



## Conference Paper

# Current Trends of Teaching Computer Programming in Undergraduate CS Programs: A Survey from Ecuadorian Universities

Gloria Arcos, Gladys L. Aguirre, Blanca Hidalgo, Raúl H. Rosero, Eduardo Villa, and Omar S. Gómez

Grupo de Investigación GrIISoft, Escuela Superior Politécnica de Chimborazo, 60155 Riobamba, Ecuador

## Abstract

Computer programming has become a relevant element of the modern era. Practically it is involved in all sectors of modern society. Currently, many universities around the globe offer bachelor's degree programs that contain computer programming related courses. In order to gain insight into how computer programming is taught in Ecuadorian Higher Education Institutions (HEIs), we have conducted the first national survey on teaching computer programming in undergraduate CS programs. In this survey participated 53 respondents from 13 Ecuadorian HEIs. We report on programming paradigms, programming languages, integrated development environments and teaching strategies used in the programming learning process. We found that the object-oriented paradigm along with the Java programming language are the most often used. The NetBeans integrated development environment is the most commonly used among the respondents. Concerning teaching strategies, e-learning platforms are commonly used as part of computer programming courses. More than half of the respondents use collaborative strategies such as pair programming in their courses. Finally, a half of the respondents use traditional computer labs for teaching programming; however, other approaches such as mobile devices, games and simulations, Arduino kits and robots start to be used as a teaching strategy. Our findings can serve as a starting point for addressing reforms in computer programming courses taught in Ecuadorian CS undergraduate programs.

**Keywords:** computer programming education, programming pedagogy, programming environments, programming, Ecuador.

## 1. Introducción

A grandes rasgos, la programación informática involucra el proceso de codificar una serie de instrucciones con el fin de ejecutarlas en un computador. Hoy en día estas

Corresponding Author:  
Omar S. Gómez  
ogomez@esPOCH.edu.ec

Received: 28 July 2017  
Accepted: 5 September 2017  
Published: 30 January 2018

Publishing services provided by  
Knowledge E

© Gloria Arcos et al. This article is distributed under the terms of the [Creative Commons](#)

[Attribution License](#), which permits unrestricted use and redistribution provided that the original author and source are credited.

Selection and Peer-review under the responsibility of the SIIPRIN Conference Committee.



instrucciones que conforman uno o varios programas informáticos pueden ejecutarse en una diversidad de dispositivos electrónicos.

Prácticamente, a día de hoy la programación informática se encuentra involucrada en todos los sectores de nuestra sociedad moderna, tales como: educación, ciencia, ocio, salud, defensa, etc. Dada su relevancia, existe una gran demanda a nivel mundial de programadores informáticos [1].

En el ámbito educativo, la programación informática comienza a incorporarse en la enseñanza preuniversitaria [2], [3], [4], [5] inclusive hay evidencia de la enseñanza de la programación informática en educación primaria y secundaria (K-12) [6], [7], [8], [9], así como en educación infantil temprana (Kindergarten) [10], [11].

En el contexto universitario, la enseñanza de la programación informática se centra principalmente en carreras afines a la informática, aunque en otras carreras no afines a ésta comienzan a incorporar cursos sobre programación informática [12], [13], [14], [15], [16], [17].

Con respecto a paradigmas de programación en los que se han reportado experiencias de enseñanza en el ámbito universitario se encuentran: el paradigma orientado a objetos [18], [19, 20], el paradigma funcional [21], [22], el paradigma lógico [23], [24], el paradigma orientado a aspectos [25], el paradigma orientado a eventos [26], [27], así como el paradigma orientado a servicios [28].

En Cuanto a los lenguajes de programación, se observa que Java es el lenguaje predominante en la enseñanza de la programación [29, 30]. Otros dos lenguajes también populares usados para tal fin son C y C++ [31]. No obstante, la enseñanza del lenguaje Python comienza a tomar cierta notoriedad en universidades estadounidenses [32], así como en universidades australianas y neozelandesas [33].

Referente a estrategias de apoyo pedagógico que se emplean en la enseñanza de la programación, se han reportado trabajos que hacen uso de herramientas TIC's de soporte a la enseñanza de la programación tales como: plataformas e-learning [34], [35], [36], [37], o enfoques mixtos como el b-learning [38], [39], [40], o el c-learning que emplea enfoques basados en redes sociales [41] o mundos virtuales [42], [43]. También se han reportado experiencias aplicando la tecnología m-learning asociada a la enseñanza basada en dispositivos móviles [44], así como en el p-learning asociado con cursos en línea masivos abiertos (en Inglés, Massive Open Online Courses o MOOC) [45], [46].

En este mismo ámbito pedagógico, también se han reportado estrategias de trabajo colaborativo para incentivar y desarrollar el aprendizaje de la programación informática tales como la realización de ejercicios de programación en equipos conformados por

2 o más personas [47], [48], así como la realización de ejercicios de programación realizados en pareja [49], [50].

Referente a ambientes de trabajo usados para el aprendizaje de la información informática, se han reportado experiencias de enseñanza utilizando kits de LEGO® [51], [52], robots [53], computadores de placa reducida como el Raspberry Pi [54], videojuegos [55], [56], [57], juegos serios [58], objetos de aprendizaje [59], programación visual [14], animación 3D [60] así como entornos de desarrollo integrado bajo un enfoque educativo [61], [62].

Con el fin de contar con mayor información sobre el estado de la enseñanza de la programación informática en Instituciones de Educación Superior (IES) en el Ecuador, en el presente trabajo reportamos hallazgos preliminares acerca de la primera encuesta nacional sobre la enseñanza de la programación. En este estudio hemos empleado una muestra de 53 docentes participantes pertenecientes a 13 IES. La finalidad de esta encuesta es analizar los paradigmas de programación, lenguajes de programación, entornos de desarrollo integrado, así como estrategias de apoyo pedagógico empleadas en la enseñanza de la programación informática en IES. Los resultados de esta encuesta servirán como referente para contrastarlos con tendencias reportadas a nivel mundial sobre esta temática (anteriormente discutido en esta sección) así como para tomar decisiones informadas para la mejora de planes de estudio de carreras afines a la informática en las IES del Ecuador.

El resto del documento se encuentra estructurado de la siguiente manera: en la Sección 2 presentamos un panorama general acerca de la programación informática. En la Sección 3 describimos el método seguido para el desarrollo y ejecución de la encuesta. En la Sección 4 presentamos los resultados obtenidos. En la Sección 5 discutimos los resultados encontrados. Finalmente, en la Sección 6 presentamos las conclusiones.

