

### Unidad 1: Introducción a la Programación y al lenguaje de programación

**OE1.1.** Analizar problemas mediante la especificación, a través de contratos, de entradas, salidas, ejemplos y casos de prueba.

**OE1.2.** Modelar información relevante a la solución del problema empleando variables, constantes, tipos de datos primitivos y cadenas de texto.

**OE1.4.** Utilizar operadores (de asignación, aritméticos, relacionales, de cadenas y lógicos), estructuras contenedoras lineales de tamaño fijo (de tipos de datos primitivos) y cadenas de texto en la construcción de soluciones.

**OE1.5.** Utilizar un ambiente de desarrollo (incluyendo la compilación y ejecución de programas desde consola) y un espacio de trabajo predefinido, para construir la solución de un problema.

**OE1.6.** Codificar en lenguaje Java la solución a un problema a partir de los contratos de solución propuestos en las etapas de análisis y diseño.

**OE1.7.** Utilizar objetos e invocar métodos estáticos de clases del API de Java en la construcción de soluciones implementadas con interfaces gráficas por consola.

### ***Problema: COP 16 1.0 y 2.0***

El Seguimiento 2 tendrá dos partes: cada una correspondiente a un porcentaje de los requerimientos de la primera y segunda entrega de la [Tarea Integradora 1](#).

#### **Parte 1: COP 16 1.0 (50%)**

Se desea que se realice un programa que tenga la capacidad de recibir el nombre, cédula del guía, mostrar la información de las caminatas e indicar el mensaje si la temperatura es favorable, es decir, dentro del enunciado de la [T11](#) que realice los siguientes requisitos marcados en azul:

Se desea realizar un programa que sea usado por un organizador voluntario de la COP 16, para registrar un grupo de personas que participará en la ruta un día en específico, que cumpla con lo siguiente:

- Se solicita el nombre y la cédula del organizador voluntario para registro del programa, además de brindarle un saludo de bienvenida.
- Posibilidad de escoger la ruta: se debe escoger una de las 3 rutas programadas. Dependiendo de la ruta que se escoja, se debe desplegar la siguiente información:
  - Punto de encuentro de la actividad.
  - Hora de inicio de la actividad.
  - Hora de finalización de la actividad.
- Se debe informar la cantidad de integrantes que harán de la actividad de la siguiente manera:
  - Cantidad de participantes.
  - Cantidad de guías.
- Ingresar los datos meteorológicos del día, los cuales son:
  - Temperatura en grados centígrados °C.
  - Porcentaje de Humedad relativa.
    - Si la temperatura está entre 20°C y 25°C y la humedad relativa entre 40% y 60%, el programa debe desplegar el mensaje de: "¡Hace un buen día para caminar por Cali!"

OJO: Debe implementar este código haciendo uso de métodos auxiliares (lo veremos en la clase del miércoles 28 de agosto).

### Ejemplo de ejecución:

```
# Bienvenido voluntario a la aplicación de Interacción de Rutas Ecológicas COP 16 Cali - Colombia. ¿Cuál es tu nombre?
> Juan Betancourt
# Por favor, digita tu cédula.
> 1123456789
# ¡Bienvenido, Juan Betancourt!
# ¿Qué ruta registrarás el día de hoy?
> Farallones
# ¡Excelente! La Ruta de los Farallones tiene como punto de encuentro Calle 16 - Universidad del Valle iniciando a las 6:40 am, y termina a las 4:00 pm.
# Ingresar la temperatura en grados centígrados °C
> 24.3
# Ingresar el porcentaje de humedad relativa
> 50
# ¡Hace un buen día para caminar por Cali!
```

## Entregables Parte 1

Debe entregar:

1. Un documento, puede ser en word o un pdf, donde se especifiquen:
  - a. Análisis: Descripción del programa, Entradas, Salidas (especificar tipo de dato para entradas y salidas), Ejemplo (el ejemplo debe contener datos distintos al del ejemplo de ejecución)
  - b. Diseño: Diagrama de flujo.
2. Archivo .java con la solución

**Rúbrica Parte 1 ([Enlace](#))**, cada rúbrica de cada parte se encuentra en una hoja dentro del archivo.

Análisis y Diseño (40%)		Codificación (60%)					Total S2 Parte 1
El análisis cuenta con una descripción, entradas, salidas y al menos un ejemplo	Se presenta un diagrama de flujo general correcto del programa.	Buenas prácticas (ver tabla de arriba)	Compilación y Ejecución del Código (ver tabla de arriba)	El programa pide y recibe correctamente por consola el nombre y cédula del guía ecológico	El programa dentro del código es desarrollado utilizando métodos (funciones) auxiliares que reciben entradas desde el main, se ejecutan y retornan las salidas de vuelta al main para ser proyectadas al usuario	El programa brinda la información de las distintas rutas y proyecta un mensaje si las condiciones climatológicas de Cali son favorables	
20,0%	20,0%	5%	5%	10%	20,0%	20,0%	100,0%

**FECHA DE ENTREGA SEGUIMIENTO 2 PARTE 1:** *Hasta el viernes 30 de agosto, 6:00 PM COT (Colombian Time).*

**ENLACE DE ENTREGA SEGUIMIENTO 2 PARTE 1:** [GitHub](#)

## Parte 2: COP 16 2.0 (50%)

Se desea que se realice un programa que tenga la capacidad de guardar la información de hasta 10 lugares biodiversos, es decir, su nombre, departamento, pero, adicional, en vez de cantidad de km cuadrados, se debe guardar el presupuesto anual que se le da al lugar para ser cuidado (suma de dinero ofrecida por el gobierno).

