

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN
FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y
SERVICIOS
ESCUELA PROFESIONAL DE INGENIERÍA DE
SISTEMAS



INTRODUCCIÓN A LA COMPUTACIÓN (A)

DOCENTE *Richard Smith Escobedo Quispe*

TEMA:

Lenguaje de Programación Ruby

PRESENTADO POR

- *Iván Rodrigo Huayhua Mayta*
- *Anthony Marcos Morocchoire Pacompía*
- *Reni Companocca Checco*

AREQUIPA – PERÚ

2021

EPÍGRAFE

Yukihiro Matsumoto: “Creo que el propósito de la vida es, al menos en parte, ser feliz. Basándose en esta creencia, Ruby está diseñado para hacer que la programación no solo sea fácil sino también divertida. Le permite concentrarse en el lado creativo de la programación, con menos estrés.

Espero ver a Ruby ayudando a todos los programadores del mundo a ser productivos, a disfrutar de la programación y a ser felices. Ese es el propósito principal del lenguaje Ruby.

No trabajé duro para hacer que Ruby fuera perfecta para todos, porque te sientes diferente a mí. Ningún idioma puede ser perfecto para todos. Traté de hacer que Ruby fuera perfecto para mí, pero puede que no sea perfecto para ti.”



MOTIVACIÓN

Una de las mejores características de Ruby es la simplicidad del lenguaje, diseñado para aumentar la productividad y para hacer divertida la programación no necesitas preocuparte de puntos y comas, en Ruby todo es un objeto. Además de ser un lenguaje simple también, es un lenguaje de alto nivel. Esto significa que aprender a programar en Ruby puede ser una gran opción para aquellos que buscan programar fácilmente. Sumado a esto es completamente gratuito, es decir, se puede utilizar, modificar y copiar sin restricciones.

Es de software libre (Open Source) y multiplataforma, se puede descargar totalmente gratis de la página oficial y ejecutarlo en diferentes sistemas operativos.

RESUMEN

Ruby es un lenguaje de programación interpretado, reflexivo y orientado a objetos, creado por el programador japonés Yukihiro Matsumoto. Ruby está diseñado para aumentar la productividad siendo su principal foco de atención la sintaxis simple con la que se maneja.

Ruby es un lenguaje que está orientado a objetos: todos los tipos de datos son un objeto, incluidas las clases y tipos que otros lenguajes definen como primitivas, (como enteros, booleanos, y nil). Toda función es un método. Las variables siempre son referencias a objetos. Combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos similares a Smalltalk. Comparte también funcionalidad con otros lenguajes de programación como Lisp, Lua, Dylan y CLU. Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una licencia de software libre.

Ruby se puede utilizar en muchos campos: desde procesamiento de texto y programación web con CGI, hasta ingeniería, genética, y programación comercial a gran escala.

Uno de los elementos que hace que Ruby sea popular es Ruby On Rails, un Framework que está muy de moda para realizar aplicaciones Web modernas. Rails está escrito en Ruby, por lo tanto, se necesita del mismo para hacer aplicaciones Web.

Es dinámico y flexible. Con estrategias como la meta programación, es decir escribir programas que escriben o manipulen otros programas. Esta flexibilidad permite modificar el código en tiempo de ejecución.

A pesar de la fama de este lenguaje, no cuenta con una comunidad muy grande ello implica que es más complicado conseguir documentación técnica, lo que dificultará el aprendizaje para las personas que recién se inician en este lenguaje.

La curva de aprendizaje de este lenguaje puede ser elevada al principio sin embargo a medida que se conoce más esta disminuirá gradualmente.

EXPECTATIVAS

Las expectativas son principalmente el poder conocer y utilizar las diferentes herramientas y/o estrategias que existen en este lenguaje para desarrollar un aplicación o proyecto, además poder conocer otros lenguajes aparte del brindado por la Universidad nos ayudará a conocer qué otras ventajas tienen los diferentes Lenguajes de Programación (LP), es decir en qué aspectos son de más ayuda.

Según nuestras fuentes de información Ruby es un lenguaje divertido y eso mismo lo hace especial, esperamos que tal afirmación sea correcta, queremos divertirnos mientras aprendemos.

