Facultad de Ciencias Exactas, Físico-Químicas y Naturales
Lic. En Ciencias de la Computación
Asignatura: Metodología de la Investigación (Código 1960)

2025

Integrantes del grupo: Nombre y Apellido y DNI: Buchieri Giovanni - 44295111

Unidad Nº 1: Inicio del proceso investigativo

- Idea de investigación, planteamiento del problema, objetivos, preguntas de investigación y justificación del estudio.

Se recomienda la lectura del texto de Hernández Sampieri, R.; Fernández Collado, C. y Baptista Lucio, M. (2017) Metodología de la Investigación. Mc Graw-Hill. México. Documento que podrás descargar en:
https://www.uca.ac.cr/wp-content/uploads/2017/10/Investigacion.pdf

1- Realiza un listado de temáticas que le interesa potencialmente investigar ligados a la temática nombrada

- Pointer Analysis
- Inteligencia Artificial
- Ciberseguridad
- Sistemas Embebido
- Análisis Estático de Código
- Optimización de Compiladores

2- Selecciona un tema de los que ha enumerado en el apartado anterior, enúncialo:
   - **Pointer Analysis**

3- Justifica según factores enumerados a continuación, cuáles considera importantes en la selección de su tema de investigación

Factores	Descripción	Justificación
Conveniencia	Utilidad presente o futura, posibilidad de acceso a fuentes, experiencia previa	Este tema es conveniente por la utilidad en la computación, especialmente en el análisis estático de programas, optimización de compiladores y detección de errores de seguridad. Además gracias a la carrera,

		tengo experiencia previa en programación de bajo nivel y lenguajes como C/C++, donde el análisis de punteros es muy importante.
Intereses	Relación con el desarrollo intelectual o profesional de los involucrados	El interés es principalmente porque el análisis de punteros es fundamental en el desarrollo de software seguro, optimizado y eficiente. Además ayuda a la eficiencia de los compiladores, algo que me interesa bastante.
Imposición/Obligación	Necesidad de cumplimiento de requisitos académicos, problemas organizacionales a solucionar	La elección de este tema también responde a una necesidad académica, ya que forma parte de mi tesis de grado en Licenciatura en Ciencias de la Computación.
Capacidades	Conjunción de factores incidentes que tienen la posibilidad de aportar un desarrollo exitoso del estudio	Cuento con una formación en lenguajes de programación de bajo nivel, como C y C++, proporcionada por la carrera.
Otros	Novedad del tema, trascendencia, etc.	Claramente el análisis de punteros no es un tema nuevo (podemos observar por los años de los artículos), pero es relevante debido a su complejidad. Además estos impactan directamente la calidad y seguridad del software.

4- Elabore un listado de palabras clave y frases cortas que describan su tema de investigación:

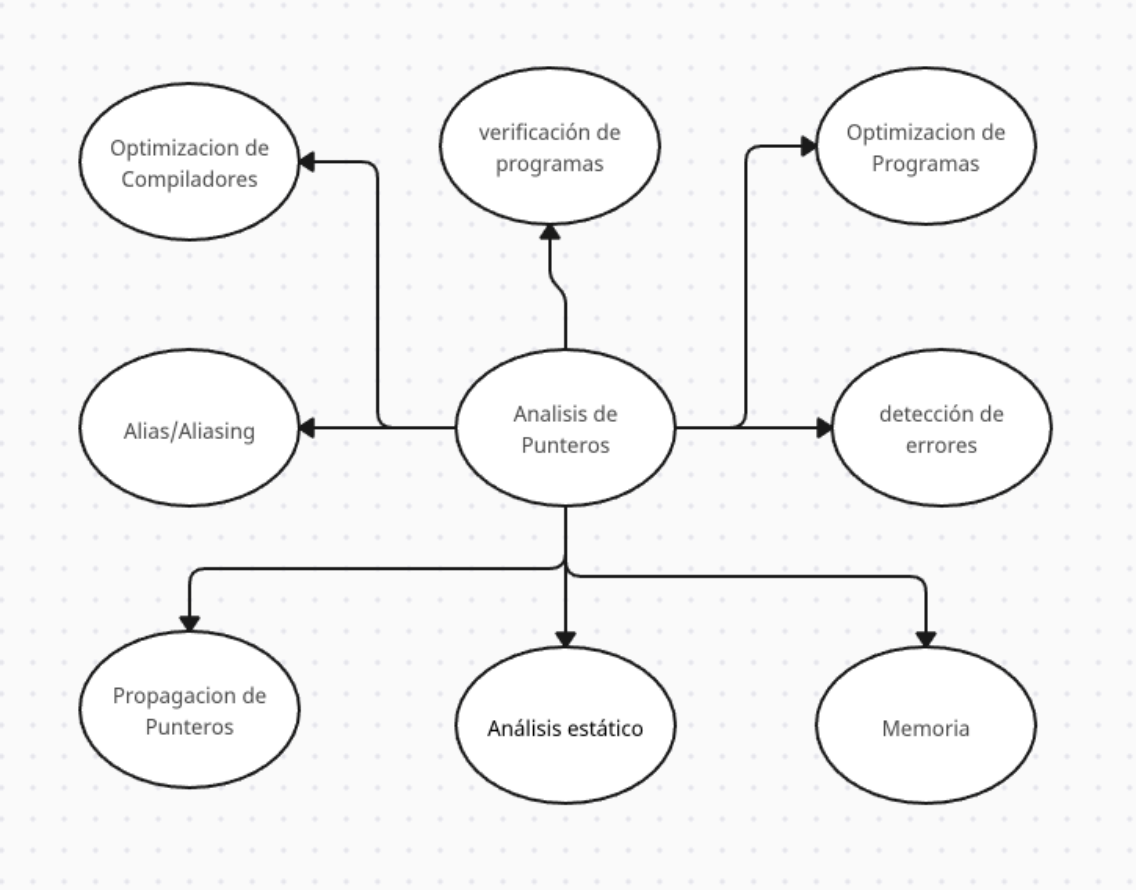
Palabras clave

- *Pointer Analysis* – Análisis de punteros
- *Static Analysis* – Análisis estático
- *Heap Analysis* – Análisis del heap
- *Aliasing / Alias*
- *Program Analysis* – Análisis de programas
- *Data Flow* – Flujo de datos
- *Program Optimization* – Optimización de programas
- *Code Verification* – Verificación de código
- Grafo de Restricciones
- Propagación de conjuntos de punteros
- Optimización de compiladores

Frases Claves:

- "Análisis de punteros para detectar alias y optimizar la ejecución de programas."
- "Evaluación de restricciones mediante grafos."
- "Colapso de ciclos para unificar conjuntos de punteros equivalentes."
- “Aplicaciones del análisis de punteros en compiladores’
- “Mejora en la eficiencia de algoritmos de análisis estático”
- "Análisis de punteros en lenguajes con manejo manual de memoria como C"
- "Aplicación del análisis de punteros en la detección de errores de seguridad."
- “Herramientas para análisis estático de software”

5- Realice un racimo asociativo o representación gráfica de las asociaciones de ideas. En el centro escriba el tema y a su alrededor en forma radial, las ideas o conceptos asociados.



6- Para elaborar el **marco de referencia y antecedentes**, realiza una búsqueda y selección de 10 trabajos académicos que estén vinculados con el tema de tu elección. (Descarga: Formato Normas APA. Recuperado de: https://normasapa.in/) USA DISTINTOS TIPOS DE BUSCADORES, ENTRE LOS COMPARTIDOS POR EL GRUPO DE FACE.

   - **Organiza tus búsquedas:**

   - En un archivo: guarda toda la información bibliográfica del artículo/palabras clave y resumen.

