

**Asunto:**

Implementación de patrones de diseño de software en el trabajo de la asignatura Arquitectura de Software, que trata sobre Análisis de secuencias genéticas.

**Implementación patrones de diseño:****1. Factory :**

Se utilizó el patrón factory para instanciar los diferentes tipos de análisis así:

**Paquete arquitecturas.analisis:**

| <b>PRODUCTO</b>               | <b>PROPÓSITO</b>  |
|-------------------------------|---|
| AnalisisSecuencia (Interface) | Define el método analizar, el cual recibe las secuencias a analizar, implementado para la clase AlineamientoSecuencia y la clase ConteoSecuencias   |
| AnalisisMotivos (Interface)   | Define el método analizar, el cual recibe una secuencia que debe ser analizada y devuelve el resultado en una lista, implementada por la clase MotivosSecuencia   |
| AnalisisFactory (clase)       | <ul style="list-style-type: none"><li>• Contiene el método crearAnalisis() de tipo AnalisisSecuencia el cual retorna la instancia de los métodos de análisis para AlineamientoSecuencia y ConteoSecuencias.</li><li>• Contiene el método crearListaMotivos() el cual retorna la instancia del método de análisis MotivosSecuencia</li></ul> |
| AlineamientoSecuencia (clase) | Implementa la interfaz AnalisisSecuencia y sobrescribe el método analizar, el cual recibe dos secuencias, muestra la secuencia más larga y el porcentaje de similitud entre ellas   |
| ConteoSecuencias (clase)      | Implementa la interfaz AnalisisSecuencia y sobrescribe el método analizar, el cual recibe una secuencia y muestre el tipo de estructura y su equivalencia   |
| MotivosSecuencia (clase)      | Implementa la interfaz AnalisisMotivos, el  |

|  |  |
|--|--|
|  | cual retorna una lista que contiene las posiciones en las que se encuentra un motivo |
|--|--|

## 2. Singleton :

Se utiliza para la conexión con la base de datos

### Paquete arquitecturas.creaciones.singleton

| PRODUCTO            | PROPÓSITO  |
|---------------------|--|
| ConexionBD ( clase) | Gestiona la conexión a MySql, asegurando que existe una sola conexión, es instanciada por la clase GuardarSecuencia del patron proxy |

## 3. Proxy:

Se utiliza para controlar el acceso a cada análisis, se validan ejecuciones

| PRODUCTO                 | PROPOSITO   |
|--------------------------|---|
| BDwrite (interfaz)       | método para guardar la secuencias   |
| BDwriteProxy (clase)     | implementa la interfaz BDwrite para guardar las secuencias subidas del archivo FASTA. Verifica el acceso, verifica si las secuencias ya estan guardadas, de lo contrario las guarda en base de datos. Verifica si un análisis ya esta guardado en bases de datos. |
| GuardarSecuencia (clase) | implementa la interfaz BDwrite para guardar la secuencias en base de datos, acá se instancia la clase singleton para establecer conexión y guardar cada análisis en base de datos.  |

## 4. Facade:

Se utiliza para leer el archivo FASTA, y solicitar al usuario a través de un menú que analisis desea realizar

| PRODUCTO                  | PROPOSITO   |
|---------------------------|---|
| Fachada (clase)           | Proporciona los métodos de lectura del archivo FASTA.<br>El método de selección de análisis donde se llama a los métodos correspondiente de cada análisis a través de Factory .                         |
| SecuenciaGenetica (clase) | clase de tipo secuencia genética, que construye a través del patrón Builder, los atributos idSecuencia y la secuencia como tal. A través de esta clase se instancia el objeto de tipo SecuenciaGenética |

## 5. Builder:

### Flujo de ejecución de los análisis:

1. Main : Contiene :
  - a. La ruta del archivo Fasta
  - b. Una lista de tipo Secuencia genética,
  - c. Un objeto de tipo Facade que instancia el método que lee el archivo Fasta y retorna una lista de secuencias que son almacenados en la lista declarada
  - d. Un objeto de tipo proxy que instancia el método de guardar secuencia, donde se verifica si las secuencias que posee el archivo ya están guardadas en bases de datos o no lo están.
  - e. Una instancia del método análisis de la clase Fachada, donde se le pregunta al usuario qué tipo de análisis desea realizar.
2. Fachada : Clase de tipo Facade, que contiene:
  - a. Un método que recibe el archivo desde la main, para ser leído línea por línea y construir el objeto de tipo secuencia genética asignándole los atributos al objeto, como lo es la identificación y la secuencia como tal.
  - b. Un método que es instanciado desde la main y muestra un menú de opciones para que el usuario elija que tipo de análisis desea realizar; dentro de este método se instancia la clase AnalisisFactory que posee un método donde según la opción elegida por el usuario, llama la clase correspondiente del análisis.
  - c. Un método sobre el análisis de alineamiento que le solicita al usuario que elija qué secuencias desea analizar, y envía a un método de

análisis de la instancia de AnalisisFactory para que realice los cálculos correspondientes

