



UNIVERSIDAD  
DE LA FRONTERA  
VICERRECTORÍA DE PREGRADO

# PROYECTOS DE DESARROLLO E INNOVACIÓN DOCENTE 2018



FONDO CONCURSABLE



## Innovación metodológica mediante la incorporación de TIC para el mejoramiento de los resultados de aprendizaje de la asignatura Programación de Computadores

Nombre Docente(s) Responsable Nombre docentes colaboradores:	Juan Ignacio Huircan Q. Mauricio Dieguez Patricia Muñoz, Erik Soto. Prof. Módulo – Matthias Clein
Departamento	Departamento de Ingeniería Eléctrica

### I. RESUMEN

El proyecto implementa una innovación en la asignatura de Programación de Computadores, perteneciente al Plan Común de las Ingenierías Civiles de la Facultad de Ingeniería y Ciencias. En este caso puntual, se aplica al módulo correspondiente a la carrera ICE. Los problemas de la asignatura son una alta tasa de reprobación, deserción y eliminación de estudiantes. Existen reclamos tanto de estudiantes que cursaron la asignatura como académicos de especialidad de asignaturas que requieren de estos resultados de aprendizaje.

La propuesta consiste aplicar una Metodología Activa mediante la incorporación de un elemento didáctico al computador (tarjeta microcontroladora y set de actividades prácticas) que permite orientar las tareas y trabajos a resolución de casos y problemas (desafíos) más tangibles para el estudiante; además de intercambiar experiencias y opiniones con sus pares, de tal forma analizar lo que hacen para resolver el problema, con lo hacen y finalmente que resultados logran.

La incorporación de esta metodología, permitió al estudiante cambiar la percepción de la asignatura, disponer de material docente como herramienta para el auto-aprendizaje, realizar evaluaciones más adecuadas y mejora en los rendimientos de los estudiantes.

## II. INTRODUCCIÓN

Después de la innovación curricular aplicada a las Carreras de Ingeniería Civil en el año 2012, la asignatura **Programación de Computadores** quedó como asignatura común a todas las Ingenierías. El programa de asignatura se hizo para satisfacer los requerimientos y de acuerdo a los resultados de aprendizaje, para que los alumnos aprendieran a programar determinado lenguaje. En el caso particular de la carrera Ingeniería Civil Electrónica (ICE), existen diversas asignaturas que requieren habilidades de programación (no tan exigentes) tales como **Herramientas de Análisis de Señales, Procesamiento Digital de Señales**, los profesores comentan que al solicitarle a los estudiantes que programen ciertos algoritmos (habitualmente en lenguaje C) los resultados no son los mejores. Los profesores de las asignaturas manifiestan su disconformidad respecto de las habilidades de programación de los estudiantes. Otra asignatura que requiere habilidades de programación es **Protocolos de Comunicación**, sin embargo, debido a la distancia (cinco semestres de diferencia) se “pierde” en el tiempo el *know how* adquirido en la asignatura de programación.

Los indicadores de aprobación han variado desde el año 2012, para el caso particular de ICE, éstos fluctúan durante los últimos cuatro semestres entre 50%, 55.6%, 14.3% y 40% el primer semestre del 2017. De acuerdo a información recabada por la dirección de carrera, la percepción de los alumnos es que han bajado las calificaciones y se plantea que “es una asignatura fome, ...no los motiva etc., muy teórica...”. Las tareas y ejemplos desarrollados en el curso por parte de los alumnos no son percibidas como una aplicación real. Desde el punto de vista del alumno, éste debe percibir que la tarea y/o el ejemplo “sirva para algo”, “tenga sentido” o tenga alguna aplicación práctica.

### OBJETIVO.

Implementar una metodología innovadora y estrategias didácticas para un aprendizaje significativo usando tecnologías en el aula (TIC) en función de un instrumento para evaluar resultados de aprendizaje.

