Equipo de Fútbol

PROGRAMACIÓN II

UADE

Introducción

Equipo TDA

En programación, nos enfrentamos a desafíos complejos que requieren **organización, eficiencia y estructura**.

Imaginemos una analogía entre la **programación** y un deporte que muchos de nosotros conocemos: el **fútbol**.

Somos el DT

El equipo

Imagina que estamos a cargo de organizar un equipo de fútbol.

Nuestro enfoque se centra en la organización eficiente del equipo y en la gestión de los jugadores para alcanzar el éxito.

En programación, esta misma organización y gestión se reflejan en los **Tipos de Datos Abstractos** (**TDA**).

TDA

Tipo de datos abstractos

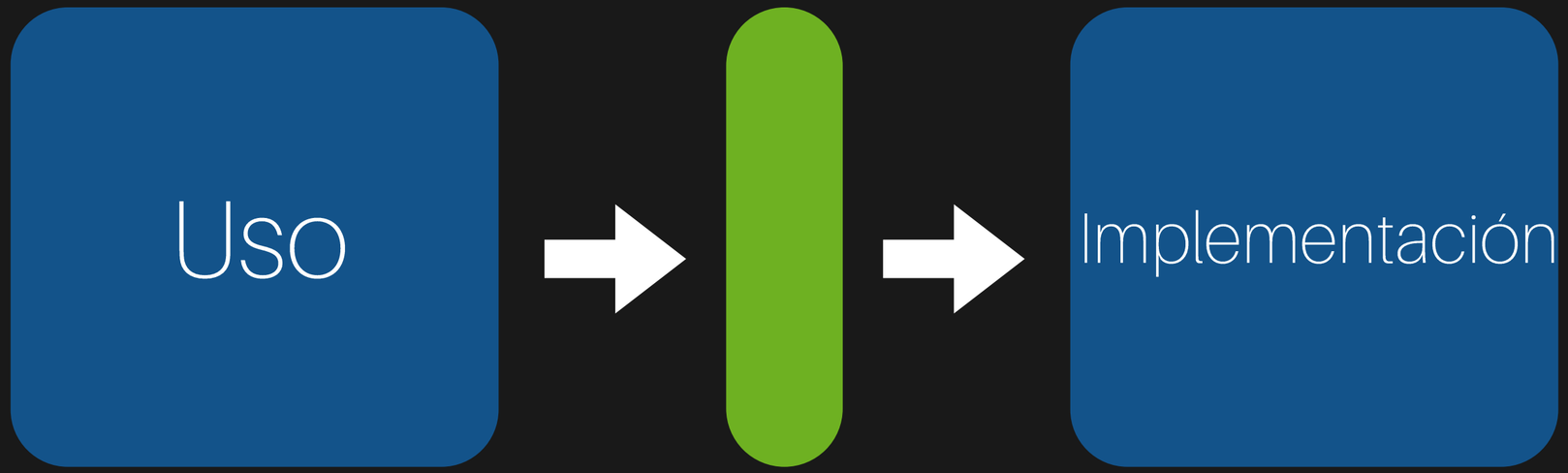
Un Tipo de Dato Abstracto (TDA) es una abstracción que especifica qué datos manipulamos y qué operaciones podemos realizar con ellos, delineando el "qué" sin adentrarse en el "cómo" de las operaciones.

TDA

3 Etapas Cruciales:

- **Especificación: (Interfaz)**
  - Define qué operaciones el TDA debe soportar.
- **Implementación:**
  - Detalla cómo se realizan las operaciones especificadas. En esta etapa, desarrollamos la lógica interna.
- **Uso:**
  - Aplicación de los TDAs en contextos reales. Aquí es donde utilizamos las operaciones definidas por los TDAs para construir la lógica de nuestra aplicación.

Interfaz



CLASE que utiliza el TDA

CLASE que implementa el TDA

Flexibilidad y Mantenibilidad: La separación clara entre especificación, implementación y uso permite modificar cómo se realizan internamente las operaciones (implementación) sin afectar la forma en que los TDAs son utilizados en el programa. Esto mejora la flexibilidad y facilita el mantenimiento del código.

Interfaz

Equipo

La interfaz en un equipo puede compararse con cómo manejamos el equipo y cada jugador. En programación, la interfaz de nuestro TDA establece cómo interactuamos con él. Definiremos operaciones claras, como "**agregarJugador**", "**quitarJugador**", "**cantidadDeJugadores**" y "**estaLleno**", como las jugadas tácticas que guían nuestro equipo.

Preguntas

Qué se imaginan que hace cada una? Qué devuelve? Para qué nos sirve?

Interfaz

Declaramos los métodos que nuestro TDA soportará, indicando las **operaciones que se pueden realizar**, los parámetros que reciben, el tipo de dato que retornan y una descripción de su funcionalidad.