## 2. Panorama general de la programación informática

En esta sección presentamos un breve panorama general sobre la programación informática donde se presentan algunos conceptos sobre paradigmas de programación, entornos de programación así como ambientes de apoyo al proceso de enseñanza aprendizaje de la programación informática.

La programación informática es considerada como el proceso de diseñar, codificar, depurar y mantener el código fuente en programas informáticos cuyo código fuente es escrito en uno o varios lenguajes de programación. El proceso de escribir código fuente requiere frecuentemente conocimientos sobre matemáticas discretas, el dominio del

problema a abordar, el paradigma y lenguaje de programación a utilizar, así como el empleo de algoritmos especializados (pensamiento algorítmico).

## 2.1. Paradigmas de programación

En la literatura existen diversos paradigmas de programación [63], [64], [65], [66] o estilos de codificación de programas. Los lenguajes de programación, por naturaleza, se enmarcan en uno o varios paradigmas. Entre los principales paradigmas de programación se encuentran los siguientes:

- Imperativo. Los programas se componen de un conjunto de sentencias que cambian su estado. Son secuencias de comandos que ordenan acciones a la computadora.
- Declarativo. Opuesto al imperativo. Los programas describen los resultados esperados sin listar explícitamente los pasos a llevar a cabo para alcanzarlos.
- Lógico. El problema se modela con enunciados de lógica de primer orden.
- Funcional. Los programas se componen de funciones, es decir, implementaciones de comportamiento que reciben un conjunto de datos de entrada y devuelven un valor de salida.
- Orientado a objetos. El comportamiento del programa es llevado a cabo por objetos, entidades que representan elementos del problema a resolver y tienen atributos y comportamiento.

No obstante, han surgido nuevos paradigmas de programación como son los siguientes:

- Dirigido por eventos. El flujo del programa está determinado por sucesos externos (por ejemplo, una acción del usuario).
- Orientado a aspectos. Apunta a dividir el programa en módulos independientes, cada uno con un comportamiento bien definido.
- Orientado a servicios. En donde a través de la codificación de servicios, se diseñan e implementan programas informáticos, principalmente aplicaciones de negocios. En un servicio se empaqueta cierta funcionalidad que puede ser usada por múltiples sistemas localizados en diferentes ubicaciones.

## 2.2. Entornos de programación

Un entorno de programación (o entorno de desarrollo integrado, en Inglés IDE) se considera como un conjunto de herramientas para automatizar y apoyar diferentes tareas que involucra el desarrollo de un programa informático. Entre las principales funciones de un entorno de programación se encuentran las siguientes: edición del código fuente, interpretación directa, compilación y ejecución del código, refactorización de código fuente, analizar la consistencia y calidad del código, depuración de código, ejecución automática de pruebas unitarias, control de versiones, generación documentación, entre otras. Ejemplos de entornos de programación populares que soportan diferentes lenguajes de programación son: Microsoft Visual Studio, NetBeans y PyCharm [67].

## 2.3. Ambientes de apoyo al proceso enseñanza aprendizaje

Las tecnologías de la información y la comunicación (TIC) son un conjunto de servicios que permiten el aprendizaje interactivo y la educación a distancia; Se han incorporado en la educación en todos los niveles apoyando el proceso enseñanza-aprendizaje, logrando en los estudiantes: colaboración, interés, autonomía. Entre estas tecnologías de apoyo se encuentran el e-learning, b-learning, c-learning, m-learning y p-learning [68], [69]. A continuación describimos brevemente estas tecnologías que se han utilizado como apoyo en el proceso de enseñanza de la programación informática.

**e-learning.** Conocido también como tele-formación, tele-educación o aprendizaje en red, el cual permite suministrar material educativo en línea (a través del Internet) a los usuarios.

**b-learning.** Conocido también como aprendizaje combinado, donde a través del uso de las TIC se aplican enfoques de enseñanza presencial (físico) y no presencial (virtual) con el fin de mejorar el proceso de enseñanza aprendizaje.

**c-learning.** Es un sistema de distribución y absorción de contenidos, basándose en aspectos como la comunicación, comunidad, colaboración y conexión. Se apoya en las redes sociales tales como: Facebook, Twitter, Microblogs, etc; en la realidad virtual como: Second Life y OpenSim. Esta tecnología se centra en el trabajo colaborativo sin tener la necesidad de encontrarse en el mismo lugar, se suelen emplear sistemas de comunicación como audio y video en tiempo real, así como las redes sociales.

**m-learning.** Se apoya en la tecnología móvil como son los dispositivos móviles inteligentes. El m-learning permite a un usuario aprender en cualquier lugar y momento dado que los contenidos educativos están disponibles para cualquier dispositivo móvil.

**p-learning.** Conocido también como aprendizaje personalizado. En éste, el proceso de aprendizaje puede ser guiado o no guiado (siguiendo un proceso de autoformación). La modalidad MOOC (En Inglés, Massive Open Online Course, o cursos en línea masivos abiertos) es un ejemplo de p-learning.

### 3. Método

En esta sección presentamos las preguntas de investigación planteadas en este trabajo así como las características del cuestionario aplicado. El objetivo de este trabajo se centra en conocer el estado actual de la enseñanza de la programación informática en una muestra de Instituciones de Educación Superior (IES) en el Ecuador.

De acuerdo al objetivo antes descrito, hemos derivado las siguientes preguntas de investigación:

- **RQ1.** ¿Qué paradigmas de programación se aplican en la enseñanza de la programación?
- **RQ2.** ¿Qué lenguajes de programación se utilizan en el proceso de enseñanza de la programación?
- **RQ3.** ¿Cuáles son los entornos de desarrollo integrado que se utilizan en la enseñanza de la programación?
- **RQ4.** ¿Cuáles son las estrategias de apoyo pedagógico que se emplean en la enseñanza de la programación?

Para responder a estas preguntas de investigación se desarrolló un cuestionario dividido en dos apartados, uno relacionado con información general sobre la institución y asignaturas impartidas por el encuestado, y el otro apartado conformado por seis preguntas relacionadas con la enseñanza de la programación informática. Este cuestionario se encuentra disponible en el Apéndice A de este documento. La unidad de análisis serán docentes pertenecientes a IES del Ecuador quienes impartan asignaturas relacionadas con la programación informática en carreras de pregrado afines a la informática.