Con este dato nuevo, se plantea que el programa pueda:

- Calcular el promedio de presupuesto de lugares biodiversos, según un departamento dado.

Es decir, dentro del enunciado de la [T11](#) que realice los siguientes requisitos marcados en rojo:

### Segunda fase: Gestión de Lugares Biodiversos

En medio del discurso ambiental que aborda la COP 16, se hace casi que obligatorio hablar sobre lugares especiales en el ambiente que resultan ser biodiversos por naturaleza y que debemos identificarlos y conocerlos para poder protegerlos de la mejor manera posible. A estos lugares, se les llama *lugares con diversidad ecológica*.

De los *lugares con diversidad biológica* se deberá almacenar la siguiente información:

- Nombre del lugar.
- Nombre del depto (Chocó, Valle, Cauca, Nariño).
- Cantidad de kilómetros cuadrados.

Claramente, cambiando el campo de “Cantidad de kilómetros cuadrados” por “Presupuesto nacional en pesos colombianos” y conservando (nombre del lugar y departamento) de la T11.

**Ejemplo de ejecución (recuerde que # indica mensaje en consola del programa y > entrada de usuario por consola) :**

```
# Bienvenido voluntario a la aplicación de Gestión de Lugares Biodiversos COP 16 Cali - Colombia.
# Te presentamos las siguientes opciones, ingresa:
# 1. Para registrar un lugar con diversidad biológica
# 2. Para consultar, según un departamento dado, el promedio de presupuestos de lugares biodiversos
# 3. Para salir del programa.
> 1
# Ingresa el nombre del lugar biodiverso
```

```

> Parque Nacional Natural Puracé
# Ingresa el departamento en el cual está el lugar
> Cauca
# Ingresa el presupuesto nacional otorgado para este lugar
> 125000000
# Lugar ingresado con éxito.
# 1. Para registrar un lugar con diversidad biológica
# 2. Para consultar, según un departamento dado, el promedio de presupuestos de lugares biodiversos
# 3. Para salir del programa.
> 1
# Ingresa el nombre del lugar biodeverso
> Parque Nacional Natural Farallones de Cali
# Ingresa el departamento en el cual está el lugar
> Valle
# Ingresa el presupuesto nacional otorgado para este lugar
> 250000000
# Lugar ingresado con éxito.
# 1. Para registrar un lugar con diversidad biológica
# 2. Para consultar, según un departamento dado, el promedio de presupuestos de lugares biodiversos
# 3. Para salir del programa.
> 1
# Ingresa el nombre del lugar biodeverso
> Parque Nacional Natural Uramba Bahía Málaga
# Ingresa el departamento en el cual está el lugar
> Valle
# Ingresa el presupuesto nacional otorgado para este lugar
> 75000000
# Lugar ingresado con éxito.
# 1. Para registrar un lugar con diversidad biológica
# 2. Para consultar, según un departamento dado, el promedio de presupuestos de lugares biodiversos
# 3. Para salir del programa.
> 2
# Seleccione uno de los cuatro departamentos a revisar el promedio presupuestal: Valle, Chocó, Cauca o Nariño
> Valle
# El promedio presupuestal para el departamento de Valle en los lugares biodiversos ingresados es de 162500000 COP
# 1. Para registrar un lugar con diversidad biológica
# 2. Para consultar, según un departamento dado, el promedio de presupuestos de lugares biodiversos
# 3. Para salir del programa.
> 2

```

# Seleccione uno de los cuatro departamentos a revisar el promedio presupuestal: Valle, Chocó, Cauca o Nariño  
 > Cauca  
 # El promedio presupuestal para el departamento de Valle en los lugares biodiversos ingresados es de 125000000 COP

## Entregables Parte 2

Debe entregar:

3. Un documento, puede ser en word o un pdf, donde se especifiquen:
  - a. Análisis: Descripción del programa, Entradas, Salidas (especificar tipo de dato para entradas y salidas), Ejemplo (el ejemplo debe contener datos distintos al del ejemplo de ejecución)
  - b. [Diseño](#): Diagrama de flujo.
4. Archivo .java con la solución

**Rúbrica Parte 2 ([Enlace](#)), cada rúbrica de cada parte se encuentra en una hoja dentro del archivo.**

Análisis, Diseño y Documentación (30%)			Codificación (70%)				Total S2 Parte 2	Bono (10%)	Total S2 Parte 2 + Bono
El análisis cuenta con una descripción, entradas, salidas y al menos un ejemplo	Se presenta un diagrama de flujo general correcto del programa.	Todos los métodos a excepción del main, tienen un contrato y el contrato es acorde a los parámetros y retorno del método. Se genera la documentación con javadoc	Buenas prácticas y uso de estructura de carpetas bin, src (ver tabla de arriba)	Compilación y Ejecución del Código (ver tabla de arriba)	El programa pide y recibe correctamente por consola la información del lugar biodiverso	El programa utiliza contenedores de tamaño fijo (arreglos) para dar solución al problema, y según un departamento dado, calcula de manera correcta el promedio presupuestal de lugares para ese departamento		Uso del idioma inglés (ver tabla de arriba)	
10,0%	10,0%	10%	5%	10%	20,0%	35,0%	100,0%	10%	110,0%

**FECHA DE ENTREGA SEGUIMIENTO 2 PARTE 1:** Hasta el jueves 12 de septiembre, 11:59 PM COT (Colombian Time).

**ENLACE DE ENTREGA SEGUIMIENTO 2 PARTE 1:** [GitHub](#)