

UNIVERSIDAD VERACRUZANA

Facultad de Contaduría y Administración.

# Campus Ixtaczoquitlan.



Título del proyecto

**PROTOCOLO DE INVESTIGACIÓN**

PARA ACREDITAR LA EXPERIENCIA EDUCATIVA DE EXPERIENCIA RECEPCIONAL DE LA:

**LICENCIATURA EN INGENIERÍA DE SOFTWARE**

PRESENTAN:

**Fernández Fernández Francisco de Jesús**

INSTRUCTOR DEL TALLER DE METODOLOGÍA:

**Oscar**

IXTACZOQUITLAN, VER. MES DE 202X

TÍTULO DE LA INVESTIGACIÓN

Desarrollo de un compilador que tome un subconjunto del lenguaje Java, para simplificar el aprendizaje de sus conceptos básicos.

# PLANTEAMIENTO DEL PROBLEMA

Los mecanismos de enseñanza de programación se ven afectados por las variantes en los lenguajes de programación, esto debido a que los mismos contienen elementos que son complejos o que son muy avanzados para los estudiantes que desean aprender a programar, por tanto, cuando un estudiante analiza y aprende un lenguaje requiere entender conceptos que se encuentran fuera del alcance de la experiencia educativa, esto provoca desesperación y frustración debido a que el estudiante es forzado a entender ideas muy avanzadas para su formación.

Uno de los problemas recurrentes es que algunos lenguajes de programación contienen elementos que resultan ser excesivamente complejos o avanzados para los estudiantes que están dando sus primeros pasos en la programación. Estos elementos incluyen sintaxis complicadas o características específicas que requieren un conocimiento previo sólido. Cuando un estudiante se enfrenta a estos elementos, es común que llegue a sentirse abrumado y desanimado, esto debido a la cantidad excesiva de información para su nivel actual

En este sentido, el proceso de aprender un lenguaje de programación no se limita únicamente a comprender su sintaxis y estructura básica, sino que a menudo implica la necesidad de abordar conceptos más avanzados que están relacionados con la programación en sí. Esto deben incluir temas como algoritmos complejos, estructuras de datos avanzadas o paradigmas de programación que requieren un nivel más profundo de comprensión.

# OBJETIVO

Objetivo general

desarrollo de una herramienta diseñada para el aprendizaje de programación básica en Java, enfocándose en conceptos básicos

## Objetivos específicos:

* Analizar la sintaxis de java
* Determinar el subconjunto del lenguaje necesario para aprender programación básica.
* Seleccionar la herramienta para la construcción del meta-compilador.
* Diseñar el subconjunto con la herramienta seleccionada.
* Implementar la funcionalidad del subconjunto de la herramienta.
* Realizar pruebas en el subconjunto.

## Variables dependientes:

* Herramienta para el aprendizaje de la programacion básica.

## Variable independiente:

* Aprendizaje de los fundamentos de programación.

# JUSTIFICACIÓN DE LA INVESTIGACIÓN

El rezago académico de los estudiantes en cuanto al aprendizaje de la programación es muy notable en lo que a la práctica se refiere. La aplicación de los conocimientos es ineficiente y restringe de manera notable el avance académico esperado. Estas situaciones son derivadas de las malas prácticas de enseñanza y aprendizaje. Como bien sabemos, la programación es un entorno lleno de lenguajes y herramientas diseñadas para la elaboración del software, cada una con ventajas y desventajas diseñadas para diferentes tareas. No obstante, cada una de estas herramientas contiene una curva de aprendizaje. Esta curva de aprendizaje resulta muy elevada para estudiantes que apenas inician en esta disciplina debido a que comúnmente las primeras etapas de aprendizaje son abordadas mediante la implementación del seudo-código, que si bien es una herramienta adecuada para la enseñanza de los principios de programación, carece de las características necesarias para abordar los diferentes métodos y posibles errores de un lenguaje convencional. La transición de un seudo-código a un lenguaje tradicional conlleva una curva de aprendizaje mayor debido a la diferencia de sintaxis y estructura de un proyecto. Por lo tanto, enseñar las bases de la programación convencional en un lenguaje que contenga una herramienta diseñada para el aprendizaje básico, que conserve la sintaxis y estructura de un lenguaje real, es más eficaz que un modelo de seudo-código.

