



La solución propuesta utiliza 3 patrones de diseño:

- 1. Patrón Adapter:** Se definió la interfaz **MTA** y las clases **SendmailAdapter** y **PostfixAdapter** que la implementan, con el fin de convertir los métodos **getMail()** y **lookupEmail()** que proveen las clases **SendmailAdapter** y **PostfixAdapter** a un mensaje que espera el cliente (**recibirMail()**).
- 2. Patrón Composite:** Se aplica este patrón para las clases **Mail** y **Carpeta**, tratando a los mails como objetos simples y a las carpetas como objetos complejos que contiene otras carpetas o mails y permitir crear una jerarquía de elementos.
- 3. Patrón Strategy:** Se define el atributo **formatoExport** para cada carpeta y mail, que determina el formato en el que se desea exportar el elemento. Es conveniente utilizar este patrón conociendo que para cada formato el algoritmo de **exportar()** es diferente, la utilización de este patrón permite cambiar de algoritmo en forma dinámica y añadir nuevos formatos de exportación con mayor facilidad (agregando subclases de **FormatoExportacion**).