

PROGRAMACIÓN LÓGICA Y FUNCIONAL (UNIDAD 1)



Título de la actividad:

Realización de los programas correspondientes a cada paradigma de programación abordado mediante una bitácora de ejercicios.

Nombre del alumno:

Br. Luis Adrián Balam Espadas.

Carrera, semestre y grupo:

Ingeniería en Sistemas Computacionales.
(8vo Semestre, Grupo "A").

Número de matrícula:

16070008.

Actividad propuesta:

Realizar un programa con un lenguaje basado en cada paradigma de programación: Procedural, Orientado a Objetos, Funcional y Lógico.

Nombre del profesor:

M.M.M.D. José Leonel Pech May.

Fecha de entrega:

Jueves, 13 de febrero de 2020.

Nombre del curso:

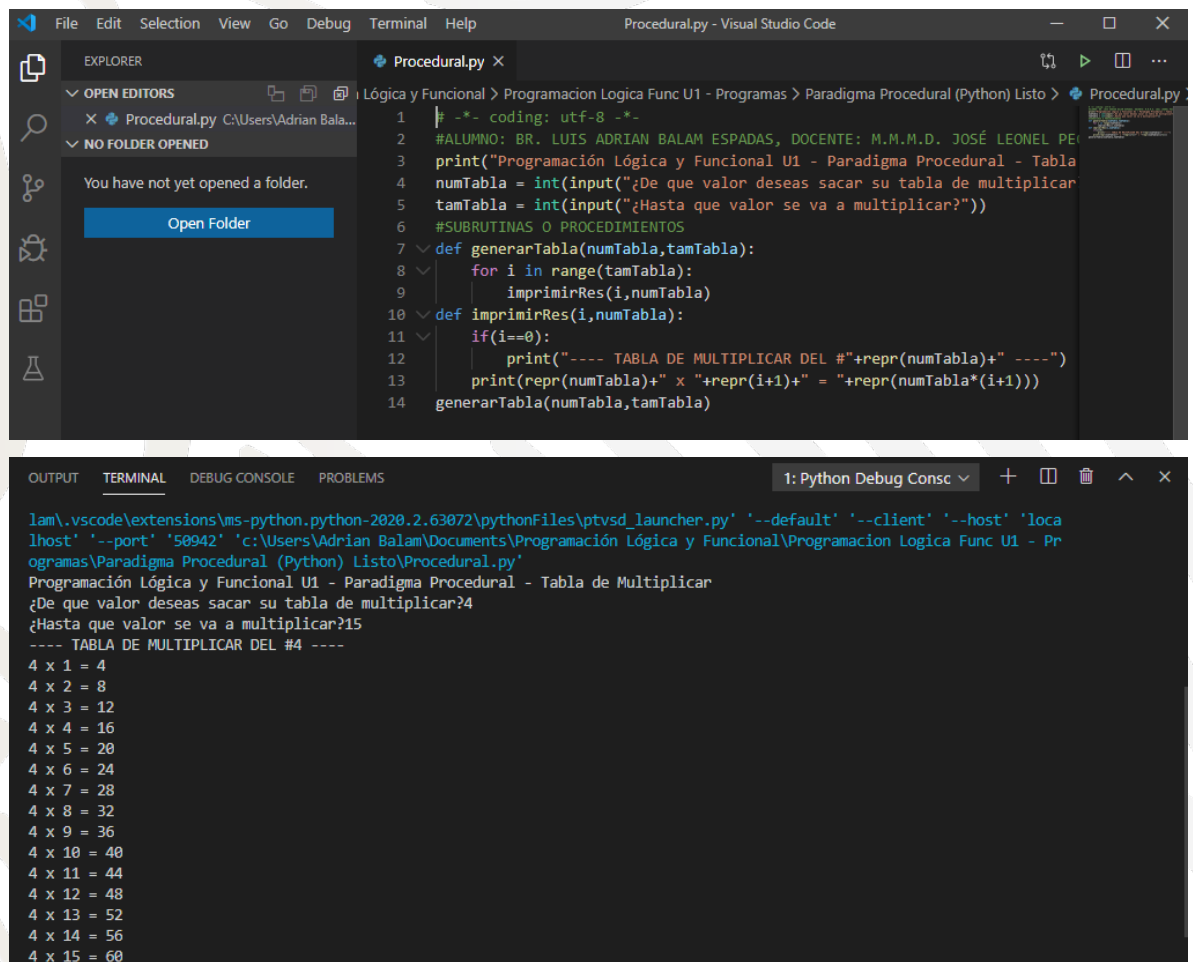
Programación Lógica y Funcional.

DESARROLLO.

BITÁCORA DE EJERCICIOS. Programas.

1.- Tablas de multiplicar (basado en el paradigma procedural, Python).

Para el caso del primer programa realizado, se trata de un algoritmo que permite definir dos valores numéricos, el primero es para el número y el último define hasta que valor se va a multiplicar, formando así una tabla de multiplicar con el uso de subrutinas o procedimientos.



```
Procedural.py - Visual Studio Code

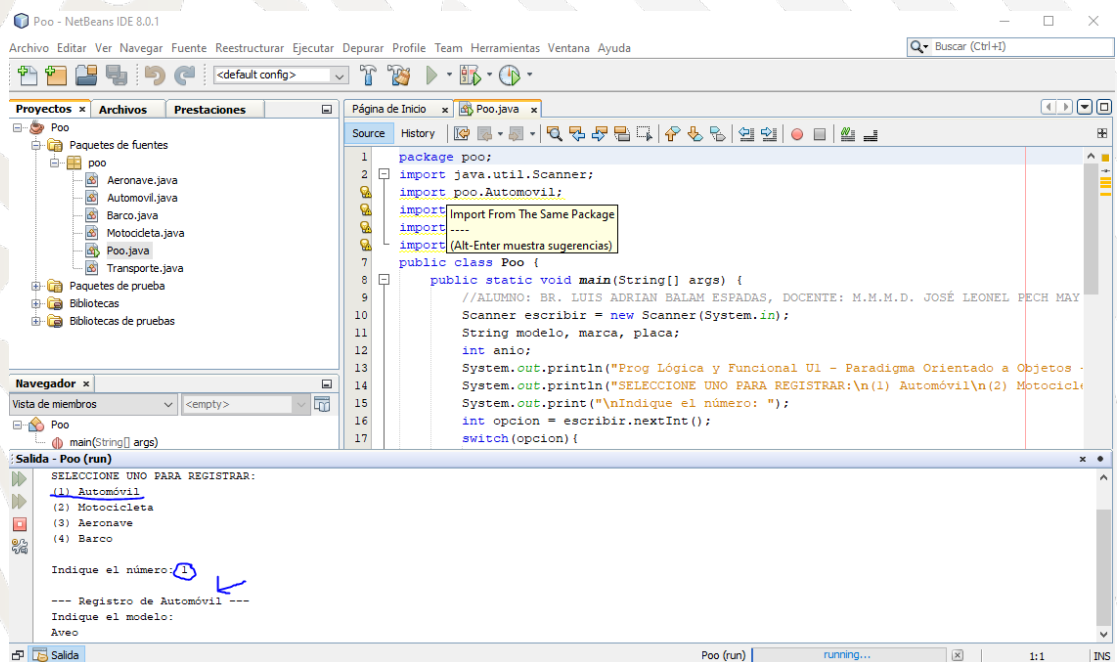
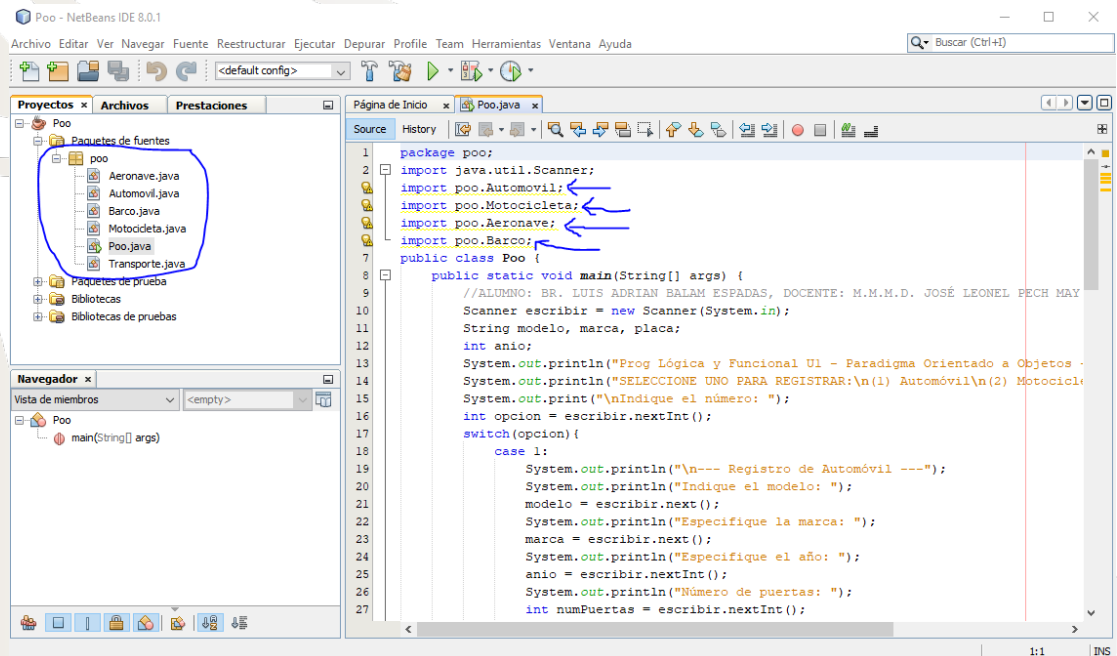
EXPLORER
  OPEN EDITORS
    Procedural.py C:\Users\Adrian Bala...
  NO FOLDER OPENED
    You have not yet opened a folder.
    Open Folder

Procedural.py
1  # -*- coding: utf-8 -*-
2  #ALUMNO: BR. LUIS ADRIAN BALAM ESPADAS, DOCENTE: M.M.M.D. JOSÉ LEONEL PE
3  print("Programación Lógica y Funcional U1 - Paradigma Procedural - Tabla
4  numTabla = int(input("¿De que valor deseas sacar su tabla de multiplicar
5  tamTabla = int(input("¿Hasta que valor se va a multiplicar?"))
6  #SUBROUTINAS O PROCEDIMIENTOS