## III. MÉTODOS

Para el logro del objetivo propuesto se aplicó una Metodología Activa en la cual mediante la incorporación de un elemento didáctico al computador (tarjeta microcontroladora y set de actividades prácticas) ayudó a orientar las tareas y trabajos a resolución de casos y problemas (desafíos) más tangibles para el estudiante. Esto permitió la participación del estudiante en actividades prácticas intercambiando experiencias y opiniones con sus pares, desarrollando la capacidad de análisis para resolver el problema, trabajo en equipo y finalmente el logro de los resultados. Finalmente permite que el estudiante sea responsable de su propio aprendizaje.

La inclusión de estos elementos didácticos no es nueva en el área. Las incorporaciones de herramientas adicionales para el uso del computador para la enseñanza de la programación van

desde recursos web (Yan et al., 2016) hasta incorporación de herramientas de hardware abierto (Ariza, 2016). Este último aspecto resulta muy atractivo para los estudiantes debido a que la interacción con plataformas externas permite al alumno visualizar sus aplicaciones (Lawson and Park, 2016; Wirth et al., 2014; Zhong and Liang, 2016). El hardware abierto consiste en una tarjeta electrónica en la cual el estudiante dispone de los diagramas esquemáticos de los circuitos, puede ver los dispositivos electrónicos, realizar conexiones y evaluar los programas realizados mediante tareas que pueden ser visualizadas en forma física.

En el caso puntual se utilizó la tarjeta llamada Raspberry PI 3, la que puede ser programada mediante lenguaje C a través de un computador personal usado como consola. Al conectar la tarjeta al computador, ésta es vista como “otra máquina” que utiliza un Sistema Operativo basado en Linux sobre la cual implementar los programas y aplicaciones. A través de esta plataforma el estudiante puede desarrollar aplicaciones que “tienen sentido”, las tareas realizadas sobre este nuevo hardware puedan ser “vistas” en forma más tangible, sobre todo cuando las aplicaciones deben interactuar con el medio externo ya sea por el uso de elementos sensores y elemento actuadores. Esto genera el entusiasmo para seguir profundizando en los tópicos respectivos pudiendo experimentar sin que se produzcan grandes riesgos. Dado que la metodología activa contempla la resolución de casos, esto también provoca un alto entusiasmo en los alumnos provocando un impacto en el aprendizaje. La propuesta resulta innovadora, pues los estudiantes de manera temprana podrán trabajar con pequeños problemas asociados a la programación interactuando con diversos periféricos y el sistema operativo.

## **Metodología**

Se divide el proceso en tres etapas:

### Etapa 1: Preparación

- Inicialmente el equipo de trabajo revisó los contenidos del programa de asignatura de tal forma de re-ordenar estos para compatibilizar los tópicos con las actividades prácticas. Esto permite realizar una alineación con los resultados de aprendizaje.
- Para ello se desarrollaron los distintos contenidos en set de diapositivas no extensas para las clases expositivas, intercalando ejercicios con el fin de reforzar los temas.
- Se consideró que todas las actividades prácticas se desarrollaran sobre sistema operativo Linux, esto debido a que la tarjeta microcontroladora requiere del manejo de este SO. De acuerdo a esto las primeras actividades prácticas de programación se realizan sobre computadores personales disponibles en el laboratorio con SO Linux. Una vez familiarizado con el entorno de programación, se procede a la incorporación de la tarjeta Raspberry, usando ahora el computador como elemento terminal de la tarjeta.

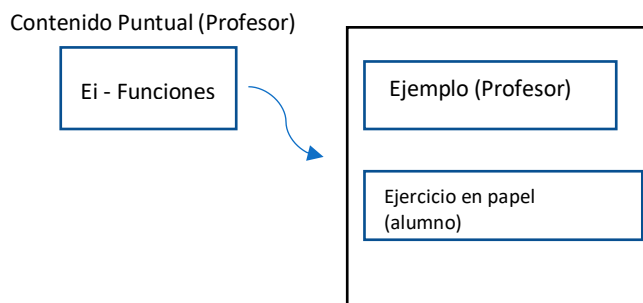
El hecho de empezar a trabajar en un computador personal con Linux en forma inicial, fue debido a que en el primer semestre no estaba disponible la tarjeta Raspberry, pero esta fue incorporada una vez avanzado el semestre. Sin embargo, en el segundo semestre a pesar de disponer de la nueva plataforma se decide mantener el mismo esquema, pues el traspaso a la tarjeta Raspberry resulta “más natural” para el estudiante.

