

Lenguajes de Programación Orientada a Objetos

Camacho Ignacio Violeta
Ingeniería en Computación
Facultad de Ingeniería
CDMX, México
camachoignaciovioleta@gm
ail.com

Luna Alberto Darwin
Leonel
Ingeniería en Computación
Facultad de Ingeniería
CDMX, México
darwin.luna@ingenieria.un
am.edu

Vences Santillán Carlos
Eduardo
Ingeniería en Computación
Facultad de Ingeniería
CDMX, México
carlos.vences.santillan@gm
ail.com

Villegas Flores Nadia
Ingeniería en Computación
Facultad de Ingeniería
CDMX, México
nadiavilleflo@gmail.com

I. INTRODUCCIÓN

La POO se basa en crear un modelo que brinde solución a un problema de manera más eficiente dentro de su programa, haciendo que la ejecución del programa sea más eficaz al reutilizar código, organizándolo y haciéndolo fácil de mantener, disminuyendo así los posibles errores que puedan llegar a presentarse al momento de la ejecución de un programa al reemplazar el código lineal simple durante la ejecución.

Este paradigma de programación se ha convertido en uno de los más importantes para nuestro desarrollo tecnológico, pues favorece la creación de programas de alta calidad que sostienen un fuerte mantenimiento de la codificación al reutilizarlo y haciéndolo más eficiente. Gracias a que su creación se basó en la manera en la que las partes involucradas piensan, es decir, usuario y máquina, nos permite traducir las cosas de la vida real a lenguaje máquina de una forma sencilla.

II. HISTORIA DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

La Programación Orientación a Objetos (P.O.O.) surge en Noruega en 1967 con un lenguaje llamado Simula 67 desarrollado por Krinsten Nygaard y Ole-Johan Dahl, en el centro de cálculo noruego. Simula 67 introdujo por primera vez los conceptos de clases, corrutinas y subclases, conceptos muy similares a los que conocemos para lenguajes POO hoy en día.

En los primeros años de los 70s, científicos del centro de investigación en Palo Alto Xerox (Xerox park) inventaron el lenguaje Small talk, siendo el primer

lenguaje Orientado a Objetos puro de los lenguajes Orientados a Objetos, el cuál únicamente utilizaba clases y objetos.

La POO se fue convirtiendo en el estilo de programación dominante a mediados de los años 1980, en gran parte debido a la influencia de C++, una extensión del lenguaje de programación C. Su dominación fue consolidada gracias al auge de las interfaces gráficas de usuario para las cuales la POO está particularmente bien adaptada. En este caso, se habla también de programación dirigida por eventos.

Las características de orientación a objetos fueron agregadas a muchos lenguajes existentes durante ese tiempo, incluyendo Ada, BASIC, Lisp más Pascal, entre otros. La adición de estas características a los lenguajes que no fueron diseñados inicialmente para ellas condujo a menudo a problemas de compatibilidad y en la capacidad de mantenimiento del código. Los lenguajes orientados a objetos "puros", por su parte, carecían de las características de las cuales muchos programadores habían venido a depender. Para saltar este obstáculo, se hicieron muchas tentativas para crear nuevos lenguajes basados en métodos orientados a objetos, pero permitiendo algunas características imperativas de maneras "seguras".

III. CARACTERÍSTICAS

El paradigma de la programación orientada a objetos consiste en la representación de la realidad. En éste se manejan algunos conceptos básicos como son clases, objetos, atributos, métodos y se caracteriza por emplear la abstracción de datos, herencia, encapsulamiento y

polimorfismo. Estas características deben ser estudiadas y comprendidas para su aplicación en la programación orientada a objetos.

Abstracción: La abstracción es un procedimiento que permite la elección de una determinada entidad de la realidad, sus características y funciones que desempeñan, la cual es representada mediante clases que contienen atributos y métodos de dicha clase.

Encapsulamiento: En programación el encapsulamiento orienta a objetos, se acostumbra a proteger la información o el estado de los atributos para que no se pueda ver o modificar la información del objeto sin el mecanismo adecuado. Para ello, se utilizan métodos para recuperar la información (getters) y asegurar que la información proporcionada sea consistente con el objeto; y a su vez, poder asignar (setters) un nuevo valor y verificar que no afecte la integridad del objeto.

Herencia: La herencia nos permite reutilizar el código programado en cada clase “heredando” o extendiendo las características de un objeto a sus “descendientes” o derivados.

