

**EL PROTOCOLO  
INALÁMBRICO DE  
APLICACIONES (WAP)**

---

# ÍNDICE

<b><i>¿QUÉ ES EL PROTOCOLO DE APLICACIONES INALÁMBRICAS?</i></b>	<b>4</b>
<b><i>COMPONENTES DE LA ARQUITECTURA WAP</i></b>	<b>7</b>
<b><i>CAPA DE APLICACIÓN (WAE)</i></b>	<b>7</b>
<b><i>CAPA DE SESIÓN (WSP)</i></b>	<b>8</b>
<b><i>CAPA DE TRANSACCIONES (WTP)</i></b>	<b>8</b>
<b><i>CAPA DE SEGURIDAD (WTLS)</i></b>	<b>9</b>
<b><i>CAPA DE TRANSPORTE (WDP)</i></b>	<b>9</b>
<b><i>EL ENTORNO INALÁMBRICO DE APLICACIONES</i></b>	<b>10</b>
<b><i>EL PROTOCOLO INALÁMBRICO DE SESIÓN</i></b>	<b>12</b>
<b><i>EL PROTOCOLO INALÁMBRICO DE TRANSACCIÓN</i></b>	<b>14</b>
<b><i>LA CAPA INALÁMBRICA DE SEGURIDAD DE TRANSPORTE</i></b>	<b>17</b>
<b><i>EL PROTOCOLO INALÁMBRICO DE DATAGRAMAS</i></b>	<b>19</b>
<b><i>APÉNDICE A: GLOSARIO DE TÉRMINOS</i></b>	<b>23</b>
<b><i>BIBLIOGRAFÍA</i></b>	<b>24</b>

---

# ÍNDICE DE FIGURAS

<i>Figura 1: Modelo de funcionamiento del WAP</i>	4
<i>Figura 2: Ejemplo de una red WAP</i>	5
<i>Figura 3: Arquitectura de WAP</i>	7
<i>Figura 4: Ejemplo de capas en WAP</i>	10
<i>Figura 5: Componentes del Cliente de WAE</i>	11
<i>Figura 6: Ejemplo intercambio de primitivas entre capa Sesión y Transacción</i>	16
<i>Figura 7: Secuencia de Primitivas para el establecimiento de una sesión segura</i>	19
<i>Figura 8: Arquitectura del Protocolo Inalámbrico de Datagramas</i>	20
<i>Figura 9: WDP sobre GSM SMS</i>	21
<i>Figura 10: WDP sobre GSM Canal de Datos de Circuitos Conmutados</i>	21
<i>Figura 11: WDP sobre Servicios Portadores CDMA</i>	22

---

# ÍNDICE DE TABLAS

<i>Tabla 1: Primitivas de Servicio de Sesión</i>	13
<i>Tabla 2: Tipos de Primitivas de Servicio.</i>	14
<i>Tabla 3: Primitivas de Servicio de Transacción</i>	15
<i>Tabla 4: Primitivas de Servicio de Capa de Seguridad</i>	18
<i>Tabla 5: Primitivas de Servicio de la Capa de Datagramas</i>	20

---

## ¿Qué es el *Protocolo de Aplicaciones Inalámbricas*?

El *Protocolo de Aplicaciones Inalámbricas* surge como la combinación de dos tecnologías de amplio crecimiento y difusión durante los últimos años: *Las Comunicaciones Inalámbricas* e *Internet*. Mas allá de la posibilidad de acceder a los servicios de información contenidos en Internet, el protocolo pretende proveer de servicios avanzados adicionales como, por ejemplo, el desvío de llamadas inteligente, en el cual se proporcione una interfaz al usuario en el cual se le pregunte la acción que desea realizar: aceptar la llamada, desviarla a otra persona, desviarla a un buzón vocal, etc.

Para ello, se parte de una arquitectura basada en la arquitectura definida para el *World Wide Web (WWW)*, pero adaptada a los nuevos requisitos del sistema. En la Figura 1 se muestra el esquema de la arquitectura WAP.

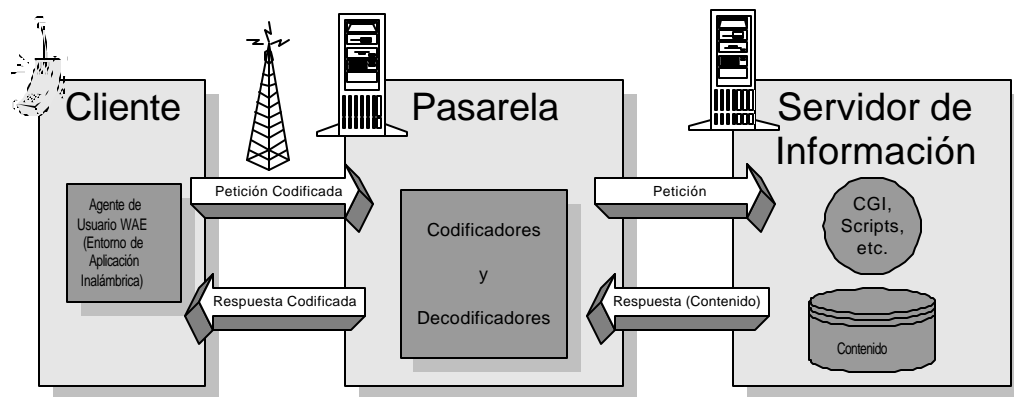


Figura 1: Modelo de funcionamiento del WAP

De esta forma, en el terminal inalámbrico existiría un “*micro navegador*”<sup>1</sup> encargado de la coordinación con la pasarela, a la cual la realiza peticiones de información que son adecuadamente tratadas y redirigidas al servidor de información adecuado. Una vez procesada la petición de información en el servidor, se envía esta información a la pasarela que de nuevo procesa adecuadamente para enviarlo al terminal inalámbrico.

Para conseguir consistencia en la comunicación entre el terminal móvil y los servidores de red que proporcionan la información, WAP define un conjunto de componentes estándar:

---

<sup>1</sup> Se pretende que este *micro navegador* actúe de interfaz con el usuario de la misma forma que lo hacen los navegadores estándar.

- ✓ Un modelo de nombres estándar. Se utilizan las URIs<sup>2</sup> definidas en WWW para identificar los recursos locales del dispositivo (tales como funciones de control de llamada) y las URLs<sup>3</sup> (también definidas en el WWW) para identificar el contenido WAP en los servidores de información.
- ✓ Un formato de contenido estándar, basado en la tecnología WWW.
- ✓ Unos protocolos de comunicación estándares, que permitan la comunicación del *micro navegador* del terminal móvil con el servidor Web en red.

Veamos ahora un modelo global de funcionamiento de este sistema en la Figura 2.

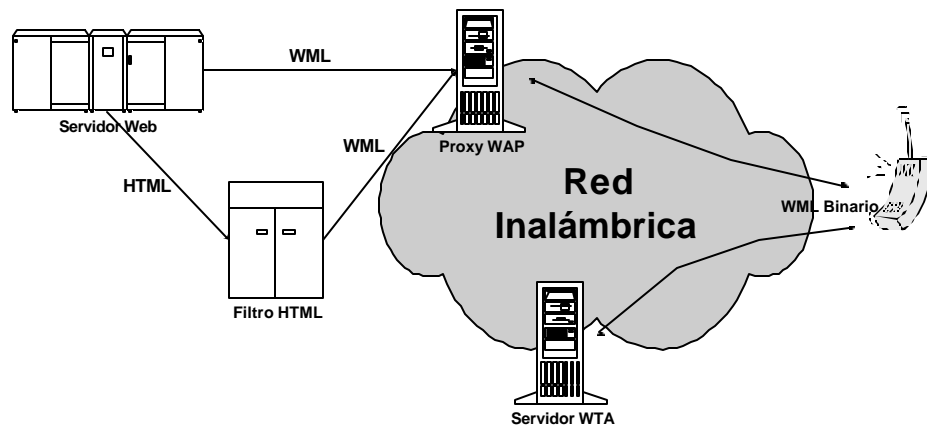


Figura 2: Ejemplo de una red WAP

En el ejemplo de la figura, nuestro terminal móvil tiene dos posibilidades de conexión: a un proxy WAP, o a un servidor WTA. El primero de ellos, el proxy WAP traduce las peticiones WAP a peticiones Web, de forma que el cliente WAP (el terminal inalámbrico) pueda realizar peticiones de información al servidor Web. Adicionalmente, este proxy codifica las respuestas del servidor Web en un formato binario compacto, que es interpretable por el cliente. Por otra parte, el segundo de ellos, el Servidor WTA<sup>4</sup> está pensado para proporcionar acceso WAP a las facilidades

<sup>2</sup> *Universal/Uniform Resource Identifier* ó Identificador Uniforme/Universal de Recurso

<sup>3</sup> *Universal/Uniform Resource Location* ó Localización Universal/Uniforme de Recurso

<sup>4</sup> *Wireless Telephony Application* ó Aplicación de Telefonía Inalámbrica

---

proporcionadas por la infraestructura de telecomunicaciones del proveedor de conexiones de red.