- Se define un set de actividades prácticas que el estudiante debe realizar en el horario de laboratorio, para ello se contempla el desarrollo de 10 a 12 Sesiones de Laboratorio o Actividades prácticas en el semestre que contemplan:
  - Aplicaciones de interacción entre el estudiante y la maquina (Desarrollo de código básico de prueba para que estudiante se entrene en la detección de errores y corrección del programa, ejemplos básicos para la verificación de las sentencias de código fuente, etc.)
  - Aplicaciones básicas relacionadas con otras disciplinas (Manejo de matrices, evaluación de fórmulas, funciones matemáticas, etc.)
  - Aplicaciones para la resolución de un problema real (Manejo de entrada-salida, uso de sensores, manejo de archivos, etc.)

### Etapa 2: Actividades en Aula/Laboratorio

Utilizando el set de diapositivas para las clases expositivas se presentan los conceptos al estudiante, intercalando ejercicios con el fin de reforzar los temas, esto bajo el siguiente esquema (Figura 1):

- Teoría
  - Clase Teórica Expositiva
    - Contenido puntual 1 – Ejemplo – Ejercicio en papel.
    - Contenido puntual 2 – Ejemplo – Ejercicio en papel.
    - Contenido puntual  $n$  – Ejemplo – Ejercicio en papel.



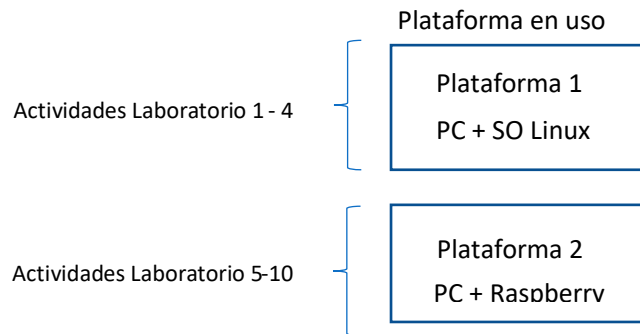
**Figura 1.** Esquema clase expositiva y trabajo en aula.

El recurso pedagógico utilizado en la parte teórica permite que los largos tiempos ocupados en pasar la teoría sean reemplazados por tiempos menores, complementado la teoría con la incorporación de ejemplos, ejercicios y tareas que el estudiante puede ir revisando en la misma clase.

De acuerdo a las guías de laboratorio, el estudiante desarrolla las actividades ya sea sobre el PC o sobre la tarjeta microcontroladora (Figura 2).

- Laboratorio
  - Guía Actividad Práctica 1 (realizada en computador/Tarjeta Raspberry PI)
    - Ejercicio 1 Práctico
    - Ejercicio 2 Práctico
  - Guía Actividad Practica 2
    - Ejercicio 1 Práctico

▪ Ejercicio 2 Práctico



**Figura 2.** Uso de plataforma de hardware.

Adicionalmente como parte del laboratorio se desarrollaron las siguientes actividades complementarias de tal forma que el estudiante pudiera hacerse parte desde el primer minuto de la actividad.

- Instalación y configuración de la plataforma Raspberry PI
- Mantención y actualización de software
- Desarrollo de los ejercicios en el aula tanto en papel como sobre la plataforma de hardware
- Desarrollo de actividades prácticas del laboratorio propuestas por el profesor

Etapa 3 - Evaluación

Parte Teórica: Se consideraron dos pruebas teóricas, más un porcentaje de consistente en un test de entrada semanal de acuerdo a la Tabla 1.