7  def generarTabla(numTabla,tamTabla):
8      for i in range(tamTabla):
9          imprimirRes(i,numTabla)
10 def imprimirRes(i,numTabla):
11     if(i==0):
12         print("---- TABLA DE MULTIPLICAR DEL #"+repr(numTabla)+" ----")
13         print(repr(numTabla)+" x "+repr(i+1)+" = "+repr(numTabla*(i+1)))
14     generarTabla(numTabla,tamTabla)

OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS
1: Python Debug Consc
lam\.vscode\extensions\ms-python.python-2020.2.63072\pythonFiles\ptvsd_launcher.py' '--default' '--client' '--host' 'loca
lhost' '--port' '50942' 'c:\Users\Adrian Balam\Documents\Programación Lógica y Funcional\Programacion Logica Func U1 - Pr
ogramas\Paradigma Procedural (Python) Listo\Procedural.py'
Programación Lógica y Funcional U1 - Paradigma Procedural - Tabla de Multiplicar
¿De que valor deseas sacar su tabla de multiplicar?4
¿Hasta que valor se va a multiplicar?15
---- TABLA DE MULTIPLICAR DEL #4 ----
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
4 x 11 = 44
4 x 12 = 48
4 x 13 = 52
4 x 14 = 56
4 x 15 = 60
```