Como estrategia de validación del cuestionario utilizamos la revisión de expertos en conjunto con el enfoque de grupo focal [70]. Durante varias sesiones de trabajo, los autores del presente documento confeccionaron las preguntas del cuestionario, se realizó una pequeña prueba piloto con los mismos autores midiendo los tiempos en completar los ítems de dicho cuestionario.

En cuanto al método de muestreo usado, seleccionamos el muestreo accidental (no probabilístico, por conveniencia). Este tipo de muestreo es frecuentemente utilizado en encuestas hechas en el ámbito de la ingeniería de software [70]. En este tipo de muestreo los participantes son seleccionados con respecto a una serie de contactos previamente conocidos.

En nuestro caso, enviamos invitaciones para completar el cuestionario a diversos docentes y coordinadores de carreras afines a la informática adscritas a diferentes IES en el Ecuador.

**Ejecución.** Una vez diseñado el cuestionario, éste se pasó a la plataforma Survey-Monkey, la cual permite la creación y llenado de encuestas en línea. Éste se mantuvo en línea durante los meses de abril y mayo del 2017.

El cuestionario se envió a varios docentes de varias IES del Ecuador así como a coordinadores de carreras afines a la informática donde divulgaron la encuesta a otros docentes. En total completaron de manera satisfactoria el cuestionario 53 docentes pertenecientes a 13 IES del Ecuador. En la Tabla 1 se muestra la distribución de participantes por IES, así como la categoría y posición de la IES participante.

TABLA 1: Características de las IES participantes en el estudio.

Institución	Categoría	Posición	Participantes
Escuela Politécnica Nacional	A	1 / 8	3
Escuela Superior Politécnica del Litoral	A	2 / 8	2
Universidad de Cuenca	A	4 / 8	5
Universidad de las Fuerzas Armadas ESPE	A	5 / 8	11
Escuela Superior Politécnica de Chimborazo	B	1 / 27	6
Pontificia Universidad Católica del Ecuador	B	2 / 27	4
Universidad Nacional de Loja	B	8 / 27	3
Universidad Técnica de Ambato	B	11 / 27	5
Universidad Técnica Particular de Loja	B	14 / 27	3
Universidad Estatal de Bolívar	C	4 / 16	3
Universidad Nacional de Chimborazo	C	7 / 16	2
Universidad Estatal Península de Santa Elena	C	11 / 16	4
Universidad Técnica de Cotopaxi	C	14 / 16	2
<b>Total</b>			<b>53</b>

En cuanto a las categorías y posiciones de IES en el Ecuador, el CEAACES (Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior),

organismo Ecuatoriano, ha realizado una serie de evaluaciones a las IES del Ecuador, y en función de los resultados obtenidos de dichas evaluaciones, en la actualidad existen 55 IES clasificadas en 5 categorías (A, B, C, D, E), donde la categoría A corresponde a aquellas IES con las mejores puntuaciones con respecto a los siguientes criterios: organización, academia, investigación, vinculación con la sociedad, recursos e infraestructura, y estudiantes.

Para nuestra encuesta, seleccionamos una muestra de universidades clasificadas en las primeras tres categorías del CEAACES.

## 4. Resultados

Tras completar el llenado de los cuestionarios, se procedió con el pre-procesamiento de la información. Las respuestas de los cuestionarios se pasaron a una hoja de cálculo donde se efectuaron una serie de análisis.

De los 53 participantes en esta encuesta, 30% corresponden a mujeres mientras que el restante 70% a hombres. Las edades de los participantes oscilan entre los 29 y 65 años de edad, con un promedio de 43 años de edad. Las preguntas planteadas fueron multi-respuesta, razón por la cual, se ha considerado en el procesamiento de la información todas las respuestas proporcionadas por los encuestados.

Los resultados de la investigación fueron analizados de acuerdo a las preguntas de investigación antes planteadas. A continuación presentamos los resultados organizados de acuerdo a las preguntas de investigación.

### 4.1. RQ1. ¿Qué paradigmas de programación se aplican en la enseñanza de la programación?

El paradigma de *programación orientado a objetos* es el que más se enseña en las carreras de Informática del Ecuador, el 75% de los encuestados, enseñan este paradigma. La *programación estructurada* que corresponde al paradigma de programación procedural es aplicada por el 18,2% de los encuestados. En la Tabla 2, se presentan los porcentajes de utilización de los paradigmas incluidos en la encuesta en la que se evidencia que los 5 paradigmas más utilizados son: *orientado a objetos*, *procedural*, *funcional*, *basado en servicios* y *dirigido por eventos*. El paradigma de programación simbólico, es enseñado por el 0,8% de los encuestados. Otros paradigmas como la *programación orientada a aspectos* es enseñado por el 1,5% de los encuestados.



TABLA 2: Resultados de paradigmas de programación utilizados.

Paradigmas de Programación	Respuestas	
	N	Porcentaje
Orientados a Objetos	40	30,3%
Procedural	20	15,2%
Funcional	17	12,9%
Basado en Servicios	14	10,6%
Dirigido por eventos	13	9,8%
Lógico	12	9,1%
Declarativo	7	5,3%
Imperativo	4	3,0%
Orientado a Aspectos	2	1,5%
Simbólico	1	0,8%
Otros	2	1,5%
<b>Total</b>	<b>132</b>	<b>100,0%</b>

#### 4.2. RQ2. ¿Qué lenguajes de programación se utilizan en el proceso de enseñanza de la programación?

En cuanto a los lenguajes de programación, observamos que Java y Javascript son los lenguajes que más se enseñan en las instituciones de educación superior, un 33,7% de las respuestas obtenidas incluyen a estos lenguajes. El 29% de las respuestas corresponden a C, C++ y C#. Otro resultado importante es la enseñanza de PHP para el desarrollo de aplicaciones Web. Los lenguajes.Net, C# y Visual Basic son usados por un 13,1% de encuestados. De acuerdo a nuestros hallazgos, los 5 lenguajes que más se enseñan en las carreras de informática, corresponden a: Java, Javascript, C++, PHP y C# (ver Tabla 3). Por otro lado, en la encuesta fueron incorporados los siguientes lenguajes: Ensamblador, Labview G-Language, Ruby, Perl, Cobol, Objective C, Fortran y Logo que no tuvieron respuestas que evidencien su utilización, algunos de ellos son lenguajes que ya no están vigentes en el mercado y otros por su reciente aparición, aún no han sido incorporados en el proceso de enseñanza de la programación en las universidades ecuatorianas.

TABLA 3: Resultados de lenguajes de programación utilizados.