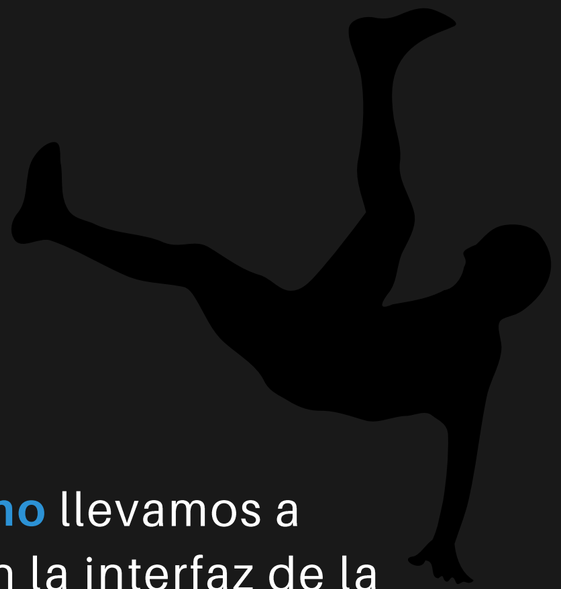
```java
public interface EjemploTDA {
    // Agrega un elemento al TDA
    void agregarElemento(tipoDato elemento);
    // Quita un elemento del TDA
    void quitarElemento(tipoDato elemento);
    // Verifica si el TDA está vacío
    boolean estaVacio();
}
```

Genérica

```java
public interface FormacionTDA {
    void agregarJugador(String nombre);
    void quitarJugador(String nombre);
    boolean estaCompleta();
    int cantidadJugadores();
    String obtenerPrimerJugador();
}
```

Equipo

Implementación



Equipo

Implementación en el equipo:
La implementación se refiere a **cómo** llevamos a cabo las acciones que definimos en la interfaz de la escaloneta. En programación, construir la implementación del TDA implica definir cómo se añaden y quitan jugadores, cómo se consulta la cantidad y cómo verificamos si está lleno.

Preguntas

Cómo les parece que hariamos cada acción si estuviesen guardadas en una lista?

"agregarJugador", "quitarJugador", "cantidadDeJugadores" y "estaLleno"

Implementación

Genérica

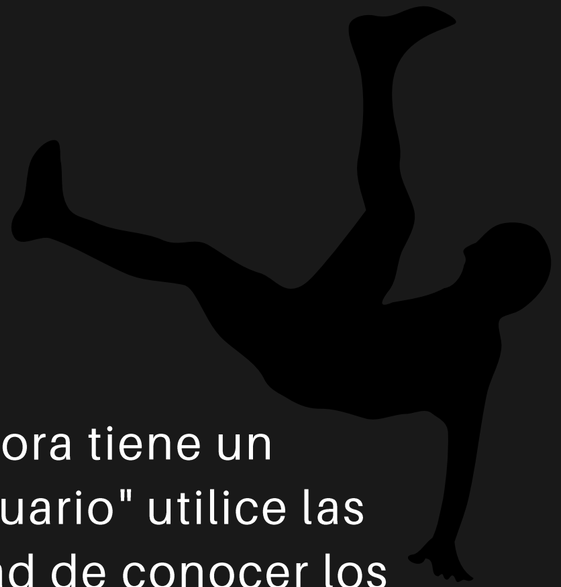
```
public class MiTDA implements EjemploTDA {
    private int[] elementos; // Para una implementación estática
    private int contador = 0;

    @Override
    public void agregarElemento(int elemento) {
        elementos[contador++] = elemento;
    }

    @Override
    public void quitarElemento(int elemento) {
        // Lógica para quitar el elemento
    }

    @Override
    public boolean estaVacio() {
        return contador == 0;
    }
}
```

Equipo

Implementación

```java
public class Formacion implements FormacionTDA {
    private ArrayList<String> jugadores = new ArrayList<>();

    public void agregarJugador(String nombre) {
        jugadores.add(nombre);
    }

    public void quitarJugador(String nombre) {
        jugadores.remove(nombre);
    }

    public boolean estaCompleta() {
        return jugadores.size() >= 11;
    }

    public int cantidadJugadores() {
        return jugadores.size();
    }

    public String obtenerPrimerJugador() {
        return jugadores.get(0);
    }
```

Uso



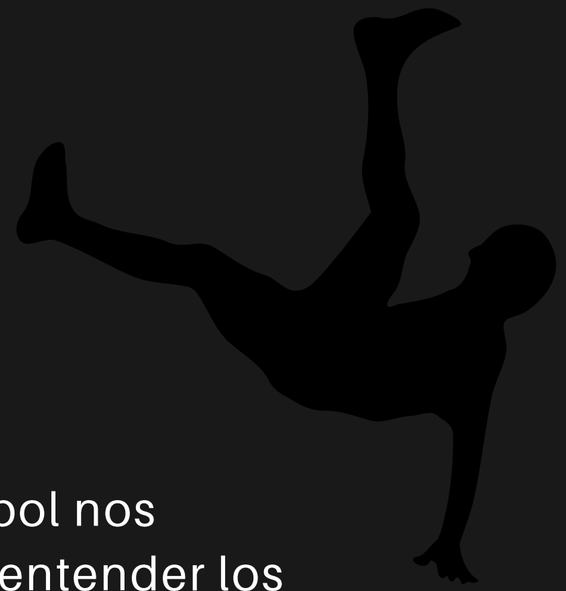
Equipo

Todo lo que hemos hecho hasta ahora tiene un propósito clave: permitir que el "usuario" utilice las operaciones definidas sin necesidad de conocer los detalles de implementación. Similar a cómo un entrenador le indica a un jugador qué hacer en el campo, el usuario simplemente utiliza la interfaz del TDA para agregar, quitar jugadores, consultar la cantidad y verificar si está lleno, sin necesidad de entrar en detalle de cómo se logra.

Qué tipo de usos le darían a estas operaciónes? En qué contexto podrían usarse?

"agregarJugador", "quitarJugador", "cantidadDeJugadores" y "estaLleno"

Conclusión



Equipo

Conclusión:
La analogía de un equipo en el fútbol nos proporciona una manera visual de entender los conceptos de TDA, implementación, uso e interfaz en la programación. Así como en el equipo, cada operación en un TDA tiene un propósito específico y trabaja en conjunto para lograr un objetivo programático.

Equipo

Lo vemos en JAVA?