



CAPA 6 MODELO OSI.

LUIS EMMANUEL LOPEZ ORTIZ	20760619
CALLERO SANCHEZ RUBEN	19760767



La capa de presentación

Se encarga de la representación y el formato de los datos que se intercambian entre aplicaciones.

Esto implica

1. Formateo: La traducción implica convertir datos de un formato específico utilizado por la aplicación a una representación comúnmente aceptada que puede ser interpretada por la capa de sesión y la capas posteriores del modelo OSI.
2. Compresión y descompresión : son técnicas utilizadas para reducir el tamaño de los datos antes de enviarlos a través de una red y luego restaurarlos a su forma original una vez que se reciben en el destino.
3. Encriptación y desenscriptación:



1. FORMATEO

Algunos de los métodos para traducción de formatos son:

1. **Mapeo Directo (Direct Mapping):** Si dos aplicaciones utilizan diferentes códigos para representar colores (RGB y HEX), la Capa de Presentación puede mapear directamente los valores de un formato al otro.
2. **Uso de Tablas (Look-up Tables):** Por ejemplo, un servidor de bases de datos puede utilizar una tabla de traducción para convertir códigos numéricos en nombres de países.
3. **Parsing y Serialización:**
4. **Negociación de Contenido (Content Negotiation):** Por ejemplo, una aplicación cliente puede incluir una lista de formatos de datos que puede aceptar en su solicitud y el servidor seleccionará el formato más adecuado para la respuesta en función de las preferencias del cliente.
5. **Transcodificación:** Por ejemplo, la transcodificación de un archivo de audio de formato WAV a formato MP3 implica cambios significativos en la estructura de datos y el tamaño del archivo.
6. **Normalización:** Por ejemplo, en el procesamiento de texto, se pueden normalizar los espacios y mayúsculas/minúsculas antes de la traducción.

Capa 6: Estándares

- ✦ *El uso de formatos estándares permiten el intercambio de datos de aplicación entre diferentes Sistemas Operativos.*
- ✦ *Para texto se tiene las siguientes representaciones:*
 - ✦ *ASCII (Código americano normalizado para el intercambio de la información)*
 - ✦ *EBCDIC (Código ampliado de caracteres decimales codificados en binario)*
- ✦ *Las imágenes gráficas tienen sus estándares de capa 6:*
 - ✦ *PICT : Formato de imagen utilizado para transferir gráficos QuickDraw entre programas MAC OS*
 - ✦ *TIFF (Formato de archivo de imagen etiquetado): Formato para imágenes con asignación de bits de alta resolución*
 - ✦ *JPEG (Grupo conjunto de expertos fotográficos): Formato gráfico utilizado para comprimir imágenes fijas de ilustraciones o fotografías complejos*

EJEMPLO


Como ejemplo daremos seguimiento a un mensaje proveniente de la capa 7 aplicación ,que envía mensajes entre dos usuarios, Alice y Bob

Paso 1: Formateo

1. Mensaje Original (Texto sin formato): Alice quiere enviar el mensaje "¡Hola Bob! ¿Cómo estás?" a Bob.
2. Conversión a formato JSON: La Capa de Presentación toma el mensaje original y lo convierte a un formato JSON estructurado para que pueda ser transmitido de manera más eficiente y comprensible a través de la red.

Conversión a formato JSON

```
{  
  "sender": "Alice",  
  "recipient": "Bob",  
  "content": "¡Hola Bob! ¿Cómo estás?"  
}
```



Ejemplo 2: Consideremos un escenario donde una aplicación cliente en una computadora envía una solicitud HTTP a un servidor web para obtener una página web. La solicitud incluye datos en formato de texto plano, y el servidor web espera recibir los datos en formato JSON.

Solicitud del cliente (texto plano):

```
GET /index.html HTTP/1.1
Host: www.example.com
Accept: text/plain
```

Solicitud traducida por la Capa de Presentación (JSON):

```
{
  "method": "GET",
  "path": "/index.html",
  "protocol": "HTTP/1.1",
  "headers": {
    "Host": "www.example.com",
    "Accept": "text/plain"
  }
}
```



2. COMPRESIÓN Y DESCOMPRESIÓN DE DATOS

1. **Selección del algoritmo:** Se elige un algoritmo de compresión adecuado, como gzip, deflate o zip, que se encargará de comprimir los datos. Cada algoritmo tiene sus propias características y eficiencia en diferentes tipos de datos.
2. **División en bloques:** Los datos se dividen en bloques más pequeños para que el algoritmo de compresión pueda trabajar de manera eficiente. Esto también permite que la compresión sea paralelizada, lo que puede acelerar el proceso en sistemas con múltiples núcleos o procesadores.
3. **Compresión:** El algoritmo de compresión busca patrones repetitivos y redundancias dentro de los bloques de datos y reemplaza esas repeticiones por referencias más cortas o códigos. De esta manera, el tamaño total de los datos se reduce, lo que ahorra ancho de banda durante la transmisión.



ALGORITMOS DE COMPRESION Y DESCOMPRESION EXISTENTES

1. Deflate
2. Gzip
3. Bzip2
4. LZMA (Lempel-Ziv-Markov chain Algorithm)
5. Zip:
6. JPEG (Joint Photographic Experts Group)
7. PNG (Portable Network Graphics)
8. FLAC (Free Lossless Audio Codec)



EJEMPLO

Paso 2: Compresión

Compresión del mensaje: La Capa de Presentación comprime el mensaje JSON utilizando un algoritmo de compresión, como gzip, para reducir su tamaño y ahorrar ancho de banda durante la transmisión.

Mensaje Comprimido (ejemplo simplificado):

xK1TYr8lcgy5kuyuWjAqp9YTYQ==

2. ENCRYPTACIÓN Y DESENCRIPTACIÓN

La encriptación y desencriptación son técnicas utilizadas para proteger la confidencialidad de la información durante su transmisión o almacenamiento. La encriptación convierte los datos en un formato ilegible (texto cifrado) utilizando un algoritmo y una clave, mientras que la desencriptación es el proceso inverso para recuperar los datos originales.





EJEMPLO

Paso 3: Encriptación

Generación de clave secreta: Alice y Bob acuerdan previamente una clave secreta compartida que solo ellos conocen. Esta clave será utilizada para encriptar y desencriptar los mensajes.

Clave secreta compartida (ejemplo simplificado):
mG9sN4zPbD3kH6rT

Encriptación del mensaje: La Capa de Presentación encripta el mensaje comprimido utilizando un algoritmo de encriptación simétrico, como AES, y la clave secreta compartida:

Mensaje Encriptado (texto cifrado, ejemplo simplificado):
fGJ83Lm9zp/WzdtlZJ78S6x0wEzx5JwpP8enQ1Ht+vMA0JGleBvZ9LV2Q==