LA TERRAZA DEL SABOR

MANUAL DE ESTANDARES

Índice

---

1. Convenciones de Nombres

Clases

- Los nombres de las clases deben ser sustantivos en CamelCase. Ejemplo: ClientesDTO, ReservasDAO.

Métodos

- Los nombres de los métodos deben ser verbos en camelCase. Ejemplo: getClienteById(), updateReserva().

Variables

- Las variables deben utilizar nombres descriptivos en camelCase. Ejemplo: totalReservas, descuento.

Constantes

- Las constantes deben estar en mayúsculas con palabras separadas por guiones bajos. Ejemplo: MAX_RESERVAS, IVA_TASA.

2. Estructura del Código

Organización de Archivos

- Las clases se deben organizar en carpetas según su funcionalidad. Ejemplo: DTOs en DataAccess/DTO, DAOs en DataAccess/DAO.

Métodos

- Los métodos deben ser cortos y realizar una única tarea específica.

3. Documentación

Comentarios

- Utilizar comentarios descriptivos para explicar bloques de código complejos.
- Utilizar Javadocs para documentar clases y métodos públicos. Ejemplo:

```
java
Copiar código
/**
 * Calcula el total con IVA incluido.
 * @param monto El monto sin IVA.
 * @return El monto con IVA.
 */
public double calcularIVA(double monto) { ... }
```

4. Manejo de Excepciones

Estrategia de Manejo

- Utilizar bloques try-catch para manejar excepciones.
- Lanzar excepciones específicas y evitar el uso de Exception genérico.
- Registrar los errores utilizando una clase de log. Ejemplo:

```
java
Copiar código
try {
    // código
} catch (SQLException e) {
    Logger.log("Error al acceder a la base de datos: " + e.getMessage());
    throw new CustomException("Error en la operación de base de datos", e);
}
```

5. Estilo de Codificación

Indentación

- Utilizar 4 espacios para la indentación. No usar tabulaciones.

Formato

- Mantener líneas de código con un máximo de 80 caracteres.
- Utilizar una línea en blanco para separar métodos y bloques de código.

6. Pruebas

Tipos de Pruebas

- Implementar pruebas unitarias para cada método.
- Realizar pruebas de integración para verificar la interacción entre componentes.

Marco de Pruebas

- Utilizar JUnit para pruebas unitarias.
- Utilizar Mockito para pruebas de interacción y simulación.

7. Seguridad

Mejores Prácticas

- Validar todas las entradas del usuario.
- Utilizar cifrado para datos sensibles.
- Implementar controles de acceso y permisos adecuados.

8. Control de Versiones

Ramas

- Utilizar ramas para cada nueva funcionalidad o corrección de errores. Ejemplo: feature/nueva-funcionalidad, bugfix/correccion-error.

Commits

- Hacer commits pequeños y descriptivos.
- Utilizar mensajes de commit claros y concisos. Ejemplo: Añadida validación de entrada en ClienteDTO.