



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

### **Tarea 1**

#### INTEGRANTES

**Torres Valencia Kevin Jair - 318331818**  
**Aguilera Moreno Adrián - 421005200**  
**Rivera Silva Marco Antonio - 318183583**

#### PROFESORA

**Karla Ramírez Pulido**

#### AYUDANTES

**Alan Alexis Martínez López**  
**Manuel Ignacio Castillo López**  
**Alejandra Cervera Taboada**

#### ASIGNATURA

**Lenguajes de Programación**

1 de septiembre de 2022

---

## Pregunta 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.

Aquí pueden comenzar a poner sus respuestas.

Prueba de respuesta.

---

## Pregunta 2

LISP es considerado el primer lenguaje de programación funcional. Y esta basado en el Cálculo- $\lambda$  que fue desarrollado por Alonzo Church.

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

▷ Estos elementos<sup>1</sup> realizan procedimientos principalmente con funciones, aunque una notable diferencia sería que en el cálculo- $\lambda$  no se trabaja con algún tipo de datos, pues estos no existen, y en LISP tenemos tipos de datos heterogéneos como los átomos. Los átomos podrían biyectarse con la aplicación a términos de una función lambda (sin embargo no son exactamente igual), por ejemplo

$$(\lambda_x \cdot x) a = a$$

puede equivaler al átomo  $a$  en LISP.

Una característica que puede ser común entre el cálculo- $\lambda$  y LISP es la recursión, pues mientras en LISP se utiliza principalmente la recursión para operar sus funciones, en el cálculo- $\lambda$  se utilizan las  $\beta$ -reducciones (estas muchas veces son recursivas con algunas reglas ya definidas).

La característica de usar paréntesis en su sintaxis es común, sin embargo, esta no se ve reflejada en su semántica. Pues en LISP representan listas y en el cálculo- $\lambda$  representa un orden jerárquico en el que se aplican las funciones base.

En ambos modelos de computo se pueden tener funciones sin aridad definida<sup>2</sup>. También se sabe que LISP trabaja propiamente con representaciones  $\lambda$ 's.

◁

- 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?

▷ En JAVA como en muchos otros lenguajes, podemos realizar recursión (tal vez no siempre sea lo más eficiente).

A partir de la versión 8 de JAVA se añadieron  $\lambda$ 's al lenguaje, mencionaremos 2 razones importantes para que esto haya pasado.

- Para que más personas se interesen en el lenguaje. El conjunto de personas que programan tal vez no es el más grande comparado con la humanidad, sin embargo, entre este subconjunto de personas existen otros tantos, por ejemplo, hay personas que sólo programan de manera funcional, algunos solo hacen orientación a objetos, y otros pocos son fan de la programación lógica (y luego están los otros tipos de lenguajes dentro de ambos paradigmas). La intersección no es vacía, pero el que JAVA tenga  $\lambda$ 's hace que el lenguaje sea más atractivo para quienes gustan de manejarlas.
- Facilitan el uso de funciones cuando no hay necesidad de usarlas múltiples veces y su cuerpo es pequeño. Normalmente se usan como funciones anónimas (aunque son equivalentes, no son necesariamente iguales a las  $\lambda$ 's en cuanto a poder computacional).

◁

---

<sup>1</sup>Cálculo- $\lambda$  y LISP.

<sup>2</sup>Múltiples parámetros.

## Pregunta 3

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

▷ Para esta pregunta, explicamos cada parádigma por separado, esto es



### Parádigma Imperativo

Describe la programación en términos del estado del programa y sentecnias que cambian dicho estado. Se le debe explicar el como hacer las cosas.

- **Orientado a objetos.** Las principales características en las orientación a objetos son; el polimorfismo (diseñar objetos que compartan comportamientos, y los puedan heredar), encapsulamiento (la información referente a un objeto esta contenida en este y puede no ser accesible al exterior), reutilización de código (modularizar nos permite evitar duplicar código), los conceptos de clases y objetos (nuestros objetos tendrán comportamientos definidos en clases).