- d. Un método sobre el análisis de motivos, donde se le solicita al usuario que elija una secuencia de las que están guardadas en bases de datos y se le envía al método instanciado de AnalisisFactory para que realice los cálculos correspondientes.
  - e. Un método sobre análisis de predicciones, donde se le solicita al usuario que elija una secuencia de las que están guardadas en bases de datos y se le envía el método a AnalisisFactory para que realice los cálculos correspondientes.
- 3. Entidad Secuencia Genética : Clase donde que contiene los atributos id y secuencia los cuales son construidos con el método Builder. Esta clase es instanciada desde Fachada para crear la lista de tipo secuencia genética.
  - 4. BDwrite : Interza de contiene un método de tipo boolean que recibe la lista de la secuencia genética llamado guardarSecuencia .
  - 5. BDwirteProxy : Clase que contiene los métodos de validación e implementa la interfaz BDwrite :
    - a. Sobreescribe el método de la interfaz BDwrite guardarSecuencias, el cual solicita el nombre del archivo donde se guarda el reporte, verifica si la secuencia ya existe en la BD, si ya existe retorna un true, de lo contrario llama el método guardarSecuencias de la clase GuardarSecuencia y envía la lista de secuencias para ser guardada.
    - b. Un método que verifica si el análisis de alineamiento ya está guardado en BD, si no está guardado llama el método guardarAnalisisAli de la clase GuardarSecuencia y envía todos los resultados del análisis.
    - c. Un método que verifica si el análisis de motivos ya está guardado en BD, donde se llama el método verificarAnalisisMotivos de la clase GuardarSecuencia y retorna una lista de enteros que contiene las posiciones donde se encuentra el motivo en la secuencia.
    - d. Un método que guarda los motivos llamando el método guardarAnalisisMotivos de la clase GuardarSecuencia
    - e. Un método que verifica si el análisis de tipo de estructura ya está guardado en BD llamando el método verificarTipoEstructura de la clase GuardarSecuencia,
    - f. Un método que guarda el análisis de tipo de estructura llamando el metodo guardarTipoEstructura de la clase GuardarSecuencia,
  - 6. GuardarSecuencia : clase que instancia la clase ConexionBD de tipo Singleton, y contiene:
    - a. Sobreescribe el método de la interfaz BDwrite guardarSecuencias, el cual guarda en BD las secuencias de tipo SecuenciasGeneticas y retorna un boolean si la información pudo ser guardada.
    - b. Método verificarSecuencias que recibe las secuencias de tipo SecuenciasGeneticas y realiza la consulta a BD, si existen retorna un true, de lo contrario retorna un false.

- c. Método `verificarAnálisisAlineamiento` que recibe las dos secuencias elegidas para el análisis, realiza la consulta en BD para verificar si el análisis ya existe por las dos secuencias elegidas.
  - d. Método `guardarAnálisisAli` que recibe las dos secuencias elegidas para el análisis, la longitud de cada secuencia, la secuencia más larga y la similitud y los guarda en BD si no existe ese análisis por las dos secuencias elegidas.
  - e. Método `verificarAnálisisMotivos` que recibe la secuencia elegida para el análisis y el motivo que desea analizar, realiza la consulta en BD para verificar si el análisis por esa secuencia y motivo ya existe, si existe retorna la lista de las posiciones que ocupa el motivo en la secuencia, de lo contrario retorna una lista vacía.
  - f. Método `guardarAnálisisMotivos` que recibe la secuencia elegida para el análisis, el motivo, y la lista de tipo entero que guarda las posiciones en los que encuentra el motivo en la secuencia y los guarda en BD si no existe ese análisis.
  - g. Método `verificarTipoEstructura` que recibe la secuencia elegida para el análisis, realiza la consulta a BD si el análisis por esa secuencia ya existe.
  - h. Método `guardarTipoEstructura` que recibe la secuencia elegida para el análisis, el tipo de estructura, el porcentaje equivalente a alfa y el equivalente al beta y los guarda en BD si no existe el análisis.
7. `ConexionBD` : Clase de tipo Singleton donde se configura la conexión a la BD y que es instanciada por la clase `GuardarSecuencia`.
8. `AnalisisFactory` : Clase de tipo Factory que contiene los métodos:
- a. `crearAnálisis` que recibe de la opción de análisis elegida por el usuario, retorna la creación de una instancia de la clase que se encarga del análisis correspondiente:
    - i. `AlineamientoSecuencia`, es una clase construida por la Factory, la cual sobrescribe el método `analizar` de la interfaz `AnalisisSecuencia`, la cual se encarga de realizar los cálculos para determinar longitud de las secuencias elegidas, secuencia más larga y la similitud entre las dos secuencias.
    - ii. `ConteoSecuencia`, es una clase construida por la Factory la cual sobrescribe el método `analizar` de la interfaz `AnalisisSecuencia`, la cual se encarga de realizar los cálculos para determinar el tipo de estructura, si es alfa, beta o mixta.
  - b. `crearListaMotivos` que se encarga de la creación de la clase `MotivosSecuencia`, la cual sobrescribe el método `analizar` de la interfaz `AnalisisMotivos` la cual se encarga de determinar en qué posiciones de la secuencia se encuentra un motivo elegido.
9. `ReporteBuilder`: Interfaz que contiene los métodos `reporteAlineamientos`, `reporteMotivos`, `reporteConteos`.

10. GeneradorReporteBuilder : Clase que implementa ReporteBuilder y sobre escribe los métodos:

- reporteAlineamiento: genera la parte del reporte que tiene que ver con el análisis de alineamiento realizado
- reporteMotivos: genera la parte del reporte que tiene que ver con el análisis de la cantidad y posiciones en la que se encuentra el motivo elegido.
- reporteConteos: genera la parte del reporte que tiene que ver con el análisis que determina el tipo de estructura.

MR:



