**GUÍA DE APRENDIZAJE Nº 2**

1. **IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Programa de Formación:  ANALISIS Y DESARROLLO DE SISTEMAS DE INFORMACION | Código:  Versión: | 228106  102 | | |
| Nombre del Proyecto:  IMPLEMENTACION DE UN APLICATIVO WEB PARA EL SEGUIMIENTO Y EVALUACION DE LA ETAPA PRODUCTIVA DE LOS APRENDICES SENA A TRAVES DE UNA BITACORA VIRTUAL | Código: | 1107755 | | |
| Fase del proyecto: | | ANALISIS | | |
| Actividad (es) del Proyecto:  Analizar el sistema | Actividad (es) de Aprendizaje:   * Implementar programas con la instrucción condicional if/then/else | Ambiente de formación ESCENARIO Aula 2030A | MATERIALES DE FORMACIÓN | |
| DEVOLUTIVO Televisor, tablero | CONSUMIBLE (unidades empleadas durante el programa) |
| **Resultados de Aprendizaje:**  Construir el modelo conceptual del macrosistema frente a los requerimientos del cliente, | Competencia:  Analizar los requerimientos del cliente para construir el sistema de información. | | | |
| Duración de la guía ( en horas): | 24 | | | |

1. **INTRODUCCIÓN**

|  |
| --- |
| Las funciones en python, también llamadas subrutinas, son instrucciones agrupadas bajo un mismo nombre, lo que permite modularizar el código y ordenarlo, de manera tal que una determinada tarea se puede “empaquetar” en una función para posteriormente ser llamada cada vez que sea necesario. |

1. **ESTRUCTURACION DIDACTICA DE LAS ACTIVIDADES DE APRENDIZAJE**

|  |
| --- |
| * 1. **Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje.)** * Las funciones se pueden crear en cualquier punto de un programa, escribiendo su definición.   La primera línea de la definición de una función contiene: la palabra reservada **def,** el nombre de la función (la guía de estilo de Python recomienda escribir todos los caracteres en minúsculas separando las plabras por guiones bajos) paréntesis (que pueden incluir los argumentos de la función, como se explica más adelante)   * Las instrucciones que forman la función se escriben con sangría con respecto a la primera línea. * Por comodidad, se puede indicar el final de la función con la palabra reservada return (más adelante se explica el uso de esta palabra reservada), aunque no es obligatorio. * Para poder utilizar una función en un programa se tiene que haber definido antes. Por ello, normalmente las definiciones de las funciones se suelen escribir al principio de los programas   A continuación, se muestran dos ejemplos de funciones. Digítelos, compílelos y córralos |