Lenguajes de programación	Respuestas	
	N	Porcentaje
Java	35	20,7%
Java Script	22	13,0%
C++	21	12,4%
PHP	17	10,1%
C#	16	9,5%
C	12	7,1%
Python	7	4,1%
Visual Basic	6	3,6%
R	6	3,6%
ASP	4	2,4%
Prolog	3	1,8%
Scala	2	1,2%
Lisp	2	1,2%
MatLab	2	1,2%
Turbo Pascal	1	0,6%
F#	1	0,6%
Otros	11	6,5%
<b>Total</b>	<b>169</b>	<b>100,0%</b>

### 4.3. RQ3. ¿Cuáles son los entornos de desarrollo integrado que se utilizan en la enseñanza de la programación?

NetBeans y Eclipse, son los IDE más utilizados en la enseñanza de la programación, un 54,3% de los encuestados utilizan estos entornos; esto puede deberse a que son software de código abierto y al soporte de Java, C y PHP que éstos incluyen. Otro IDE utilizado es Microsoft.Net que se ubica como el tercero que más que enseña en las carreras de informática del Ecuador. Entre otros IDEs utilizados y que no fueron incluidos en la encuesta constan: Code::Blocks, un entorno de desarrollo integrado libre y multiplataforma para el desarrollo de programas en lenguaje C y C++, Dev-C++ que tiene el mismo fin, Pycharm, un IDE utilizado exclusivamente para Python y IntelliJ IDEA, un IDE para Java; entre otros. La Tabla 4 muestra los resultados obtenidos en el estudio.

TABLA 4: Resultados de entornos de programación.

Entornos de Desarrollo Integrado	Respuestas	
	N	Porcentaje
NetBeans	34	29,3%
Microsoft Visual Studio	18	15,5%
Eclipse	29	25,0%
JDeveloper	5	4,3%
R Studio	4	3,4%
Xcode	3	2,6%
MatLab	2	1,7%
JBuilder	2	1,7%
Ruby Mine	1	0,9%
Otros	18	15,5%
<b>Total</b>	<b>116</b>	<b>100,0%</b>

#### 4.4. RQ4. ¿Cuáles son las estrategias de apoyo pedagógico que se emplean en la enseñanza de la programación?

Esta pregunta hemos examinado las herramientas TICs de soporte al proceso de enseñanza-aprendizaje, estrategias de trabajo colaborativo empleadas así como ambientes de aprendizaje utilizados para tal fin.

**Herramientas TICs de soporte.** Como se muestra en la Tabla 5, el e-learning y b-learning son las plataformas utilizadas por el 77,9% de los docentes ya sea para la enseñanza tanto a distancia como presencial, el uso de redes sociales a través del c-learning es utilizado por un 10,8% de los docentes. Las tendencias de utilización de tecnologías móviles y moocs aún no son aplicadas de forma considerable en el proceso de enseñanza de la programación informática, el 4,8% y el 6% respectivamente la usan.

TABLA 5: Resultados de las herramientas TICs utilizadas.

Herramientas TICs	Respuestas	
	N	Porcentaje
e-Learning	39	47,0%
b-Learning	24	28,9%
c-Learning	9	10,8%
p-Learning	5	6,0%
m-Learning	4	4,8%
Otros	2	2,4%
<b>Total</b>	<b>83</b>	<b>100,0%</b>

**Estrategias de trabajo colaborativo.** Observamos que los encuestados aplican varias estrategias de trabajo colaborativo en el aula siendo el trabajo individual el más utilizado por los docentes (ver Tabla 6), sin embargo, esta estrategia es combinada por el trabajo en equipo, ya sea en parejas o en equipos de más de dos personas. El 63% de los encuestados aplican el trabajo colaborativo en el proceso de la enseñanza de la programación.

TABLA 6: Resultados de las estrategias de trabajo colaborativo.

Trabajo Colaborativo	Respuestas	
	N	Porcentaje
Individual	37	37,0%
En parejas	34	34,0%
En equipos de más de dos personas	29	29,0%
<b>Total</b>	<b>100</b>	<b>100,0%</b>

**Ambientes de aprendizaje.** El 44,8% de los encuestados utilizan ambientes de trabajo distintos al uso de laboratorios de cómputo, es importante recalcar la tendencia hacia la inclusión de dispositivos móviles y software de juegos y simuladores en el proceso de enseñanza de la programación, un 38,5% de los encuestados utilizan estos ambientes. Observamos también que se han empezado a utilizar otros ambientes como el uso de robots y kits Arduino para introducir a los estudiantes en el aprendizaje de la programación. Los resultados se muestran en la Tabla 7.

TABLA 7: Resultados de los ambientes de trabajo.

Ambientes de Trabajo	Respuestas	
	N	Porcentaje
Uso de computadoras	53	55,2%
Uso de dispositivos móviles	20	20,8%
Software de juegos y simuladores	17	17,7%
Uso de kits Arduino	4	4,2%
Uso de robots	1	1,0%
Otros	1	1,0%
<b>Total</b>	<b>96</b>	<b>100,0%</b>

## 5. Discusión

En esta sección discutimos los resultados encontrados en relación con las preguntas de investigación. Con respecto a los paradigmas de programación (**RQ1**), observamos que el paradigma de programación más popular usado en la enseñanza de la programación es el paradigma orientado a objetos. En segundo y tercer lugar se encuentran los paradigmas procedural y funcional. Estos resultados son corroborados por otro estudio recientemente realizado en Reino Unido [29]. En este estudio, los autores observan que los tres paradigmas de programación más populares usados en la enseñanza de la programación son: el paradigma orientado a objetos, procedural y funcional.

Referente a la segunda pregunta de investigación (**RQ2**), en el cuestionario diseñado se incluyeron 25 lenguajes de programación, no obstante sólo 17 de ellos fueron escogidos por los encuestados; es decir, que sólo se escogió un 68% de los lenguajes. La opción incluida de *Otros* lenguajes corresponde al 6,5% de los docentes encuestados; sin embargo, el análisis de estas respuestas ha permitido determinar la confusión que existen entre los lenguajes de programación, IDEs, lenguajes de marcado que fueron incluidos en la descripción de esta opción, así mismo se determinaron dos lenguajes que no habían sido incluidos en la encuesta y que fueron parte de las respuestas, es el caso de *Typescript*, un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft basado en JavaScript y SQL (Structured Query Language), que es un lenguaje declarativo de acceso a bases de datos relacionales.