# MARCO TEÓRICO

## 1. Identificación del Problema y Objetivos de Investigación

Utilizar una versión simplificada del lenguaje de programación Java facilita la comprensión de los conceptos básicos de este idioma.

## 1.2. objetivos específicos

* Identificar los factores que componen la base de un aprendizaje solido que permita comprender elementos básicos y características únicas del lenguaje de programación java
* Abordar la curva de aprendizaje inicial de forma dinámica, con la finalidad de formalizar las bases del pensamiento lógico requerido para comprensión de los elementos informáticos
* Identificar los factores de riesgo que generar un deficis en la comprensión de los diferentes paradigmas de programación y los diferentes métodos existentes
* Hacer un énfasis en la resolución de problemas por medio de la comprensión lógica y matemática empleando lenguajes de programación
* Seleccionar el subconjunto de Java
* Seleccionar una herramienta para la generación de gramáticas
* Desarrollar la gramática en la herramienta
* Implementar la semántica de la herramienta

## 2.1. literatura relacionada

Los entornos de programación enfocados a la educación son muy útiles y es común encontrar estudios que traten de descubrir cuál es el lenguaje de programación más adecuado para el aprendizaje, así como también se enfocan en identificar una estrategia para la enseñanza adecuada de los fundamentos de programación (López García, 2023).

El correcto aprendizaje de la programación es fundamental para los tiempos actuales, y es una parte importante a considerar en el currículum para aplicar a cualquier puesto relacionado con el campo. No obstante, los fundamentos de programación se pueden volver complicados para algunos estudiantes en aspectos como aplicar conceptos básicos de la programación. Esto puede ser causado por diferentes razones, entre las cuales se destacan la falta de experiencia, deficiencia de aprendizaje o malas prácticas de enseñanza (Sánchez, 2012).

El problema más recurrente entre las carreras relacionadas al ámbito de la programación es el aprendizaje de algoritmos y conceptos básicos de los fundamentos de programación. Esto se debe a que la mayoría de estudiantes que cursan las materias introductorias de programación llegan con un déficit en el entendimiento de esta rama y de la informática en general. Esta situación puede generar un índice de abandono mayor al 30% en la mayoría de los casos. Por esta razón, se plantea utilizar herramientas más adecuadas para un aprendizaje básico del ámbito de la programación, como la herramienta PSeInt. No obstante, el uso de esta herramienta es meramente de aprendizaje y dista mucho del ámbito de desarrollo de un lenguaje de programación convencional (Anderson Huerta & González-Bañales, ).

La complejidad que trae consigo la curva de aprendizaje de cualquier lenguaje de programación produce la necesidad de utilizar técnicas y herramientas adecuadas y efectivas. Para ello, debemos considerar los aspectos y requerimientos necesarios para alcanzar el éxito de una herramienta de este estilo. Como algunos estudios demuestran, una de las mejores formas de aprender a programar es mediante la implementación de algoritmos informáticos como recursos esquemáticos para plasmar un modelo de resolución del problema, como lo podrían ser los diagramas de flujo. Si bien estos conceptos nos ayudan a entender el funcionamiento teórico de los procesos que se ejecutan, distan mucho de una solución verídica. Esto se debe a los factores de errores lógicos y de implementación tecnológica que presentan los retos informáticos a los que los estudiantes pueden enfrentarse en la curva de aprendizaje de un lenguaje de programación (Moroni & Señas, 2005).