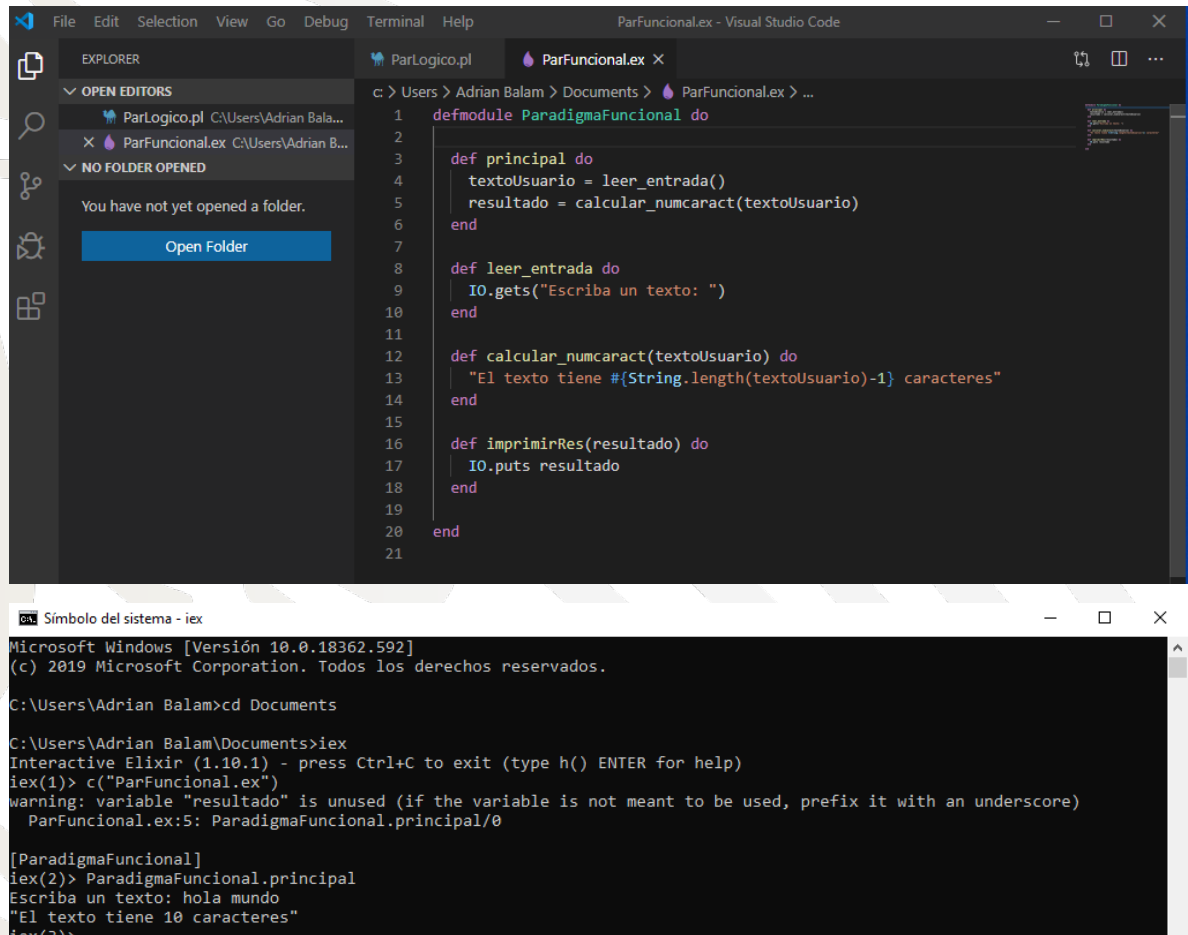
2.- Registro de transporte (basado en el paradigma orientado a objetos, Java).

En cuanto al paradigma orientado a objetos, se ha elaborado un programa que se encarga de registrar distintos tipos de transporte, haciendo pleno uso de la herencia entre clases, la instanciación y creación de objetos con sus atributos y métodos correspondientes.



3.- Contador de caracteres (basado en el paradigma funcional, Elixir).