Por otro lado, de acuerdo a nuestros resultados, los cinco lenguajes populares más enseñados por docentes de IES en el Ecuador son: Java, Javascript, C++, PHP y C#. A nivel mundial la popularidad de los lenguajes de programación suele medirse de acuerdo a tres indicadores que son: el índice TIOBE, el índice PYPL y el ranking de RedMonk [71].

El índice TIOBE [72] mide la popularidad de los lenguajes de programación de acuerdo al número de búsquedas hechas en diferentes motores de búsqueda como: Google, Bing, Yahoo!, Wikipedia, Amazon, Youtube y Baidu. En el caso del índice PYPL [73], éste se basa de acuerdo a tutoriales buscados en el Internet, se asume entonces que entre más tutoriales se busquen sobre un lenguaje particular, éste es más popular. Por otra parte el ranking de RedMonk [74] se basa en extraer indicadores de los sitios GitHub y Stack Overflow para calcular la popularidad de los lenguajes de programación. En la Tabla 8 se muestran las cinco posiciones en cuanto a la popularidad de los lenguajes de programación para contrastarlos como nuestros resultados.

TABLA 8: Popularidad de los lenguajes de programación.

Posición	TIOBE (Ago, 2017)	PYPL (Ago, 2017)	RedMonk (3er Trim., 2017)	IES Ecuador (2017)
1	Java	Java	JavaScript	Java
2	C	Python	Java	JavaScript
3	C++	PHP	Python	C++
4	C#	C#	PHP	PHP
5	Python	JavaScript	C#	C#

Como se observa en la Tabla 8, el lenguaje Java se encuentra en las primeras posiciones en los tres índices antes mencionados, así como en nuestros resultados.

Como dato de interés, cuando en 1996, Chandler y Hand [75] informaron sobre la aceptación de la programación orientada a objetos en las universidades del Reino Unido, aproximadamente un tercio de los departamentos académicos afines a la informática expresaron interés por este paradigma, y ninguna institución del Reino Unido había adoptado Java como su principal lenguaje de programación; más de dos décadas después, observamos que la programación orientada a objetos es ampliamente utilizada y que Java es uno de los lenguajes de programación dominante, en nuestro caso, que se enseña en el 93% de las instituciones que participaron en nuestra encuesta.

Por otra parte, observamos que el lenguaje Python comienza a cobrar relevancia a nivel mundial, en los tres índices éste se encuentra dentro de las primeras cinco posiciones. No obstante, en nuestros resultados éste aparece en la séptima posición. Este hallazgo sugiere que de manera paulatina comienzan a incorporarse cursos sobre el lenguaje Python en IES del Ecuador.

En cuanto a la tercera pregunta de investigación (**RQ3**), nuestros resultados indican que el entorno de desarrollo integrado más popular usado en el proceso de enseñanza-aprendizaje de la programación es NetBeans, seguido de Microsoft Visual Studio. En la tercera posición se encuentra el IDE Eclipse. Con respecto al estudio reportado en [29], los autores observan que Eclipse es el IDE más popular seguido de BlueJ y Netbeans. Cabe señalar que BlueJ es un IDE para el lenguaje de programación Java, desarrollado principalmente para un propósito educativo. Dada la naturaleza de este IDE valdría la pena comenzar a utilizarlo para el proceso de enseñanza-aprendizaje de la programación informática en IES del Ecuador.

Con respecto a la cuarta pregunta de investigación (**RQ4**), observamos que las plataformas e-learning son ampliamente utilizadas como apoyo al proceso de enseñanza-aprendizaje de la programación informática. No obstante, observamos

también que los docentes comienzan a explorar otras alternativas como son el b-learning, c-learning, p-learning y m-learning.

Observamos también que los docentes suelen utilizar estrategias de trabajo colaborativo como el trabajo en parejas y en equipos de más de dos personas. Existe evidencia a favor del uso de estrategias colaborativas en el aprendizaje de la programación informática [76], [77]. Dada esta evidencia positiva, recomendamos a los docentes que en la medida de lo posible utilicen estrategias colaborativas en el proceso de enseñanza-aprendizaje de la programación informática.

Finalmente en cuanto a los ambientes de trabajo, observamos que la mayoría de docentes utiliza laboratorios de cómputo para la enseñanza de la programación, no obstante el uso de otros ambientes como son: dispositivos móviles, juegos y simuladores, kits Arduino y robots comienzan también a utilizarse en la enseñanza de la programación informática.

## 5.1. Limitaciones

A continuación describimos cómo minimizamos algunas de las amenazas que suelen reportarse en estudios de este tipo. El sesgo en la selección de los participantes se convierte en una posible amenaza por haber usado un método de muestreo no convencional, en total participaron 13 IES de 55 IES clasificadas por el CEAACES. No obstante, en el muestro utilizado recabamos información de universidades clasificadas en las tres primeras categorías del CEAACES y que están geográficamente ubicadas en distintas regiones del País. No obstante, los resultados aquí presentados pueden servir como un referente inicial sobre el estado de la enseñanza de la programación informática en IES del Ecuador.

Para garantizar la confidencialidad de los participantes, los cuestionarios fueron respondidos de manera anónima garantizando la información sensible de los participantes. La validación del cuestionario se realizó a través de reuniones tipo grupo focal en donde se fueron refinando las distintas preguntas. Las preguntas diseñadas en nuestro cuestionario son similares a las reportadas en otros estudios [29], [33].

## 6. Conclusiones

En este trabajo, hemos reportado resultados iniciales acerca de la primera encuesta nacional sobre la enseñanza de la programación informática en carreras afines a la informática pertenecientes a IES del Ecuador. Los resultados encontrados sirven como

un referente inicial para tener un mejor entendimiento acerca del rol y efectividad de la enseñanza de la programación en IES del Ecuador. Estos resultados pueden también servir como soporte para la toma de decisiones en la actualización de contenidos en asignaturas afines con esta temática de estudio.

Como trabajo futuro, los resultados de la encuesta aquí reportada servirán para la realización de un estudio longitudinal, en donde de manera periódica se mejore y aplique esta encuesta con la finalidad de contar con mayor información acerca de la evolución en la enseñanza de la programación informática en IES del Ecuador.