---

## Componentes de la Arquitectura WAP

---

Una vez introducido el sistema, vamos a ver la arquitectura que le da consistencia. La arquitectura WAP está pensada para proporcionar un “*entorno escalable y extensible para el desarrollo de aplicaciones para dispositivos de comunicación móvil*”. Para ello, se define una estructura en capas, en la cual cada capa es accesible por la capa superior así como por otros servicios y aplicaciones a través de un conjunto de interfaces muy bien definidos y especificados. Este esquema de capas de la arquitectura WAP la podemos ver en la Figura 3.

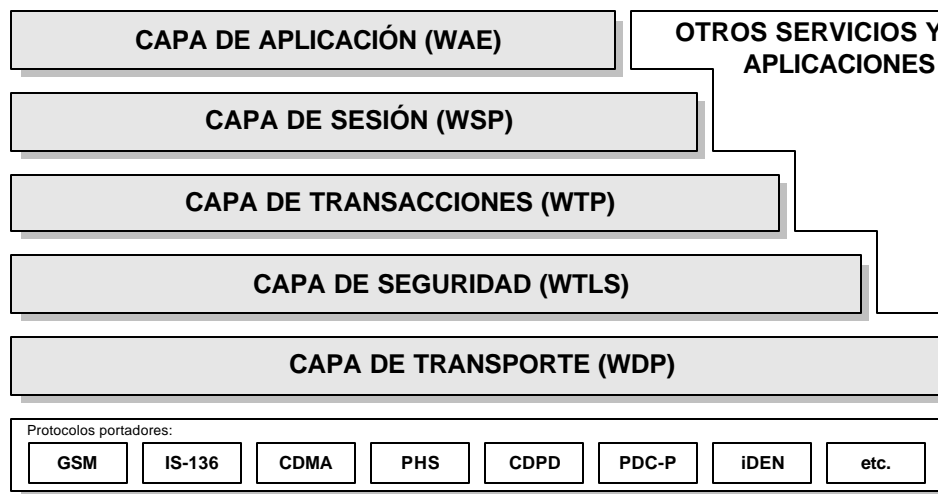


Figura 3: Arquitectura de WAP

Hagamos un recorrido por estas capas de forma breve, antes de pasar a analizarlas con más profundidad.

### CAPA DE APLICACIÓN (WAE<sup>5</sup>)

El *Entorno Inalámbrico de Aplicación (WAE)* es un entorno de aplicación de propósito general basado en la combinación del *World Wide Web* y tecnologías de Comunicaciones Móviles.

Este entorno incluye un *micro navegador*, del cual ya hemos hablado anteriormente, que posee las siguientes funcionalidades:

- ✓ Un lenguaje denominado WML<sup>6</sup> similar al HTML, pero optimizado para su uso en terminales móviles.

---

<sup>5</sup> *Wireless Application Environment* ó Entorno Inalámbrico de Aplicación

<sup>6</sup> *Wireless Markup Language*

- 
- ✓ Un lenguaje denominado *WMLScript*, similar al *JavaScript* (esto es, un lenguaje para su uso en forma de *Script*)
  - ✓ Un conjunto de formatos de contenido, que son un conjunto de formatos de datos bien definidos entre los que se encuentran imágenes, entradas en la agenda de teléfonos e información de calendario.

### **CAPA DE SESIÓN (WSP<sup>7</sup>)**

El *Protocolo Inalámbrico de Sesión (WSP)* proporciona a la Capa de Aplicación de WAP interfaz con dos servicios de sesión: Un servicio orientado a conexión que funciona por encima de la Capa de Transacciones y un servicio no orientado a conexión que funciona por encima de la Capa de Transporte (y que proporciona servicio de datagramas seguro o servicio de datagramas no seguro)

Actualmente, esta capa consiste en servicios adaptados a aplicaciones basadas en la navegación Web, proporcionando las siguientes funcionalidades:

- ✓ Semántica y funcionalidades del HTTP/1.1 en una codificación compacta.
- ✓ Negociación de las características del Protocolo.
- ✓ Suspensión de la Sesión y reanudación de la misma con cambio de sesión.

