



Buena práctica

Enseñanza de los fundamentos de la programación a todos los estudiantes de ESPOL

Escuela Superior Politécnica del Litoral
Guayaquil, Ecuador

Responsable:

Rafael Bonilla Armijos

Profesor Titular Auxiliar I

Facultad de Ingeniería en Electricidad y Computación, Escuela Superior Politécnica del Litoral

Email: rabonilla@espol.edu.ec

[Sitio web de la práctica](#)

Palabras clave:

Innovación en metodologías enseñanza | Adaptaciones curriculares | Estudiantes | Gestión de la docencia | Tecnologías para la enseñanza

Resumen ejecutivo

La Escuela Superior Politécnica del Litoral a partir del año 2016 introdujo en las mallas curriculares de todas las carreras la materia Fundamentos de Programación. Fundamentos de Programación se dicta utilizando el lenguaje Python y a lo largo del semestre se introducen problemas a resolver en distintos dominios: ingeniería, nutrición, economía, deportes, entre muchos otros.

El objetivo principal es ayudar a los estudiantes a desarrollar habilidades de resolución de problemas y pensamiento crítico utilizando estrategias de «Computational» que son aplicables a problemas propuestos en muchas áreas de conocimiento. Adicionalmente la materia incluye un módulo de procesamiento de datos que busca presentar a los estudiantes herramientas que permiten automatizar la analítica de información, una habilidad que se ha convertido en un pilar fundamental a la hora de convertir datos en conocimiento en todas las industrias.

A lo largo del curso los estudiantes aprenden a representar la información con diferentes estructuras de datos y a utilizar el paradigma «Divide y vencerás» descomponiendo problemas complejos en tareas que son fáciles de resolver y que permiten enfocar la actividad de programación.

Los estudiantes del curso tienen la posibilidad de participar en un hackathon denominado PyWeekend. El PyWeekend se organiza de tal manera que, estudiantes de diferentes carreras forman grupos interdisciplinarios que buscan resolver retos propuestos semanas antes de que la competencia se lleve a cabo. A lo largo de los cuatro años anteriores, se ha trabajado con retos provenientes de profesores de diversos departamentos de la Universidad, socios de la industria, socios del sector público usando a menudo datasets anonimizados para la competencia. Durante el fin de semana de la competencia, los estudiantes cuentan con la ayuda de mentores que pueden aportar sugerencias o resolver dudas pero no participan en la implementación de código.

La evaluación de la materia es unificada de tal manera que alrededor de 900 estudiantes que cursan la materia cada semestre resuelven los mismos problemas usando las mismas herramientas.

Los cursos permiten la inscripción de estudiantes de varias carreras puesto que no hay paralelos dirigidos a una determinada área del conocimiento. Este modelo permite reunir a estudiantes con perfiles y antecedentes diversos en grupos que resuelven problemas usando un lenguaje común.

Descripción detallada de la práctica

Dimensión: Contexto y origen de la práctica

Caracterización del contexto y la situación que origina y en la que tiene lugar la práctica

En el Ecuador hay un déficit de alrededor de 9.000 puestos de trabajo con requisitos de manejo de tecnología, incluyendo programación inicial, que no pueden ser llenados por falta de profesionales capacitados. La situación en el mundo es similar. ESPOL, reconociendo esta necesidad presente y futura, decidió en el 2016 añadir Fundamentos de Programación como materia obligatoria para todos sus estudiantes indistinto de la carrera que hayan seleccionado. El objetivo es formar a nuestros estudiantes en el manejo de la tecnología y la creación de programas para computadoras que les permita cubrir estas necesidades y convertirse en profesionales líderes y distinguidos en cada una de sus profesiones.

Aprender a programar ofrece múltiples beneficios que incluyen el desarrollo de nuevas habilidades para la resolución de problemas, incorporación de las matemáticas en la vida cotidiana, pensamiento lógico, sistémico y sistemático y mejor entendimiento de cómo funcionan las cosas.

Planteamiento de pertinencia de la práctica a partir de la situación, problema u oportunidad detectada

Nuestra práctica es innovadora porque fuimos la primera universidad del país en enseñar a programar a todos sus estudiantes, no solo a los estudiantes de carreras técnicas o de Ingeniería. Esto nos llevó a diseñar nuevos esquemas y metodologías de enseñanza para poder llegar a una audiencia que no es necesariamente tecnológica. Dentro de los cambios que se implementaron están el lenguaje de programación (Python), los tipos de ejercicios a revisar en clases (no técnicos o matemáticos sino de un contexto diverso e incluyente a todas las carreras) e identificar al análisis de altos volúmenes de datos como un elemento común y de fácil entendimiento para todos nuestros estudiantes. Es así que empezamos a discutir con los estudiantes problemas como análisis de cadenas de ADN, factores de nutrición, medio ambiente, planificación urbana, entre otros.

Esperamos compartir nuestra experiencia de éxito con otras instituciones evitando que encuentren los mismos problemas que nosotros ya superamos

Dimensión: Planificación de la práctica

Planteamiento de objetivos de la práctica

Nuestra práctica presenta a los estudiantes estrategias para el desarrollo del pensamiento lógico y la resolución de problemas comunes en diversos campos profesionales por medio del diseño e implementación de soluciones basadas en el uso de un lenguaje de programación. Cubre los principios básicos para que el estudiante pueda leer y escribir programas; haciendo énfasis en el diseño y análisis de algoritmos. Además, introduce a los estudiantes en el uso de herramientas de desarrollo y depuración.

Proceso de diseño y planificación, identificación de los actores involucrados y definición de recursos

En el diseño de la práctica participaron profesores de la carrera Computación quienes mantuvieron reuniones con todas las autoridades académicas (Vicerrectora, Subdecanos y Coordinadores Académicos) y profesores claves de cada carrera para escuchar sus necesidades y expectativas de enseñar a programar. Esta información se utilizó para seleccionar los tópicos a enseñar, el lenguaje de programación y el tipo de problemas que se tratarían en las clases de la materia Fundamentos de Programación logrando llegar a una audiencia más diversa y no necesariamente tecnológica.

Estas reuniones sirvieron también para determinar la demanda de estudiantes a la materia en cada semestre junto con la necesidad de profesores y personal de apoyo (técnicos docentes y ayudantes de cátedra) para dictar la materia. Finalmente, las reuniones permitieron evaluar la capacidad de infraestructura (aulas de clases y laboratorios de computadoras) con los que contaba la institución, llegando a la conclusión que se necesitaba potenciar ligeramente a la misma adquiriendo nuevos equipos (computadoras, proyectores, entre otros.)

Se delegó a los profesores de la carrera Computación la responsabilidad de implementar la práctica, coordinarla semestre a semestre y evaluar sus resultados. Los beneficiarios directos de la práctica son todos los estudiantes de nuestra institución y los indirectos los profesores quienes ahora pueden abordar los problemas de sus materias de manera diferente y esperar soluciones usando programas de computadoras.

Mecanismos de evaluación

Para medir la efectividad de la práctica se decidió utilizar tres medidores:

1. Cantidad de estudiantes de otras carreras que toman materias opcionales e itinerarios de Computación: Dado que los estudiantes no están obligados a llevar esta materia, su registro voluntario es un reflejo de su deseo de aprender más sobre Programación y Computación.
2. Cantidad de estudiantes que participan en competencias de programación tipo Hackathon dentro y fuera de nuestra universidad: Estas actividades son también de registro voluntario y netamente de programación. La participación de los estudiantes en ellas se origina de su interés luego de tomar nuestra materia Fundamentos de Programación y por eso la utilizamos para medir la efectividad de la práctica.
3. Encuestas de satisfacción: Durante el transcurso de la práctica se han corrido encuestas anónimas al final del semestre sobre el aporte de la materia a desarrollar la habilidad de comprender, descomponer y solucionar problemas en los estudiantes. Los semestres en los que se corrieron estas encuestas han sido seleccionados aleatoriamente para no generar expectativas en los estudiantes que pudieran alterar los resultados observados.

Dimensión: Implementación de la práctica

Descripción de la implementación de la práctica, indicando etapas, acciones y actividades específicas

La práctica se dividió en las siguientes etapas:

1. Diseño
2. Pilotaje
3. Implementación
4. Evaluación

En el diseño de la práctica participaron profesores de la carrera Computación quienes mantuvieron reuniones con todas las autoridades académicas (Vicerrectora, Subdecanos y Coordinadores Académicos) y profesores claves de cada carrera para escuchar sus necesidades y expectativas de enseñar a programar.