## Referencias

- [1] J. Bailey and R. B. Mitchell, "Industry Perceptions of the Competencies Needed by Computer Programmers: Technical, Business, and Soft Skills," *J. Comput. Inf. Syst.*, vol. 47, no. 2, pp. 28-33, Dec. 2006.
- [2] C. Beerli, A. Yehudai, J. Gal-Ezer, and D. Harel, "A High School Program in Computer Science," *Computer*, vol. 28, no. 10, pp. 73-80, 1995.
- [3] D. M. Kurland, R. D. Pea, C. Clement, and R. Mawby, "A Study of the Development of Programming Ability and Thinking Skills in High School Students," *J. Educ. Comput. Res.*, vol. 2, no. 4, pp. 429-458, Nov. 1986.
- [4] L. Grandell, M. Peltomäki, R.-J. Back, and T. Salakoski, "Why Complicate Things?: Introducing Programming in High School Using Python," in *Proceedings of the 8th Australasian Conference on Computing Education - Volume 52*, Darlinghurst, Australia, Australia, 2006, pp. 71-80.
- [5] T. C. Wang, W. H. Mei, S. L. Lin, S. K. Chiu, and J. M. C. Lin, "Teaching programming concepts to high school students with Alice," in *2009 39th IEEE Frontiers in Education Conference*, 2009, pp. 1-6.
- [6] S. Kanemune, T. Nakatani, R. Mitarai, S. Fukui, and Y. Kuno, "Dolittle - experiences in teaching programming at K12 schools," in *Proceedings. Second International Conference on Creating, Connecting and Collaborating through Computing*, 2004., 2004, pp. 177-184.
- [7] F. Kalelioğlu, "A new way of teaching programming skills to K-12 students: Code.org," *Comput. Hum. Behav.*, vol. 52, pp. 200-210, Nov. 2015.
- [8] S. Grover and R. Pea, "Computational Thinking in K-12: A Review of the State of the Field," *Educ. Res.*, vol. 42, no. 1, pp. 38-43, Jan. 2013.
- [9] H. Hong, J. Wang, and S. H. Moghadam, "K-12 Computer Science Education Across the U.S.," in *Informatics in Schools: Improvement of Informatics Knowledge and*



- Perception, vol. 9973, A. Brodnik and F. Tort, Eds. Cham: Springer International Publishing, 2016, pp. 142–154.
- [10] A. Strawhacker, M. Lee, C. Caine, and M. Bers, "ScratchJr Demo: A Coding Language for Kindergarten," in Proceedings of the 14th International Conference on Interaction Design and Children, New York, NY, USA, 2015, pp. 414–417.
- [11] T. Meadthaisong, S. Meadthaisong, and S. Chaowaskoo, "Tangible Programming for Basic Control System New Frameworks to Engineering Education for Children," *Adv. Mater. Res.*, vol. 931–932, pp. 1298–1302, 2014.
- [12] P. K. Chilana et al., "Perceptions of non-CS majors in intro programming: The rise of the conversational programmer," in 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 2015, pp. 251–259.
- [13] A. Regalado-Méndez, F. K. Delgado-Vidal, R. E. Martínez-López, and E. Peralta-Reyes, "BALANCEO DE ECUACIONES QUÍMICAS INTEGRANDO LAS ASIGNATURAS DE QUÍMICA GENERAL, ALGEBRA LINEAL Y COMPUTACIÓN: UN ENFOQUE DE APRENDIZAJE ACTIVO," *Form. Univ.*, vol. 7, no. 2, pp. 29–40, 2014.
- [14] W. L. Honig, "Teaching and Assessing Programming Fundamentals for Non Majors with Visual Programming," in Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education, New York, NY, USA, 2013, pp. 40–45.
- [15] B. Furman and E. Wertz, "A first course in computer programming for mechanical engineers," in Proceedings of 2010 IEEE/ASME International Conference on Mecha-tronic and Embedded Systems and Applications, 2010, pp. 70–75.
- [16] W. X. Shandong, "Strategies for teaching java programming language to economics and management majors," presented at the WIT Transactions on Information and Communication Technologies, 2014.
- [17] G. Kiss, "Teaching Programming in the Higher Education not for Engineering Students," *Procedia - Soc. Behav. Sci.*, vol. 103, pp. 922–927, Nov. 2013.
- [18] K. Beck and W. Cunningham, "A laboratory for teaching object oriented thinking," 1989, pp. 1–6.
- [19] C. Griggio, G. Leiva, G. Polito, G. Decuzzi, and N. Passerini, "A Programming Environment Supporting a Prototype-based Introduction to OOP," in Proceedings of the International Workshop on Smalltalk Technologies, New York, NY, USA, 2011, p. 5:1–5:5.
- [20] G. M. Ferreira, M. Z. Nascimento, K. D. R. Assis, and R. P. Ramos, "Teaching object oriented programming computer languages: learning based on projects," in

- International Conference on Software Engineering Advances (ICSEA 2007), 2007, pp. 81–81.
- [21] S. Joosten, K. Van Den Berg, and G. Van Der Hoeven, "Teaching functional programming to first-year students," *J. Funct. Program.*, vol. 3, no. 01, pp. 49–65, Jan. 1993.
- [22] S. Thompson, "Where do I begin? A problem solving approach in teaching functional programming," in *Programming Languages: Implementations, Logics, and Programs*, vol. 1292, H. Glaser, P. Hartel, and H. Kuchen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 323–334.
- [23] S. Friese, "Measuring of and reacting to learners' progress in logic programming courses," 2010, p. 152.
- [24] S. Vosinakis, P. Koutsabasis, and G. Anastassakis, "A Platform for Teaching Logic Programming Using Virtual Worlds," 2014, pp. 657–661.
- [25] I. Boticki, M. Katic, and S. Martin, "Exploring the Educational Benefits of Introducing Aspect-Oriented Programming Into a Programming Course," *IEEE Trans. Educ.*, vol. 56, no. 2, pp. 217–226, May 2013.
- [26] K. B. Bruce, A. P. Danyluk, and T. P. Murtagh, "Event-driven programming is simple enough for CS1," *ACM SIGCSE Bull.*, vol. 33, no. 3, pp. 1–4, Sep. 2001.
- [27] H. B. Christensen and M. E. Caspersen, "Frameworks in CS1: a different way of introducing event-driven programming," *ACM SIGCSE Bull.*, vol. 34, no. 3, p. 75, Sep. 2002.
- [28] X. Liu, R. Raj, T. Reichmayr, C. Liu, and A. Pantaleev, "Incorporating Service-Oriented Programming techniques into undergraduate CS and SE curricula," 2013, pp. 1369–1371.
- [29] E. Murphy, T. Crick, and J. H. Davenport, "An Analysis of Introductory Programming Courses at UK Universities," *Art Sci. Eng. Program.*, vol. 1, no. 2, Apr. 2017.
- [30] B. Chalk and K. Fraser, "A Survey on the teaching of introductory programming in Higher Education," *ICS High. Educ. Acad*, 2005. [Online]. Available: [https://www.researchgate.net/publication/228383736\\_A\\_Survey\\_on\\_the\\_teaching\\_of\\_introductory\\_programming\\_in\\_Higher\\_Education](https://www.researchgate.net/publication/228383736_A_Survey_on_the_teaching_of_introductory_programming_in_Higher_Education). [Accessed: 30-Jul-2017].
- [31] A. Pears et al., "A survey of literature on the teaching of introductory programming," 2007, p. 204.
- [32] P. Guo, "Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities." [Online]. Available: <https://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-u-s-universities/fulltext>. [Accessed: 30-Jul-2017].

