

Módulo formativo: Programación y base de datos

Objetivo: Desarrollar sistemas informáticos con lenguajes de programación y base de datos, aplicando diferentes metodologías según los requerimientos de funcionalidad.

Ficha N.º BT54I2	
Contenidos	Actividades de aprendizaje
<p>Tema: Estructuras de control</p> <p>Contenido procedural: Determinar las estrategias de programación orientada a objetos, utilizando módulos, unidades o paquetes de desarrollo de programas.</p> <p>Lenguajes de programación Ahora bien, a una computadora no se puede dar instrucciones como «reconoce ese rostro». ¡Ella no sabría qué hacer! A una computadora se debe dar instrucciones lógicas y matemáticas precisas, y en un lenguaje que ella entienda. A este tipo de lenguajes se denomina «lenguajes de programación».</p> <p>Código fuente Cuando se escribe un conjunto de instrucciones para la computadora, se genera un texto usando cierto lenguaje de programación. A este texto se denomina «código fuente».</p> <p>Clases, atributos Un tipo especial de programación, con enormes ventajas prácticas, es la llamada «programación orientada a objetos» (POO). Se basa en la observación de que cosas similares comparten características comunes. Por ejemplo, todas las personas tienen un nombre, una estatura, una edad, un peso, un número de cédula. Estas características (nombre, estatura, edad, peso, número de cédula) se denominan «atributos» y pertenecen a la «clase» persona.</p> <p>Métodos Aparte de atributos, una persona puede ser caracterizada por las actividades que realiza. Supongamos que debemos realizar un sistema para gestionar el quehacer de una cafetería. Para ello, sabemos que en la cafetería</p>	<p>Tareas: Tomar como caso de estudio algún negocio familiar o de alguien cercano. Estudiar los distintos roles (clases) que hay en el negocio y enumerar los respectivos atributos y métodos de cada clase. Luego instanciar cada clase con los datos de los «objetos» (personas) reales que trabajen en el negocio.</p>

trabajarán personas, todas con los atributos mencionados. Pero, además, algunas tendrán la capacidad de preparar café y otras de recibir a los clientes y cobrar. A estas acciones, en el mundo de la programación, se les denomina «métodos».

Objetos

Hasta aquí, todo sería un poco abstracto. ¡Pero en una cafetería no se trabaja con entes abstractos, sino con personas concretas! Entonces contratamos a una persona de carne y hueso cuyo nombre es «Lucía»; estatura: 1,65; edad: 23; etcétera; que va a realizar la acción «atender las mesas». Decimos entonces que hemos «instanciado» la clase persona, creando un «objeto» real, Lucía.

Cuando llegue otro trabajador, solo deberemos instanciar la misma clase, sin necesidad de empezar todo desde cero. Este es el poder de la POO.

Funciones

La idea de reutilizar código, de no escribir una y otra vez las mismas cosas, da origen a las llamadas «funciones». Una función nos sirve para realizar cálculos matemáticos repetitivos, pero cuyos «argumentos» pueden cambiar. Por ejemplo, en nuestra cafetería queremos una función para calcular el monto que deben pagar los clientes. Esta función (que podemos llamar «calculaMonto») debería recibir como argumentos los costos de los productos consumidos (costo1, costo2, costo3...). ¿Cómo calcula el monto a pagar nuestra función? Pues suma todos los costos y multiplica el resultado por 1.12 (para añadir el IVA). Así, cuando un cliente llega a la caja, «llamamos» a nuestra función «enviándole» como argumentos los valores específicos para ese cliente, y ella calcula el monto y nos «devuelve» el valor correspondiente. ¡Muy práctico!

Documentación

Las apps que usas en tu celular no fueron desarrolladas por una sola persona, sino por un ejército de programadores, casi seguro. ¿Cómo se entienden entre todos ellos? Para indicar a su siguiente colega qué se ha hecho y qué se debe hacer a continuación, un programador debe redactar una «documentación» adecuada, que narra, por así decirlo, el trabajo realizado. La documentación es una herramienta muy importante para coordinar equipos de trabajo.

Actividades recomendadas

Invitar a los estudiantes a pensar en otro ejemplo de clase, la clase «auto». ¿Qué atributos debería tener esta clase? (motivarlos a dar las respuestas, que deberían ser del tipo: color, placa, año de fabricación, cilindraje, etc.)

Con la clase generada en consenso por todo el grupo (y anotadas en la pizarra), los estudiantes en equipos instanciarán un total de cinco objetos carro por equipo. Estos deben ser reales, del entorno cercano del estudiante, por ejemplo, el auto del papá, del tío, del amigo, etc. Si algún estudiante no sabe la placa o algún otro dato, deberá llamar ese momento para adquirir esa información.

Con la ayuda de sus estudiantes, escriba en pseudocódigo una función «prom» que tome como argumentos cuatro aportes de un alumno (ap1, ap2, ap3, ap4) y devuelva un número o si reprueba y 1 si aprueba.

(Incluimos a continuación un ejemplo, que sirve si la nota mínima para aprobar es 7).

```
función prom(ap1, ap2, ap3, ap4)
```

```
    p = (ap1 + ap2 + ap3 + ap4) / 4
```

```
    if p > 7
```

```
        devolver 1
```

```
    caso contrario
```

```
        devolver 0
```