INDICE:

EPÍGRAFE.....	3
MOTIVACION.....	4
RESUMEN.....	5
EXPECTATIVAS.....	6
1. HISTORIA.....	08
2. PARADIGMA.....	08
3. CARACTERÍSTICAS GENERALES	09
4. VENTAJAS DEL USO DEL LENGUAJE DE PROGRAMACIÓN RUBY	09
5. DESVENTAJAS DEL USO DEL LENGUAJE DE PROGRAMACIÓN RUBY	10
6. CURVA DE APRENDIZAJE.....	11
7. APLICACIONES.....	12
8. EJEMPLOS EN EL LENGUAJE RUBY.....	13
9. EJERCICIOS EN EL LENGUAJE RUBY.....	14
10. APLICACIÓN.....	18
11. EXPERIENCIAS VIVIDAS	
11.1. RECOMENDACIONES.....	20
11.2. CONCLUSIONES.....	20
12. ENTREGABLES.....	21
13. REFERENCIAS.....	21

INTRODUCCIÓN AL LENGUAJE DE PROGRAMACIÓN RUBY

1. HISTORIA

El lenguaje fue creado por Yukihiro "Matz" Matsumoto, quien empezó a trabajar en Ruby el 24 de febrero de 1993, y lo presentó al público en el año 1995. Desde su liberación pública en 1995, Ruby ha atraído devotos desarrolladores de todo el mundo. En el 2006, Ruby alcanzó reconocimiento masivo, formándose grupos de usuarios activos en las ciudades más importantes del mundo y llenando las capacidades de las conferencias relacionadas a Ruby. El índice TIOBE, que mide el crecimiento de los lenguajes de programación, ubica a Ruby en la posición #13 del ranking mundial. Refiriéndose a su crecimiento, predicen, "Todo indica que Ruby llegará a estar entre los 10 primeros en menos de 6 meses". Gran parte de su crecimiento se atribuye a la popularidad alcanzada por aplicaciones desarrolladas con Ruby, en particular el Framework de desarrollo web Ruby on Rails.

2. PARADIGMA

- Ruby ha sido descrito como un lenguaje de programación multiparadigma: permite programación procedural (definiendo funciones y variables fuera de las clases haciéndolas parte del objeto raíz Object), con orientación a objetos (todo es un objeto), o funcionalmente (tiene funciones anónimas, clausuras o closures, y continuations; toda la sentencia tiene valores, y las funciones devuelven la última evaluación).
- Soporta introspección, reflexión y metaprogramación, además de soporte para hilos de ejecución gestionados por el intérprete. Ruby tiene tipado dinámico, y soporta polimorfismo de tipos (permite tratar a subclases utilizando la interfaz de la clase padre). Ruby no requiere de polimorfismo de funciones al no ser fuertemente tipado (los parámetros pasados a un método pueden ser de distinta clase en cada llamada a dicho método).

2.1. PARADIGMAS DEL LENGUAJE RUBY

- Reflexivo: Es una característica de algunos lenguajes para determinar y modificar su comportamiento. Ruby no posee la capacidad de controlar estos objetos, pero sí puede controlar estructuras y bloques. Es un lenguaje de programación de alto nivel o procedimental. Es capaz de expresar el algoritmo de una manera correcta a la capacidad cognitiva humana. Puede tratar con variables, objetivos, expresiones booleanas.
- Reflexividad en Ruby:
 - Programación Imperativa Son un conjunto de instrucciones que le indican al computador cómo realizar una tarea. Esto quiere decir que se describen sentencias que modifican el estado de un programa.
 - Programación Orientada a Objetos Es la utilización de funciones aritméticas que no maneja datos mutables o de estado.
- Paradigma de Programación Funcional:

- Los lenguajes de programación funcional han sido enfatizados en el ambiente académico principalmente y no tanto en el desarrollo de software comercial. También es utilizada en la industria a través de lenguajes de dominio específico como R (estadística), Mathematica (matemáticas simbólicas), J y K (análisis financiero).

3. CARACTERÍSTICAS GENERALES

- Orientado a objetos
- Cuatro niveles de ámbito de variable: global, clase, instancia y local.
- Manejo de excepciones
- Iteradores y clausuras o closures (pasando bloques de código)
- Expresiones regulares nativas similares a las de Perl a nivel del lenguaje
- Posibilidad de redefinir los operadores (sobrecarga de operadores)
- Recolección de basura automática
- Altamente portable
- Hilos de ejecución simultáneos en todas las plataformas usando green threads
- Carga dinámica de DLL/bibliotecas compartidas en la mayoría de las plataformas.
- Introspección, reflexión y metaprogramación
- Amplia librería estándar
- Soporta inyección de dependencias
- Soporta alteración de objetos en tiempo de ejecución
- Continuaciones y generadores

4. VENTAJAS DEL USO DEL LENGUAJE PROGRAMACIÓN RUBY

- Es código libre y no debes preocuparte de pagar licencias por su uso, e incluso es gratis y tampoco tienes que tener dinero para costear la descarga desde la web. De esta manera puedes utilizar, copiar, modificar y distribuir este lenguaje de programación sin preocuparte por las leyes nacionales e internacionales.
- Te permite programar en la web viendo todo como un objeto, ya que su estructura y sintaxis es orientada a objetos. De esta manera puedes asignar propiedades y acciones a cualquier forma de código que escribas usando a Ruby como base. Esta es una excelente forma de usar las variables de instancias y clases, establecer métodos para cualquier proyecto online que se te ocurra, sin importar si usas números y otros tipos primitivos.
- Cuenta con flexibilidad para modificarlo a tu gusto, ya que te permite eliminar o redefinir cualquier parte del código esencial de Ruby. Esto significa que tienes acceso a lo que necesites para adicionar funciones extras a las formas de código que ya existen, de modo que no te limites a programar. Esto último lo puedes hacer hasta el punto de modificar el operador “+” para sumar por la palabra “suma” con tan sólo adicionar un método a la clase Numeric.

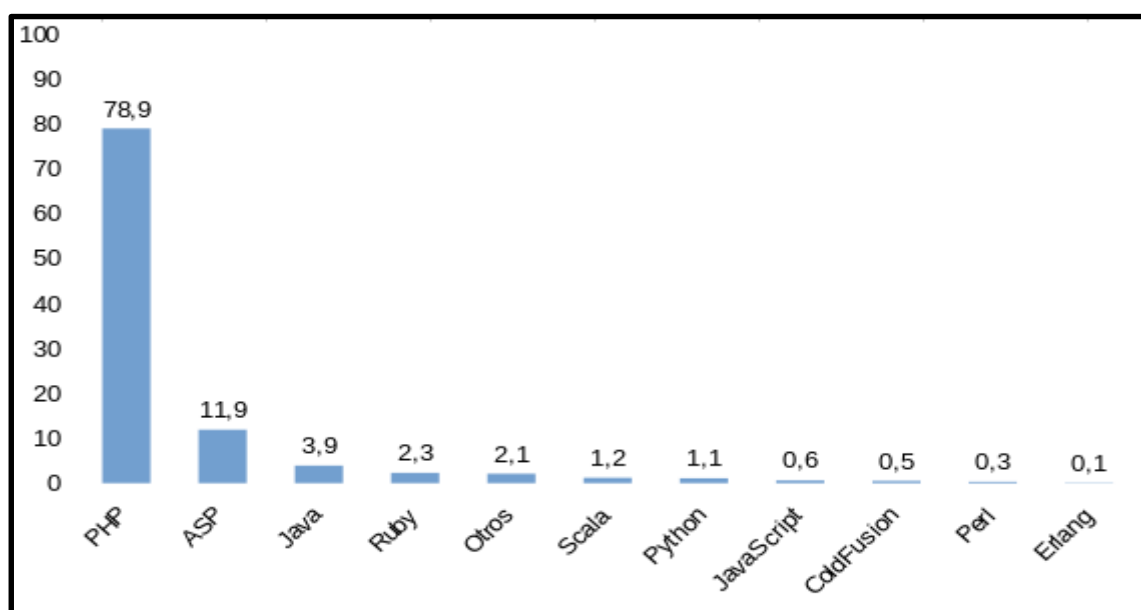
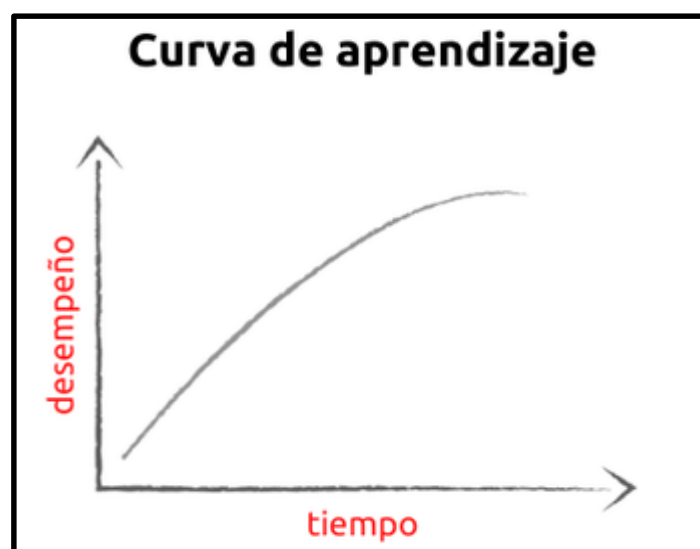
- Te ofrece bloques para que agregues cláusulas a cualquier método dentro del código de programación. Para muchos desarrolladores, tanto novatos como experimentados, se trata de una funcionalidad que aporta flexibilidad. Esto último te permite describir de manera natural como debe actuar cada fragmento del lenguaje que esté relacionado con aquellos. Además, estos últimos están inspirados en lenguajes funcionales como Lisp o Perl.
- Utiliza convenciones simples para determinar y mencionar a las variables, lo que mejora su legibilidad frente a otros lenguajes.

