos doce años de experiencia.

VirPLC es un *software* libre de simulación de PLC (*Programmable Logic Controller*), con el que se puede diseñar y comprobar el funcionamiento de cualquier sistema de control. Trabaja bajo Windows (desde XP hasta W10) y se puede descargar desde la web del autor (<http://www.xtec.cat/~aferrer8/virplc.htm>). Su posibilidad de simulación inmediata permite no solo trabajar en el aula sin limitación de entrenadores, sino también practicar en casa para preparar, ampliar o reforzar alguna de las

propuestas, proyectos o retos. En la Figura 1 podemos observar las dos ventanas de trabajo: a la izquierda el entorno de programación y a la derecha, el de simulación.

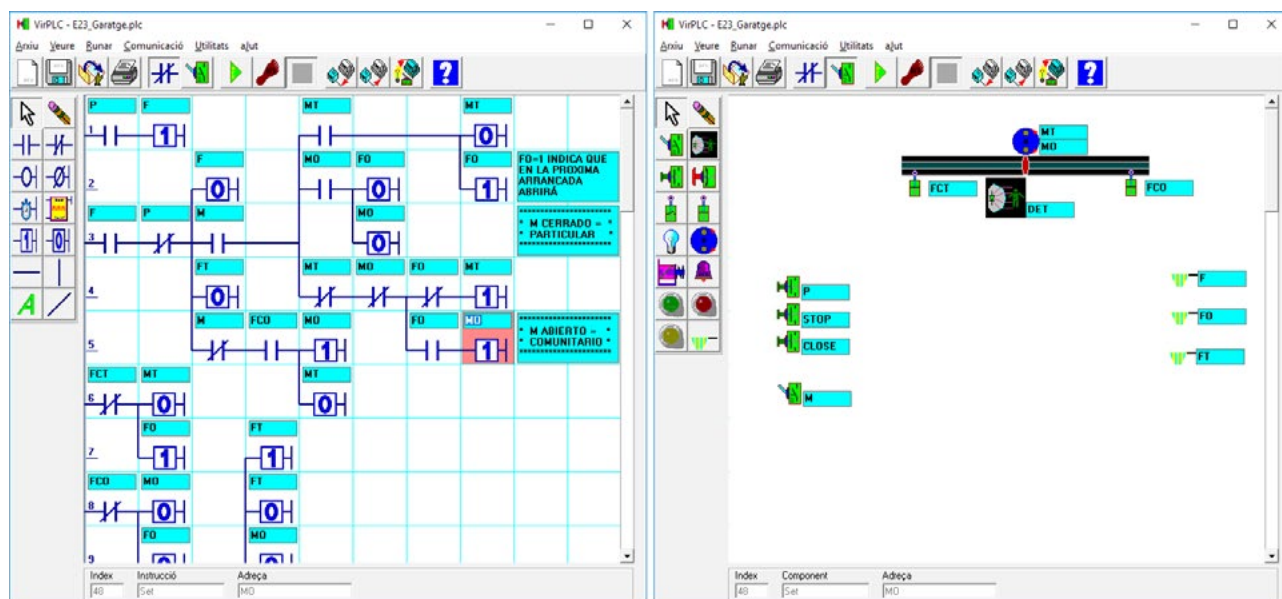


Figura 1. Ventanas de trabajo de VirPLC

## 2. Objetivos fundamentales de VirPLC

En el año 2005, los Ministros de Educación de la OCDE decían que: "...El desarrollo sostenible y la cohesión social dependen de forma crítica de las competencias de toda nuestra población..." (OCDE, 2005: 3). Sin embargo, para determinar cuáles son las competencias a desarrollar, hay que tener presente que la adquisición efectiva de conocimientos y destrezas a través de la enseñanza formal se convierte en una competencia cuando se transfiere a la vida cotidiana (Izquierdo, Caamaño, & Sarramona, 2016).

VirPLC es una herramienta para la introducción a la programación de sistemas. No está únicamente enfocada a los estudiantes más avanzados, sino muy especialmente a aquellos que potencialmente podrían serlo, pero que, por sus vicisitudes personales o por las exigencias específicas del sistema educativo actual, nunca lo han sido.