Polimorfismo: El polimorfismo, en una colección de objetos con herencia, si las clases especializadas de una superclase tienen un método con la misma definición o firma, reaccionarán de la manera adecuada cuando reciban el mismo mensaje.

IV. JAVA

A. Historia

El lenguaje de programación Java fue creado en 1991 por James Gosling para un proyecto compuesto por un equipo de 13 personas llamado *the Green Project* para Sun Microsystems, dicho proyecto tenía la intención de crear un control remoto para diferentes dispositivos electrónicos como la televisión y el teléfono en uno solo, el prototipo creado fue nombrado Star7 y fue ensamblado a partir de piezas de diversos dispositivos electrónicos, para este proyecto se utilizaba el lenguaje C++ , pero en un intento de crear una versión de C más segura, Gosling terminó por crear un nuevo lenguaje de programación el

cual fue llamado Oak, este nuevo lenguaje se pensó para ser independiente de la plataforma con la intención de que fuera posible ejecutarlo en diferentes dispositivos sin tener que reescribir el código generado, con el nuevo desarrollo del internet los creadores notaron el gran potencial que poseía este lenguaje para el desarrollo de páginas web más dinámicas por lo que lo renombraron Java y comenzaron a desarrollar un navegador web que sería nombrado Hot Java, finalmente se comercializó en 1995 y en 1996 se lanza java 1.0 siendo su primera implementación pública. Posteriormente Java fue adquirido por Oracle en el 2010.

B. Características

Java es un lenguaje de programación fuertemente tipado orientado a objetos debido a que permite crear clases a través de las cuales se pueden crear objetos que contendrán atributos y métodos que podrán utilizarse para manipular las acciones de dicho objeto, esto es especialmente útil ya que permite la reutilización del código para diferentes proyectos donde se necesiten estructuras de la misma naturaleza. Java también se le considera una plataforma o entorno de desarrollo, el JRE (Java Runtime Environment) es el software requerido para la ejecución de aplicaciones creadas en java, así como también existe el JDK (Java Development Kit) el cual contiene al JRE al igual que un compilador y Javadoc para la documentación.

C. Aplicaciones

Java cuenta con múltiples ediciones entre las que se encuentran:

- a. Java Standard Edition (Java SE): principalmente utilizada para desarrollo de páginas web y aplicaciones de escritorio, cuenta con bibliotecas de clases que se enfoca a interfaces gráficas, acceso a bases de datos, seguridad, entre otras.
- b. Java Micro Edition (Java ME): enfocada al desarrollo de aplicaciones para dispositivos móviles, consolas de videojuegos y televisores inteligentes.

- c. Java Enterprise Edition (Java EE): Se suele utilizar para la creación de aplicaciones con arquitectura cliente servidor, pensado para el desarrollo de aplicaciones empresariales, también funciona para crear aplicaciones web como webservices, webstocking, entre otras.

Entre las empresas que utilizan Java se encuentran Twitter, Netflix, Uber, eBay, square, Amazon entre otras. Pese a ser usado principalmente para Apps escritorio, aplicaciones móviles y páginas Web, java ha alcanzado también el campo de los videojuegos como es el caso de minecraft.

V. TYPESCRIPT

A. Historia

JavaScript nació como una solución a los problemas que presentaba JavaScript, ya que al ser un lenguaje tan sencillo de utilizar con el avance de los proyectos entre más grandes se volvían más difícil era mantenerlos y cambiar partes importantes código original, por lo cual en el 2010 Microsoft comenzó a desarrollar un nuevo lenguaje similar a JavaScript pero que extendiera sus funcionalidades, incluyendo tipos de datos, soporte de módulos, compilador, herramientas de desarrollo entre otras cosas, el objetivo era crear un lenguaje que utilizara la sintaxis de JavaScript pero que pudiera sostener proyectos de gran magnitud y que los programas desarrollados en este nuevo lenguaje fueran capaces de ser utilizados por JavaScript.

TypeScript fue finalmente publicado en octubre del 2012 tras dos años de desarrollo y desde entonces ha tenido adiciones como el soporte para genéricos y un nuevo compilador, actualmente se ha convertido en uno de los lenguajes de programación más populares en GitHub donde sus desarrolladores se mantienen pendientes de los usuarios de este lenguaje.

