

Patrón de diseño Builder

Juan Pablo Rincón Romero – 20231020101

Facultad de Ingeniería, Universidad Distrital Francisco José de Caldas



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS**

Modelos de programación

Docente José Joaquin Bocanegra

25 de febrero de 2026

1. Análisis del problema

El taller plantea el desarrollo de un módulo para un sistema de pedidos de comida rápida, donde el producto principal es una hamburguesa configurable. Cada hamburguesa debe incluir ciertos componentes obligatorios, pero también puede tener ingredientes adicionales según las preferencias del cliente.

El problema principal radica en que la hamburguesa puede tener múltiples combinaciones posibles. Si se intentara modelar esta situación utilizando constructores tradicionales con diferentes combinaciones de parámetros, se generaría el problema del constructor telescopico, es decir, múltiples constructores sobrecargados con distintos conjuntos de parámetros.

Esto provoca:

- Código difícil de leer.
- Mayor probabilidad de errores.
- Baja mantenibilidad.
- Confusión sobre el orden y significado de los parámetros.

Por ejemplo:

```
new Hamburguesa ("Ajonjolí", "Res", true, "Lechuga, Tomate");
```

No es evidente qué representa cada valor sin revisar la documentación, lo que reduce la claridad del código.

2. Identificación del objeto y atributos

- **Objeto complejo:**

Hamburguesa

Debe ser inmutable, es decir:

- I. Atributos privados y finales.
- II. Constructor privado.
- III. Sin métodos setters
- IV. Creación únicamente mediante el Builder.

- Atributos obligatorios:

I. tipoPan (breadType)

Tipo de pan utilizado en la hamburguesa.

II. tipoCarne (meatType)

Tipo de carne seleccionada.

Estos se reciben en el constructor del Builder para asegurar que siempre estén presentes.

- Atributos opcionales:

I. queso (cheese)

Indica si incluye queso (valor por defecto: falso).

II. ingredientesAdicionales (extraIngredients)

Lista de ingredientes adicionales como lechuga, tomate, cebolla, etc.

Se configuran mediante métodos encadenables del Builder.

3. Salida por consola

==== First Hamburger ===

Hamburger Details:

Bread Type: Sesame

Meat Type: Beef

Cheese: Yes

Extra Ingredients: [Lettuce, Tomato, Onion]

==== Second Hamburger ===

Hamburger Details:

Bread Type: Whole Wheat

Meat Type: Chicken

Cheese: No

Extra Ingredients: [Bacon]

4. Diagrama UML

Builder Pattern - Hamburger Example