VirPLC es un simulador de PLC mediante un sistema de programación simbólico de diagrama de contactos, dispuestos en forma de escalera (*ladder*) en que se puede operar con diversos niveles de dificultad. Sin embargo, su propósito no es el de convertir a los alumnos/as en expertos en automatización sino, más bien, facilitar un primer contacto entre el alumno y la lógica, en este caso aplicada a la tecnología, a fin de hacerles conscientes de sus posibilidades lógico-deductivas.

---

En general VirPLC pretende contribuir a desarrollar las competencias básicas del ámbito científico-técnico (Izquierdo, Caamaño, & Sarramona, 2016) y a aumentar los valores de autoestima personal del alumno mediante la valoración de capacidades a menudo olvidadas y muy poco estimuladas: la lógica y el sentido común.

Y estos dos enormes retos se trabajan a partir de los siguientes objetivos generales:

- Contribuir al desarrollo del pensamiento lógico-deductivo-razional, que a menudo permanece aletargado y a punto de oxidarse en una parte muy profunda de cerebro, y que adquiere potencial para despuntar de forma abrupta a partir de los 15 años.
- Considerando que VirPLC requiere unos requisitos conceptuales previos prácticamente nulos, mejorar el sentimiento de autoestima a quien no ha podido nunca demostrar, ni demostrarse, su propia valía respecto a unas capacidades personales que nunca le han sido ponderadas ni reconocidas, para así motivar y reincorporar al sistema educativo a todo aquel alumnado que, desmotivado con la oferta educativa anterior, había abandonado todo interés.
- Hacer disfrutar al alumno en clase, mediante un aprendizaje totalmente funcional, basado en la operatividad de sistemas de control simples, que él mismo utiliza en la vida cotidiana (ascensor, puerta de supermercado, puerta de garaje...), con una tecnología atractiva, muy actual y de enorme implantación en la industria.
- Contribuir a la orientación profesional del alumno (objetivo particularmente importante si situamos esta metodología en los últimos cursos de la etapa de enseñanza secundaria obligatoria).

Además, aunque ninguno de los objetivos generales de este recurso pretende convertir a los alumnos/as en expertos en automatización, indirectamente VirPLC permite también abarcar una multitud de objetivos secundarios y mucho más específicos de la tecnología como son: Sentirse capaz de diseñar pequeños programas de automatización de sistemas elementales; Facilitar un primer contacto entre el alumno y las tecnologías de control, haciéndoles conscientes de sus enormes posibilidades y gran campo de aplicación; Adquirir la capacidad para detectar disfunciones y posibles mejoras en la secuencia operativa en un sistema; Capacitarlos para superar grandes problemas con pequeñas ideas, utilizando la lógica y el sentido común. Actitud crítica en la valoración las ventajas e inconvenientes de soluciones diversas en un mismo problema técnico o reto; Adquirir la autocrítica necesaria para valorar el resultado obtenido puede ser mejorado; Hábito de planificar ordenadamente la ejecución de las partes de un proyecto.

---

### 3. Metodología

Es más que trivial que la metodología de cualquier recurso está condicionada a sus características. Las de VirPLC son las siguientes:

- **Motivador:** Al requerir un nivel de conocimientos iniciales igual a cero, permite a aquel alumno/a que por diversas circunstancias se ha “descolgado” de la escolarización y que ha “perdido el tren”, que pronto se dé cuenta de que en este entorno puede destacar respecto de ese otro compañero/a que siempre ha sido buen/a estudiante y que siempre ha sacado las mejores notas. Este hecho le otorga un carácter fuertemente motivador ya que, en muchos casos, permite al alumno/a recuperar la confianza en sí mismo/a.
- **Funcional, significativo y cotidiano:** el control de sistemas reales comúnmente utilizados por el alumno, desde los más sencillos a los más complejos, a partir de un sistema de programación gráfico y no secuencial.
- **Interactivo:** posee un dinamismo inmediato en el continuo haz / prueba.
- **Adictivo:** a medida que vamos evolucionando en las diversas sesiones, el alumno ve más claras sus posibilidades y, en general, adopta un interés creciente ante cada reto que el profesor propone.
- **Formativo:** es especialmente interesante para aquellos alumnos que podrían orientar su futuro hacia las ciencias o las ingenierías, ya sea porque tengan intención de cursar Bachillerato Científico-técnico o prevean acceder a un Ciclo Formativo de carácter tecnológico o simplemente tengan vocación de trabajar en un entorno donde unos conocimientos básicos de programación le puedan ser útiles.