Artículo 1:

Título: Wave Propagation and Deep Propagation. A description of two new algorithms for Inclusion Based Points-to Analysis. (Año: 2009)

Autores: Fernando Magno Quintão Pereira, Daniel Berlín.

Palabras clave: Wave Propagation, Deep Propagation, Points-to Analysis, Algorithms, Memory Efficiency.

Pereira y Berlín desarrollaron dos nuevos algoritmos para el análisis de inclusión basado en punteros. La Propagación de Ondas mejora la eficiencia en comparación con técnicas previas, mientras que la Propagación Profunda minimiza el uso de memoria, presentando un equilibrio entre precisión y rendimiento.

URL: https://homepages.dcc.ufmg.br/~fernando/publications/papers/CGO09.pdf

Artículo 2:

Título: Online Cycle Detection and Difference Propagation for Pointer Analysis. (Año: 2004)

Autores: David J. Pearce, Paul H.J. Kelly, Chris Hankin.

Afiliacion: Department of Computing, Imperial College, London

Palabras clave: Cycle Detection, Difference Propagation, Set Constraints, Worklist Algorithm.

Este artículo presenta y evalúa diversas técnicas para mejorar el tiempo de ejecución del análisis de punteros en el contexto de grandes programas en C. El análisis se formula como un grafo de restricciones de conjunto y se resuelve mediante un algoritmo de lista de trabajo.

URL: https://whileydave.com/publications/PKH03_SCAM_preprint.pdf

Artículo 3:

Título: Flow-Sensitive Pointer Analysis for Millions of Lines of Code. (Año: 2011)

Autores: Ben Hardekopf, Cavin Lin.

Palabras clave: Flow-Sensitive Analysis, Large Codebases, Program Verification, Static Analysis.

Presenta un nuevo algoritmo de análisis de punteros sensible al flujo, mucho más rápido que muchos algoritmos que existen, lo que permite, por primera vez, el análisis de punteros sensible al flujo para programas con millones de líneas de código.

URL: https://www.cs.utexas.edu/~lin/papers/cgo11.pdf

Artículo 4:

Titulo: PUS: A Fast and Highly Efficient Solver for Inclusion-based Pointer Analysis (Año: 2022)

Autores: Peiming Liu, Brad Swain, Yanze Li y Jeff Huang.

Palabras clave: Análisis Estático - Análisis de Punteros - *Causality Subgraph*

Este artículo presenta a Pus, un solucionador muy eficiente para análisis de punteros basados en inclusiones. A diferencia de *deep y wave propagation*, solo considera restricciones parciales que afecten al resultado final. En programas complejos de gran tamaño da resultados muy comprometedores.

URL: https://peimingliu.github.io/asset/pic/PUS.pdf

Artículo 5:

Título: An Effective Approach of Points-To Analysis (Año: 2015)

Autores: Zhang Yuping, Deng Zhaori, Zhang Xiaoning, Ma Yan

Afiliación: College of Information Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China

Este artículo aborda una mejora de eliminación de ciclos para el análisis de punteros sensible al contexto basado en gráficos de invocación. Utiliza los métodos de propagación por ondas y propagación profunda, que son técnicas para la optimización del análisis de punteros basado en inclusión Se propone un algoritmo de propagación por ondas y propagación profunda sensible al contexto. Primero, se presenta la definición e inicialización de un nuevo grafo de restricciones. Luego, mediante un ejemplo, se describe cómo la propagación sensible al contexto puede realizar un análisis de punteros de forma precisa y eficiente.

Palabras clave:
Optimización, Invocación, Proceso sensible al contexto, Análisis de punteros, Eliminación de ciclos, Grafo de llamadas

URL: https://www.atlantis-press.com/article/17650.pdf

Artículo 6

Título: Pointer Analysis. (Año: 2015)

Autores: Yannis Smaragdakis, George Balatsouras.

Afiliación: University of Athens.

Palabras clave: Static Analysis, Logical Specifications, Pointer Approximation, Program Analysis.