Se ha llevado a cabo la construcción de un programa con el lenguaje de programación Elixir que sea capaz de contar el número de caracteres que tiene una entrada hecha por el usuario de texto. Como es de esperarse, la lógica se basa en el concepto llamado “función” puramente.



```
File Edit Selection View Go Debug Terminal Help ParFuncional.ex - Visual Studio Code
EXPLORER
OPEN EDITORS
ParLogico.pl C:\Users\Adrian Bala...
ParFuncional.ex C:\Users\Adrian B...
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder

ParFuncional.ex
1 defmodule ParadigmaFuncional do
2
3   def principal do
4     textoUsuario = leer_entrada()
5     resultado = calcular_numcaract(textoUsuario)
6   end
7
8   def leer_entrada do
9     IO.puts("Escriba un texto: ")
10  end
11
12  def calcular_numcaract(textoUsuario) do
13    "El texto tiene #{String.length(textoUsuario)-1} caracteres"
14  end
15
16  def imprimirRes(resultado) do
17    IO.puts resultado
18  end
19
20 end
21

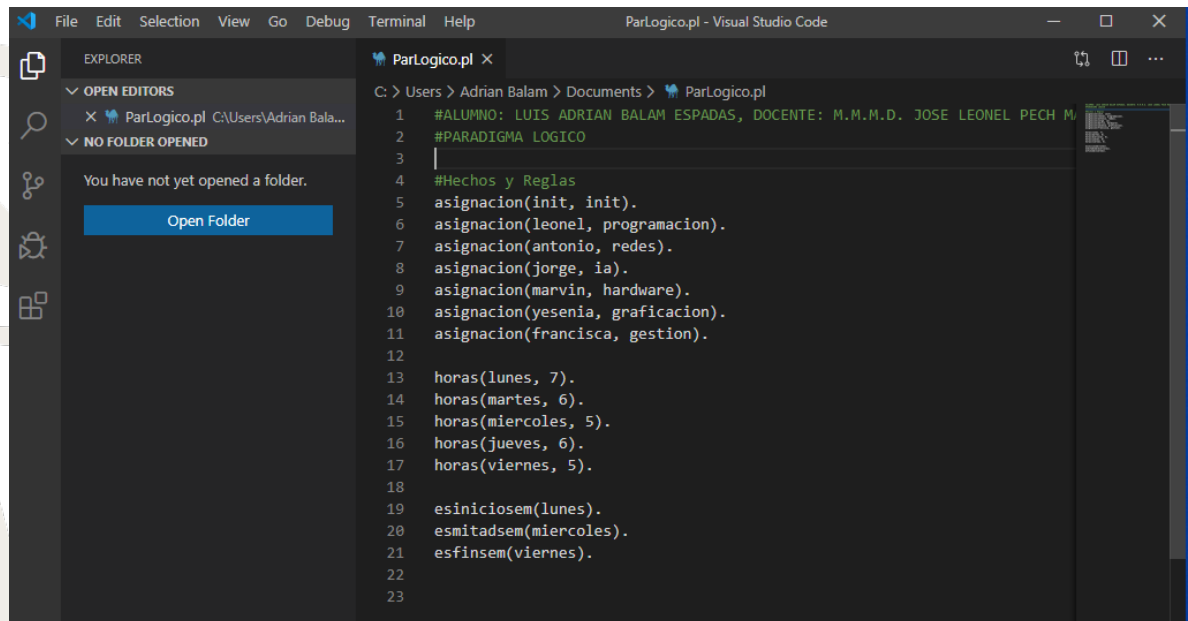
Simbolo del sistema - iex
Microsoft Windows [Versión 10.0.18362.592]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.
C:\Users\Adrian Balam>cd Documents
C:\Users\Adrian Balam\Documents>iex
Interactive Elixir (1.10.1) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> c("ParFuncional.ex")
warning: variable "resultado" is unused (if the variable is not meant to be used, prefix it with an underscore)
ParFuncional.ex:5: ParadigmaFuncional.principal/0

[ParadigmaFuncional]
iex(2)> ParadigmaFuncional.principal
Escriba un texto: hola mundo
"El texto tiene 10 caracteres"
iex(3)>
```

4.- Asignación de materias, horas por día e identificación de partes de la semana escolar (basado en el paradigma lógico, Prolog).