Para el pilotaje se seleccionaron 160 estudiantes cursando su primer semestre en 10 carreras no de Ingeniería como Economía, Turismo, Acuicultura, Nutrición y Producción de Medios Audiovisuales. El objetivo era probar

la nueva metodología y afinar detalles antes de la etapa de implementación. La participación de estos estudiantes en la materia fue voluntaria. El grupo se dividió en 4 paralelos que fueron dictados por los profesores de Computación a cargo de la práctica. Los resultados fueron favorables y las conclusiones permitieron llevar la implementación con mayor agilidad.

Durante la etapa de implementación, la participación de los estudiantes fue a gran escala. Al inicio se atendió en promedio más de mil estudiantes por semestre. Para esto fue necesario contratar nuevos profesores y planificar entre 35 y 45 paralelos por semestre. Desde el punto de vista académico fue necesario coordinar todos los esfuerzos. Se realizó una detallada planificación de todas las actividades que era necesarias realizar por cada profesor. Mantener reuniones periódicas (cada dos semanas) de seguimiento fue un acierto pues se convirtieron en espacios para responder inquietudes de los profesores sobre la metodología de enseñanza y compartir experiencias positivas (buenas prácticas que fueron incorporadas a la metodología) y negativas. Las experiencias negativas fueron analizadas en detalle para buscar soluciones prácticas e inmediatas a los problemas. Una vez más, la experiencia de otros profesores de la materia en resolver problemas similares ayudó a superar todos estos inconvenientes. Finalmente se creó un hackathon (PyWeekend), evento semestral, con el propósito de incrementar la motivación de los estudiantes hacia la materia y crear la oportunidad para ellos de enfrentarse a problemas reales de la industria y comunidad local fuera del aula de clases.

La etapa de evaluación es un proceso constante que se realiza para medir el impacto y beneficios de la materia, realizando los ajustes necesarios. Los medidores más significativos para la evaluación son:

1. La participación de estudiante que no son de Computación en las materias de Computación de libre ingreso como Análisis Exploratorio de Datos, Introducción a la Ciencia de Datos, Desarrollo de Juegos de Vídeos, entre otras.
2. La participación de estudiantes de ESPOL, especialmente que no son de Computación, en competencias de programación (Hackathons, GameJams, AppJams, DataJams, entre otras) considerando también su ubicación final entre todos los concursantes.
3. Encuestas de anónimas de satisfacción.

Identificación y justificación de eventuales cambios durante la implementación

Podemos mencionar dos cambios importantes.

1. La adopción de paralelos masivos (60+ estudiantes). Se probaron cuatro paralelos masivos dictados exclusivamente por un profesor. Inmediatamente se notó que el manejo del aula no era el más apropiado, especialmente en los laboratorios donde el profesor no podía cubrir toda la necesidad. Entonces se sumaron técnicos docentes como personal de apoyo y aunque la experiencia mejoró, terminamos por eliminar esta opción pues no ofrecía una experiencia óptima de enseñanza al compararla con grupos más pequeños (de hasta 30 estudiantes).
2. Realización de un hackathon (PyWeekend). Este evento nació con el propósito de incrementar la motivación de los estudiantes hacia la materia y crear la oportunidad para ellos de enfrentarse a problemas reales de la industria y comunidad local. Se realiza todos los semestres exclusivamente para los estudiantes de Fundamentos de Programación y para muchos es su primer hackathon (pero no el último).

Dimensión: Resultados de la práctica

Proceso de evaluación y descripción de resultados

Para evaluar la práctica se planificaron los siguientes tres medidores:

1. Cantidad de estudiantes de otras carreras que toman materias opcionales e itinerarios de Computación
2. Cantidad de estudiantes que participan en competencias de programación tipo Hackathon dentro y fuera de nuestra universidad
3. Encuestas de satisfacción

Medidor 1

Desde el 2018 la carrera de Computación ofrece materias opcionales para estudiantes de otras carreras en las

cuales ha recibido a 57 estudiantes que no tienen obligación de tomarlas. La distribución de los estudiantes por materia es la siguiente:

| | |
|---------------------------------------|----|
| 1. Análisis Exploratorio de Datos | 7 |
| 2. Minería de Datos | 7 |
| 3. Desarrollo de Juegos de Vídeo | 23 |
| 4. Procesamiento Digital de Imágenes | 6 |
| 5. Introducción a la Ciencia de Datos | 14 |

Medidor 2

Gracias a la iniciativa del PyWeekend cada semestre los estudiantes de la materia de Fundamentos de Programación tienen la oportunidad de participar en un evento tipo Hackathon en el que ponen a pruebas su habilidad de resolución de problemas, su habilidad de programación y trabajo en equipos multidisciplinarios. A continuación mostramos el número de estudiantes que han participado por semestre desde que se inició con este proyecto junto con el número de estudiantes que no eran de la carreras de computación:

| año | # participantes | # participantes no computacion |
|---------|-----------------|--------------------------------|
| 2017-2T | 109 | 95 |
| 2018-1T | 135 | 115 |
| 2018-2T | 156 | 145 |
| 2019-1T | 148 | 134 |
| 2019-2T | 157 | 137 |
| 2020-1T | 160 | 144 |
| 2020-2T | 138 | 112 |

Algunos otros eventos tipo Hackathon en las que estudiantes que han tomado la materia han tenido una participación relevante son:

- Hackathon Banco del Pacífico (Septiembre de 2018), DATA VADERS, ganadores del primer lugar (algunos de los integrantes fueron mentores o participaron en el pyweekend)
- NASA Space Apps Challenge 2019, Team Ideas , finalista globales (algunos de los integrantes fueron mentores o participaron en el PyWeekend)
- NASA Space Apps Challenge 2020, Galatic turtles , ganadores del primer lugar (algunos de los integrantes fueron mentores o participaron en el PyWeekend)

Medidor 3

A lo largo de la práctica hemos realizado encuestas de satisfacción a los alumnos, al menos una vez por año. La participación en estas encuestas es opcional y anónima. Los resultados obtenidos muestran que en promedio el 75% de los estudiantes consideran que la materia ha aportado a desarrollar la habilidad de comprender, descomponer y solucionar problemas usando un lenguaje de programación.

Análisis de la práctica

Dimensión: Proyección de la práctica a futuro

Propuesta de proyección y sostenibilidad de la práctica a futuro; descripción de eventuales ajustes y modificaciones de la práctica para futuras implementaciones

Luego de casi cinco años de llevar adelante la práctica, sentimos que la hemos logrado estabilizar al punto de haber podido sobrellevar reformas académicas y curriculares y más de un año de pandemia que nos obligó a recurrir a la enseñanza mediante medios digitales. El cambio a la enseñanza virtual nos ha dejado muchas lecciones a nosotros como responsables de la práctica. Ahora sabemos qué funciona mejor y qué simplemente no es una buena práctica. A continuación algunas de nuestras principales reflexiones.

Pensando ya en el futuro y el regreso a las aulas, el consenso de los profesores es que la materia

Fundamentos de Programación se desarrolla mejor en un ambiente presencial que virtual. Estar físicamente en un aula permite una mejor interacción con los estudiantes y poder observar sus reacciones frente al material que se discute.

Es evidente que existen estudiantes que aprovechan más de la enseñanza en línea pero no son muchos. La presencia del profesor en el aula es necesaria para poder hacer un seguimiento en directo del grupo a medida que se revisan nuevos temas y se realizan las prácticas en los laboratorios. Esto permite poder responder dudas y orientar a los estudiantes en momentos más oportunos. La enseñanza presencial y la metodología de la materia no impide el estudio autónomo complementario usando medios virtuales.

El mayor reto que tenemos es reincorporarnos a la educación presencial y adoptar las mejores prácticas de la enseñanza en línea como: un mayor aprendizaje autónomo y el uso de herramientas tecnológicas (ej. Ambientes Integrados de Desarrollo Colaborativos en las nubes) que pueden ser utilizados desde prácticamente cualquier dispositivo. De esa manera, un estudiante puede programar inclusive desde su celular mientras participa de una clase presencial con su profesor en el aula.

A los profesores nos queda por diseñar un plan de retorno. Definitivamente nuestras clases presenciales no pueden ser como antes de la pandemia y desde ya estamos trabajando en este plan, discutiendo ideas y lineamientos para el dictado de nuestras clases bajo esta nueva modalidad. Creemos que el espíritu y los objetivos de la materia no van a variar pero las técnicas de enseñanza si deberán ser ajustadas.

Dimensión: Apreciación global de la práctica

Análisis de cumplimiento de objetivos de la práctica

Actualmente la universidad tiene 13 carreras que cuentan con la acreditación internacional ABET. La carrera de Computación forma parte de la comisión de Computing Accreditation Commission (CAC) la cual propone 6 resultados de aprendizaje que deben ser alcanzados a lo largo de la malla, los cuales se encuentran distribuidos en todas las materias.

Adicional a los objetivos propios de la materia Fundamentos de Programación previamente mencionados en la sección 3.2, dentro de la clase se realizan actividades para alcanzar los siguientes resultados de aprendizaje ABET:

1. «Diseñar, implementar y evaluar una solución basada en computación para cumplir con un conjunto dado de requisitos de computación en el contexto de la disciplina del programa.»
2. Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas para la disciplina del programa.»
3. Los objetivos propios de la materia se complementan con los resultados de aprendizaje ABET ya que ambos forman a los estudiantes en el desarrollo del pensamiento lógico para la resolución de problemas, diseñando e implementando soluciones utilizando un lenguaje de programación, además de manejar grupos de estudiantes de diversas carreras lo cual contribuye al trabajo en equipos multidisciplinarios.

A lo largo del semestre se realizan evaluaciones periódicas como talleres, lecciones, evaluaciones, entre otras, para evidenciar el cumplimiento de dichos objetivos.

Además, de manera institucional, las mediciones de los resultados de aprendizaje ABET se realizan cada dos años, el proceso de evaluación consiste en:

- Diseñar herramientas donde se pueden evaluar los diferentes criterios de las rúbricas.
- Dar retroalimentación a los estudiantes sobre los trabajos realizados.
- Realizar un análisis y su posterior reporte en cuanto a los resultados obtenidos.
- Compartir los resultados con el cuerpo docente para realizar un plan de mejoras.
- Implementar el plan de mejoras.

El proceso de medición contribuye directamente a la realización de un plan de mejoras, el cual a su vez nos ayuda a plantearnos nuevos procesos de enseñanza, aplicación de técnicas e integración con nuevos sistemas.

Evaluación reflexiva sobre la práctica implementada, con énfasis en su potencial de transformación

Durante las clases presenciales, los talleres prácticos eran la mejor herramienta para reafirmar conceptos y adquirir nuevas destrezas en programación, los estudiantes realizaban mayor cantidad de consultas que durante las horas teóricas. Este es un aspecto clave que no ha podido ser igualado en la modalidad virtual.

A raíz de la pandemia, la figura ha cambiado, el uso constante de aulas virtuales ha permitido que los estudiantes vayan desarrollando sus códigos a la par, logrando identificar en una etapa temprana los errores comunes y desarrollando habilidades que permiten un mejor entendimiento del contenido. Pensando en nuestro regreso a modalidad presencial, este es uno de los aspectos que queremos incorporar de la modalidad virtual a la modalidad presencial. El uso de herramientas de programación colaborativas en las nubes es pieza fundamental para lograrlo dado que los estudiantes las pueden usar desde cualquier dispositivo, incluyendo celulares o tablets.

Otras ventajas de la virtualidad que nos buscamos mantener en las clases presenciales incluyen: fomentar el trabajo en equipo, mejor integración con los problemas de la vida real y retroalimentación inmediata y precisa.

Como lo mencionamos anteriormente, la práctica también contribuye al descubrimiento y desarrollo de nuevas habilidades, es así como los estudiantes deciden libremente participar en concursos de programación (PyWeekend) propios de la materia, armando sus propios grupos de trabajo e interesándose en nuevas áreas de estudio donde aplican los conocimientos adquiridos en Fundamentos de Programación. Luego, conociendo ya como funcionan estos eventos y movidos por su propia curiosidad e iniciativa, buscan participar de concursos similares organizados fuera de la universidad y la materia, incluyendo eventos de la ciudad e internacionales, como ya lo hemos mencionado anteriormente.

Al final, la práctica ha demostrado ser un factor determinante de transformación en el tipo de personas y profesionales que forma ESPOL. No solo formamos a estudiantes con una mejor preparación académica que les da una ventaja competitiva en el ámbito profesional; prácticas como el PyWeekend, incrementan la motivación y el auto estima de nuestros estudiantes lo que los ha llevado a participar de eventos internacionales (NASA Space Apps Challenge, Google Summer of Code) o a buscar oportunidades profesionales (pasantías laborales) fuera del país en institutos de gran prestigio como el CERN (2 pasantes).

©2021

Centro Interuniversitario de Desarrollo - CINDA

Esta obra está bajo una Licencia de Creative Commons Reconocimiento 4.0.