

Bing

Por supuesto, aquí tienes un ejemplo más detallado y realista de cómo implementar un Singleton en Java utilizando una clase que maneja la configuración de una aplicación web:

```
public class AppConfig {
    private static AppConfig instance;
    private String apiUrl;
    private String apiKey;
    private boolean debugMode;

    private AppConfig() {
        // Configuración inicial (puedes cargarla desde un archivo, base de datos, etc.)
        apiUrl = "https://api.example.com";
        apiKey = "your-api-key";
        debugMode = false;
    }

    public static synchronized AppConfig getInstance() {
        if (instance == null) {
            instance = new AppConfig();
        }
        return instance;
    }

    public String getApiUrl() {
        return apiUrl;
    }

    public String getApiKey() {
        return apiKey;
    }

    public boolean isDebugMode() {
        return debugMode;
    }

    public void setDebugMode(boolean debugMode) {
        this.debugMode = debugMode;
    }

    public static void main(String[] args) {
        // Uso del Singleton para obtener la configuración
        AppConfig appConfig = AppConfig.getInstance();

        // Obtener valores de configuración
        System.out.println("API URL: " + appConfig.getApiUrl());
        System.out.println("API Key: " + appConfig.getApiKey());

        // Cambiar al modo de depuración
        appConfig.setDebugMode(true);
        System.out.println("Debug Mode: " + (appConfig.isDebugMode() ? "Enabled" : "Disabled"));
    }
}
```

En este ejemplo:

1. Creamos la clase `AppConfig` que almacena la configuración global de nuestra aplicación.
2. El constructor privado inicializa la configuración.
3. El método `getInstance` garantiza que solo haya una instancia activa de `AppConfig`.
4. Utilizamos los métodos `getApiUrl`, `getApiKey`, `isDebugMode` y `setDebugMode` para acceder y modificar los valores de configuración.

Adapta este ejemplo según tus necesidades específicas. El patrón Singleton es útil cuando necesitas una única instancia de una clase durante todo el ciclo de vida de una solicitud en una aplicación web. 😊.