

Actividad: Principios Fundamentales de la Programación

Descripción General

Esta actividad está diseñada para introducir a los estudiantes en los **principios fundamentales de la programación**. A través de ejercicios prácticos, los estudiantes comprenderán los conceptos básicos que son comunes a todos los lenguajes de programación y desarrollarán habilidades de resolución de problemas.

Objetivos de Aprendizaje

- Comprender los principios fundamentales de la programación.
 - Aplicar estos principios en la resolución de problemas simples.
 - Desarrollar habilidades de pensamiento lógico y algorítmico.
 - Familiarizarse con la sintaxis básica de un lenguaje de programación.
-

Principios Fundamentales de la Programación

1. Abstracción

La **abstracción** es el proceso de identificar las características esenciales de un objeto o sistema, ignorando los detalles menos relevantes. En programación, esto significa crear modelos simplificados de problemas complejos.

Ejemplo:

javascript

CopyEdit

```
// Abstracción de un estudiante
class Estudiante {
  constructor(nombre, edad, matricula) {
    this.nombre = nombre;
```

```
        this.edad = edad;
        this.matricula = matricula;
    }

    mostrarInformacion() {
        return `${this.nombre}, ${this.edad} años, Matrícula:
    ${this.matricula}`;
    }
}
```

2. Descomposición

La **descomposición** consiste en dividir un problema complejo en partes más pequeñas y manejables. Cada parte puede resolverse de manera independiente.

Ejemplo:

javascript

CopyEdit

```
// Descomposición de un programa para calcular promedios
function calcularPromedio(numeros) {
    return sumarNumeros(numeros) / numeros.length;
}

function sumarNumeros(numeros) {
    let suma = 0;
    for (let i = 0; i < numeros.length; i++) {
        suma += numeros[i];
    }
    return suma;
}
```

3. Reconocimiento de Patrones

El **reconocimiento de patrones** permite identificar similitudes entre problemas para reutilizar soluciones existentes.

Ejemplo:

javascript

CopyEdit

```
// Patrón para procesar diferentes tipos de datos
function procesarDatos(datos, procesador) {
  const resultados = [];
  for (let dato of datos) {
    resultados.push(procesador(dato));
  }
  return resultados;
}

const numeros = [1, 2, 3, 4, 5];
const duplicados = procesarDatos(numeros, x => x * 2);
const cuadrados = procesarDatos(numeros, x => x * x);
```

4. Algoritmos

Un **algoritmo** es una secuencia finita de instrucciones bien definidas para resolver un problema específico.

Ejemplo:

javascript

CopyEdit

```
// Algoritmo de búsqueda binaria
function busquedaBinaria(array, objetivo) {
  let inicio = 0;
  let fin = array.length - 1;

  while (inicio <= fin) {
    let medio = Math.floor((inicio + fin) / 2);

    if (array[medio] === objetivo) {
      return medio; // Encontrado
    } else if (array[medio] < objetivo) {
      inicio = medio + 1;
    } else {
      fin = medio - 1;
    }
  }

  return -1; // No encontrado
}
```

5. Modularidad

La **modularidad** implica dividir un programa en módulos o funciones independientes que pueden ser desarrollados, probados y mantenidos por separado.

Ejemplo:

javascript

CopyEdit

```
// Programa modular para gestionar una tienda
function calcularPrecioTotal(productos) {
  return productos.reduce((total, producto) => total +
    calcularPrecioProducto(producto), 0);
}

function calcularPrecioProducto(producto) {
  return producto.precio * (1 - producto.descuento) * (1 +
    producto.impuesto);
}

function aplicarDescuento(producto, porcentaje) {
  producto.descuento = porcentaje;
  return producto;
}
```

Actividad Práctica

Parte 1: Análisis de Principios

Identifica y explica qué principios de programación se aplican en el siguiente código:

javascript

CopyEdit

```
function calcularEstadisticas(numeros) {
  return {
    promedio: calcularPromedio(numeros),
    maximo: encontrarMaximo(numeros),
    minimo: encontrarMinimo(numeros)
  };
};
```

```
}

function calcularPromedio(numeros) {
  return numeros.reduce((a, b) => a + b, 0) / numeros.length;
}

function encontrarMaximo(numeros) {
  return Math.max(...numeros);
}

function encontrarMinimo(numeros) {
  return Math.min(...numeros);
}

// Uso
const datos = [5, 3, 8, 1, 9, 4];
const estadisticas = calcularEstadisticas(datos);
console.log(estadisticas);
```

Parte 2: Aplicación de Principios

Desarrolla un programa que gestione una biblioteca simple aplicando los principios de programación vistos. El programa debe:

1. Crear una **abstracción** para representar libros (título, autor, año, género).
 2. Implementar funciones para:
 - Agregar libros a la biblioteca.
 - Buscar libros por título o autor.
 - Filtrar libros por género.
 - Mostrar todos los libros.
 3. Utilizar la **descomposición** y **modularidad** para organizar el código.
 4. Implementar al menos un **algoritmo eficiente** (por ejemplo, para ordenar o buscar libros).
-

Parte 3: Reflexión

Escribe una breve reflexión (máximo 300 palabras) sobre:

1. ¿Cómo aplicaste cada uno de los principios en tu solución?
 2. ¿Qué dificultades encontraste al aplicar estos principios?
 3. ¿Cómo estos principios pueden ayudarte a desarrollar software de mejor calidad?
-

Criterios de Evaluación




Criterio	Porcentaje
Comprensión de los principios	30%
Aplicación práctica	40%
Calidad del código	20%
Reflexión	10%

Fecha de Entrega

La actividad debe ser entregada en formato digital (archivo `.js` o `.py` con el código y un documento `.pdf` o `.md` con la reflexión) antes del:

[FECHA DE ENTREGA]

Recursos Adicionales

-  [Clean Code por Robert C. Martin](#)
 -  [Curso online: "Principios de Programación" en Coursera](#)
 -  [Documentación de JavaScript / Python](#)
-

¡Buena suerte con la actividad! Recuerda que estos principios son fundamentales no solo para esta tarea, sino para toda tu carrera como programador.