Resumen: Este trabajo presenta una introducción y un estudio de las técnicas de *pointer analysis*, con énfasis en destilar la esencia de los algoritmos de análisis más comunes. Para ello, se enfoca en una presentación común de análisis de punteros: los algoritmos son modelados como especificaciones lógicas configurables y de fácil comprensión.

URL: https://yanniss.github.io/points-to-tutorial15.pdf

Artículo 7

Titulo: *Making context-sensitive inclusion-based pointer analysis practical for compilers using parameterised summarisation*. (Año: 2013)

Autores: Yulei Sui, Sen Ye, Jingling Xue, Jie Zhang

Afiliación: University of New South Wales & Beijing University of Chemical Technology.

Palabras Clave: Pointer analysis- Inclusion-based analysis - Context-sensitive analysis

Resumen
Debido a su alta precisión como análisis de punteros insensible al flujo, el análisis de Andersen se ha implementado en algunos compiladores de optimización modernos. Para lograr una mayor precisión, describimos cómo añadir sensibilidad al contexto al análisis de Andersen. El análisis resultante, denominado ICON, es eficiente para analizar programas grandes y, al mismo tiempo, lo suficientemente preciso como para impulsar las optimizaciones del compilador. Su novedad radica en resumir los efectos secundarios de un procedimiento mediante una función de transferencia en variables virtuales que representan ubicaciones completamente parametrizadas a las que se accede mediante sus parámetros formales. Como resultado, se logra un buen equilibrio entre eficiencia y precisión, lo que resulta en un ICON más potente que un análisis sensible a un solo sitio de llamada y menos que un análisis sensible a la ruta de llamada (cuando los ciclos de recursión de un programa se colapsan en todos los casos).

URL: https://yuleisui.github.io/publications/spe14.pdf

Artículo 8

Titulo: *Scaling Context-Sensitive Points-to Analysis*. (Año: 2012)

Autor: Rupesh Nasre
Afiliación: *Computer Science and Automation Indian Institute of Science*

Palabras Clave

Pointer analysis - Context-sensitive analysis - Inclusion-based analysis - Program analysis scalability - Multibloom filter - Randomized algorithms
- Linear system modeling -Constraint prioritization

URL:
https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e7fd6928faafdb950026bbb4ae2a597561772d26

Resumen

Rupesh Nasre propone cuatro técnicas innovadoras orientadas a mejorar la escalabilidad de este tipo de análisis de punteros en programas C/C++. Introduce el uso de filtros *multibloom* para representar información de punteros de forma eficiente. Esta técnica, reduce significativamente el tiempo de ejecución y los requisitos de memoria. También desarrolla un algoritmo que agrupa restricciones para procesarlas con distintas precisiones.

Artículo 9

Titulo: Points-To Analysis with Efficient Strong Updates (Año: 2011)

Autores: Ondřej Lhoták, Kwok-Chiang Andrew Chung

Afiliación: D. R. Cheriton School of Computer Science University of Waterloo Waterloo, Ontario, Canada

Palabras Clave: Points-to analysis, Flow sensitivity, Strong updates, Andersen's analysis, LLVM

Resumen

Los autores proponen un enfoque intermedio entre el análisis de punteros *flow-insensitive* (eficiente) y el *flow-sensitive* (preciso). La propuesta, llamada “Strong Update” analysis, combina la eficiencia del análisis insensible al flujo con la precisión de las actualizaciones fuertes, típicas del análisis sensible al flujo.

URL: https://plg.uwaterloo.ca/~olhotak/pubs/popl11.pdf

Artículo 10

Título: *Efficient field-sensitive pointer analysis of C (*Año: 2006*)*

Autores: David J. Pearce, Paul H.J. Kelly, Chris Hankin
Publicado en: *ACM Transactions on Programming Languages and Systems (TOPLAS)*,

Palabras Clave: Restricciones de conjuntos - Análisis de punteros

Resumen

El tema de este artículo es el análisis de punteros insensible al flujo y al contexto. Presentamos un enfoque novedoso para modelar con precisión variables de estructura y llamadas indirectas a funciones. Nuestro método prioriza la eficiencia y la simplicidad, y amplía el lenguaje de las restricciones de conjuntos.