Antes de su implementación es recomendable una sesión de unas seis horas donde el profesor expone los fundamentos de Álgebra de Boole y de la lógica binaria, y donde ya se pueda detectar el nivel de capacidad intrínseca lógica del alumnado. Después es necesario invertir un par o tres de sesiones, para analizar los diversos dispositivos de entrada y salidas del mercado (interruptores, pulsadores NO / NT, finales de carrera NT, detectores, sirenas, puntos de luz, motores con dos sentidos de giro...) y donde se detalle la filosofía de los PLC.

Posteriormente, es aconsejable hacer una primera sesión, donde el alumnado descargue e instale VirPLC y su *dossier*, para concienciarlo así de que también lo puede hacer en casa. De hecho, la gran mayoría de alumnos/as se lo instala en casa y se prepara los retos que se van proponiendo en clase.

En el manual del alumno (<http://www.xtec.cat/~aferrer8/MANUAL.pdf>) se remarca que, para

---

introducir cada reto, primero hay que crear la necesidad provocada a partir de un problema práctico de automatización y posteriormente hay que buscar la solución que resuelve esta necesidad con un grado de posibilidades variables. Así, la metodología de aplicación en el aula puede ser muy diversa en función de las características del alumnado y de las del profesor, pero la que el autor ha testado y aplica en el aula después de doce años de experiencia es la siguiente:

- El profesor propone un reto concreto, útil y cotidiano, que representa el control de un sistema. Este sistema empieza siendo muy cercano y en cada sesión aumenta de complejidad (por ejemplo: una alarma, un semáforo, una puerta de supermercado, un ascensor, una puerta de garaje comunitario).
- Con todo el grupo-clase se analiza y se debate los pros y contra de los diversos *modus operandi* posibles. En este punto se constata la complejidad de sistemas que cotidianamente usamos sin darnos cuenta y juntos determinamos tanto el funcionamiento, como la elección de los actuadores (salidas), así como los elementos de detección (entradas), requeridos en la pantalla de *hardware*. También a partir del debate grupo/profesor, se estipulan las posibles mejoras que determinarán los diferentes niveles de perfección y que se valorarán como Regular, Ok, Ok+, Ok++ y, a veces, Ok+++ (o solo para genios).
- Individualmente o por parejas, en función de la conveniencia al alumno y de las posibilidades del aula, el alumno/a comienza a elaborar el programa insertando las instrucciones muy paulatinamente, mediante secuencias continuas y reiteradas haz / prueba, para controlar el sistema o reto. En esta fase el profesor interviene de forma muy dosificada y se encarga de controlar y anotar los niveles adquiridos por el alumno, el cual, a menudo, solo dispone de una hora para culminar cada reto.
- A medida que se avanza, el profesor puede mostrar situaciones de conflicto que requieran mayor depuración del programa y, además, tanto el alumno como el profesor pueden aportar ideas de mejora por el sistema a controlar.
- Al finalizar cada reto o sistema de control el profesor hace una demostración práctica con el proyector, donde se ha de acabar adquiriendo la fase de máxima complejidad (Ok+++). En dicha exposición, el profesor se ayuda del alumnado de manera que esta fase “genial” se consiga con la colaboración de todos; dicho de otro modo: con la fusión de todas las mentes.

Evidentemente los primeros retos deben ser muy cerrados, muy simples y con un requerimiento mínimo de instrucciones de *software*: LOAD, LOAD NOT, OUT y OUT NOT. Posteriormente empezamos a dotar al PLC de memoria con las instrucciones SET y RSET, que se irán describiendo a medida que sean necesarias. Finalmente, con los TIMERS y los CONTADORES, dotamos al autómatas de máximas prestaciones y nos enfrentamos a sistemas de control completos, complejos y poderosos.

