|  |  |
| --- | --- |
| ASCII | Unicode |
| **Ventajas:**   * Se puede utilizar muy bien con sistemas ajenos * Adecuado para cantidades de datos medias (<= 4096 Bytes) * Muy buen rendimiento, ya que no hay ni cabecera ni elaboración de ningún procedimiento * Protocolo libre para la transferencia de todos los formatos de carácter   **Desventaja:**   * La transferencia de datos no se acusa * Seguridad de datos baja (distancia Hamming = 1, ya que sólo se utiliza el bit de paridad) * Necesita la llamada coordinada de las funciones de envío y recepción en ambas partes | **Ventajas**   * UTF-8 permite codificar cualquier carácter Unicode * Es compatible con US-ASCII, la codificación del repertorio de 7 bits es directa. * Fácil identificación. Es posible identificar claramente una muestra de datos como UTF-8 mediante un sencillo algoritmo. La probabilidad de una identificación correcta aumenta con el tamaño de la muestra. * UTF-8 ahorrará espacio de almacenamiento para textos en caracteres latinos, donde los caracteres incluidos en US-ASCII son comunes, cuando se compara con otros formatos como UTF-16 * Una secuencia de bytes para un carácter jamás será parte de una secuencia más larga de otro carácter por contener información de sincronización.   **Desventajas**   * UTF-8 utiliza símbolos de longitud variable; eso significa que diferentes caracteres pueden codificarse con distinto número de bytes. Es necesario recorrer la cadena desde el inicio para encontrar el carácter que ocupa una determinada posición. * Los caracteres ideográficos usan 3 bytes en UTF-8, pero solo 2 en UTF-16. Así, los textos chinos, japoneses o coreanos ocupan más espacio cuando se representan en UTF-8. * UTF-8 ofrece peor rendimiento que UTF-16 y UTF-32 en cuanto a coste de computación, por ejemplo, en operaciones de ordenación. |