B. Características

TypeScript es un lenguaje de programación de código abierto desarrollado por Microsoft y que fue diseñado para ser una extensión del ya conocido Java script, al

extender la sintaxis de este cualquier programa escrito en java script funciona en TypeScript, así como a través de un compilador se puede pasar de TypeScript a JavaScript, soporta ficheros similares a los de C que permiten a diferentes programas utilizar valores ya definidos como entidades, además de que toma un enfoque orientado a objetos debido al uso de tipos estáticos, clases, y objetos.

El tipado estático hace más predecible al código, al contrario de Java Script, pese a que tenga que escribirse más código, y es debido a esta separación de los tipos de datos que los editores de texto son capaces de autocompletar las sentencias que se pueden realizar para dicho tipo de dato, ahorrando tiempo para los programadores, facilitando el desarrollo y así ayudando a detectar errores que en JavaScript no se mostrarían.

C. Aplicaciones

Entre las empresas que utilizan TypeScript hoy en día se encuentran YouTube, Facebook, Google, entre otras. Una de las principales razones por las que estas empresas deciden implementar TypeScript es porque ya cuentan con piezas de código importantes en JavaScript y la compatibilidad con este hace que puedan mudar de un lenguaje a otro sin tener que reescribir por completo sus códigos.

VI. C#

A. Historia

C# ha sido desarrollado por Microsoft con el objetivo de permitir a los desarrolladores crear una multitud de aplicaciones ejecutadas en .NET *Framework* (una tecnología que admite la compilación y ejecución de aplicaciones y servicios web XML).

Durante el desarrollo del .NET Framework, sus bibliotecas de clases fueron escritas mediante un compilador de código administrado denominado Simple Managed C (SMC). En enero de 1999, Anders Hejlsberg, creador de lenguajes como Turbo Pascal y Delphi, formó un equipo para diseñar un nuevo lenguaje, cuyo nombre inicial fue Cool (C-like Object Oriented Language). Microsoft quiso mantener dicho nombre como definitivo,

pero por razones de registro de marca tuvo que echarse para atrás.

En julio de 2000, durante la Professional Developers Conference, el lenguaje había sido renombrado a C# y todas las librerías de clases y el runtime de ASP.NET fueron portadas a este. Los creadores del lenguaje Java decían que C# era un clon de Java sin ninguna novedad ni innovación a lo que Anders Hejlsberg contestó que «C# es ‘no un clon de Java’ y es mucho más cercano a C++ en su diseño».

B. Características

C# es considerado como una evolución y necesidad de ciertas circunstancias; evolución por sus lenguajes antecesores que son C y C++ y necesidad a la hora en que la compañía tuvo problemas con la empresa creadora del lenguaje Java. Es por lo anterior que C Sharp presenta los atributos positivos de C++, Java y Visual Basic y los mejora otorgando un lenguaje fuerte y actualizado para los tiempos actuales.

C# es un lenguaje de programación moderno, orientado a objetos y con seguridad de tipos, permite a los desarrolladores crear muchos tipos de aplicaciones sólidas y seguras que se ejecutan en .NET.

Es un lenguaje de programación orientado a objetos y orientado a *componentes*. Proporciona construcciones de lenguaje para respaldar directamente estos conceptos, lo que convierte a C# en un lenguaje natural en el que crear y usar componentes de software. Desde su origen, C# ha agregado funciones para admitir nuevas cargas de trabajo y prácticas de diseño de software emergentes. En esencia, C# es un *lenguaje orientado a objetos*. Ya que el programador define los tipos y su comportamiento.

Varias características de C# ayudan a crear aplicaciones sólidas y duraderas. *La recolección de basura* recupera automáticamente la memoria ocupada por objetos no utilizados inalcanzables. *Los tipos anulables* protegen contra variables que no hacen referencia a objetos asignados. *El manejo de excepciones*

proporciona un enfoque estructurado y extensible para la detección y recuperación de errores. *Las expresiones lambda* admiten técnicas de programación funcional. *La sintaxis de Language Integrated Query (LINQ)* crea un patrón común para trabajar con datos de cualquier origen. El soporte de lenguaje para *operaciones asincrónicas* proporciona sintaxis para construir sistemas distribuidos. C# tiene un sistema de *tipo unificado*.

C. Aplicaciones

C# es un lenguaje de programación creado para diseñar aplicaciones en la plataforma .Net que, aunque no es el único lenguaje de programación que acepta .Net para realizar aplicaciones, C# si es el más recomendable y sencillo de usar.