URL:
https://www.doc.ic.ac.uk/~phjk/Publications/EfficientFieldSensitivePointerAnalysisforCPASTE04.pdf

- Lee y selecciona los artículos que consideres pertinente, organizándolos de mayor a menor generalidad en otro archivo, sigue conservando la información bibliográfica.
- Busca información académica sobre los autores de dichos trabajos (esto te permitirá establecer dónde se estudia tu tema).

Artículo 1
Título: *Wave Propagation and Deep Propagation. A Description of Two New Algorithms for Inclusion-Based Points-to Analysis* (Año: 2009)

Autores: Fernando Magno Quintão Pereira y Daniel Berlín

- **Fernando Magno Quintão Pereira:**
    - Obtuvo su doctorado en la Universidad de California, Los Ángeles (UCLA) en 2008. Actualmente es profesor asociado en el Departamento de Ciencia de la Computación en la Universidade Federal de Minas Gerais (UFMG), Brasil. Sus principales áreas de investigación incluyen generación de código, optimización de compiladores y análisis estático de programas.
- **Daniel Berlín:** La única información encontrada de él es la siguiente:

- Afiliación: Google, Inc., EE.UU.

- Temas de publicación: análisis de punteros, consumo y uso de memoria, optimización de compiladores (por ejemplo, GCC), algoritmos y métodos de análisis (incluyendo topología de grafos y tiempos de ejecución).
-

Artículo 2 y Articulo 10:

Título Art. 2: *Online Cycle Detection and Difference Propagation for Pointer Analysis. (Año: 2004)*
Título Art. 10: *"Efficient Field-Sensitive Pointer Analysis for C" (2006)*
Autores: David J. Pearce, Paul H.J. Kelly y Chris Hankin

- **David J. Pearce, Paul H.J. Kelly y Chris Hankin:**

Afiliación: Se los asocia con el Departamento de Computing del Imperial College London.

David J. Pearce:

Afiliación: ConsenSys, equipo de Trustworthy Smart Contracts.

Intereses de investigación: lenguajes de programación, compiladores, análisis estático y verificación formal.

Durante su doctorado, desarrolló algoritmos innovadores para el análisis de punteros estáticos.

- **Paul H.J. Kelly**
  - Profesor de Tecnología de Software en el Imperial College London (desde 1989).
  - Dirige el grupo de investigación de Optimización del Rendimiento de Software del Imperial College.
  - Codirector del Centro de Métodos Computacionales en Ciencia e Ingeniería del Imperial College.
  - Temas de investigación: compiladores, arquitectura de computadoras, teoría de la computación, IA y procesamiento de imágenes.

- **Chris Hankin**

  Chris Hankin es miembro del Instituto de Ciencia y Tecnología de Seguridad y profesor de Ciencias de la Computación del Imperial College de Londres.

Su investigación se centra en la ciberseguridad, el análisis de datos y el análisis de programas basado en semántica.

Lidera proyectos multidisciplinares centrados en el desarrollo de análisis visuales avanzados y en proporcionar un mejor soporte de decisiones para defenderse de los ciberataques.

Dirige el Instituto de Investigación NCSC/EPSRC en Sistemas Ciber Físicos Interconectados Confiables.

Artículo 3:

Título: Flow-Sensitive Pointer Analysis for Millions of Lines of Code. (Año: 2011)
Autores: Ben Hardekopf, Cavin Lin.

- **Ben Hardekopf**

  El profesor Hardekopf se licenció en Ciencias de la Computación e Ingeniería Eléctrica en la Universidad de Duke en 1997. Cursó su doctorado en Ciencias de la Computación en la Universidad de Texas en Austin (con la supervisión del profesor Calvin Lin).

