

Lenguajes de Programación

Tarea 1

Karla Ramírez Pulido Alan Alexis Martínez López

Semestre 2023-2 **Fecha de inicio:** 13 de febrero 2023
Facultad de Ciencias UNAM **Fecha de entrega:** 20 de febrero 2023

Integrantes:

Dania Paula Gongora Ramírez
Salgado Tirado Diana Laura

Instrucciones

Resolver los siguientes ejercicios de forma clara y ordenada de acuerdo a los lineamientos de entrega de tareas disponibles en la página del curso. Esta tarea en particular se realizará en **equipos de 2 personas**.

Ejercicios

1. **Elige 4 lenguajes de programación (uno por cada paradigma), e indica para cada uno de ellos el año de creación, paradigma al que pertenece y principales características.**
 - **Java** – 1991 – Orientado a objetos
 - **Portable:** El código Java es compilado en bytecode que puede ser ejecutado en cualquier plataforma que tenga una máquina virtual Java (JVM).
 - **Seguro:** Java está diseñado para ser un lenguaje seguro, ya que se ejecuta en una JVM que controla el acceso a los recursos del sistema y evita la ejecución de código malicioso.
 - **Multihilo:** Java soporta la ejecución de múltiples hilos de forma concurrente.
 - **Robusto**
 - **Fácil de aprender:** Java tiene una sintaxis sencilla y clara, lo que lo hace fácil de aprender y entender.
 - **Interpretable:** Java permite la interpretación del código en lugar de la compilación.
 - **Prolog** – 1972 – Lógico
 - **Declarativo:** En lugar de dar una secuencia de pasos para resolver un problema, Prolog describe el problema de una manera declarativa
 - **Interpretado:** Prolog es un lenguaje interpretado, lo que significa que se ejecuta directamente desde el código fuente.
 - **Eficiente:** Prolog es un lenguaje de programación eficiente que utiliza la resolución de unificación para encontrar soluciones a problemas lógicos complejos.
 - **Manejo de listas:** Prolog tiene una gran facilidad para manejar listas y otros tipos de datos complejos.
 - **Integración con bases de datos:** Prolog puede integrarse fácilmente con sistemas de gestión de bases de datos y se utiliza a menudo para la programación de bases de datos.
 - **Haskell** – 1990 – Funcional
 - **Funciones puras:** todas las funciones en Haskell son funciones puras, lo que significa que no tienen efectos secundarios y no modifican el estado del programa.

- **Evaluación perezosa:** Haskell utiliza una técnica llamada evaluación perezosa, lo que significa que las expresiones se evalúan sólo cuando es necesario.
- **Sintaxis clara y concisa:** la sintaxis de Haskell es clara y concisa, lo que hace que el código sea fácil de leer y escribir.
- **Pattern matching:** Haskell utiliza una técnica llamada pattern matching, que permite descomponer una estructura de datos en sus componentes individuales.
- **Python** – 1991 – Imperativo
 - **Multiparadigma:** Python admite múltiples paradigmas de programación, como programación imperativa, programación orientada a objetos y programación funcional.
 - **Amplia biblioteca estándar:** Python tiene una amplia biblioteca estándar que proporciona módulos y funciones predefinidas para realizar tareas comunes, como trabajar con archivos, redes y bases de datos.
 - **Portabilidad:** Python es un lenguaje de programación portátil que se puede ejecutar en múltiples plataformas, como Windows, macOS y Linux.
 - **Interactivo:** Python tiene un intérprete interactivo que permite probar y depurar código de forma rápida y sencilla.

2. **LISP es considerado el primer lenguaje de programación funcional. Y está basado en el Cálculo λ que fue desarrollado por Alonzo Church.**

- a. **Investiga y explica brevemente qué elementos del Cálculo λ están presentes en LISP e indica por qué crees que pueden usarse en un lenguaje de programación.**

El cálculo lambda es un sistema formal diseñado en base de la noción de aplicación de funciones y la recursión, los lenguajes de programación basados en cálculo lambda, son llamados lenguajes de paradigma funcional, en estos lenguajes en comparación de los imperativos no existen estructura de iteración, por lo que lleva a que las definiciones sean creadas en base a recursión.

Estas son características de LISP como lenguajes de programación, el cual esta constituido por definiciones y funciones matemáticas,

De las principales características es la habilidad de expresar algoritmos de manera recursiva, otro elemento importante en LISP es la capacidad de permitir que la funciones sean pasadas y devueltas como parámetros, además de basarse fuertemente en la idea utilizar formulas como funciones, además de que en LISP como lenguaje de programación esta implementado las expresiones lambda, las cuales las cuales son funciones anónimas.