Unity: Es un programa para crear videojuegos, muy popular últimamente por todas las posibilidades que tiene. Su fama la tiene porque su motor es multiplataforma, por lo que permite exportar los videojuegos a ordenadores, dispositivos móviles con Android o iOS, y las diferentes videoconsolas del mercado. Además, es posible programar de diversos lenguajes, dentro de los cuales esta C#, gracias a MonoDevelop.

Es un entorno de desarrollo de software para sistemas operativos Windows. Este conjunto de herramientas se utiliza para crear sitios y aplicaciones web, así como generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles.

VII. RUBY

A. Historia

Ruby es un lenguaje con un balance cuidado. Su creador, Yukihiro “Matz” Matsumoto (quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995), mezcló partes de sus lenguajes favoritos (Perl, Smalltalk, Eiffel, Ada y Lisp) para formar un nuevo lenguaje que incorporara tanto la programación funcional como la imperativa. A menudo ha manifestado que está

“tratando de hacer que Ruby sea natural, no simple”, de una forma que se asemeje a la vida real.

Desde su liberación pública en 1995, Ruby ha atraído devotos desarrolladores de todo el mundo. En el 2006, Ruby alcanzó reconocimiento masivo, formándose grupos de usuarios activos en las ciudades más importantes del mundo y llenando las capacidades de las conferencias relacionadas a este.

La Licencia Ruby (en inglés, Ruby License) es la licencia de código abierto aplicada al lenguaje de programación Ruby y también en otros proyectos. Esta licencia es aprobada por Free Software Foundation aunque todavía no ha sido aprobada como licencia de código abierto por la Open Source Initiative.

B. Características

Es un lenguaje de propósito general, es decir, con Ruby se pueden desarrollar todo tipo de aplicaciones diferentes: aplicaciones de servicio web, clientes de correo electrónico, procesamiento de datos en Backend, aplicaciones de red, etc.

Es un lenguaje interpretado, es decir, no es compilado, se necesita que el intérprete de Ruby evalúe el código y lo traduzca en lenguaje de máquina entendible por un ordenador, pero no existe un proceso previo de compilación como en C o Java.

Es dinámico y flexible. Con estrategias como la metaprogramación (escribir programas que escriben o manipulan otros programas). Incluso se puede utilizar el propio lenguaje Ruby para alterar el código en tiempo de ejecución.

Es de software libre (Open Source) y multiplataforma, se puede descargar totalmente gratis de la página oficial y ejecutarlo en diferentes sistemas operativos.

Es 100% Orientado a objetos; todo es un objeto, es decir, todo tiene una serie de propiedades (variables de instancia) y sobre el objeto podemos realizar acciones (o ejecutar métodos).

C. Aplicaciones

Muchas aplicaciones famosas utilizan el lenguaje Ruby, como los sitios web de Airbnb y SoundCloud, la red social Twitch y el sitio web de streaming de series y películas Hulu. El lenguaje Ruby puede ser muy útil, ya que este lenguaje se puede utilizar para crear varios programas para desktop. Hoy en día, es uno de los lenguajes más utilizados para el desarrollo de aplicaciones back-end en el mundo.

Ruby on rails fue creado para desarrollar aplicaciones web para startups y se inspiró totalmente en el desarrollo ágil utilizado en este tipo de empresas. Hoy en día, este framework también se usa ampliamente para desarrollar aplicaciones como aplicaciones comerciales y corporativas, sistemas de ventas y otras aplicaciones utilizadas por Internet.

Usar Ruby on rails tiene muchas ventajas, como el hecho de que también es un framework de programación open source, tiene una gran comunidad activa de usuarios y también se puede usar en múltiples plataformas. Esta herramienta tiene como objetivo facilitar el desarrollo web y se utiliza para resolver problemas en menos tiempo. También se puede aplicar para crear MVP (producto mínimo viable) y validar nuevas hipótesis comerciales para aplicaciones web.

VIII. PYTHON

A. Historia

Lanzado en los primeros años de la década de los 90s por Guido Van Rossum en Holanda, a petición del centro de investigación Centrum Wiskunde & Informatica (CWI), Python surgió como un lenguaje alternativo al lenguaje ABC para su implementación en el *proyecto Amoeba* bajo la dirección de CWI tras haber encontrado problemas en el uso del lenguaje ABC.