5. DESVENTAJAS DEL USO DEL LENGUAJE DE PROGRAMACIÓN RUBY

- No cuenta con comunidades y soporte de atención al cliente tan amplio como otros lenguajes de programación. Esto es un punto negativo ya que si estás empezando con la programación web y apenas usas Ruby vas a necesitar libros y documentación técnica a la que no puedes tener fácil acceso. Ello comparado con otros códigos tales como PHP y C#, los cuales tienen amplia comunidad y ayuda extendida online, incluyendo muchos más recursos literarios.
- Tim Bray, desarrollador web, ha mencionado en reiteradas oportunidades que el lenguaje de programación Ruby es más lento en cuanto al procesamiento se refiere. Además, esto mismo se ha comprobado a través de varios sitios web que miden la respuesta de los códigos informáticos. Esto implica que necesitas más tiempo de trabajo para comprobar el funcionamiento de tu proyecto.
- En vista de ser nuevo en el campo de los lenguajes de programación web, requieren mayor tiempo para que te familiarices con las estructuras y su funcionamiento general. Además, tiene su propio código de codificación, lo que significa que la curva de aprendizaje es mayor comparado con otros como PHP o Perl.
- Las actualizaciones son más lentas comparadas con otros lenguajes de programación como PHP, que una vez creado se ha actualizado para corregir errores varias veces al año. Esto te ocasiona retrasos al desarrollar un proyecto web si esperas que corrijan aspectos de seguridad o funcionalidad que sean esenciales para que tu programa funcione correctamente.
- No encuentras muchos framework creados con Ruby, lo que limita un poco el desarrollo de aplicaciones para la web. Sin embargo, una de las mejores es Ruby on Rails, muy aceptada por la comunidad de desarrolladores ya que favorece la convención antes que la configuración.

6. CURVA DE APRENDIZAJE

Con Ruby podemos desarrollar todo tipo de aplicaciones diferentes, desde aplicaciones de servicio web, clientes de correo electrónico, procesamiento de datos, aplicaciones de red, y cualquier cosa que se nos ocurra podemos hacer utilizando Ruby. Y por lo tanto necesitamos un intérprete de Ruby, que evalúe nuestro código y lo traduzca a un lenguaje máquina entendible por un ordenador, pero necesita un proceso previo de compilación. Ruby es un lenguaje dinámico y flexible con estrategias como la meta programación incluso podemos utilizar el propio lenguaje Ruby para alterar el código en tiempo de ejecución; además, es un lenguaje de alto nivel esto es porque está lejos del lenguaje máquina y es muy cercano incluso a inglés, a veces parecerá que estamos programando en inglés. Es de software libre y multiplataforma, podemos descargarlo totalmente gratis y ejecutarlo en distintos sistemas operativos.