**Tabla 1.** Porcentaje Evaluación Teórica

Actividad de Evaluación	Porcentaje	
Prueba Teórica 1	40%	Individual
Prueba Teórica 1	40%	Individual
Test (promedio)	20%	Individual

Parte Práctica: Se consideraron tres trabajos prácticos de acuerdo a los porcentajes indicados en la Tabla 2.

**Tabla 2.** Porcentaje Evaluación Práctica

Actividad de Evaluación	Porcentaje	
Trabajo Práctico 1	20%	En Grupo
Trabajo Práctico 2	20%	En Grupo
Trabajo Práctico 3	50%	Individual
Evaluación Actitudinal	10%	Individual

Las actividades de la metodología activa están alineadas con los distintos Resultados de Aprendizaje. Cada una de las actividades realizadas tributan en distinto grado al cumplimiento de cada uno de los resultados de aprendizaje.

**Tabla 3.** Actividades y resultados de aprendizaje.

Actividad	Resultado de Aprendizaje
Instalación de las herramientas de software sobre las plataformas de hardware	Identificar los componentes de hardware y software como introducción al funcionamiento de computadores.
Actividad de laboratorio usando la tarjeta microcontroladora	Utilizar Tecnología de información y comunicación para el aprendizaje
Actividades Teórica, tareas, ejercicios de clases	- Construir programas a partir de algoritmos utilizando lenguajes de programación de alto nivel.  - Reconocer los conceptos propios de las herramientas de diseño aplicables a un problema de programación.
Proyectos de laboratorio	- Diseñar algoritmos a través de herramientas tecnológicas y de diseño para la solución de problemas.  - Aplicar el método de Análisis Orientado al Objetivo para la resolución de problemas de ingeniería.
Tareas y ejercicios de clases. Actividades de laboratorio	- Resolver problemas de diversa índole.

- IV. **RESULTADOS** en relación al problema que generó esta experiencia de innovación ¿de qué manera se evidenció el cambio? ¿Cuáles fueron los hallazgos? Aporte evidencias tales como: encuesta a estudiantes u otros. Se pueden incluir gráficos o tablas (máximo 2 páginas).

El primer cambio que evidencia la innovación es el Ranking de asignatura, de acuerdo al informe de gestión de carreras de pregrado de 2019 se tiene:

**Tabla 4.** Ranking de asignaturas

Semestre	Promedio	Categoría
2017-2	3.72	S
2018-1	4.56	B
2018-2	4.75	MB

Dado que la innovación se empezó a desarrollar el primer semestre del año 2018, ya se observa un incremento durante el primer semestre. Para el segundo semestre se observa un nuevo incremento, el cual resulta ser superior a un punto respecto del original. Es importante mencionar que este resultado no estaba considerado dentro de los resultados originales, pero refleja la visión del estudiante respecto de la intervención de la asignatura.

El proyecto originalmente contempla cuatro resultados:

**R1.** Puesta en marcha de una Metodología Activa para mejorar el cumplimiento de los resultados de aprendizaje.

La metodología fue implementada de acuerdo a lo indicado en el apartado anterior. Esta permitió inicialmente un cambio en la estructura de la clase teórica, haciendo que el traspaso de los conceptos de programación resulta ser más inmediato para el alumno. La incorporación en la parte práctica de la tarjeta Raspberry PI como elemento pedagógico se realiza pasada la sexta semana después de iniciado el primer semestre del año 2018, esto debido a un pequeño

retraso en las compras. Durante el segundo semestre del año 2018 la plataforma fue incorporada ya a partir de la cuarta semana.

## **R2. Material docente validado como herramienta para el auto-aprendizaje.**

Se desarrolló material en formato electrónico de las clases teóricas y las actividades de laboratorio, las cuales quedaron disponibles en la plataforma del curso respectiva.