Guido decidió nombrar a este lenguaje de programación, Python, en honor a una serie televisiva titulada Monty Python's Flying Circus, definiéndolo como un lenguaje de programación divertido de usar. La

primera versión fue creada en 1991, sin embargo, su publicación oficial fue hasta 1994 aún bajo la dirección de CWI; el lanzamiento de Python 1.2 en 1995 fue la última versión publicada bajo este instituto. Tiempo después, tras la búsqueda de Van Rossum por un nuevo equipo, Python 1.6 salió al mercado bajo la dirección de Corporation for National Research Initiatives (CNRI) de Virginia, Estados Unidos. Posteriormente con la creación de Python Software Foundation en el año 2000 y bajo la completa dirección de desarrollo a cargo de Guido, se crea Python 2.0, versión que ya incluía la recolección de basura y soporte a Unicode entre sus funcionalidades.

Este lenguaje *open source* comenzó con una simple licencia que se ajustaba a la *General Public License* (GPL), permitiendo la modificación del código fuente a los usuarios programadores para así crear un código derivado del lenguaje sin la necesidad de ser open source. Actualmente podemos hacer uso de la tercera y más reciente versión de este lenguaje de alto nivel y multiplataforma, Python 3, siendo que finalmente se colocó como un software open source.

En la actualidad, *Python Software Foundation* es quien se encarga del desarrollo y distribución de Python bajo la dirección del mismísimo Guido, quien se encuentra trabajando en Google como desarrollador.

B. Características

Siendo un lenguaje de alto nivel interpretado y dinámico, Python ha demostrado ser uno de los lenguajes de programación más atractivos y preferidos entre desarrolladores y programadores, pues además de ser un lenguaje flexible gracias a que el diseño de algunos de sus componentes fue desarrollado en otros lenguajes de programación, proporciona un entorno simple que facilita su comprensión y funcionamiento.

Este lenguaje, considerado multi-paradigma, se clasifica como un lenguaje interpretado al poseer un intérprete que facilita la ejecución de código, utilizando también un tipado dinámico y combinaciones de referencia. Se considera multiparadigma ya que no sólo

soporta la programación estructurada y orientada a objetos, sino que también brinda la posibilidad de implementar otros paradigmas al emplear ciertas extensiones específicas.

Dentro de sus características principales, nos encontramos con la portabilidad, pues al ser un lenguaje sumamente flexible, los scripts de Python (códigos fuente) pueden migrar e implementarse en diferentes sistemas operativos sin necesidad de reescribir el código. Por mencionar algunos de los sistemas operativos con los que es compatible, tenemos a Windows, Linux, UNIX, Amigo, MacOS, etcétera.

También tenemos en la lista la antes mencionada y característica simpleza de este lenguaje, pues provee a los programadores las herramientas necesarias para codificar de forma eficiente, utilizando como ejemplo las librerías internas estándar que nos posibilitan la realización de operaciones básicas y la implementación de funciones avanzadas.

Finalmente, como otra de sus características, nos brinda una resolución dinámica para unir métodos y nombres de variables durante la ejecución del programa.

Cabe mencionar que dentro de sus funcionalidades podemos hallar un recolector de basura que se encarga del mantenimiento en memoria, y adicionalmente, debemos mencionar las tecnologías a las que Python otorga apoyo: COM, .Net y objetos.

C. Aplicaciones

Debido a que Python es un lenguaje open source, es decir, que no hay limitaciones para su uso, distribución y cambio en su código a pesar de que todos los derechos son completamente de Python Institute, sus aplicaciones son bastante diversas, yendo desde la programación de sistemas, Graphical User Interfaces (GUIs), programación para bases de datos, programación numérica y científica, procesamiento de imágenes, videojuegos (gaming), robótica, data mining, entre muchos usos más que lo colocan como uno de los lenguajes de programación favoritos a nivel mundial.

IX. PHP

A. Historia

Siendo un software de uso libre, cuya licencia está bajo la PHP License, PHP fue lanzado en 1994, cuyos inicios se concentraban en formar parte de un paquete de programas pertenecientes y creados por Rasmus Lerdorf, siendo conocido como Personal Home Page Tools, destinado principalmente al desarrollo de la página web que en ese entonces Rasmus estaba creando. Para 1997 se lanzó una nueva versión de este lenguaje recibiendo una modificación en su nombre (PHP/FI), y presentando una nueva herramienta añadida: el interpretador de comandos SQL, Forms Interpreter.