Todas estas características del cálculo lambda son útiles de aplicar en un lenguaje de programación, ya que el cálculo lambda ya que este es equivalente a las maquinas de turing, el poder de esta teoría de cómputo radica en que este se centra en qué es lo que va a ser computado en vez de concentrarse en él como, esto hace que los lenguajes de programación con estas características su fuerte sea la recursión el cual es un recurso muy importante en la computación.

- b. **Menciona cuáles de estos elementos están presentes en el lenguaje de programación JAVA. ¿Acaso estos elementos estaban en las primeras versiones del lenguaje? De no ser así ¿porqué crees que fueron añadidos?**

El elemento que fue incluido en java acerca del cálculo lambda son las expresiones lambda, las cuales funcionan como clases anónimas, esta no fue añadida hasta la versión 8 de JAVA (Java 1.8 en Marzo de 2014), antes de esta versión no se encontraban estas expresiones, considero que fueron añadidos, ya que aunque JAVA es un lenguaje del tipo orientación a objetos, esta característica del cálculo lambda provee de un gran poder computacional muy útil ya que permite delegar fragmentos de código.

3. **Describe las principales características de los distintos paradigmas de programación (Imperativo, Orientado a Objetos, Funcional y Lógico) y da 2 ejemplos de lenguajes de programación de cada paradigma.**

Lenguajes de programación con paradigma imperativo: Las principales características de estos lenguajes de programación es que los programas escritos bajo este paradigma son secuencias concisas de instrucciones, de forma que se encadenan instrucciones una detrás de otra, los valores de las variables pueden ser modificados al tiempo de la ejecución, estos suelen utilizar estructuras de control para ejecutar las instrucciones, tiene un enfoque procedimental lo cual significa que divide una tarea grande en tareas más pequeñas y sencillas. Dos ejemplos de lenguajes de programación son C y Fortran.

Lenguajes de programación con paradigma Orientado a Objetos: La orientación a objetos se caracteriza por buscar modelar representaciones de la vida real por medio de la abstracción, otra característica de este paradigma es el uso de encapsulamiento, herencia y polimorfismo, el encapsulamiento se refiere a proteger la información o estado de los atributos para que no se pueda ver o modificar la información del objeto sin un mecanismo establecido, la herencia nos permite reutilizar código dando las características de un objeto a sus "descendientes", el polimorfismo se refiere a una cualidad de la herencia en la cual a los objetos a los que se les extiende ciertas características al invocar un método este se adapte de manera distinta a cada objeto. Dos ejemplos de lenguajes de programación son java y python.

Lenguajes de programación con paradigma Funcional: Sus principales características son; estar fuertemente basado en el cálculo lambda, definiciones de funciones de matemáticas puras, valores de variables inmutables, uso de recursión en la definición de funciones, el uso de lista como estructura de datos principal, utilizar funciones como tipos primitivos y expresiones lambda. Dos ejemplos de lenguajes de programación son Lisp y swift.

Lenguajes de programación con paradigma Lógico: Sus principales características son; estar fuertemente basado en la lógica de predicados de primer orden, no sé especifica el como, si no el que, es decir, el programa son declaraciones que describen la solución, utiliza el backtracking como control de secuencia para lograr los resultados deseados, los predicados son relaciones lo cual hace que los argumentos funciones como entrada y salida. Dos ejemplos de lenguajes de programación son Prolog y mercury.

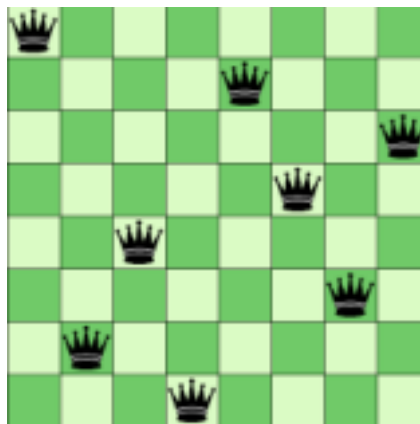
4. ¿Cuál de los paradigmas de lenguajes de programación, es el más adecuado para resolver los siguientes problemas? Justifica en cada caso.

a. Se requiere desarrollar un sistema que simule un modelo de sociedad de organización de termitas. Este sistema se compone, de manera general de: un espacio que las termitas recorrerán y en el cuál se encuentran astillas esparcidas, las termitas siguen las siguientes reglas de comportamiento.

- Caminan aleatoriamente hasta encontrar una astilla.
- Si la termita se encuentra cargando una astilla, la suelta y continúa caminando aleatoriamente.
- Si la termita no está cargando una astilla la toma y continúa caminando aleatoriamente con la astilla

Para resolver este problema el paradigma de programación más adecuado es el orientado a objetos ya que de esta manera podrían representarse las termitas como objetos con atributos que representen si esta cargando una astilla o no, además de métodos que representen otros comportamientos de las terminas como caminar o soltar la astilla.

b. El Problema de las Ocho Reinas consiste en acomodar ocho reinas de ajedrez en un tablero, sin que ninguna de éstas se ataque entre sí. Una reina, puede atacar (a) de forma vertical, (b) de forma horizontal y (c) en diagonal. Usando estas reglas, indicar si el siguiente tablero es una solución al problema de las ocho reinas.