La dificultad se manifiesta en la experiencia de los estudiantes al enfrentarse a desafíos en la depuración, navegación, pruebas y documentación durante el proceso de aprendizaje de programación. Este obstáculo se ve agravado por la constante evolución de tecnologías, paradigmas y lenguajes, lo que dificulta aún más el proceso de comprensión. Los conceptos fundamentales, como variables, parámetros, funciones y estructuras de control, a menudo se perciben como temas difíciles de abordar. La solución propuesta implica un cambio en las estrategias de enseñanza, adaptando el enfoque para abordar las dificultades específicas, logros y errores experimentados por los estudiantes. De esta manera, el aprendizaje se fundamenta en la experiencia práctica de los alumnos, permitiendo una comprensión más efectiva de algoritmos y conceptos básicos de programación (Guitierrez et al., 2022)

# Selección de Teorías y Conceptos Relevantes

La herramienta para el aprendizaje de programación propone una alternativa a la enseñanza convencional de los fundamentos de la programación. En lugar de centrarse en algoritmos y pseudocódigos, aboga por el empleo del lenguaje Java como pilar fundamental del conocimiento. Esta perspectiva busca proporcionar un entorno real donde los fundamentos teóricos se aplican directamente en la práctica. La herramienta ofrece un contexto lógico y auténtico, permitiendo a no solo comprender los conceptos abstractos, sino también aplicarlos de manera activa y significativa en situaciones prácticas. Este enfoque tiene como objetivo proporcionar una experiencia de aprendizaje más inmersiva y efectiva, donde los principios teóricos se integran de manera fluida en la resolución de problemas del mundo real utilizando el lenguaje de programación Java.

Se anticipa que la herramienta de aprendizaje generará un impacto positivo en la enseñanza de los fundamentos básicos de programación. El enfoque se centra en la comprensión teórica a través de la práctica, inmersos en un entorno real que expone errores lógicos, de sintaxis y técnicos. El objetivo principal es cultivar la tolerancia a la frustración, al mismo tiempo que se inicia a los estudiantes en la resolución de problemas informáticos y lógicos. Esta integración se considera esencial para los fundamentos de la programación, dado su papel significativo y prevalente en el mundo del desarrollo. La herramienta aspira a proporcionar una experiencia de aprendizaje integral que no solo fortalezca la comprensión teórica, sino que también fomente habilidades prácticas cruciales en el contexto del desarrollo informático.(Guerrero, Santiago Guamán y César Caiza, 2015).

El aprendizaje de los fundamentos de programación es esencial, ya que estos conocimientos no solo son necesarios para comprender nuevas tecnologías, sino que también constituyen la base de un conocimiento más profundo y especializado. Este enfoque se orienta hacia una serie de herramientas y conocimientos teóricos fundamentales que son esenciales para la comprensión lógica y técnica de las problemáticas informáticas, proporcionando una base sólida para abordar de manera efectiva los retos presentes y futuros en el ámbito de la informática. La adquisición de estos fundamentos no solo impulsa la comprensión conceptual, sino que también sienta las bases necesarias para la resolución de problemas y el desarrollo de soluciones informáticas efectivas.

Se anticipa que el uso de la herramienta educativa resultará en un significativo aumento en el nivel de comprensión y competencia en los fundamentos de programación. Se espera que esta mejora se traduzca en una mayor capacidad para resolver problemas informáticos y lógicos, así como en una mayor tolerancia a la frustración. Además, se prevé que los estudiantes inicien su camino hacia la resolución de problemas informáticos y lógicos como parte integral de los fundamentos de programación (Pulido, 2015).

Los mecanismos de enseñanza de programación se ven afectados por las variantes en los lenguajes de programación, ya que estos contienen elementos que son complejos o demasiado avanzados para los estudiantes que desean aprender a programar. Por tanto, cuando un estudiante analiza y aprende un lenguaje, necesita comprender conceptos que se encuentran fuera del alcance de la experiencia educativa, lo que provoca desesperación y frustración. Esto se debe a que el estudiante se ve obligado a entender ideas muy avanzadas para su formación. Estas variantes están estrechamente relacionadas con la formación que recibe el estudiante, así como también con los métodos de enseñanza y la disposición para aprender del estudiante. Como bien sabemos, la curva de aprendizaje para el entendimiento de un lenguaje de programación es muy grande, y por consiguiente, se requiere el entendimiento de múltiples conceptos e ideas matemáticas, lógicas e informáticas para comprender su funcionamiento (Insuasti, 2016).