Tras el desarrollo de Zeev Suraski para la creación de PHP3, este lenguaje fue capaz de soportar la orientación a objetos, dándole el poder de alcanzar algunos paquetes que brindaron a los desarrolladores la posibilidad de implementar métodos y propiedades tras incluir la herencia. Las versiones posteriores, diseñadas y desarrolladas por Zeev y Andi Gutmans como PHP 4 y 5 trajeron consigo nuevos recursos orientados a objetos, sin embargo, para el instante en el que se creó PHP 4 no se contaba con apuntadores o handlers, recursos necesarios para mantener una programación orientada a objetos que fuese estable, funciones que fueron incluidas más tarde con el lanzamiento de PHP 5.

B. Características

El lenguaje PHP, por sus siglas en inglés Hypertext Preprocessor, y antes mejor conocido como Personal Home Page, es un tipo de lenguaje interpretado y de modularización (1) que entra en la categoría de software libre u open source, cuyo dominio está definido como específico para el desarrollo web.

Al inicio de su lanzamiento su implementación se concentraba principalmente en el desarrollo de aplicaciones web capaces de interactuar en el lado del servidor, generando contenido dinámico en la World Wide Web (WWW). Tras una continua evolución en sus

distintas versiones, este lenguaje comenzó a ofrecer ciertas funcionalidades en línea de comandos, sumando a ello usos adicionales que ayudaron a crear lo que hoy conocemos como bases de datos en compañía de MySQL, teniendo como propósito principal la implementación de soluciones web que sean eficientes, simples y de alta velocidad.

Entre sus principales características podemos mencionar la portabilidad, el tipado dinámico, su estructuración orientada a objetos, velocidad y robustez, haciendo notorio el favoritismo que ha recibido para la creación de bases de datos en el desarrollo de aplicaciones web, pues otorga aquellas facilidades que todo desarrollador de internet necesita para cumplir con las expectativas y requerimientos solicitados por el cliente. Es importante mencionar que, dentro de un lenguaje, además de las características mencionadas con anterioridad, la sintaxis influye en gran parte para su elección al programar, siendo en este caso, que PHP cuenta con una sintaxis parecida a los lenguajes C, C++ y Perl, lo cual brinda un sencillo entendimiento de su funcionalidad si se cuenta con los conocimientos básicos de programación.

Tomando en cuenta que PHP trabaja con servidores web, este puede estar incrustado con HTML dependiendo del tipo de servidor, y permite la creación de módulos que funcionan como repositorios de extensiones PECL (PHP Extension Community Library) que pueden ser incluidos en futuras versiones del lenguaje.

Gracias a que es un software gratuito y de código abierto, al crear productos relacionados a PHP, para incluir dicho nombramiento dentro de los mismos creados fuera del control PHP License, se debe solicitar un permiso especial para poder incluir “PHP” en el nombre del producto.

C. Aplicaciones

Sabiendo que PHP tiene como una de sus características principales la portabilidad, instalar y

(1) Modularización: Conjunto de instrucciones que cumplen con una o más tareas específicas bien definidas.

utilizarlo en casi cualquier sistema operativo es posible, siendo compatible con Windows, Linux, MacOS, FreeBSD, Novell NetWare, RISC Los, AIX, IRIX y Solaris. Además, les otorga soporte a protocolos como IMAP, SNMP, NNTP, POP3, HTTP, LDAP, XML-RPC y SOAP, permitiendo una interacción entre estos gracias a la apertura de sockets (2).

Ya que es un lenguaje cuya programación está enfocada a la programación de scripts para servidores, puede ser empleado en una gran gama de posibilidades en las que puede brindar soporte en la recopilación de datos, generación de páginas con contenidos dinámicos, y, además, siendo posible de utilizar en la programación orientada a objetos, otorgando apoyo a varias bases de datos como Oracle, Sybase, PostgreSQL, InterBase, MySQL, SQLite, MSSQL, Firebird, entre otros.

X. CONCLUSIONES INDIVIDUALES

A. *Camacho Ignacio Violeta*

Gracias a esta investigación fui capaz de comprender más a fondo el origen de distintos lenguajes de programación orientados a objetos y cómo es que estos se han desarrollado tratando de corregir los errores de otros o ampliando su funcionalidad y han evolucionado hasta convertirse en lo que son actualmente, también me permitió observar como la programación orientada a objetos se ha abierto paso a través de los años extendiéndose cada vez más a través de estos lenguajes en la mayoría de cosas que utilizamos en nuestro día a día, esto debido a las empresas que implementan dichos lenguajes en sus programas gracias a su versatilidad e infinidad de usos, creo firmemente que esta investigación cumplió con su propósito ya que para nosotros como estudiantes de ingeniería en computación es indispensable conocer la historia, el funcionamiento y la implementación de estos lenguajes para el futuro desarrollo de proyectos propios y en equipos para nuestra vida profesional.