7. APLICACIONES

- Twitter. Quizá sea una “trampa” en la lista, pero había que mencionar a esta enorme red social. Millones de usuarios de todo el mundo la utilizan a diario. En sus inicios, se creó sobre Ruby, aunque a día de hoy cambió a Scala para gestionar mejor la gran cantidad de mensajes en tiempo real.
- Airbnb. Se trata de una empresa y una plataforma de software dedicada a la oferta de alojamientos a particulares y turísticos. El nombre es un acrónimo de airbed and breakfast (colchón inflable y desayuno). Airbnb tiene una oferta de más de 2.000.000 propiedades en 192 países y en más de 33.000 ciudades.
- Hulu. Es un sitio web de pago, apoyado por publicidad, streaming de programas de televisión y películas de NBC, FOX, y muchas otras redes y estudios. Actualmente sólo se ofrecen a los usuarios en los Estados Unidos. Es una plataforma muy parecida a Netflix o HBO.
- Groupon. El gigante de los descuentos a nivel mundial. Lanzado en noviembre de 2008 en Chicago, su base está realizada en Ruby. Millones de personas de todo el mundo se benefician a diario de los descuentos en la web.
- Bloomberg. Fundada en 1981, es una de las mayores corporaciones del mundo dedicada al mundo de las finanzas, con su propia agencia de noticias y software bursátil y con más de 15.000 empleados, una de las empresas más grandes que recurre a Ruby.
- Soundcloud. El YouTube del sonido. La plataforma sueca gestiona 175 millones de visitas únicas al mes y la subida de doce horas de audio por minuto, y se ha convertido en una referencia tanto para los usuarios como para los creadores. Originalmente fue escrita en Ruby, pero al igual que Twitter más tarde adoptó Scala para ciertas partes de la aplicación, aunque R sigue presente en el código de SoundCloud.
- Github. No podíamos olvidarnos de la mayor red social de desarrolladores a nivel mundial. El repositorio de código de Github está construido sobre Ruby, tiene más de 10 millones de usuarios y almacena más de 25 millones de repositorios.

8. EJEMPLOS EN EL LENGUAJE RUBY:

```
irb(main):002:0> #Escribir oración
=> nil
irb(main):003:0> puts "Hola Mundo"
Hola Mundo
=> nil
```

```
irb(main):001:0> año_actual = 2021 #Es el año actual
=> 2021
irb(main):002:0> "¿En que año naciste?"
=> "¿En que año naciste?"
irb(main):003:0> año_nacimiento = 2001
=> 2001
irb(main):004:0> edad = año_actual - año_nacimiento
=> 20
irb(main):005:0> puts "Tienes #{edad} años"
Tienes 20 años
=> nil
```

```
irb(main):004:0* #Imprimir números del 0 al 9
irb(main):005:0> var=0
=> 0
irb(main):006:1* while var< 10
irb(main):007:1*   puts var
irb(main):008:1*   var += 1
irb(main):009:0> end
0
1
2
3
4
5
6
7
8
9
=> nil
```

9. EJERCICIOS EN EL LENGUAJE RUBY:

1. Básicos:

1)

```
1  cadena1 = "Esta es una cadena de caracteres"
2  cadena2 = "y esta otra cadena"
3
4  #Se muestra la cadena1 y la cadena2 concatenadas
5  puts cadena1 + " " + cadena2
```

```
> bundle exec ruby main.rb
Esta es una cadena de caracteres y esta otra cadena
> []
```

2)

```
1  print "Ingrese su nombre: "
2
3  #Se lee el nombre del usuario y se imprime en pantalla
4  name = gets.chomp
5  puts "Hola "+name
6
7  #Cantidad de caracteres del nombre
8  print "Tu nombre tiene ", name.length, " letras"
9  puts "\n"
10 print "Cuántos años tienes?: "
11
12 #Se convierte el string a entero para posteriormente ser
13 #multiplicado por la la cantidad de caracteres
14
15 age = gets.chomp.to_i
16 puts "El producto es: ", age*name.length
17
18
```

```
> bundle exec ruby main.rb
Ingrese su nombre: Anthony
Hola Anthony
Tu nombre tiene 7 letras
Cuántos años tienes?: 56
El producto es:
392
> []
```