Para resolver este problema el paradigma de programación más adecuado también es el orientado a objetos pues en este los algoritmos de backtracking suelen ser más fáciles de implementar ya que las soluciones parciales pueden ser almacenadas en objetos.

El backtracking es una de las formas más efectivas de resolver el problema de las ocho reinas.

c. **Escribir un programa que indique si un número natural es *primo*. Un número es primo.**
El paradigma más adecuado sería el de programación imperativa, ya que este enfoque funciona muy bien para problemas matemáticos pues se puede dividir al problema en pequeñas operaciones lógicas.

d. **Dado un archivo con los primeros 10,000 dígitos de π , contar todas las apariciones de números primos presentes en él.**

La programación funcional es el más adecuado para este problema porque este funciona mejor con funciones, por lo tanto es el adecuado para resolver problemas matemáticos y de procesamiento de datos.

5. Investiga qué significan los siguientes conceptos en un lenguaje de programación. Y elabora un pequeño ejemplo ocupando como base al lenguaje de programación *HASKELL*

a. Sintaxis. Como el conjunto de normas a seguir al escribir el código fuente para que este pueda considerarse como correcto para ese lenguaje de programación.

Ejemplo. La sintaxis para crear nuestro propio tipo de dato es la siguiente:

```
data Cosa = Zapato
```

```
    | Barco
```

```
    | Cera
```

```
    | Repollo
```

```
    | Rey
```

```
    | Tacos
```

```
deriving Show
```

Lo anterior constituye un nuevo tipo de dato llamado 'Cosa' con cinco constructores.

b. Semántica. Se refiere al significado que se le da a la combinación de símbolos derivados de la sintaxis, es decir, el comportamiento derivado de la ejecución del programa.

Ejemplo. Tenemos el siguiente método:

```
suma :: [Int] -> Int
```

```
suma [] = 0
```

```
suma (x:xs) = x + suma xs
```

En este lo que hacemos es sumar los números de una lista utilizando recursión, lo que hace es sumar la cabeza de la lista y llamar recursivamente la función con la cola.

c. Idioms (convenciones de programación). Se refiere a un patrón de código perteneciente a un lenguaje de programación específico, son un conjunto de guías y prácticas que ayudan a generar buenas prácticas de programación para facilitar el mantenimiento de software.

Ejemplo. La convención más común es el cómo utilizar los comentarios, en el caso de haskell los comentarios se hacen con los símbolos “-” o “{-}” para comentarios largos, estos suelen escribirse arriba de la definición o función que se quiere describir:

--- Función auxiliar que elimina elementos repetidos de una lista---

```
elimina :: [Prop] -> [Prop]
```

```
elimina [] = []
```

```
elimina [x] = [x]
```

```
elimina (x:xs) = x : [ y | y <- elimina(xs), y /=x ]
```

d. Bibliotecas. Conjunto de archivos para el desarrollo de software, compuesta por código y datos, están hechas con el objetivo de programar de manera más sencilla proporcionando funcionalidades comunes.

Ejemplo. Un ejemplo es la descripción de la siguiente función del lenguaje:

```
head :: HasCallStack => [a] -> aSource#
```

O(1)

O(1). Extract the first element of a list, which must be non-empty.