---

Por ejemplo, en una sesión avanzada de VirPLC, imaginemos que se haya propuesto realizar el control de una puerta de garaje:

- ¿Cuál es el elemento actuador principal?: Un motor que gire en dos sentidos.
- ¿Puede haber actuadores secundarios?: Sí, por ejemplo, luces que se enciendan o hagan intermitencia cuando la puerta baje (o mejor, 3 segundos antes).
- ¿Cuántos pulsadores debe haber en el interior del garaje y qué función tienen en relación a cómo esté la puerta? (Aquí se puede comprobar que el *modus operandi* de una puerta de garaje no es igual en un garaje particular que en uno colectivo, donde la puerta debe cerrar de forma automática pasado un cierto tiempo...).
- ¿Qué tenemos que poner en el exterior del garaje?
- ¿Debe haber algún dispositivo de protección o de seguridad?, ¿por ejemplo, una barrera activa de infrarrojos?, y ¿cómo ha de actuar?
- Si la puerta está temporizada para bajar automáticamente después de haber subido, ¿cuánto tiempo debe estar en espera?
- ¿Qué tendría que pasar si, cuando la puerta está levantada, alguien pulsa el pulsador interior de parada?, ¿y si está bajando?, ¿y si está subiendo?, ¿y si está parada?

En definitiva, hay que determinar, entre todos, cómo sería más lógico que actuara el sistema y dejar siempre la posibilidad de que cada uno lo personalice de forma justificada.

## 4. Resultados

VirPLC ya se está utilizando en varios institutos, universidades y centros educativos. Sus objetivos son muy diversos y, por tanto, también sus resultados: no es lo mismo utilizarlo en la universidad o en un ciclo formativo de grado medio o superior, con la intención expresa de servir como introducción para familiarizarse en la programación de PLC, que ofrecerlo como una materia alternativa al final de la escolarización obligatoria, con la intención, anteriormente explicitada, de estimular el alumnado mediante los desarrollo de "otras" capacidades cognitivas, en este caso la lógica pura.

Desde ya hace tiempo, en el Institut de Palamós, aprovechando el hecho de que gran cantidad de alumnos elige asignaturas optativas relacionadas con la informática, se ofrece, junto con la típica "Informática de usuario", un crédito con el nombre "Informática de programación" destinado al

---

alumnado más audaz y atrevido. La demanda ha sido notable y el perfil del alumnado, en referencia a los resultados obtenidos en las calificaciones de los boletines de otras asignaturas, también.

Es evidente que la valoración de los resultados es compleja, como complejo es el propio ser humano. El alumno recibe una serie de interacciones de varias materias, con las respectivas exigencias de su profesorado y de su metodología. Además, el alumno también resulta influido por la familia, por sus compañeros en clase, por sus amistades y por muchas circunstancias personales. Y no solo esto; como se indica en la crítica introductoria, no se pueden analizar resultados objetivos a partir de las notas de las evaluaciones que, por más numéricas que sean, siempre son relativas. Con todo, y siempre bajo la perspectiva y las sensaciones personales, como aplicador de VirPLC durante más de doce años, evalúo los resultados obtenidos del siguiente modo:

- Por un lado, se ha podido comprobar cómo algunos alumnos/as que no destacaban especialmente por sus resultados académicos reflejados en su boletín de notas, a menudo con entornos familiares no demasiado óptimos, pueden destacar en este nuevo ámbito, en comparación con otros alumnos/as simplemente aplicados, provenientes de entornos más concienciados con la educación. Muchas veces se trataba de alumnado con tendencia al abandono del sistema educativo y que, afortunadamente, ha continuado su formación postobligatoria en un ciclo formativo de Grado Medio o en Bachillerato.
- Por otro lado, esta oferta ha tenido mucha demanda por parte de los alumnos que ya tenían en perspectiva cursar un Ciclo Formativo Técnico o un Bachillerato Científico-tecnológico. Además, la incorporación posterior de un lenguaje de programación con *scripts* orientados a objetos y clases como el Visual Basic, C#, Java o Python, ha permitido a aquellos alumnos que, varios años después, han cursado una Ingeniería o un Ciclo Formativo Superior de carácter técnico, disponer de una ventaja abismal respecto a sus compañeros de clase y obtener excelentes resultados en materias como: algoritmos de programación, programación II y similares.