2. Intermedios

1)

```
irb(main):001:1* def promedio(arr) #hallaremos el promedio de cierta cantidad de alumnos
irb(main):002:1*   suma = 0
irb(main):003:2*   arr.each do |num|
irb(main):004:2*     suma += num
irb(main):005:1*   end
irb(main):006:1*   suma/arr.length.to_f
irb(main):007:0> end
=> :promedio
irb(main):008:0> puts promedio([12, 15, 17, 19]) #Hallamos el promedio de 4 estudiantes
15.75
=> nil
```

2)

```
irb(main):001:1* def ordenar(arr)
irb(main):002:2*   loop do
irb(main):003:2*     intercambiar = false
irb(main):004:3*     for i in 0...arr.length-1
irb(main):005:4*       if arr[i] > arr[i+1]
irb(main):006:4*         arr[i], arr[i+1] = arr[i+1], arr[i]
irb(main):007:4*         intercambiar = true
irb(main):008:3*       end
irb(main):009:2*     end
irb(main):010:2*     break unless intercambiar
irb(main):011:2*   end
irb(main):012:1* end
irb(main):013:1* arr
irb(main):014:1*   arr
irb(main):015:0> end
=> :ordenar
irb(main):016:0> puts ordenar([3, 4, 5, 1, 2, 7, 8, 9]).inspect
[1, 2, 3, 4, 5, 7, 8, 9]
=> nil
```

3. Avanzados:

1)

```
#Programa utilizando el lenguaje de programación Ruby
#Programa que evalúa la ecuación de segundo grado
print "\nBienvenido al Lenguaje Ruby\n"
print "=====\n"
print "\nEvaluando la ecuacion de segundo grado\n"
print "Valor de la constante A:"
a=gets().to_i
if a!=0 then
  print "Valor de la constante B:"
  b=gets().to_i
  print "Valor de la constante C:"
  c=gets().to_i
  print "\nLa ecuacion es = ",a,"X^",2," + ",b,"X + ",c,"\n"
  d=b*b-4*a*c
  if d<0 then
    print "Tiene raiz imaginaria"
  else
    x1=(-b+Math.sqrt(d))/(2*a)
    x2=(-b-Math.sqrt(d))/(2*a)
    puts "X1="+x1.to_s
    print "X2="+x2.to_s
  end
else
  print "El valor de A debe ser diferente de 0"
end
gets()
```

```
> bundle exec ruby main.rb
[]
Bienvenido al Lenguaje Ruby
=====

Evaluando la ecuacion de segundo grado
Valor de la constante A:8
Valor de la constante B:10
Valor de la constante C:-25

La ecuacion es = 8X^2 + 10X + -25
X1=1.25
X2=-2.5
```

```
> bundle exec ruby main.rb
[]
Bienvenido al Lenguaje Ruby
=====

Evaluando la ecuacion de segundo grado
Valor de la constante A:9
Valor de la constante B:38
Valor de la constante C:1

La ecuacion es = 9X^2 + 38X + 1
X1=-0.02648188452292012
X2=-4.195740337699302
```

```
> bundle exec ruby main.rb
[]
Bienvenido al Lenguaje Ruby
=====

Evaluando la ecuacion de segundo grado
Valor de la constante A:0
El valor de A debe ser diferente de 0
```


2)

```
irb(main):001:1* def solucion(matriz_uno, matriz_dos, nf, nc)
irb(main):002:2*   if(nf == 0) && (nc == 0)
irb(main):003:2*     return matriz_uno[nf][nc] + matriz_dos[nf][nc]
irb(main):004:2*   else
irb(main):005:3*     if(nf > -1)
irb(main):006:3*       nc = nc - 1
irb(main):007:4*       if(nc >= -1)
irb(main):008:4*         return matriz_uno[nf][nc+1]+matriz_dos[nf][nc+1]+solucion(matriz_uno, matriz_dos, nf, nc)
irb(main):009:4*       else
irb(main):010:4*         return solucion(matriz_uno, matriz_dos, nf-1, 2)
irb(main):011:3*     end
irb(main):012:2*   end
irb(main):013:1* end
irb(main):014:0> end
=> :solucion
irb(main):015:0> matriz_uno = Array.new(3,0) {Array.new(3,0)}
(irb):15: warning: block supersedes default value argument
=> [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
irb(main):016:0> matriz_dos = Array.new(3,0) {Array.new(3,0)}
(irb):16: warning: block supersedes default value argument
=> [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
irb(main):017:1* for i in 0..2
irb(main):018:2*   for j in 0..2
irb(main):019:2*     matriz_uno[i][j] = rand(1001)
irb(main):020:2*     matriz_dos[i][j] = rand(1001)
irb(main):021:1*   end
irb(main):022:0> end
=> 0..2
irb(main):023:1* for i in 0..2
irb(main):024:2*   for j in 0..2
irb(main):025:2*     print matriz_uno[i][j], "\t"
irb(main):026:1*   end
irb(main):027:1*   print "\n"
irb(main):028:0> end
358      647      889
734      898      462
544      198      385
=> 0..2
irb(main):029:0> print "\n"

=> nil
irb(main):030:1* for i in 0..2
irb(main):031:2*   for j in 0..2
irb(main):032:2*     print matriz_dos[i][j], "\t"
irb(main):033:1*   end
irb(main):034:1*   print "\n"
irb(main):035:0> end
112      70      862
276      121     605
446      104     351
=> 0..2
```