- [33] R. Mason and Simon, "Introductory Programming Courses in Australasia in 2016," 2017, pp. 81-89.
- [34] P. Neve, G. Hunter, and D. Livingstone, "NoobLab: An E-learning Platform for Teaching Programming," 2014, pp. 55-62.
- [35] K. Chrysafiadi and M. Virvou, "PeRSIVA: An empirical evaluation method of a student model of an intelligent e-learning environment for computer programming," *Comput. Educ.*, vol. 68, pp. 322-333, Oct. 2013.
- [36] Y. Takemura, H. Nagumo, K.-L. Huang, and H. Tsukamoto, "Assessing the Learners' Motivation in the E-Learning Environments for Programming Education," in *Advances in Web Based Learning - ICWL 2007*, vol. 4823, H. Leung, F. Li, R. Lau, and Q. Li, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 355-366.
- [37] B. McCarthy, "Applying E-Learning Technologies to Teach Computer Programming: A Case Study," in *Proceedings of the 7th International Conference on Electronic Business*, pp. 242-245.
- [38] C. J. Olelewe and E. E. Agomuo, "Effects of B-learning and F2F learning environments on students' achievement in QBASIC programming," *Comput. Educ.*, vol. 103, pp. 76-86, Dec. 2016.
- [39] T. B. Bati, H. Gelderblom, and J. van Biljon, "A blended learning approach for teaching computer programming: design for large classes in Sub-Saharan Africa," *Comput. Sci. Educ.*, vol. 24, no. 1, pp. 71-99, Jan. 2014.
- [40] O. Deperlioglu and U. Kose, "The effectiveness and experiences of blended learning approaches to computer programming education," *Comput. Appl. Eng. Educ.*, vol. 21, no. 2, pp. 328-342, Jun. 2013.
- [41] Ö. Özyurt and H. Özyurt, "Using Facebook to enhance learning experiences of students in computer programming at Introduction to Programming and Algorithm course: USING FACEBOOK FOR COMPUTER PROGRAMMING LEARNING," *Comput. Appl. Eng. Educ.*, vol. 24, no. 4, pp. 546-554, Jul. 2016.
- [42] T. Richter et al., "ViPLab - A Virtual Programming Laboratory for Mathematics and Engineering," 2011, pp. 537-542.
- [43] M. Esteves, B. Fonseca, L. Morgado, and P. Martins, "Improving teaching and learning of computer programming through the use of the Second Life virtual world: Improving teaching and learning through Second Life," *Br. J. Educ. Technol.*, vol. 42, no. 4, pp. 624-637, Jul. 2011.
- [44] O. Ortiz, P. M. Alcover, F. Sanchez, J. A. Pastor, and R. Herrero, "M-Learning Tools: The Development of Programming Skills in Engineering Degrees," *IEEE Rev. Iberoam. Tecnol. Aprendiz.*, vol. 10, no. 3, pp. 86-91, Aug. 2015.

- [45] A. Vihavainen, M. Luukkainen, and J. Kurhila, "Multi-faceted support for MOOC in programming," 2012, p. 171.
- [46] M. (Moti) Ben-Ari, "MOOCs on introductory programming: a travelogue," *ACM Inroads*, vol. 4, no. 2, p. 58, Jun. 2011.
- [47] D. Phuong, F. Harada, and H. Shimakawa, "Collaborative Learning Environment to Improve Novice Programmers with Convincing Opinions in Computer Room," 2009, pp. 61–66.
- [48] D. Čubranić and M. A. D. Storey, "Collaboration support for novice team programming," 2005, p. 136.
- [49] S. Balik et al., "Pair Programming In Introductory Programming Labs," presented at the 2003 Annual Conference, 2003, p. 8.911.1-8.911.12.
- [50] A. Radermacher and G. Walia, "Investigating student-instructor interactions when using pair programming: An empirical study," 2011, pp. 41–50.
- [51] P. B. Lawhead et al., "A road map for teaching introductory programming using LEGO© mindstorms robots," *ACM SIGCSE Bull.*, vol. 35, no. 2, p. 191, Jun. 2003.
- [52] E. Wang, J. LaCombe, and A.-M. Vollstedt, "Teaching Structured Programming Using Lego Programmable Bricks," presented at the 2007 Annual Conference & Exposition, 2007, p. 12.1370.1-12.1370.10.
- [53] M. Sugimoto, T. Fujita, H. Mi, and A. Krzywinski, "RoboTable2: a novel programming environment using physical robots on a tabletop platform," 2011, p. 1.
- [54] M. Kolling, "Educational Programming on the Raspberry Pi," *Electronics*, vol. 5, no. 3, pp. 1–17, Mar. 2016.
- [55] P. Mozelius et al., "Game-Based Technologies in Teaching Programming in Higher Education: Theory and Practices," *Recent Pat. Comput. Sci.*, vol. 9, no. 2, pp. 105–113, Jul. 2016.
- [56] S. Mladenović, Krpan, Divna, and M. Mladenović, "Using Games to Help Novices Embrace Programming: From Elementary to Higher Education," *Int. J. Eng. Educ.*, vol. 32, no. 1B, pp. 521–531, Jan. 2016.
- [57] S. Xinogalos, C. Malliarakis, D. Tsompanoudi, and M. Satratzemi, "Microworlds, Games and Collaboration: three effective approaches to support novices in learning programming," 2015, pp. 1–8.
- [58] T. Mitamura, Y. Suzuki, and T. Oohori, "Serious games for learning programming languages," 2012, pp. 1812–1817.
- [59] L. C. Begosso, L. R. Begosso, A. Ribeiro, R. M. dos Santos, and R. H. Begosso, "The use of learning objects for teaching computer programming," 2015, pp. 1–6.