  Afiliación: Se incorporó al profesorado de la Universidad de California en Santa Bárbara en 2009.

  La investigación del profesor Hardekopf se centra en los lenguajes de programación y su diseño, análisis e implementación.

- **Cavin Lin**

  Calvin Lin es Profesor Distinguido de la Universidad de Texas (UT) y se dedica a la investigación en arquitectura informática, compiladores y seguridad.

  Áreas de investigación:

  - Arquitectura de computadoras
  - Computación paralela
  - Lenguajes de programación y compiladores
  - Seguridad y privacidad

  Intereses de investigación:

  - Compiladores
  - Educación en informática

- **Para antecedentes**: Usa citas de parafraseo o indirectas y comienza a escribir: Pepe (2023), en su investigación…; céntrate en el resumen del trabajo. Finalmente, reúne los antecedentes ubicándolos en orden cronológico y de mayor a menor especificidad.

Artículo 2 (2004- Pearce, Kelly y Hankin): Se proponen técnicas para mejorar el tiempo de los análisis interprocedural de punteros en C, formulando el problema como un grafo de restricciones. Se introduce un algoritmo para la detección en línea de ciclos y una técnica de propagación de diferencias, optimizando el proceso al registrar cambios en las soluciones de variables.

Artículo 10 (2006 – Pearce, Kelly y Hankin): Desarrollan un análisis de punteros insensible al flujo y al contexto para C, mejorando la precisión al modelar variables de estructura e invocaciones indirectas a funciones, aunque con mayor costo computacional.

Artículo 1 (2009 – Pereira y Berlin): Desarrollaron dos algoritmos para análisis de punteros basado en inclusión. El método de *Wave Propagation* (es una versión modificada de una técnica temprana presentada por Pearce et al.) mejora significativamente el tiempo de ejecución de su predecesor, mientras que *Deep Propagation* es más liviana en cuanto a uso de memoria, demostrando ventajas en *benchmarks* pequeños.

Artículo 3 (2011 – Hardekopf y Lin): Introducen un algoritmo de análisis de punteros sensible al flujo capaz de manejar programas con millones de líneas de código. Su método se fundamenta en una representación dispersa del código, obtenida a partir de un análisis por etapas e insensible al flujo.

- **Para marco teórico:** Selecciona un autor/res que trabajen sobre tu tema de interés. Introducelos al lector (resumen líneas de trabajo y enfoques que hayan desarrollado), finalmente identifica los términos teóricos centrales y defínelos (puedes hacer cita directa).

Marco Teórico

Para abordar el análisis de punteros, tomaré como referencia el trabajo de Fernando Magno Quintão Pereira y Daniel Berlín, quienes desarrollaron nuevas metodologías en la optimización del análisis de inclusión basado en punteros. Pereira, profesor en la Universidad Federal de Minas Gerais, trabajó en compiladores, generación de código y análisis estático de programas. Berlín, contribuyó en la optimización de análisis de programas en entornos de desarrollo a gran escala. Ambos investigadores diseñaron algoritmos eficientes para mejorar la precisión y el rendimiento del análisis de punteros, destacándose por la implementación de la Propagación de Ondas y la Propagación Profunda.

Términos Teóricos Centrales

- **Análisis de Punteros:** Proceso mediante el cual se determina a qué ubicaciones de memoria pueden apuntar los punteros en un programa.
- **Propagación de Ondas:** Algoritmo diseñado para mejorar la eficiencia del análisis de inclusión basado en punteros, reduciendo el tiempo de ejecución en comparación con métodos tradicionales. Aunque requiere de mucho uso de memoria, logra tiempos de ejecución muy bajos en entornos con abundantes recursos, especialmente en benchmarks de gran tamaño.
- **Propagación Profunda:** es un análisis más ligero que requiere menos memoria, sin sacrificar precisión. Presenta el mejor tiempo de ejecución promedio, menores requisitos de memoria y los tiempos más rápidos para menos de 100.000 líneas de código.