Taller - Aplicación Builder y Singleton

Santiago Cacua Villamizar – 01220371021

Nury Farelo Velásquez

Docente diseño de software 1

Universidad de Santander

Diseño Y Análisis De Algoritmos - BUC372602A

Bucaramanga

2024

Contenido

[Documentación del Patrón Builder: 2](#_Toc178515474)

[Diagrama UML: 4](#_Toc178515475)

[Documentación del Patrón Prototype: 4](#_Toc178515476)

[Diagrama UML: 6](#_Toc178515477)

# Documentación del Patrón Builder:

**Descripción:**  
El patrón **Builder** permite construir objetos complejos paso a paso. En este caso, se implementa un sistema para crear **Currículums Vitae** (CVs) personalizados utilizando diferentes estilos (cronológico o creativo).

**Clases:**

1. **CV (Producto):**

* Clase que representa un CV con secciones como información personal, experiencia laboral, educación, habilidades, proyectos y referencias.
* **Métodos:**
  + - \_\_init\_\_(): Inicializa un CV vacío con listas para las diferentes secciones.
    - \_\_str\_\_(): Devuelve una representación en texto del CV.

1. **CVBuilder (Interfaz):**

* Define los métodos que deben implementar los builders para crear un CV.
* **Métodos (abstractos):**
  + - set\_informacion\_personal(info): Añade información personal.
    - add\_experiencia\_laboral(exp): Añade experiencia laboral.
    - add\_educacion(edu): Añade educación.
    - add\_habilidades(hab): Añade habilidades.
    - add\_proyectos(pro): Añade proyectos.
    - add\_referencias(ref): Añade referencias.
    - get\_cv(): Devuelve el CV construido.

1. **CVBuilderCronologico (Builder Concreto):**

* Implementa CVBuilder para crear un CV de estilo cronológico.
* **Métodos:**
  + - Implementa todos los métodos de CVBuilder para llenar las secciones del CV cronológico.

1. **CVBuilderCreativo (Builder Concreto):**

* Implementa CVBuilder para crear un CV con un enfoque creativo.
* **Métodos:**
  + - Similar a CVBuilderCronologico, pero añade un enfoque creativo a las secciones del CV.

1. **GeneradorCV (Director):**

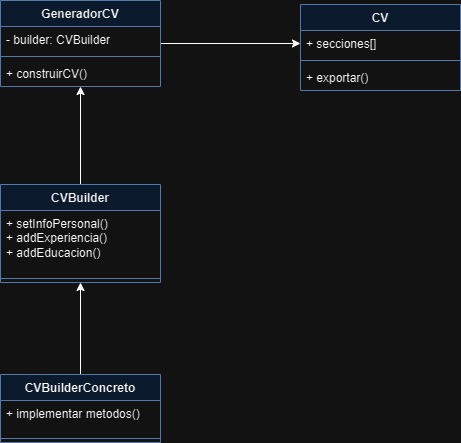
* Clase que controla el proceso de construcción del CV utilizando un CVBuilder.
* **Métodos:**
  + - \_\_init\_\_(builder): Inicializa con un builder específico.
    - construir\_cv(): Añade las diferentes secciones al CV paso a paso utilizando el builder.

Texto

Descripción generada automáticamente

Ilustración 1 Ejecución Builder + Prueba Unitaria

# Diagrama UML:



# Documentación del Patrón Prototype:

**Descripción:**  
El patrón **Prototype** permite clonar objetos a partir de un prototipo existente, sin necesidad de conocer la clase concreta del objeto. En este caso, se utiliza para clonar diferentes tipos de documentos (CV, Carta de Presentación, Informe).

**Clases:**

1. **Documento (Interfaz Prototipo):**
   * Define el método clonar() que todas las subclases deben implementar para crear copias de sí mismas.
2. **CurriculumVitae, CartaPresentacion, Informe (Prototipos Concretos):**
   * Clases que implementan Documento y que permiten la clonación de cada tipo de documento.
   * **Métodos:**
     + \_\_init\_\_(contenido): Inicializa el documento con un contenido específico.
     + clonar(): Crea una copia profunda del documento.
     + \_\_str\_\_(): Devuelve una representación en texto del documento.
3. **EditorDocumentos (Cliente):**
   * Clase que contiene una colección de plantillas de documentos y permite al usuario clonar una de ellas.
   * **Métodos:**
     + \_\_init\_\_(): Inicializa el editor con un conjunto de plantillas (CV, Carta, Informe).
     + clonar\_documento(tipo): Clona el documento correspondiente al tipo especificado.

Texto

Descripción generada automáticamente

Ilustración 2 Ejecución Prototype + Prueba Unitaria

# Diagrama UML:

Diagrama

Descripción generada automáticamente