5.1 C- JAVA Api for XML Processing

5.2 A- Simple API for XML

5.3 B-Document Object Model

5.4 B-Ocupa menos en memoria

5.5 C- es más versátil

5.6 C- Creación-almacenamiento-recuperación-procesado

5.7 C- ficheros

5.8 C- Document

5.9 B- como un campo de texto

5.10 C- Es un lenguaje de programación basado en XML

5.11 B- todos los elementos <personaje> independientemente de su ubicación

5.12 C- todos los elementos <nombre> que son hijos de un elemento <personaje>

5.13 B- todos los atributos de todos los nodos

5.14 C- XReference

5.15 Lo más recomendable seria usarlo cuando tenemos claro el contenido para poder crear los métodos correspondientes y que no nos de error, cuando lo vaya recorriendo. Un buen ejemplo sería una página que haga consultas, ya que los campos tendrán bastante control de tipos de datos, entrada de información.

5.16 Este operador es más útil cuando vamos a tener la que modificar el documento en cuestión, al usar los nodos para localizar donde quiere hacer la modificación, también es más fácil al usar las relaciones de parentesco. Como ejemplo para usarlo, probaría usar con HTML de noticas, ya que así podemos adaptar la información que le damos al cliente y adatarlo a sus preferencias.

5.17 Yo usaría SAX ya que al tener los datos de entrada claros y sin cambios, haría un método con bucle de dos partes. La primera parte del bucle ira recorriendo los canales, cada vez que tenga uno entra en el segundo bucle y mandaría el correo con los datos pedidos, tendría un Scanner para poder introducir el correo que se mandara la información.

5.18 En este caso usaría de nuevo SAX ya que con el método correcto para que recorrer la información y luego otro método para que el usuario puede pedir cierto canal y poner sus datos para recibir la información.

5.19 Dependerá de lo que necesite hacer con él, si es solo revisar, mover y leer los datos usaría SAX pero si va ser necesario modificar o darle más orden a los datos entonces usaría DOM, de partida usaría DOM ya que me permite guardad los datos en mi memoria y poder buscar cualquier parte del documento por medio de los árboles y nodos.