|  |
| --- |
| * 1. **Actividades de transferencia del conocimiento (Conceptualización y Teorización).**   **Actividad No.1**  Implemente una función que reciba dos parámetros: el primero debe ser una cadena de caracteres y en segundo un carácter. La función debe retornar cuantos caracteres del segundo parámetro este contenido en el primer parámetro.  C:\Users\usuario\Google Drive\Luis Fernando Corredor\ADSI\Portafolio\Modulos\Implementacion\Curso Python\MisProgramas\cuenta\_caracteres.py  **Actividad No.2**  Implemente una función que reciba dos parámetros: el primero debe ser una cadena de caracteres y en segundo un carácter. La función debe retornar cuantos caracteres hay en el la cadena del primer parámetro antes que aparezca el carácter del segundo parámetro  C:\Users\usuario\Google Drive\Luis Fernando Corredor\ADSI\Portafolio\Modulos\Implementacion\Curso Python\MisProgramas\cuenta\_caracteres\_validos.py  **Actividad No.3**  Implemente una función que reciba un parámetro entero; la función deberá retornar una lista con la serie de Fibonacci  C:\Users\usuario\Google Drive\Luis Fernando Corredor\ADSI\Portafolio\Modulos\Implementacion\Curso Python\MisProgramas\fibonacci\_new.py  **Diccionarios**  Los diccionarios en Python son un tipo de estructuras de datos que permite guardar un conjunto no ordenado de pares clave-valor, siendo las claves únicas dentro de un mismo diccionario (es decir que no pueden existir dos elementos con una misma clave).  A continuación, se define un diccionario con los meses del año  **Ejemplo 1**  meses = {1: "January",2: "February",3: "March",4: "April",5: "May",6: "June",7: "July",8: "August",  9: "September",10: "October",11: "November",12: "December"}  print(meses) **# Muestra todo el diccionario**  print(meses[1]) **# Muestra January**  meses2 = meses.copy() **# Copia el contenido del meses en meses2**  meses.clear() **# Elimina el contenido del diccionario**  print("meses",meses)  print("meses2",meses2)  **Ejemplo 2**  contactos = {1:'Andres',2:'Myriam',3:'Alejandra',4:'Valentina'}  contactos2 = contactos.copy()  print(contactos)  print(contactos[1])  print(contactos2)  del(contactos2[4]) **# Elimina el elemento con clave 4:”Valentina”**  print(contactos2)  **Añadir o actualizar elementos**  Para añadir o actualizar un nuevo elemento se ha de seguir el siguiente formato:  ***NombreDiccionario[llave] = valor***  contactos[5] = ‘Karen’  En el caso de que la llave no este en el diccionario, este se añade, si no esta se actualiza.  **Concatenación de diccionario – método update()**  vehiculos1 = {"Placa":"IBX001","Marca":"Mazda"}  vehiculos2 = {"Modelo":2000,"Color":"Plateado"}  vehiculos1.update(vehiculos2)  **Retornar un elemento**  print(vehiculos1.get("Placa"))  print(vehiculos1.get("Marca"))  print(vehiculos1)   * + **Estructuras de selección múltiple switch-case**   Pyhton no cuenta con la estructura de selección múltiple **switch-case,**  pero normalmente esto se implementa a través de diccionarios.  def mes(numero\_mes):  switch = {  1: "January",  2: "February",  3: "March",  4: "April",  5: "May",  6: "June",  7: "July",  8: "August",  9: "September",  10: "October",  11: "November",  12: "December"  }  return switch.get(numero\_mes,"Mes invalido")  numero\_mes = int(input("Numero del mes: "))  print(mes(numero\_mes))  En el siguiente ejemplo se puede apreciar que los diccionarios también pueden utilizar funciones como valores.  def uno(numero\_dia):  return "Lunes"  def dos(numero\_dia):  return "Martes"  def tres(numero\_dia):  return "Miercoles"  def cuatro(numero\_dia):  return "Jueves"  def cinco(numero\_dia):  return "Viernes"  def seis(numero\_dia):  return "Sabado"  def siete(numero\_dia):  return "Domingo"  def dia\_semana(numero\_dia):  switch = {1:uno(numero\_dia),  2:dos(numero\_dia),  3:tres(numero\_dia),  4:cuatro(numero\_dia),  5:cinco(numero\_dia),  6:seis(numero\_dia),  7:siete(numero\_dia)  }  return switch.get(numero\_dia,"Día no valido")  numero\_dia = int(input("Ingrese el numero del día de la semana: "))  print(f"Hoy es {dia\_semana(numero\_dia)}") |
| * 1. **Actividades de apropiacion del conocimiento (Conceptualización y Teorización).** |

1. **RECURSOS PARA EL APRENDIZAJE**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ACTIVIDADES DEL PROYECTO** | **DURACIÓN  (Horas)** | **Materiales de formación devolutivos: (Equipos/Herramientas)** | | **Materiales de formación (consumibles)** | | **Talento Humano (Instructores)** | | **AMBIENTES DE  APRENDIZAJE TIPIFICADOS** |
| **Descripción** | **Cantidad** | **Descripción** | **Cantidad** | **Especialidad** | **Cantidad** | **ESCENARIO (Aula, Laboratorio, taller, unidad productiva)** y elementos y condiciones de seguridad industrial, salud ocupacional y medio ambiente |
|  |  |  |  |  |  |  |  |  |

|  |
| --- |
| **JDK**  **J2SE**  **Java** |

1. **GLOSARIO DE TERMINOS**
2. **REFERENTES BIBLIOGRÁFICOS**

|  |
| --- |
| Harvey M. Deitel, Paul J. Deitel (2004). Como Programar en Java, Pearson Education.  [Froufe Quintas, Agustín](http://biblioteca.sena.edu.co/F/8IVEUS4EGGBXJ5SGUXR2QRQLL3TFQ63B6GFACX6M9J3XX3KCS5-24793?func=full-set-set&set_number=009595&set_entry=000006&format=999) (2009). [Java 2 : manual de usuario y tutorial](http://biblioteca.sena.edu.co/F/8IVEUS4EGGBXJ5SGUXR2QRQLL3TFQ63B6GFACX6M9J3XX3KCS5-24794?func=full-set-set&set_number=009595&set_entry=000006&format=999), Alfaomega Ra-Ma  Flórez Fernández, Héctor Arturo (2012). Programación Orientada a Objetos Usando Java,  [Ecoe Ediciones](http://www.digitaliapublishing.com.bdigital.sena.edu.co/s/e/179).  <http://www.digitaliapublishing.com.bdigital.sena.edu.co/a/30030/programacion-orientada-a-objetos-con-java> |

1. **CONTROL DEL DOCUMENTO (ELABORADA POR)**

|  |
| --- |
| **Ing. Luis Fernando Corredor Mora** |