## **R3. Instrumento de evaluación validado para verificar los resultados de aprendizaje mediante su aplicación en el módulo piloto de la asignatura.**

La evaluación contempla la medición tanto de la parte teórica como la parte práctica. Dado que la aprobación es en conjunto, 30% teoría y 70% práctico. En primera instancia no se contempla por parte de equipo realizar un cambio drástico en los instrumentos de evaluación, sin embargo, considerando la calidad práctica de la asignatura el foco estará puesto en el trabajo del alumno en dicha área. Se establecieron tres instrumentos para la evaluación del curso:

Pruebas Teóricas: Contemplan preguntas de materia y conceptos de programación además de un problema de programación a desarrollar en papel.

Test de entrada: Están basados en ejercicios desarrollados o ejercicios propuestos (variantes de ejercicios desarrollados) por el profesor que quedan de tarea y los cuales los estudiantes deben revisar fuera del horario de clases. Luego al ingreso de la clase siguiente, los estudiantes son interrogados mediante un test de 15 minutos. La lógica de estos ejercicios y es que el estudiante los desarrolle en su casa y los pruebe en el computador para verificar la sintaxis del código generado.

Trabajos Prácticos de Laboratorio: Los Trabajos Prácticos consideran problemas desarrollo de programas de mediana complejidad que inicialmente los estudiantes desarrollan en grupo (2 personas) para que finalmente desarrollen una aplicación en forma individual.

## **R4. Mejoramiento en el porcentaje de aprobación a un valor superior al 60%.**

Se realizó un cálculo de los distintos porcentajes de aprobación del curso de los tres últimos años, los cuales se muestran en la Tabla 5. Considerando que la innovación del curso comenzó el año 2018, se observa en la Tabla 5 que el primer semestre se incrementa el porcentaje de aprobación del 82% al 86%, sin embargo el segundo semestre disminuye al 58%. Se observa que este último valor es menos de lo esperado.

A pesar de que el rendimiento del segundo semestre no cumple la expectativa del 60% esperado, sigue siendo mayor que los resultados 2016 y primer semestre 2017.

**Tabla 5.** Porcentajes de aprobación últimos tres años.

Carrera ICE	2016		2017		2018	
	I semestre	II semestre	I semestre	II semestre	I semestre	II semestre
N° inscritos	18	7	20	11	21	12
N° aprobados	10	1	8	9	18	7
N° reprobados	8	6	12	2	3	5
% Aprobación	56%	14%	40%	82%	86%	58%



Por otro lado, es importante mencionar que se realizó una modificación en la evaluación de los trabajos prácticos del segundo semestre del año 2018. Durante el primer semestre de 2018 se consideran todos los trabajos prácticos en grupo. Se observa en esta situación una diferencia entre las evaluaciones prácticas respecto de las teóricas, estudiantes que tienen notas altas en los trabajos de laboratorio no tienen notas altas en la parte teórica, específicamente en lo que tiene que ver en la generación de código. Se opta para el segundo semestre de 2018 evaluar la parte práctica en forma individual. Esta situación permite sincerar finalmente los rendimientos, creemos que la baja obedece a este aspecto.

#### Discusión respecto de los resultados

- La mejora en el porcentaje de aprobación del primer semestre del año 2018 eventualmente no puede ser asociada a la incorporación de la nueva plataforma de hardware, dado que no estuvo en funcionamiento pleno el curso del segundo semestre del año 2018.
- Creemos que el cambio realizado en la forma de pasar los contenidos teóricos permite un cambio en la dinámica de la clase, evita los tiempos muertos entre la teoría y la parte práctica, dado que los ejercicios desarrollados en clases en conjunto con el test permiten generar un refuerzo en los estudiantes.
- La incorporación de la tarjeta Raspberry PI como elemento de apoyo cambió la percepción del curso, pues los estudiantes rápidamente se dieron cuenta que en otras universidades también se utiliza.
- Es muy relevante que las actividades prácticas puedan ser realizadas en grupo, dada la discusión que se produce entre los estudiantes, sin embargo, los trabajos prácticos deben ser desarrollados en forma individual lo que permite establecer en forma más clara si el alumno aprende o no.