B. *Luna Alberto Darwin Leonel*

La Programación Orientada a Objetos ha permitido un gran avance al desarrollo tecnológico actual, ya que presentan características que ayudan a la creación de soluciones más complejas ante el crecimiento de la necesidad de mejor software, así como la búsqueda de un mejor rendimiento a pesar de dicha complejidad.

Razón por la cual es importante poder entender su funcionamiento y ser capaces de crear nuevas soluciones a los problemas que el mundo actual necesita resolver, y, por ende, ampliar los conocimientos en estos lenguajes para elegir el que más se acomode personalmente, debido a que se necesita también pensar en la practicidad para los desarrolladores, tanto para crear los programas como darles mantenimiento.

C. *Vences Santillán Carlos Eduardo*

Llegué a la conclusión de que en esta investigación pude aprender mucho más sobre las características de los lenguajes de programación orientados a objetos, estas son importantes ya que nos muestran más a fondo el paradigma y cómo va a funcionar el lenguaje al momento de codificar.

El tema principal de esta investigación fue conocer algunos de los diversos lenguajes de programación orientados a objetos; considero que este objetivo se cumplió de manera correcta, ya que para cada uno pudimos recolectar y analizar información que nos lleva a su contexto histórico y a sus raíces. Lo que me pareció más interesante y enriquecedor, fue el hecho de que se nos explica cómo fueron surgiendo a lo largo de la historia por diversos desarrolladores y empresas. También es importante mencionar la evolución que han tenido con el paso del tiempo, ya que la programación orientada a objetos fue cambiando década tras década, por lo que algunos lenguajes se adaptaron correctamente y siguen siendo los mejores hoy en día.

Como resumen de la conclusión, creo que este tema es fundamental para conocer la historia y el proceso del

(2) Sockets: Mecanismo creado para la comunicación entre un programa del servidor y un programa del cliente en una red.

nuevo paradigma que estaremos usando a lo largo del curso.

D. Villegas Flores Nadia

La programación ha demostrado tener un gran impacto para el desarrollo y creación de nuevas tecnologías que hoy en día se han vuelto indispensables en la vida cotidiana, pues es gracias a esta práctica que podemos facilitar ciertas tareas para tener un estilo de vida mucho más cómodo.

A partir de la necesidad por obtener resultados más eficientes en el desempeño de software, surgen nuevas mejoras que facilitan la labor de crear algo que sea de utilidad y demuestre ser funcional en cada caso posible. Es así como el paradigma orientado a objetos surge, trayendo consigo mejoras que permiten la reutilización del código y el encapsulamiento. Además, proporciona las facilidades que todo programador o desarrollador necesita al momento de crear código, pues permite realizar ciertas funciones de tal manera que mantiene una estructura estable y gracias a la abstracción, es que es posible construir sistemas complejos de manera mucho más sencilla.

Sin duda alguna, los lenguajes de programación orientados a objetos son totalmente necesarios para continuar con esta evolución tecnológica que día a día nos persigue, y es importante conocer las características de diversos lenguajes para que así, nosotros como desarrolladores, podamos sentirnos completamente cómodos con el entorno de desarrollo y auxiliados dependiendo de lo que necesitemos realizar.

XI. REFERENCIAS

- [1] "IBM documentation". IBM - Deutschland | IBM. <https://www.ibm.com/docs/es/spss-modeler/saas?topic=language-object-oriented-programming> (accedido el 23 de septiembre de 2022).
- [2] J. A. P. C. "Historia y origen de la POO". HISOTRIA Y ORIGEN DE LA POO. <https://origendelapoo.blogspot.com/2020/08/blog->

[creado-por-estudiantes-de.html](#) (accedido el 23 de septiembre de 2022).

- [3] "Características de la POO". Portal Académico del CCH. <https://portalacademico.cch.unam.mx/cibernetica1/algoritmos-y-codificacion/caracteristicas-POO> (accedido el 23 de septiembre de 2022).
- [4] "What is TypeScript?". Microsoft Learn: Build skills that open doors in your career. <https://learn.microsoft.com/en-us/shows/web-wednesday/what-is-typescript> (accedido el 23 de septiembre de 2022).
- [5] "IBM documentation". IBM - Deutschland | IBM. <https://www.ibm.com/docs/es/spss-modeler/saas?topic=language-object-oriented-programming> (accedido el 23 de septiembre de 2022).
- [6] "[C#] un poquito de historia de C#." System.OutOfMemoryException. <https://sparraguerra.wordpress.com/2015/05/18/un-poquito-de-historia-de-c/> (accedido el 23 de septiembre de 2022).
- [7] "The history of C# - C# Guide". Microsoft Learn: Build skills that open doors in your career. <https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history> (accedido el 23 de septiembre de 2022).
- [8] J. García. "Qué es Ruby y sus características". OpenWebinars.net. <https://openwebinars.net/blog/que-es-ruby/> (accedido el 23 de septiembre de 2022).
- [9] "Ruby: ¿para qué sirve este lenguaje de programación?". HostGator. <https://www.hostgator.mx/blog/lenguaje-de-programacion-ruby/> (accedido el 22 de septiembre de 2022).
- [10] M. Nosrati. "Python: An appropriate language for real world programming". CiteSeerX. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.678.2551&rep=rep1&type=pdf> (accedido el 17 de septiembre de 2022).
- [11] Challenger, Y. Díaz y R. A. Becerra. "El lenguaje de programación Python". Sistema de Información

- Científica Redalyc, Red de Revistas Científicas.
<https://www.redalyc.org/pdf/1815/181531232001.pdf> (accedido el 17 de septiembre de 2022).
- [12] C. A. Rojas. "Python 3 al descubierto".
https://d1wqtxts1xzle7.cloudfront.net/54161186/cap1.PHYTON-with-cover-page-v2.pdf?Expires=1663366752&Signature=GZOED-v5N~LtBa7-5c9c11FhwtU5N7jfkxfR~tyOnQ84gy74tKieVmdIb5ejs2elWpOkQoTj8m5ti9JJ1auJtX0WwLTX0zfXkT7B03Sp9Q5i2xNhRUbQgoilOfpXe4c7WIt43UshQSdzTl8SYXKQ3M~IEywTOiP7bypU6lBMAD8NkvnO9OEitA7yPq79URLiV1UdRilGb0u0PMIZVbzmu7RUias~KJj754WychYA2TV9jwOfV-Vb4fEPVcZFTwVxjtPXBjXqD5k3N~W5Nk1T3KAZWHzTo5XfjX4kSQylyjCvOYbDBQ5DG0QfYiqbMgvQl6iWrzbbTKTOy8-7jxacw__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA (accedido el 17 de septiembre de 2022).
- [13] M. A. Arias. "Introducción a PHP".
https://books.google.com.mx/books?hl=es&lr=&id=oqjQCgAAQBAJ&oi=fnd&pg=PT6&dq=qué+es+PHP&ots=x3mSArDbZ7&sig=_wugNzzCY3UKQ5XAytjW7AsPkqE&redir_esc=y#v=onepage&q&f=false (accedido el 17 de septiembre de 2022).
- [14] "PHP: ¿Qué puede hacer PHP? - Manual". PHP: Hypertext Preprocessor.
<https://www.php.net/manual/es/intro-whatcando.php> (accedido el 17 de septiembre de 2022).
- [15] S. Sanderson, Java for beginners, A Simple Start To Java Programming (Written By A Software Engineer), Globalized Healing, 2016.
- [16] C. Harrison. (2022, Abr 19). What is TypeScript?. [Internet]. Disponible en <https://learn.microsoft.com/en-us/shows/web-wednesday/what-is-typescript>
- [17] B. Benz, R. Preddy. (2022, Abr 10). What is Java? | Java for Beginners. [Internet]. Disponible en <https://learn.microsoft.com/en-us/shows/java-for-beginners/what-is-java>
- [18] Ahejlsberg. (2021, Mar 23). TypeScript Language Specification. [Internet]. Disponible en <https://github.com/Microsoft/TypeScript/blob/730f18955dc17068be33691f0fb0e0285ebbf9f5/doc/spec.md>
- [19] A. Haugen. (2022, May 22). What is an OOL (Object-Oriented Language), with Examples?. [Internet]. Disponible en <https://history-computer.com/ool-object-oriented-language-guide/>