Creacionales:

Para el diagrama de clases se utilizaron dos patrones el primero fue Factory method y el segundo fue singleton.

Los demás no se utilizaron abtract method no fue necesario además ya se está implementando el patrón Factory method.

Para el prototype tampoco lo necesitamos puesto que no había necesidad de crear objetos nuevos clonando la instancia.

Patron builder permite la creación de variedad de objetos complejos desde un objeto fuente pero ya se utiliza Factory method para la creación de objetos además dentro del sistema no hay objetos complejos.

Estructurales:

Fachada no se utilizó puesto que no había un sistema complejo para ocultar ni tampoco hay subsistemas.

el patrón adapter puesto no hay necesidad de transformar una interfaz a otra ni por idioma ni por ningún otro detalle mínimo además es sistema no cumple una funcionalidad de ese tipo.

Composite : el sistema no posee muchas herencia y los objetos son simple no hay objetos complejos.

Comportamiento:

Chain of responsabilities: No había necesidad de encadenar responsabilidades,además es usualmente utilizado en interfaces graficas pero el sistema que se crea no utiliza interfaces graficas.