Ejemplos de lenguajes orientados a objetos:

1. C++.
2. JAVA.

- **Estructurado.** Las principales características en los lenguajes estructurados son; la sintaxis (orden y reglas del lenguaje), la semántica (interpretación y significado de la sintaxis), la pragmática (modo en que el contexto o situación le entrega interpretación al significado).

Ejemplos de lenguajes estructurados:

1. C.
2. PASCAL.

---

## Paradigma Declarativo

Este paradigma de programación esta basado en el desarrollo de programas “declarando” un conjunto de condiciones, proposiciones, afirmaciones, restricciones, ecuaciones o transformaciones que describen el que requiera y no el como hacer.

- **Funcional.** Las principales características de la programación funcional son; semántica limpia (no hay ambigüedades en el significado de las funciones, almacenamiento implícito de datos), inexistencia de efectos colaterales (no hay asignación a variables globales), utilización del cálculo- $\lambda$  (independencia fuera de la función).

Ejemplos de lenguajes funcionales:

1. Racket.
2. Haskell.

- **Lógico.** Las principales características de los lenguajes en el paradigma lógico son; reglas (conjunto de proposiciones lógicas escritas como clausulas de Horn), hechos (Expresiones atómicas que verifican un predicado sobre algunos entes que puede procesar el mismo), consultas (proposiciones que se verifican para algún valor de verdad), recursión.

Ejemplos de lenguajes lógicos:

1. GODEL.
2. PROLOG.

---

## Pregunta 4

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

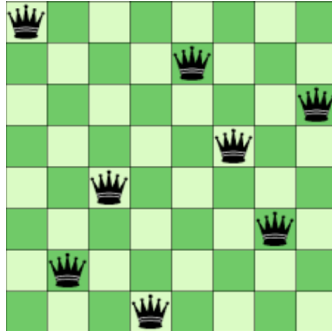
a) Se requiere desarrollar un sistema que simule un modelo de sociedad de organización de termitas. Este sistema se compone, de manera general de: un espacio que las termitas recorrerán y en el cuál se encuentran astillas esparcidas, las termitas siguen las siguientes reglas de comportamiento.

- 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 caminado aleatoriamente con la astilla

Aquí pueden comenzar a poner sus respuestas.

---

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.



Aquí pueden comenzar a poner sus respuestas.

---

c) Escribir un programa que indique si un número natural es primo. Un número es primo.

Aquí pueden comenzar a poner sus respuestas.



---

d) Dado un archivo con los primeros 10,000 dígitos de  $\pi$ , contar todas las apariciones de números primos presentes en él.

Aquí pueden comenzar a poner sus respuestas.

---

## Pregunta 5

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

a) Sintaxis.

Aqui pueden comenzar a poner sus respuestas.

b) Semántica.

Aqui pueden comenzar a poner sus respuestas.

c) Idioms (Convenciones de programación).

Aqui pueden comenzar a poner sus respuestas.

d) Bibliotecas.

Aqui pueden comenzar a poner sus respuestas.

---

## Pregunta 6

A partir de la siguiente función, crea una firma indicando el tipo de entrada que recibirá la función y el tipo de salida que se obtendrá de la función. Asigna un nombre mnemotécnico (es decir que se autodescriba) para la misma. Justifica tu respuesta.

```
(define (foo a b)
  (cond
    [(> (car a) (car b)) (sub1 (foo (cdr a) (cdr b)))]
    [else 0]))
```

Aquí pueden comenzar a poner sus respuestas.

---

## Pregunta 7

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))`

Aqui pueden comenzar a poner sus respuestas.

b) `(append '(Bue) '(n semestre))`

Aqui pueden comenzar a poner sus respuestas.

---

## Pregunta 8

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

Aqui pueden comenzar a poner sus respuestas.