10. APLICACIÓN:

CARRITO DE COMPRAS

```
irb(main):001:0> puts "CARRITO DE COMPRAS " #programa que ayude en las compras
CARRITO DE COMPRAS
=> nil
irb(main):002:1* def menuPrincipal() #creamos una opcion para capturar datos y retornarlo
irb(main):003:1*   puts "MENU PRINCIPAL" #Ahora comenzamos a imprimir las opciones
irb(main):004:1*   puts "    1. REALIZAR COMPRA "
irb(main):005:1*   puts "    2. DEVOLVER COMPRA "
irb(main):006:1*   puts "    3. ACTUALIZAR INVENTARIO "
irb(main):007:1*   puts "    4. SALIR "
irb(main):008:1*   op=gets.to_i #declaramos en variable donde vamos a capturar lo que se ingrese por el teclado
irb(main):009:1*   return op #retornamos un numero
irb(main):010:0> end
=> :menuPrincipal
irb(main):011:1* def cls() #creamos una funcion que nos permite limpiar la pantalla
irb(main):012:1*   system("cls") # este nos permite limpiar la pantalla cada que vez que llamamos este metodo
irb(main):013:0> end
=> :cls
irb(main):014:1* def menuCompra() #creamos el menu 2
irb(main):015:1*   puts "MENU COMPRA"
irb(main):016:1*   puts "    1.  SELECCIONAR PRODUCTO "
irb(main):017:1*   puts "    2.  SELECCIONAR MARCA "
irb(main):018:1*   puts "    3.  SELECCIONAR PRECIO "
irb(main):019:1*   puts "    4.  SALIR "
irb(main):020:1*   op=gets.to_i
irb(main):021:1*   return op
irb(main):022:0> end
=> :menuCompra
irb(main):023:1* def addCart() #definimos una funcion donde todo esta relacionado al carrito
irb(main):024:2*   begin #do while creamos opciones
irb(main):025:2*     opcion=menuCompra() #Creamos una variable para que retorne cuando llamamos una funcion
irb(main):026:3*     case opcion
irb(main):027:3*     when 1
irb(main):028:3*       cls #lo pondremos esta funcion al inicio para que vaya limpiando la pantalla
irb(main):029:3*       puts "opcion compra 1" #colocamos un mensaje cuando llamamos las opciones
irb(main):030:3*     when 2
irb(main):031:3*       cls
irb(main):032:3*       puts "opcion compra 2"
irb(main):033:3*     when 3
irb(main):034:3*       cls
irb(main):035:3*       puts "opcion compra 3"
irb(main):036:3*     when 4
irb(main):037:3*       cls
irb(main):038:3*       puts "opcion compra 4"
irb(main):039:2*     end
```

```

irb(main):040:1*      end while opcion>=1 and opcion<5 #para que se repita todo lo que esta dentro del begin
irb(main):041:0> end
=> :addCart
irb(main):042:1* begin
irb(main):043:2*   opcion=menuPrincipal()
irb(main):044:2*   case opcion
irb(main):045:2*   when 1
irb(main):046:2*     cls
irb(main):047:2*     addCart
irb(main):048:2*   when 2
irb(main):049:2*     cls
irb(main):050:2*     puts "opcion 2"
irb(main):051:2*   when 3
irb(main):052:2*     cls
irb(main):053:2*     puts "opcion 3"
irb(main):054:2*   when 4
irb(main):055:2*     cls
irb(main):056:2*     puts "opcion 4"
irb(main):057:1*   end
irb(main):058:0> end while opcion>=1 and opcion<5