### **CAPA DE TRANSACCIONES (WTP<sup>8</sup>)**

El *Protocolo Inalámbrico de Transacción (WTP)* funciona por encima de un servicio de datagramas, tanto seguros como no seguros, proporcionando las siguientes funcionalidades:

- ✓ Tres clases de servicio de transacciones:
  - Peticiones inseguras de un solo camino.
  - Peticiones seguras de un solo camino.
  - Transacciones seguras de dos caminos (petición-respuesta)
- ✓ Seguridad usuario-a-usuario opcional.
- ✓ Transacciones asíncronas.

---

<sup>7</sup> *Wireless Session Protocol* ó Protocolo Inalámbrico de Sesión

<sup>8</sup> *Wireless Transaction Protocol* ó Protocolo Inalámbrico de Transacción.

---

## **CAPA DE SEGURIDAD (WTLS<sup>9</sup>)**

La *Capa Inalámbrica de Seguridad de Transporte (WTLS)* es un protocolo basado en el estándar SSL, utilizado en el entorno Web para la proporción de seguridad en la realización de transferencias de datos. Este protocolo ha sido especialmente diseñado para los protocolos de transporte de WAP y optimizado para ser utilizado en canales de comunicación de banda estrecha. Para este protocolo se han definido las siguientes características:

- ✓ Integridad de los datos. Este protocolo asegura que los datos intercambiados entre el terminal y un servidor de aplicaciones no ha sido modificada y no es información corrupta.
- ✓ Privacidad de los datos. Este protocolo asegura que la información intercambiada entre el terminal y un servidor de aplicaciones no puede ser entendida por terceras partes que puedan interceptar el flujo de datos.
- ✓ Autenticación. Este protocolo contiene servicios para establecer la autenticidad del terminal y del servidor de aplicaciones.

Adicionalmente, el WTLS puede ser utilizado para la realización de comunicación segura entre terminales, por ejemplo en el caso de operaciones de comercio electrónico entre terminales móviles.

## **CAPA DE TRANSPORTE (WDP<sup>10</sup>)**

El *Protocolo Inalámbrico de Datagramas (WDP)* proporciona un servicio fiable a los protocolos de las capas superiores de WAP y permite la comunicación de forma transparente sobre los protocolos portadores válidos.

Debido a que este protocolo proporciona un interfaz común a los protocolos de las capas superiores, las capas de Seguridad, Sesión y Aplicación pueden trabajar independientemente de la red inalámbrica que dé soporte al sistema.

Antes de pasar a estudiar en más profundidad cada uno de estos protocolos, veamos tres ejemplos de interconexión de estas capas en la Figura 4:

---

<sup>9</sup> *Wireless Transport Layer Security* ó Capa Inalámbrica de Seguridad de Transporte

<sup>10</sup> *Wireless Datagram Protocol* ó Protocolo Inalámbrico de Datagramas

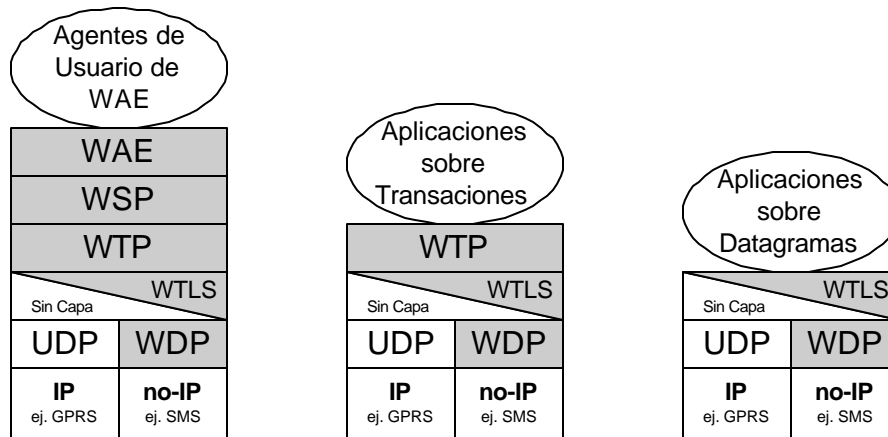


Figura 4: Ejemplo de capas en WAP

Así pues, dependiendo de la aplicación en cuestión, la comunicación se realizará con una determinada capa de la estructura de WAP.