Para el caso del último programa realizado, se empleó el uso del lenguaje de programación Prolog para la especificación de hechos y reglas que caracterizan a la asignación de materias del 8vo semestre, definiendo también las horas de clase por día y estableciendo reglas que permiten saber

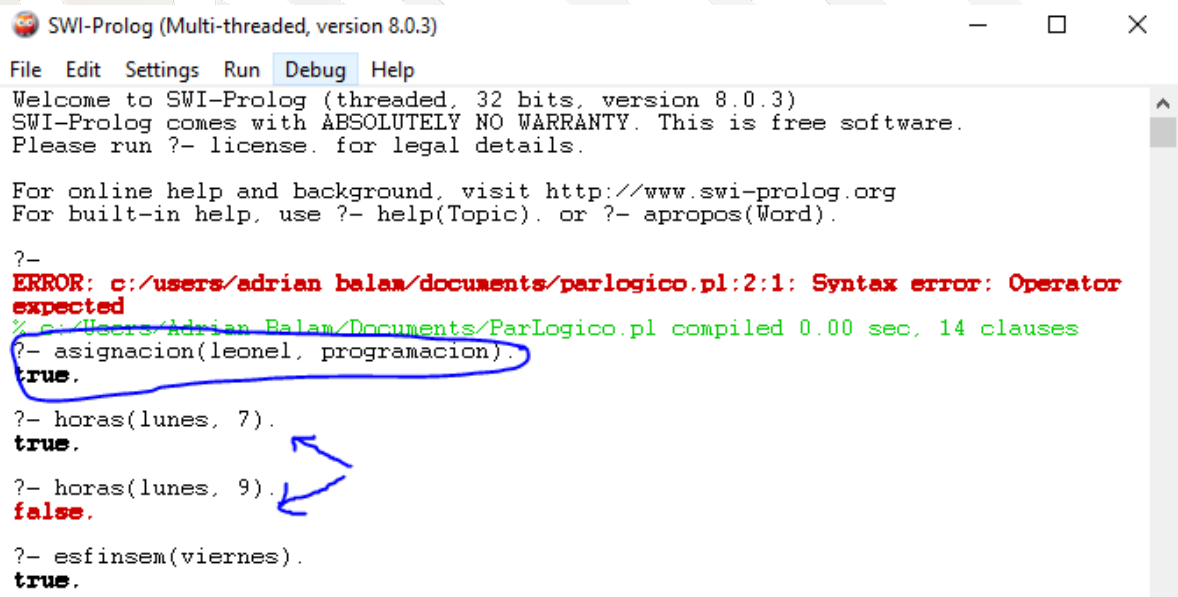
si un determinado día es o no inicio de semana, mitad de semana o fin de semana.



```
File Edit Selection View Go Debug Terminal Help
ParLogico.pl - Visual Studio Code

EXPLORER
OPEN EDITORS
ParLogico.pl C:\Users\Adrian Bala...
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder

ParLogico.pl X
C:\Users\Adrian Balam\Documents> ParLogico.pl
1 #ALUMNO: LUIS ADRIAN BALAM ESPADAS, DOCENTE: M.M.M.D. JOSE LEONEL PECH M.
2 #PARADIGMA LOGICO
3
4 #Hechos y Reglas
5 asignacion(init, init).
6 asignacion(leonel, programacion).
7 asignacion(antonio, redes).
8 asignacion(jorge, ia).
9 asignacion(marvin, hardware).
10 asignacion(yesenia, graficacion).
11 asignacion(francisca, gestion).
12
13 horas(lunes, 7).
14 horas(martes, 6).
15 horas(miercoles, 5).
16 horas(jueves, 6).
17 horas(viernes, 5).
18
19 esiniciosem(lunes).
20 esmitadsem(miercoles).
21 esfinsem(viernes).
22
23
```



```
SWI-Prolog (Multi-threaded, version 8.0.3)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 32 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
ERROR: c:/users/adrian balam/documents/parlogico.pl:2:1: Syntax error: Operator expected
% c:/Users/Adrian Balam/Documents/ParLogico.pl compiled 0.00 sec, 14 clauses
?- asignacion(leonel, programacion).
true.
?- horas(lunes, 7).
true.
?- horas(lunes, 9).
false.
?- esfinsem(viernes).
true.
```

Link del repositorio en GitHub:

<https://github.com/AdrianBaEs/CotejoEjerciciosParadigmasLABE>