```

```

MENU COMPRA
1. SELECCIONAR PRODUCTO
2. SELECCIONAR MARCA
3. SELECCIONAR PRECIO
4. SALIR

MENU PRINCIPAL
1. REALIZAR COMPRA
2. DEVOLVER COMPRA
3. ACTUALIZAR INVENTARIO
4. SALIR

```

11. EXPERIENCIAS VIVIDAS:

1. RECOMENDACIONES:

- a. Su instalación es algo dificultoso dado que aparte de descargar el programa de la página oficial, se le tiene que añadir 2 programas extras los cuales son el NODE.Js y el YARN, además a parte de la culminación de la instalación, habrá dificultades para su uso, como el que usará como interfaz el CMD, o el uso del IRB en versiones recientes.
Pero a pesar de esas dificultades, podemos usar IDE´S en línea como el reptil.com, entre otras páginas web.
- b. Ruby, como lenguaje de programación, presenta grandes ventajas: combina característica de lenguajes dinámicos, pero incorpora soluciones típicas de lenguajes estáticos. Ruby es un lenguaje apto para aquellos que les gusta realizar aplicaciones sin escribir líneas interminables de código, es decir, Ruby es un lenguaje que apuesta por la productividad.
- c. Una cosa mala del lenguaje es que para poder manejar ciertas características de la framework Ruby on Rails es necesario aprender PHP, no es necesario que se domine todo el lenguaje de programación, sólo hace falta un conocimiento básico. Sin embargo, si no dominas PHP, el proceso de desarrollo en Ruby on Rails se dificulta aún más ya que tu proceso de aprendizaje de esta framework será más lento.

2. CONCLUSIONES:

- a. Al principio se aprende rápidamente gracias a manuales subidos a la red, pero donde se te enseña a crear lo más básico, como crear un blog o un recetario. Creo que el aprendizaje de Ruby On Rails se complica a medida que intentas compilar tus aplicaciones y crear cosas nuevas debido a la falta de información y programadores conocidos a los que poder preguntar dudas.
- b. Aun así, considero que es un lenguaje con un gran futuro, en el que la inmensa mayoría programará próximamente a medida que los programadores vayan conociendo su potencialidad. Así es, que debido a su estabilidad y características ha surgido ya una prometedora comunidad a su alrededor. Ya en el ranking TIOBE TOP 20, que mide popularidad de lenguajes de programación, indica un claro ascenso para Ruby, pero se encuentra actualmente muy por debajo de PHP, y aún por debajo de Python y Perl, lenguajes con más antigüedad que Ruby.

12. ENTREGABLES:

GitHub:

<https://github.com/123ihuayhua/TIF-IC-A-RUBY/tree/main>

Exposición videos:

Introducción:

<https://drive.google.com/file/d/1Yn2caLvZIS0y4iSRof2L2EIY8DEns7sk/view?usp=sharing>

Ejercicios:

<https://drive.google.com/file/d/1gbZZW3MLd73TqiFfWN83zURTDZr2XDue/view?usp=sharing>

Aplicación:

<https://drive.google.com/file/d/1vI8Jrk5YqDkthf7VleFS002QzxyIbN3C/view?usp=sharing>

13. REFERENCIAS

<https://openwebinars.net/blog/que-es-ruby/>

<https://es.wikipedia.org/wiki/Ruby>

<https://www.hostgator.mx/blog/lenguaje-de-programacion-ruby/>

<https://replit.com/languages/ruby>

<https://www.ruby-lang.org/es/downloads/>

<https://www.youtube.com/watch?v=R3MhA8vWGsg>

<https://www.youtube.com/watch?v=oWoyqlmrUmk>

<https://www.youtube.com/watch?v=wSNTT8MLI90>

<https://www.crehana.com/pe/blog/desarrollo-web/ruby-lenguaje-programacion/>

<https://openwebinars.net/blog/que-es-ruby/>

<https://rubyonrails.org/>

<https://cloud.google.com/ruby?hl=es>

<https://www.digital55.com/desarrollo-tecnologia/ruby-on-rails-desarrollo-software/>