### **El Entorno Inalámbrico de Aplicaciones**

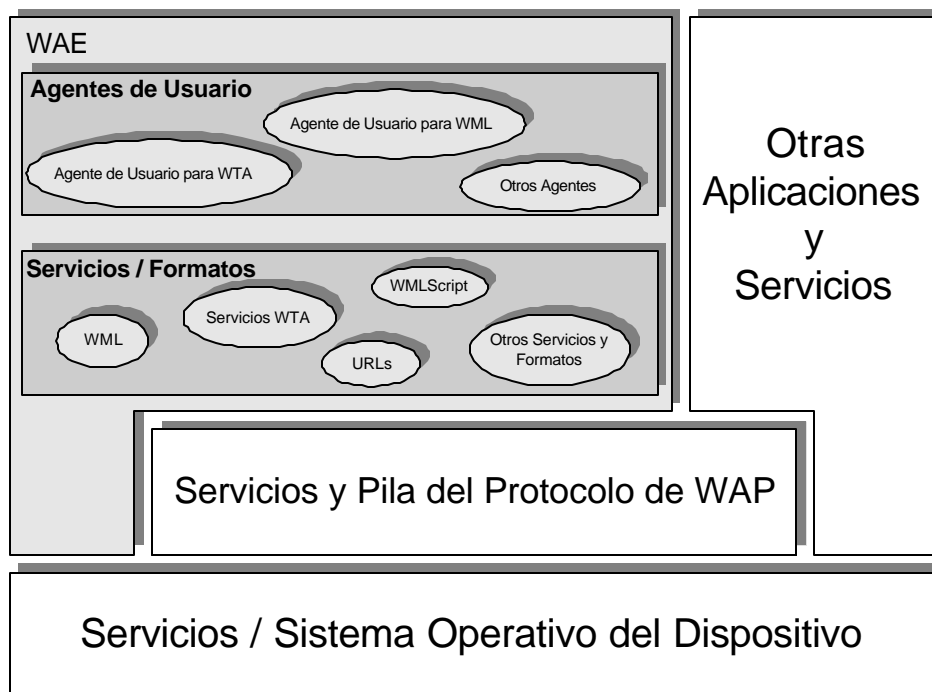
El objetivo del *Entorno Inalámbrico de Aplicaciones* es construir un entorno de aplicación de propósito general, basado fundamentalmente en la filosofía y tecnología del World Wide Web (WWW). Principalmente, se pretende establecer un entorno que permita a los operadores y proveedores de servicios construir aplicaciones y servicios que puedan utilizarse en una amplia variedad de plataformas inalámbricas de forma útil y eficiente.

De esta forma, la arquitectura del Entorno Inalámbrico de Aplicaciones (en adelante WAE) está enfocado principalmente sobre los aspectos del cliente de la arquitectura del sistema de WAP, esto es, de los puntos relacionados con los agentes de usuario<sup>11</sup>. Esto es debido a que la parte que más interesa de la arquitectura es aquella que afecta principalmente a los terminales móviles, esto es, a aquellos puntos en los cuales van a estar ejecutándose los diversos agentes de usuario.

Si volvemos sobre la *Figura 1*, vemos que entre los agentes de usuario localizados en el cliente (en el terminal móvil) y los servidores de información se define un nuevo elemento: *Las Pasarelas*. Su función es codificar y decodificar la información intercambiada con el cliente, para así minimizar la cantidad de datos radiados, así como minimizar el proceso de la información por parte del cliente.

<sup>11</sup> **Agentes de usuario:** Un agente de usuario es todo aquel software o dispositivo que interpreta un contenido, p. e. WML. Esto incluye navegadores de texto, navegadores de voz, sistemas de búsqueda, etc.

Basándonos en esta arquitectura, vamos a profundizar un poco más en los componentes de este Entorno Inalámbrico de Aplicación. Tal y como podemos observar en la Figura 5, se divide en dos partes, dos capas lógicas:



**Figura 5: Componentes del Cliente de WAE**

- ✓ Los Agentes de Usuario, que incluye aquellos elementos como navegadores, agendas telefónicas, editores de mensajes, etc.
- ✓ Los Servicios y Formatos, que incluyen todos aquellos elementos y formatos comunes, accesibles a los Agentes de Usuario, tales como WML, WMLScript, formatos de imagen, etc.

