

1. Por cada uno de los miembros del equipo, buscar dos lenguajes de programación distintos e indicar los siguientes datos:

- (a) Nombre del lenguaje.
- (b) Nombre de las o los diseñadores.
- (c) Año de creación.
- (d) Clasificación del lenguaje de acuerdo al:
 - (1) Nivel,
 - (2) Propósito,
 - (3) Estilo/Paradigma.
- (e) Ejemplo de sintaxis y semántica sencillo.
- (f) Ejemplo de biblioteca.
- (g) Ejemplo de convención de programación (idiom).

▪ **Camila:**

- **Nombre del lenguaje:** Wolfram Language.
- **Nombre del diseñador:** Stephen Wolfram.
- **Año de creación:** 1988, hace 33 años.
- **Clasificación del lenguaje de acuerdo al:**
 - a) **Nivel:** Lenguaje de alto nivel.
 - b) **Propósito:** Propósito general.
 - c) **Estilo/Paradigma:** Multiparadigma, simbólico, funcional y lógico.
- **Ejemplo de sintaxis y semántica sencillo:**

```
(* Esto es un comentario. *)
```

```
4 + 3  
(* = 7 *)
```

```
1 + 2 * (3 + 4)  
(* = 15 *)  
(* La multiplicación se puede omitir: 1 + 2 (3 + 4) *)
```

```
(* Las divisiones regresan números racionales: *)  
3 / 2  
(* = 3/2 *)
```

- **Ejemplo de biblioteca:**

```
fun = LibraryFunctionLoad["demo", "demo_I_I", {Integer}, Integer]
```
- **Ejemplo de convención de programación (idiom):**

```
(*En vez de cambiar tu nube con la URL completa se puede hacer  
con una palabra que la explique.*)
```

```
$CloudBase = "https://www.test.wolframcloud.com/"  
Output: https://www.test.wolframcloud.com/  
  
$CloudBase = "prd"  
Output: https://www.test.wolframcloud.com/
```

- **Nombre del lenguaje:** Python.
- **Nombre del diseñador:** Guido van Rossum.
- **Año de creación:** 1991, hace 33 años.
- **Clasificación del lenguaje de acuerdo al:**
 - a) **Nivel:** Lenguaje de alto nivel.
 - b) **Propósito:** Propósito general.
 - c) **Estilo/Paradigma:** Multiparadigma, orientado a objetos, funcional.
- **Ejemplo de sintaxis y semántica sencillo:**

```
# Este es un comentario  
midpoint = 5  
  
# Se crean dos listas vacías  
lower = []; upper = []  
  
# La indentación importa para el control de los bloques de código  
# separa los números en inferiores y superiores  
for i in range(10):  
    if (i < midpoint):  
        lower.append(i)  
    else:  
        upper.append(i)  
  
print("lower:", lower)  
print("upper:", upper)
```

- **Ejemplo de biblioteca:**
`import matplotlib.pyplot as plt`
- **Ejemplo de convención de programación (idiom):**

```
#Se puede cambiar el valor de dos variables por tuplas  
x = True  
y = False  
x, y = y, x  
x  
# False  
y  
# True
```

■ Oscar

- **Nombre del lenguaje:** Swift.
- **Nombre de los diseñadores:** Chris Lattner, Doug Gregor, John McCall, Ted Kremenek, Joe Groff, y Apple Inc..
- **Año de creación:** 2010.
- **Clasificación del lenguaje de acuerdo al:**
 - a) **Nivel:** Lenguaje de alto nivel.
 - b) **Propósito:** Programación de sistemas, aplicaciones para móviles y de escritorio, llegando a servicios en la nube.
 - c) **Estilo/Paradigma:** Multiparadigma (Orientado a protocolos, objetos, funcional, programación imperativa).
- **Ejemplo de sintaxis y semántica sencillo:**

```
func printMessage(message: String) {  
    println(message)  
}
```

- **Ejemplo de biblioteca:**
- **Ejemplo de convención de programación (idiom):**

```
for i in 1...5 {  
    # Código  
}
```

- **Nombre del lenguaje:** R.
- **Nombre de los diseñadores:** Ross Ihaka y Robert Gentleman.
- **Año de creación:** 1993.
- **Clasificación del lenguaje de acuerdo al:**
 - a) **Nivel:** Lenguaje de alto nivel.
 - b) **Propósito:** Investigación científica, machine learning, minería de datos, investigación biomédica, bioinformática y matemáticas financieras.
 - c) **Estilo/Paradigma:** Orientado a objetos.
- **Ejemplo de sintaxis y semántica sencillo:**

```
while(condicion) {  
    # Código  
}
```

- **Ejemplo de biblioteca:**
- **Ejemplo de convención de programación (idiom):**

```
library(installr)
```

```
for (i in lista) {  
    # Código  
}
```

▪ **Ethan**

- **Nombre del lenguaje:** Rust.
- **Nombre de los diseñadores:** Graydon Hoare, Dave Herman, Brendan Eich.
- **Año de creación:** 2010.
- **Clasificación del lenguaje de acuerdo al:**
 - a) **Nivel:** Lenguaje de medio nivel.
 - b) **Propósito:** Propósito general.
 - c) **Estilo/Paradigma:** Multiparadigma.
- **Ejemplo de sintaxis y semántica sencillo:**
 - a) **Sintaxis:** Comúnmente, para representar una variable de alguno tipo numérico en varios lenguajes se usa la palabra `int` o `float`. En Rust, primero nos debemos de preguntar de cuantos bits queremos nuestro número, o si lo queremos con signo. A continuación se muestra una tabla de todos los tipos de números en Rust:

Tamaño	Con signo	Sin signo
8 bits	i8	u8
16 bits	i16	u16
32 bits	i32	u32
64 bits	i64	u64
128 bits	i128	u128
arch	isize	usize

Entonces, para declarar debemos de usar la palabra clave `let`, seguida de un identificador, el simbolo `:`, el tipo de número, el simbolo `=` y un valor.

- b) **Semántica:** Siguiendo la sintaxis anterior, obtenemos lo siguiente:

```
let n: i32 = 42;
```

El significado dado es que a la variable `n` se le asigna un tipo entero de 32 bits con signo, el cual tiene un valor numérico de 42.

- **Ejemplo de biblioteca:** El equivalente a Rust a una biblioteca sería un `crate`. Un `crate` bastante utilizado es `rand`, que nos permite generar números aleatorios.
- **Ejemplo de convención de programación (idiom):** Rust no tiene constructores como tal, en general, se usa una función estática para crear un nuevo objeto. Por ejemplo:

```
struct Rectangulo {  
    base: u32,  
    altura: u32,  
}
```

```
impl Rectangulo {  
    fn new() -> Rectangulo {  
        Rectangulo {  
            base: 0u32,  
            altura: 0u32,  
        }  
    }  
}
```

- **Nombre del lenguaje:** C++.
- **Nombre del diseñador:** Bjarne Stroustrup.
- **Año de creación:** 1983.
- **Clasificación del lenguaje de acuerdo al:**
 - a) **Nivel:** Lenguaje de medio nivel.
 - b) **Propósito:** Propósito general.
 - c) **Estilo/Paradigma:** Multiparadigma.
- **Ejemplo de sintaxis y semántica sencillo:**
 - a) **Sintaxis:** Para poder imprimir algún texto a la consola, utilizamos `std::cout` seguido del operador `<<` y los caracteres o números como salida. Cabe mencionar que, para utilizar `std::cout`, es necesario incluir la directiva del preprocesador para poder utilizar la biblioteca `iostream`
 - b) **Semántica:** Siguiendo la sintaxis anterior, obtenemos lo siguiente:

```
#include <iostream>  
  
int main()  
{  
    std::cout << "Hola, Mundo!";  
    return 0;  
}
```

- **Ejemplo de biblioteca:** Una de las bibliotecas más usadas es `iostream`, que define a los objetos de entrada y salida estándar.
- **Ejemplo de convención de programación (idiom):** Al igual que en C, en C++ las llaves de una función se escriben en una línea aparte:

```
#include <iostream>  
  
int main()  
{  
    std::cout << "Hola, Mundo!";  
    return 0;  
}
```

- **Nombre del lenguaje:** Java.
- **Nombre del diseñador:** Sun Microsystems.
- **Año de creación:** 1995.
- **Clasificación del lenguaje de acuerdo al:**
 - a) **Nivel:** Lenguaje de alto nivel.
 - b) **Propósito:** Propósito general.
 - c) **Estilo/Paradigma:** Orientación a objetos, crear aplicaciones (empresariales, web y escritorio), permite ejecutar un mismo programa en múltiples sistemas operativos y hace posible ejecutar el código en sistemas remotos de manera segura.
- **Ejemplo de sintaxis y semántica sencillo:**
 - a) **Sintaxis:** Un ejemplo de sintaxis es el siguiente:

```
//Ejemplo en java  
int var = 10;
```