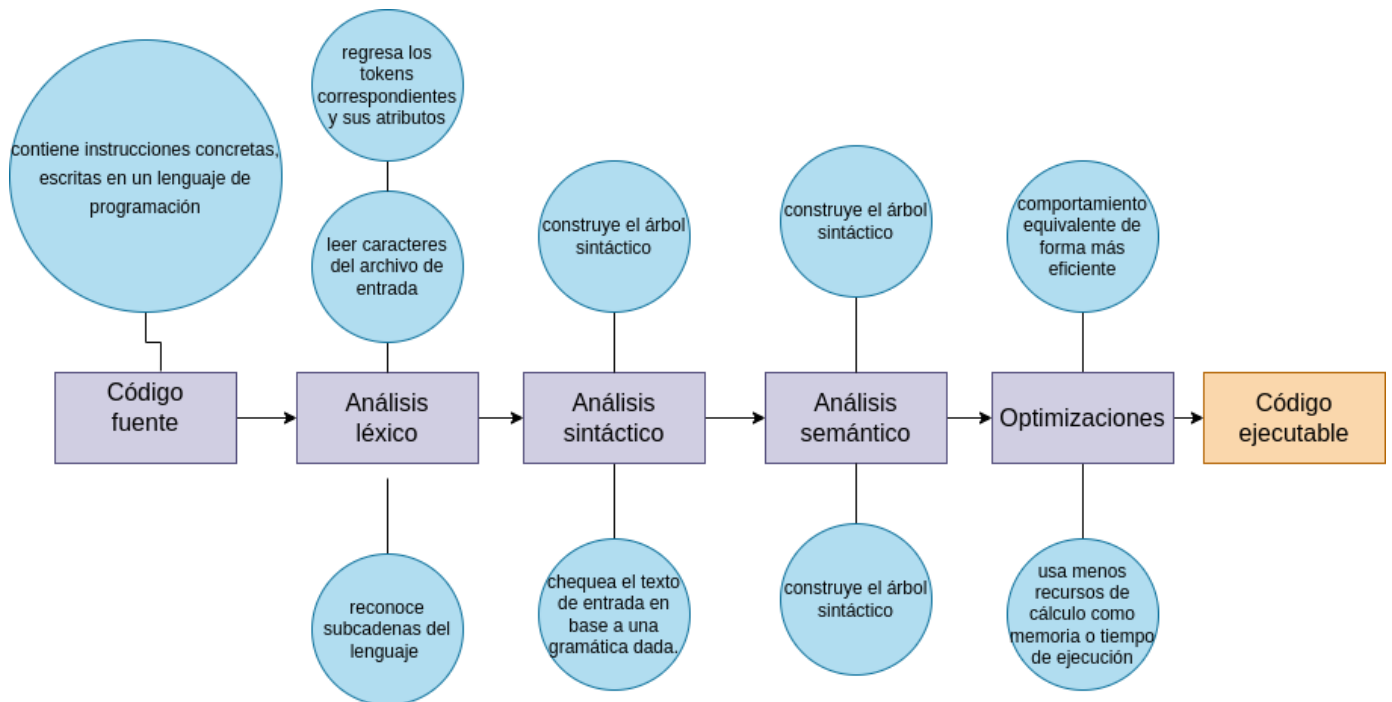
6. Calcula el resultado de las siguientes funciones en el lenguaje de programación *Racket* y muéstralo. Posteriormente realiza tu propia implementación de cada función.

a. (second '(1 7 9 4 5 6))

b. (append '(Me) '(mes))

```
> (second '(1 7 9 4 5 6))
7
> (append '(Me) '(Mes))
'(Me Mes)
> (segundo '(1 7 9 4 5 6))
7
> (unir '(Me) '(Mes))
'(Me Mes)
```

7. Dibuja un mapa mental que muestre las fases de generación código ejecutable, sus principales características y elementos involucrados.



Fuentes:

Para la pregunta 1:

<https://cognosonline.com/mx/blog-mx/que-son-paradigmas-de-programacion/>

<https://okhosting.com/blog/paradigma-de-programacion/>

<https://www.monografias.com/trabajos107/paradigmas-programacion/paradigmas-programacion>

Para la pregunta 2 se sacó información de:

<https://cursos.aiu.edu/LENGUAJES%20DE%20PROGRAMACION/Sesi%C3%B3n%209/PDF/Lenguajes%20de%20Programaci%C3%B3n%20Sesio%C2%B4n9.pdf>

Para la pregunta 3:

<https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/programacion-imperativa/>

<https://portalacademico.cch.unam.mx/cibernetica1/algoritmos-y-codificacion/caracteristicas-POO>

<https://domingogallardo.github.io/apuntes-lpp/teoria/tema02-programacion-funcional/tema02-programacion-funcional.html>

<http://blogramacion5.blogspot.com/2015/09/paradigma-logico.html>

Para la pregunta 5:

<http://odin.fi-b.unam.mx/salac/practicaspoo/P02-POO-Fundamentos.pdf>

<https://gus.chat/blog/semantica-en-programacion-tipos-y-usos-mas-comunes/#:~:text=La%20sem%C3%A1ntica%20en%20programaci%C3%B3n%20describe,dos%20partes%3A%20sintaxis%20y%20sem%C3%A1ntica.>

https://es.wikipedia.org/wiki/Convenciones_de_programaci%C3%B3n

Para la pregunta 7:

<https://www.jvs-informatica.com/blog/glosario/codigo-fuente/#:~:text=El%20c%C3%B3digo%20fuente%20es%20un,identificada%20como%20ingenier%C3%ADa%20de%20software.>

<https://blogs.uoc.edu/informatica/optimizacion-de-codigo-un-codigo-mas-eficiente/>

<http://www.lcc.uma.es/~galvez/ftp/tci/tictema3.pdf>

<http://arantxa.ii.uam.es/~alfonsec/docs/compila5.htm>

<https://dc.exa.unrc.edu.ar/staff/fbavera/papers/TesisJTLex-Bavera-Nordio-02.pdf>