## 5. Conclusiones

A nadie le viene mal desarrollar la capacidad lógica deductiva ni el sentido común inherente y propio de la humanidad; al contrario, esta capacidad también nos hace crecer como personas y nos prepara mejor para enfrentar cualquier reto futuro del que nos provenga la misma vida.

Muchos alumnos han descubierto una poderosa capacidad propia que nunca han podido desarrollar de manera tan completa. Con ella han mejorado también en algo tan importante como su propia autoestima.

---

Así pues, sin considerar las ventajas de la inclusión en centros más concretos de formación tecnológica, recomiendo VirPLC, o cualquier otra aplicación similar basada en la programación simbólica y no secuencial, como paso introductorio a los lenguajes de programación de tipo script con bucles de control, objetos, clases, eventos y multitud de otros conceptos mucho más sofisticados.

Son del todo imprescindibles la rapidez de respuesta en la interacción haz / prueba, así como la funcionalidad y cotidianidad de los distintos retos.

Finalmente, volver a recalcar que demasiado a menudo se premia la innovación sin comparar los resultados mediante pruebas externas, como las Competencias Básicas de la ESO o las Pruebas de Acceso a la Universidad de Bachillerato. En lugar de proclamar al viento teorías pedagógicas varias, si un instituto considerado de entorno difícil, durante más de doce años de estadística, en determinadas materias tiene un resultado muy superior a la media en dichas pruebas externas, con las mismas horas lectivas y recursos que los otros institutos, ¿por qué no se pregunta a este instituto por sus metodologías, sus estrategias y sus consejos? ¿Por qué?

## 6. Referencias

Márquez, A. M., Garrido, M. T. & Moreno, M. C. (2006). La innovación tecnológica en la enseñanza universitaria: análisis de un caso de utilización de foro y chat. *Revista Latinoamericana de Tecnología Educativa*, 5(1), 31-57.

ORDEN 1275/2010, de 8 de marzo, (BOCM 18 de marzo de 2010) por la que se implanta el proyecto de institutos de innovación tecnológica en la Comunidad de Madrid. [http://www.educa2.madrid.org/web/institutos\\_it/ies](http://www.educa2.madrid.org/web/institutos_it/ies)

Luján-Mora, S. (2013). De la clase magistral tradicional al MOOC: doce años de evolución de una asignatura sobre programación de aplicaciones web. From the traditional lecture to the MOOC: twelve years of evolution of a subject about web application programming. *Revista de Docencia Universitaria*, 11, 279-300.

Nóvoa, A. (2009). Para una formación de profesores construida dentro de la profesión. *Revista de Educación*, 350, 203-218.

Nieto-Isidro, S. & Ramos-Calle, H. (2013). Uso de los errores como estrategia didáctica en el aprendizaje de las matemáticas en el nivel universitario. *Memoria de realización del Proyecto de Innovación Docente ID2013/215*, Escuela Politécnica Superior de Zamora. <http://www.elmundo.es/universidad/2004/01/27/campus/1075223354.html>

---

Caballero Ruiz, M. C. (2011). Fracaso escolar. Una realidad en nuestras aulas. *Revista Autodidacta*, 1(5), 154-159.

Llorens-Largo, F. (2015). Dicen por ahí. . . . que la nueva alfabetización pasa por la programación. *ReVisión*, 8(2), 11-14.

OCDE (2005). La definición y selección de competencias clave. Resumen ejecutivo.

Izquierdo, M., Caamaño, A., & Sarramona, J. (2016). Competències bàsiques de l'àmbit científicotecnològic. Generalitat de Catalunya. Departament d'Ensenyament. Direcció General d'Educació Secundària Obligatòria i Batxillerat. Barcelona: Servei de Comunicació i Publicacions. <http://ensenyament.gencat.cat/web/.content/home/departament/publicacions/colleccions/competencies-basiques/eso/eso-cientificotecnic.pdf>