Como se puede ver en la Figura, dentro de WAE se separan Servicios de Agentes de Usuario, lo que proporciona flexibilidad para combinar varios Servicios dentro de un único Agente de Usuario, o para distribuir los Servicios entre varios Agentes de Usuario.

Los dos Agentes de Usuario más importantes son el Agente de Usuario para WML y el Agente de Usuario para WTA.

El Agente de Usuario para WML es el Agente de Usuario fundamental en la arquitectura del Entorno Inalámbrico de Aplicación. A pesar de su importancia, este Agente de Usuario no está definido formalmente dentro de esta arquitectura, ya que sus características y capacidades se dejan en manos de

---

los encargados de su implementación. El único requisito de funcionalidad que debe cumplir este Agente de Usuario, es el proporcionar un sistema intérprete a los lenguajes WML y WMLScript, de forma que se permita la navegación desde el terminal móvil.

Por otra parte, el Agente de Usuario para WTA permite a los autores acceder e interactuar con las características de los teléfonos móviles (p. e. Control de Llamada), así como otras aplicaciones supuestas en los teléfonos, tales como agendas de teléfono y aplicaciones de calendario.

### ***El Protocolo Inalámbrico de Sesión***

---

El *Protocolo Inalámbrico de Sesión* constituye la capa que se sitúa por debajo de la capa de Aplicación, proporcionando la capacidad necesaria para:

- ✓ Establecer una conexión fiable entre el cliente y el servidor, y liberar esta conexión de una forma ordenada.
- ✓ Ponerse de acuerdo en un nivel común de funcionalidades del protocolo, a través de la negociación de las posibilidades.
- ✓ Intercambiar contenido entre el cliente y el servidor utilizando codificación compacta.
- ✓ Suspender y recuperar la sesión.

Hoy por hoy, este protocolo ha sido definido únicamente para el caso de la navegación, definiéndose como WSP/B<sup>12</sup>. Esta implementación está realizada para el establecimiento de una conexión sobre la base de un protocolo compatible con HTTP1.1.

De esta forma, se han definido un conjunto de primitivas de servicio<sup>13</sup> para permitir la comunicación entre la capa de sesión integrada dentro del equipo cliente y la capa de sesión integrada en el equipo servidor. Estas primitivas, junto con una pequeña descripción de las mismas, puede verse en la Tabla 1:

<b>Nombre de la primitiva</b>	<b>Descripción</b>
<i>S-Connect</i>	Esta primitiva se utiliza para iniciar el establecimiento de la conexión, y para la notificación de su éxito
<i>S-Disconnect</i>	Esta primitiva se utiliza para desconectar una sesión, y para notificar al usuario de una

---

<sup>12</sup> *Wireless Session Protocol -- Browsing*

<sup>13</sup> Una primitiva de servicio representa el intercambio lógico de información entre la capa de Sesión y capas adyacentes.

	sesión que esa sesión no se puede establecer, que ha sido desconectada
<i>S-Suspend</i>	Esta primitiva se utiliza para solicitar la suspensión de la sesión
<i>S-Resume</i>	Esta primitiva se utiliza para solicitar que se recupere la sesión utilizando para las direcciones el nuevo identificador de punto de acceso de servicio.
<i>S-Exception</i>	Esta primitiva se utiliza para notificar aquellos eventos que no están asignados a una transacción en particular, ni provocan la desconexión o suspensión de la sesión.
<i>S-MethodInvoke</i>	Esta primitiva se utiliza para solicitar una operación que deba ser ejecutada en el servidor.
<i>S-MethodResult</i>	Esta primitiva se utiliza para devolver una respuesta a una petición de operación.
<i>S-MethodAbort</i>	Esta primitiva se utiliza para abortar una solicitud de ejecución de operación, que no haya sido aún completada.
<i>S-Push</i>	Esta primitiva se utiliza para enviar información no solicitada desde el servidor, dentro del contexto de una sesión de forma y sin confirmación.
<i>S-ConfirmedPush</i>	Esta primitiva realiza las mismas funciones que la anterior, pero con confirmación.
<i>S-PushAbort</i>	Esta primitiva se utiliza para anular una primitiva anterior del tipo <i>S-Push</i> o <i>S-ConfirmedPush</i> .