#### V. CONCLUSIONES

La forma de enseñar un lenguaje de programación va más allá de traspasar al estudiante una cantidad de símbolos (palabras reservadas del nuevo lenguaje) en conjunto con una serie de reglas que representan el nuevo lenguaje. El contexto sobre el cual se va a utilizar este nuevo lenguaje será tan relevante como el código mismo, pues el saber ¿Para qué estoy aprendiendo esto?, ¿Dónde lo puedo utilizar? y ¿Por qué? Será un factor que ayuda y facilita el aprendizaje del mismo.

Se requiere para ello intercalar en el curso charlas de profesores/investigadores/desarrolladores que utilicen la programación para la solución de los problemas en los cuales trabajan (Se hizo una presentación al curso de un investigador que le muestra al estudiante en que aplicaciones se requiere lo que están aprendiendo). Esta actividad pudiera ser ampliada a ramos de matemática donde el especialista de ingeniería podría “aterrizar” lo abstracto a algo más concreto.

La intervención de la asignatura se seguirá realizando durante el año 2019 pero mejorando los temarios prácticos y rehaciendo los instrumentos de evaluación. Esta es la propuesta de incorporación de un segundo ramo de programación para la mejora de la carrera.

Para la mejora de los instrumentos de evaluación se incluirá a profesores de las asignaturas que requieren de la programación.

También se propone incluir a dichos profesores de la especialidad para generar una batería de temas y/o problemas prácticos para las actividades de “Trabajo Práctico”

La incorporación de esta innovación puede ser extendida en primera instancia a carreras afines, ya sea Ingeniería Civil Telemática en Ingeniería Civil Eléctrica. La discusión respecto de la incorporación en otras carreras de ingeniería civil no se ha realizado dado que según comentarios dichas disciplinas “no necesitan programación” a pesar de que el ramos es común para todas.

Dentro de las potenciales amenazas en esta área representan las modificaciones que trae la propuesta de “innovación” de las Ingenierías Civiles que la Facultad de Ingeniería y Ciencias está proponiendo, con un ciclo común de dos años. Creemos que esta situación aleja a los estudiantes de la especialidad que tanto ha costado acercar para el logro de la identidad.

## VI. REFERENCIAS

Ariza, Jonathan Álvarez. 2016. “A Proposal for Teaching Programming Languages through Open Hardware Tools.” In *2016 IEEE 8th International Conference on Engineering Education: Enhancing Engineering Education Through Academia-Industry Collaboration, ICEED 2016, 202–7*. <https://doi.org/10.1109/ICEED.2016.7856072>.

Lawson, Wesley, Stephen Secules, Shuvra Bhattacharyya, and Ayush Gupta. 2016. “An Application-Based Learning Approach to C Programming Concepts and Methods for Engineers.” In *ASEE’s 123rd Annual*. New Orleans. <https://doi.org/10.18260/p.26567>.

Wirth, Michael, and Judi McCuaig. 2014. “Making Programs With The Raspberry Pi.” In *WCCCE 2014: The 19th Western Canadian Conference on Computing Education - In-Cooperation with ACM SIGCSE*, 1–5. Richmond. <https://doi.org/10.1145/2597959.2597970>.

Yan, Yu, Hiroto Nakano, Kohei Hara, Takenobu Kazuma, and Aiguo He. 2016. “A Web Service for C Programming Learning and Teaching.” *Proceedings - 2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2016*, 414–19. <https://doi.org/10.1109/CISIS.2016.70>.

Zhong, Xiaoyang, and Yao Liang. 2016. “Raspberry Pi: An Effective Vehicle in Teaching the Internet of Things in Computer Science and Engineering.” *Electronics* 5 (4): 56. <https://doi.org/10.3390/electronics5030056>.