- b) **Semántica:** En este caso usando este mismo ejemplo tenemos una variable de nombre "var" de tipo **int** que se refiere a tipo entero con un valor de **10**.
- **Ejemplo de biblioteca:** Java provee una amplia funcionalidad para crear nuevas aplicaciones. Un ejemplo es **javax.jms**.
- **Ejemplo de convención de programación (idiom):** Uno de los idioms que posee el lenguaje de programación java es los nombres de las clases deben ser sustantivos, en mayúsculas y minúsculas, con la primera letra de cada palabra interna en mayúscula. El nombre de las interfaces también debe estar en mayúscula (la primera) al igual que los nombres de las clases. Use palabras completas y debe evitar acrónimos y abreviaturas. Por ejemplo:

```
public class Lenguaje {  
    // código  
}
```

- **Nombre del lenguaje:** Ruby.
- **Nombre del diseñador:** Yukihiro Matsumoto.
- **Año de creación:** 1995.
- **Clasificación del lenguaje de acuerdo al:**
 - a) **Nivel:** Lenguaje de alto nivel.
 - b) **Propósito:** Principalmente en la programación web a través del framework Ruby on Rails.
 - c) **Estilo/Paradigma:** Orientado a objetos.
- **Ejemplo de sintaxis y semántica sencillo:**
 - a) **Sintaxis:** Un ejemplo de sintaxis es el siguiente:

```
#Ejemplo en ruby  
x = 10;
```

b) **Semántica:** En este caso, debemos de leer esta asignación como el definir una variable de nombre "x" a la que le estamos asignando el valor de 10.

- **Ejemplo de biblioteca:** Las dependencias para las aplicaciones de Ruby se declaran en un archivo Gemfile. Por ejemplo:

```
source "https://rubygems.org"
gem "rails"
```

- **Ejemplo de convención de programación (idiom):** En Ruby, para declarar constantes se utilizan tenemos las siguientes opciones:

`THIS_CONSTANT`, `This_Constant`, `ThisConstant`.

2. Define una función en Racket que calcule la longitud de un número natural, es decir, que calcule el número de dígitos que lo conforman.

El código es el siguiente:

```
(define (digitos n)
  (let ([m (floor (/ (log n) (log 10)))]
        (+ m 1)))
```

3. Define una función en Racket que filtre los números positivos de una lista. Es decir, dada una lista con números enteros, devolver una nueva lista con sólo números positivos.

El código es el siguiente:

```
(define (positivos l)
  (cond
    [(empty? l) l]
    [else (if (< (first l) 0)
              (positivos (rest l))
              (cons (first l) (positivos (rest l))))])
```

Referencias

- [1] R (lenguaje de programación). (n.d.). Wikipedia. Retrieved October 1, 2021, from [https://es.wikipedia.org/wiki/R_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/R_(lenguaje_de_programaci%C3%B3n))
- [2] A. (2021). Statements and the structure of a program. Learncpp. <https://www.learncpp.com/cpp-tutorial/statements-and-the-structure-of-a-program/>
- [3] rust-unoficial. (2021). Constructors. Rust Design Patterns. <https://rust-unofficial.github.io/patterns/idioms/ctor.html>
- [4] The Rust Programming Language - The Rust Programming Language. (2018). The Rust Programming Language. <https://doc.rust-lang.org/stable/book/>

- [5] Ramirez, K. (2021, septiembre). Nota de clase 1. Conceptos generales.
- [6] Swift (lenguaje de programación). (2021, June 6). Wikipedia. [https://es.wikipedia.org/wiki/Swift_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Swift_(lenguaje_de_programaci%C3%B3n))
- [7] Tutorial: Enlace de una biblioteca Swift de iOS. (2020, February 11). Microsoft. <https://docs.microsoft.com/es-es/xamarin/ios/platform/binding-swift/walkthrough>
- [8] Tipos de biblioteca estándar en Swift: NO PRIMITIVOS. (n.d.). ICHL.PRO. Retrieved October 1, 2021, from <https://ichi.pro/es/tipos-de-biblioteca-estandar-en-swift-no-primativos-224372004350148>
- [9] Santana, A., & Nieves Hernández, C. (n.d.). Librerías en R. ULPGC. Retrieved October 1, 2021, from https://estadistica-dma.ulpgc.es/cursosR4ULPGC/5-librerias.html#a%C3%Bladir_paquetes_a_nuestra_instalaci%C3%B3n_de_r
- [10] Capítulo 5 Manejo de la biblioteca y gestión de paquetes. (n.d.). <https://Bookdown.Org>. Retrieved October 1, 2021, from <https://bookdown.org/gboccardo/manual-ED-UCH/manejo-de-la-biblioteca-y-gestion-de-paquetes.html#descargar-paquetes>