

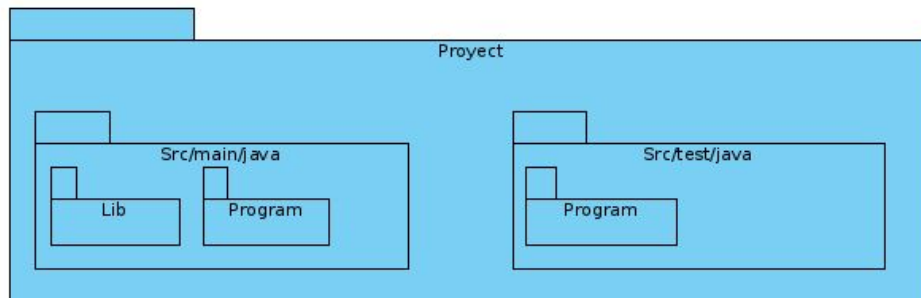
## Diseño del Sistema

### Diseño del Sistema

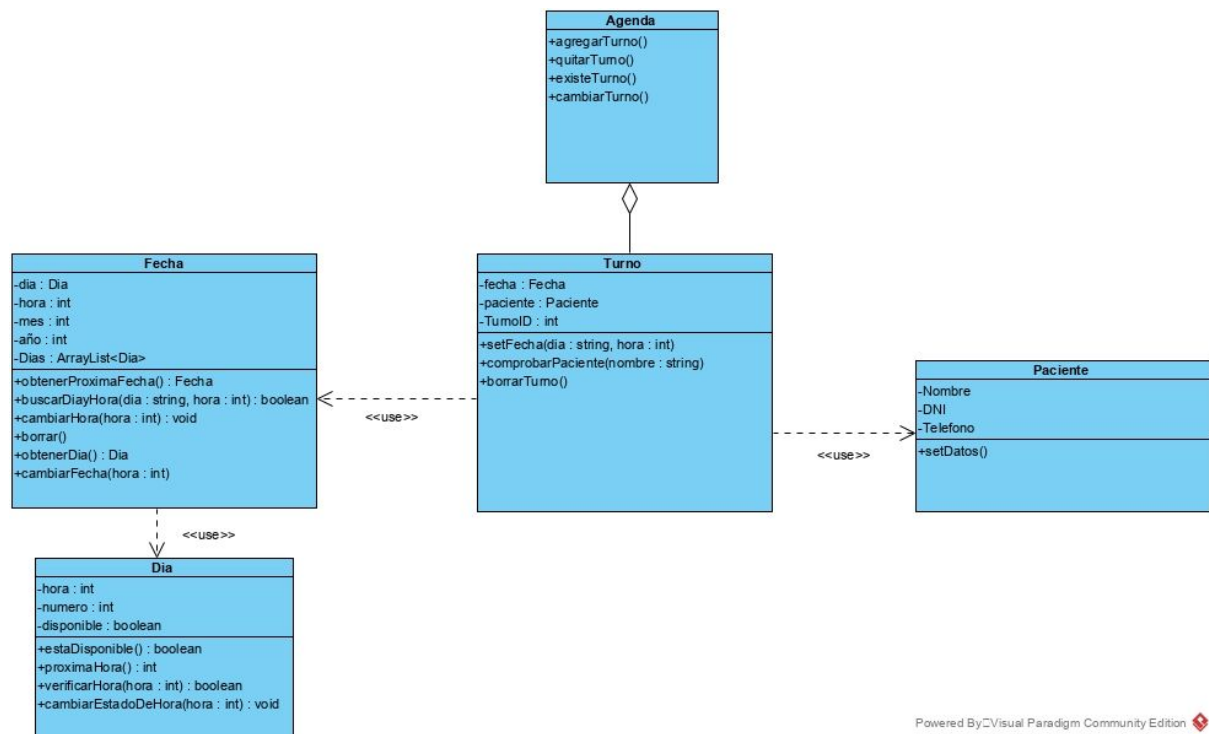
Se debe presentar un **documento de diseño** en el cual se incluyan diagramas de paquetes, diagramas de clases y objetos, diagramas de secuencia y todo aquel diagrama que sirva para explicar el diseño del software a construir.

Se debe mostrar también la aplicación de patrones de diseño como el Singleton, Observer, Strategy, etcétera usando diagramas de clases e indicar por qué se utilizan y qué problemas se solucionan con ellos.

Por otro lado, se deben generar **pruebas unitarias automáticas para el código**, explicando cómo correrlas y verificar su estado. Actualizar la Matriz de trazabilidad para incluir módulos o clases y casos de pruebas unitarias.



### Diagrama de paquetes



Powered By Visual Paradigm Community Edition

Diagrama de clases

Diagrama de objetos

Diagrama de secuencia

Parte de patrones de diseño

Observador/ Observer es un patrón de diseño de software que define una dependencia del tipo uno a muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, notifica este cambio a todos los dependientes.

Se implementará el patrón de diseño Observer para poder modificar y actualizar las estadísticas de los tiempos ocupados del médico. Lo que harán las vistas del sistema, es suscribirse a la de esta forma se evita que la clase enemigo pregunte constantemente si También se implementó este patrón para actualizar a la clase App de los cambios de HP del personaje para mostrarlos en pantalla.

Pruebas unitarias