A pesar de la complejidad que engloba el entendimiento de un lenguaje de programación, no es este el motivo por el cual la mayoría de estudiantes decide abandonar sus estudios o proyectos. El problema se suscita durante el entendimiento y aprendizaje de los fundamentos más básicos de la programación. Durante este periodo, el estudiante tratará de relacionar los conocimientos con saberes empíricos relacionados con otras experiencias educativas. Es por este motivo que el aprendizaje inicial con diagramas tiende a generar un entendimiento más completo en cuanto a los procesos lógicos que se llevan a cabo durante la ejecución de alguna solución informática.

No obstante, esta solución gráfica no puede enfatizar en los errores lógicos y técnicos que engloban una solución informática real. Si bien el pseudocódigo es una herramienta adecuada para la comprensión de los fundamentos más básicos de la programación, no puede ahondar en cuanto a la implementación y uso adecuado de estos conceptos debido a las limitantes que el propio pseudocódigo contiene. La aplicación práctica de estos conceptos en un entorno tangible facilita la comprensión, mientras que el uso de un lenguaje de programación real proporciona un contexto auténtico que aumenta la relevancia y la motivación del estudiante (Perazo, 2013).

GRAMATICAS BNF

Estas gramáticas, también denominadas gramáticas de tipo 2 o gramáticas independientes del contexto, son responsables de la generación de lenguajes libres o independientes del contexto. Estos últimos son aquellos que pueden ser identificados por un autómata de pila, ya sea determinista o no determinista.

Frecuentemente, las gramáticas independientes del contexto se expresan utilizando una notación llamada BNF (Forma de Backus-Naur). BNF es la metodología más común para definir la sintaxis de los lenguajes de programación. En esta notación, se deben seguir ciertas convenciones:

* Los elementos no terminales se indican entre paréntesis angulares < >.
* Los elementos terminales se representan mediante cadenas de caracteres sin paréntesis angulares.
* Cada regla debe tener un único elemento no terminal en el lado izquierdo, dado que se trata de una gramática libre del contexto. [referencia apa7 2link ]

Árbol de derivación

Un árbol de derivación permite mostrar gráficamente cómo se puede derivar cualquier cadena de un lenguaje a partir del símbolo distinguido de una gramática que genera ese lenguaje. Un árbol es un conjunto de puntos, llamados nodos, unidos por líneas, llamadas arcos. Un arco conecta dos nodos distintos. Para ser un árbol un conjunto de nodos y arcos debe satisfacer ciertas propiedades:

- hay un único nodo distinguido, llamado raíz (se dibuja en la parte superior) que no tiene arcos incidentes.

- todo nodo c excepto el nodo raíz está conectado con un arco a otro nodo k, llamado el padre de c (c es el hijo de k). El padre de un nodo, se dibuja por encima del nodo.

- todos los nodos están conectados al nodo raíz mediante un único camino.

- los nodos que no tienen hijos se denominan hojas, el resto de los nodos se denominan nodos interiores.

El árbol de derivación tiene las siguientes propiedades:

- el nodo raíz está rotulado con el símbolo distinguido de la gramática;

- cada hoja corresponde a un símbolo terminal o un símbolo no terminal;

- cada nodo interior corresponde a un símbolo no terminal.

GRAMATICAS LIBRES DEL CONTEXTO. (2007).

6: Definición y Explicación de Conceptos Clave

* Herramienta: Un instrumento o aplicación diseñada para realizar tareas específicas relacionadas con la programación.
* Lenguaje: Un conjunto de reglas y símbolos utilizados para comunicarse con un ordenador y escribir programas.
* Programación: El acto de escribir, diseñar y mantener el código fuente de un programa de computadora.
* Gramática: En el contexto de la programación, un conjunto de reglas que define la estructura y la composición correcta de las instrucciones en un lenguaje de programación.
* Fundamentos de Programación: Los principios básicos y conceptos esenciales que subyacen en la creación y comprensión de programas informáticos.
* Mecanismos de Enseñanza: Métodos o enfoques utilizados para facilitar la comprensión y el aprendizaje de los conceptos de programación, que pueden incluir tutoriales interactivos, ejemplos prácticos y proyectos.
* Pseudocódigo: Una representación intermedia entre el lenguaje natural y el código fuente de un programa, utilizada para planificar y describir algoritmos antes de la implementación real.

# HIPÓTESIS

Utilizar una herramienta diseñada para comprender y asimilar los fundamentos de la programación en un lenguaje real puede contribuir a suavizar la curva de aprendizaje

## Pregunta de investigacion:

¿Cómo influye una herramienta de aprendizaje centrada en un entorno de desarrollo local en la mejora del rendimiento académico de los estudiantes universitarios?

## Solucion tentativa:

Emplear una herramienta específicamente diseñada para comprender y asimilar los conceptos fundamentales de la programación en un lenguaje de programación real puede ser una estrategia efectiva para facilitar el proceso de aprendizaje.

# ASPECTOS METODOLÓGICOS DE LA INVESTIGACIÓN

* 1. Tipo de Estudio

La elección de emplear un estudio observacional se fundamenta en la necesidad de capturar y comprender los fenómenos o comportamientos en su entorno natural, sin alterar las condiciones naturales. Este enfoque se considera apropiado cuando la intervención directa del investigador podría distorsionar los resultados o cuando la manipulación de variables resulta impracticable o éticamente cuestionable. Al optar por este tipo de estudio, se busca obtener una visión detallada y genuina de los patrones de comportamiento, las interacciones o las características específicas de la población bajo estudio. En el ejemplo concreto de observar el comportamiento de niños en un parque de juegos, la elección de un enfoque observacional garantiza una comprensión más auténtica de sus acciones, sin influencias externas, lo que contribuye a la validez y la fidelidad de los resultados obtenidos.

## Métodos de investigación

Estudio Observacional:

Un estudio observacional se caracteriza por la observación y registro sistemático de eventos, comportamientos o fenómenos tal como ocurren en su entorno natural, sin intervención activa por parte del investigador.

* 1. Fuentes y Técnicas para la recopilación de información

Encuestas

1. ÍNDICE TENTATIVO

[PLANTEAMIENTO DEL PROBLEMA 2](#_Toc2)

[OBJETIVO 2](#_Toc3)

[Objetivos específicos: 3](#_Toc4)

[Variables dependientes: 3](#_Toc5)

[Variable independiente: 3](#_Toc6)

[JUSTIFICACIÓN DE LA INVESTIGACIÓN 3](#_Toc7)

[MARCO TEÓRICO 4](#_Toc8)

[1. Identificación del Problema y Objetivos de Investigación 4](#_Toc9)

[1.2. objetivos específicos 4](#_Toc10)

[2.1. literatura relacionada 5](#_Toc11)

[Selección de Teorías y Conceptos Relevantes 7](#_Toc12)

[HIPÓTESIS 11](#_Toc13)

[Pregunta de investigacion: 11](#_Toc14)

[Solucion tentativa: 11](#_Toc15)

[ASPECTOS METODOLÓGICOS DE LA INVESTIGACIÓN 11](#_Toc16)

[Métodos de investigación 12](#_Toc17)

1. FUENTES BIBLIOGRAFICAS

Muñoz, R., M., & Barría, M., B. (2012). Determinando las diicultades en el aprendizaje de la primera asignatura de programación en estudiantes de ingeniería civil informática. Congreso Internacional de Informática Educativa, TISE. <https://www.tise.cl/volumen8/TISE2012/17.pdf>

Guitierrez, L., E,G, Guerrero, C., A,G, & Lopez Ospina, H., A,L,O. (2022). Ranking of problems and solutions in the teaching and learning of object-oriented programming. Education and Information Technologies, Vol.:(0123456789).