- [60] A. Salim et al., "On using 3D animation for teaching computer programming in Cairo University," in 2010 The 7th International Conference on Informatics and Systems (INFOS), 2010, pp. 1–5.
- [61] K. Johnsgard and J. McDonald, "Using Alice in Overview Courses to Improve Success Rates in Programming I," 2008, pp. 129–136.
- [62] M. Kölling, B. Quig, A. Patterson, and J. Rosenberg, "The BlueJ System and its Pedagogy," *Comput. Sci. Educ.*, vol. 13, no. 4, pp. 249–268, Dec. 2003.
- [63] A. L. Ambler, M. M. Burnett, and B. A. Zimmerman, "Operational versus definitional: a perspective on programming paradigms," *Computer*, vol. 25, no. 9, pp. 28–43, Sep. 1992.
- [64] R. W. Floyd, "The paradigms of programming," *Commun. ACM*, vol. 22, no. 8, pp. 455–460, Aug. 1979.
- [65] O. S. Gómez, *Paradigma de programación dirigida por eventos: Una visión general*. Amazon KDP, 2015.
- [66] A. Sillitti, T. Vernazza, and G. Succi, "Service Oriented Programming: A New Paradigm of Software Reuse," in *Software Reuse: Methods, Techniques, and Tools*, vol. 2319, C. Gacek, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 269–280.
- [67] C. Arsenault, "Best IDE Software - a List of the Top 10," *KeyCDN Blog*, 08-Dec-2016. [Online]. Available: <https://www.keycdn.com/blog/best-ide/>. [Accessed: 03-Aug-2017].
- [68] S. Martin, R. Gil, G. Diaz, E. Sancristobal, M. Castro, and J. Peire, "From e-learning to m-learning through b-learning and s-learning," in 2008 50th International Symposium ELMAR, 2008, vol. 2, pp. 341–344.
- [69] H. Gardner, J. B., *E-learning: Concepts and practice*. Londres: SAGE, 2006.
- [70] J. Linåker, S. M. Sulaman, R. M. de Mello, and M. Höst, "Guidelines for Conducting Surveys in Software Engineering," 2015.
- [71] Thibaud, "Top Programming Languages to Learn in 2017," *CodinGame Blog*, 18-Jan-2017. [Online]. Available: <https://www.codingame.com/blog/top-programming-languages-to-learn-in-2017/>. [Accessed: 05-Aug-2017].
- [72] "TIOBE Index | TIOBE - The Software Quality Company." [Online]. Available: <https://www.tiobe.com/tiobe-index/>. [Accessed: 05-Aug-2017].
- [73] "PYPL PopularitY of Programming Language index." [Online]. Available: <http://pypl.github.io/PYPL.html>. [Accessed: 05-Aug-2017].
- [74] S. O'Grady, "The RedMonk Programming Language Rankings: June 2017," *tecosystems*, 08-Jun-2017. [Online]. Available:

<http://redmonk.com/sogrady/2017/06/08/language-rankings-6-17/>. [Accessed: 05-Aug-2017].

- [75] J. Chandler and S. Hand, "Teaching Object Technology in Britain," in Proceedings of TOOLS'96, California, 1996.
- [76] C. McDowell, L. Werner, H. Bullock, and J. Fernald, "The effects of pair-programming on performance in an introductory programming course," 2002, p. 38.
- [77] D. Preston, "PAIR Programming As a Model of Collaborative Learning: A Review of the Research," J Comput Sci Coll, vol. 20, no. 4, pp. 39-45, Apr. 2005.

## Apéndice A. Cuestionario sobre la enseñanza de la programación informática en las carreras de sistemas, informática y computación en Instituciones de Educación Superior (IES) del Ecuador

**Objetivo.** Recopilar información a una muestra seleccionada de varias IES del Ecuador que cuentan con carreras en el ámbito de la informática. De manera particular, este cuestionario se ha desarrollado para obtener información preliminar sobre la situación actual de la enseñanza de la programación informática en la educación superior ecuatoriana en las carreras de Ingeniería de Sistemas, Informática y Computación.

A continuación se presenta la encuesta conformada por dos apartados, donde el primer apartado está compuesto por ítems de información general; mientras que el segundo contiene ítems relacionados a la situación actual de la enseñanza de la programación informática en las IES.

### [Información general]

**Instrucciones.** Complete los ítems que a continuación se muestran:

**IG1.** Nombre de la IES (Institución de Educación Superior) en la que labora:

**IG2.** Nombre de la carrera en la que colabora:

**IG3.** Indique las asignaturas a su cargo involucradas en la programación informática:

**IG4.** Edad del encuestado:

**IG5.** Género del encuestado:

## [Información específica]

**Instrucciones.** Seleccione y/o complemente las opciones que considere pertinentes.

**IE1.** Seleccione los lenguajes de programación que utiliza en sus asignaturas.

Posibles opciones: C, C++, Objective C, Java, PHP, Python, ASP, Ruby, Google Go, Perl, LOGO, Lisp, Prolog, Java Script, Scala, R, Visual Basic, C#, F#, Turbo Pascal, Ensamblador, MATLAB scripting language, Fortran, Cobol, Labview G-Language, otros especifique.

**IE2.** Seleccione los paradigmas de programación que utiliza en sus asignaturas.

Posibles opciones: Imperativo, declarativo, lógico, funcional, orientado a objetos, procedural, simbólico, dirigido por eventos, orientado a aspectos, basado en servicios, otros especifique.

**IE3.** Seleccione los entornos de desarrollo integrado (IDE) que suele utilizar en sus asignaturas.

Posibles opciones: Microsoft Visual Studio, Netbeans, Eclipse, Rstudio, JDeveloper, Delphi, Matlab, JBuilder, Labview, Xcode, RubyMine, otros especifique.

**IE4.** Seleccione las herramientas TICs de soporte que suele utilizar como apoyo al proceso de enseñanza-aprendizaje de la programación informática.

Posibles opciones: e-learning, blend-learning, c-learning (empleando redes sociales), m-learning (a través de dispositivos móviles), p-learning (mooc), otros especifique.

**IE5.** Seleccione los ambientes de aprendizaje que suele utilizar en el proceso de enseñanza de la programación informática.

Posibles opciones: uso de computadoras, uso dispositivos móviles, uso de robots, uso de kits arduino, software de juegos y simuladores, otros especifique.

**IE6.** Seleccione las estrategias colaborativas para desarrollar el aprendizaje de la programación informática que suele utilizar.

Posibles opciones: Individual, en parejas, en equipos de más de tres personas, otros especifique.