Tecnologías de la Información en Educación: Sistematización de experiencias docentes (Primera Edición: junio de 2020). (2020). Red Durango de Investigadores Educativos (Redie). https://www.researchgate.net/profile/Dora-Gonzalez-Banales/publication/342243248\_Tecnologias\_de\_informacion\_en\_educacion\_Sistematizacion\_de\_experiencias\_docentes/links/5eea3e96a6fdcc73be84b5aa/Tecnologias-de-informacion-en-educacion-Sistematizacion-de-experiencias-docentes.pdf#page=99

López García, A., & Urquiza-Fuentes, J. (2022). Experiencias de uso del Pseudocódigo y Java en la enseñanza de programación en Ciclos Formativos y Bachillerato. Actas de las Jenui, 7, Páginas: 379-382. <https://aenui.org/actas/pdf/JENUI_2022_050.pdf>

Moroni, M., N., & Señas, P. (2005). Estrategias para la enseñanza de la programación. Repositorio Institucional De La UNLP. <http://sedici.unlp.edu.ar/handle/10915/18901>

GRAMATICAS LIBRES DEL CONTEXTO. (2007). CIENCIAS DE LA COMPUTACION I. <https://users.exa.unicen.edu.ar/catedras/ccomp1/ApunteGLC.pdf>

Insuasti, J. (2016). Problemas de enseñanza y aprendizaje de los fundamentos de programación. Revista Educación Y Desarrollo Social, 10(2), 234–246. <https://doi.org/10.18359/reds.1966>

Perazo, C. (2013). El 80% de los estudiantes de carreras informáticas abandona sus estudios. www.lanacion.com.ar. Recuperado 24 de noviembre de 2023, de <https://www.lanacion.com.ar/tecnologia/el-80-de-los-estudiantes-de-carreras-informaticas-abandonan-sus-estudios-nid1632045/>

Pulido, V. (2015). El uso de herramientas de programación gráfica en alumnos de CFGM en sistemas microinformáticos y redes [Tesis de Maestria]. Universidad Internacional de La Rioja.

Guerrero, M., Santiago Guamán, D., & César Caiza, J. (2015). Revisión de Herramientas de Apoyo en el Proceso de Enseñanza-Aprendizaje de Programación. Revista Politecnica, 35(1). <https://revistapolitecnica.epn.edu.ec/ojs2/index.php/revista_politecnica2/article/view/430>

1. CRONOGRAMA

Aquí se coloca un cronograma tentativo con las actividades a realizar, considerar la división por semanas o meses. Considerar espacios para entrega al asesor para su correcta revisión.

|  |
| --- |
| **ACTIVIDADES A REALIZAR:**  **Investigacion,**  **Redaccion,**  **Estudios para reespaldar la investigacion,**  **Comparar Resultados,**  **Fabricacion de prototipo** |
| **SEMANAS PROGRAMADAS** | | | | | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| Investigación | . | . |  | . |  |  |  |  |  |  |  |
| Redacción |  |  |  |  | . | . |  |  |  |  |  |
| Estudios para respaldar la investigación | . | . |  |  |  |  |  | . | . |  |  |
| Comparar Resultados |  |  |  |  |  |  |  |  |  | . |  |
| Fabricación de prototipos |  |  |  |  |  | . | . | . | . | . | . |

1. PRESUPUESTO

Desglose de cuánto costará el desarrollo de este proyecto.

1. DATOS PERSONALES

Información personal de quienes desarrollarán el proyecto.

Nombre: Francisco de jesus fernandez fernandez

Domicilio: Calle sinaloa #31 col. Ricardo flores magon

Teléfono: 272 229 7752

E-mail: fernandezfrancisco261@gmail.com