**Tabla 1: Primitivas de Servicio de Sesión**

Adicionalmente, existen cuatro tipos de cada una de estas primitivas, tal y como puede verse en la Tabla 2:

<b>Tipo</b>	<b>Abreviación</b>	<b>Descripción</b>
<i>Request</i>	req	Se utiliza cuando una capa superior solicita un servicio de la capa inmediatamente inferior
<i>Indication</i>	ind	Una capa que solicita un servicio utiliza este tipo de primitiva para notificar a la

<i>Response</i>	res	capa inmediatamente superior de las actividades relacionadas con su par, o con el proveedor del servicio Este tipo de primitiva se utiliza para reconocer la recepción de la primitiva de tipo <i>Indication</i> de la capa inmediatamente inferior
<i>Confirm</i>	cnf	La capa que proporciona el servicio requerido utiliza este tipo de primitiva para notificar que la actividad ha sido completada satisfactoriamente.

**Tabla 2: Tipos de Primitivas de Servicio.**

Por último, reseñar que cada una de estas primitivas está perfectamente definida dentro de la especificación, tanto desde el punto de vista del diagrama de tiempos en el que se tienen que invocar las primitivas, como desde el punto de vista de los parámetros intercambiados.

## ***El Protocolo Inalámbrico de Transacción***

El *Protocolo Inalámbrico de Transacción* se establece para proporcionar los servicios necesarios que soporten aplicaciones de “navegación” (del tipo petición/respuesta). Es a este dúo petición/respuesta, lo que vamos a denominar como transacción. Este protocolo se sitúa por encima del *Protocolo Inalámbrico de Datagramas* y, de forma opcional, de la *Capa Inalámbrica de Seguridad de Transporte*, que serán estudiados posteriormente.

Las características de este protocolo son:

- ✓ Proporciona tres clases de servicios de transacción:
  - Clase 0: mensaje de solicitud no seguro, sin mensaje de resultado.
  - Clase 1: mensaje de solicitud seguro, sin mensaje de resultado.
  - Clase 2: mensaje de solicitud seguro, con, exactamente, un mensaje de resultado seguro.
- ✓ La seguridad se consigue a través del uso de identificadores únicos de transacción, asentimientos, eliminación de duplicados y retransmisiones.
- ✓ Seguridad opcional usuario a usuario.
- ✓ De forma opcional, el último asentimiento de la transacción puede contener algún tipo de información adicional

---

relacionada con la transacción, como medidas de prestaciones, etc.

- ✓ Se proporcionan mecanismos para minimizar el número de transacciones que se reenvían como resultado de paquetes duplicados.
- ✓ Se permiten las transacciones asíncronas.

Al igual que en el protocolo anterior (el protocolo inalámbrico de sección), en la Tabla 3 vamos a ver las primitivas de servicio<sup>14</sup> que sustentan la comunicación entre dos capas de transacciones situadas en dos equipos distintos:

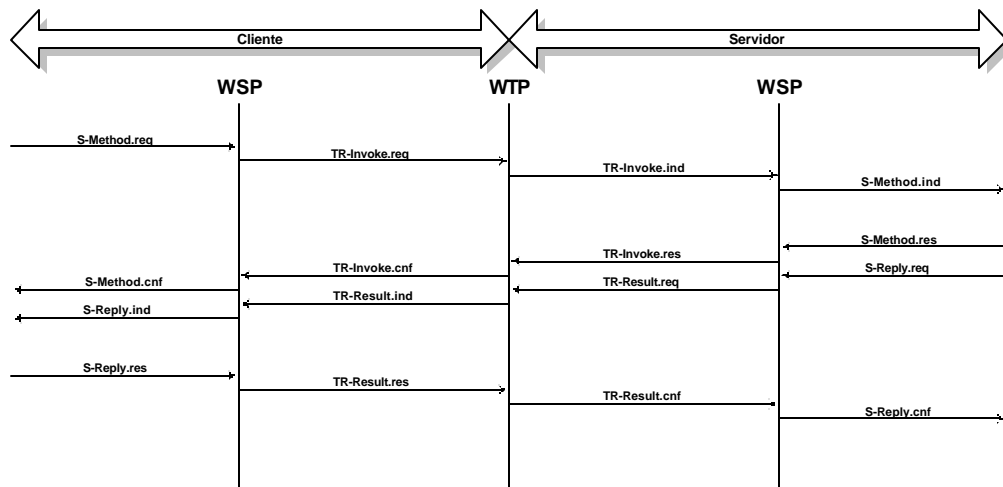
Nombre de la primitiva	Descripción
<i>TR-Invoke</i>	Esta primitiva se utiliza para iniciar una nueva transacción.
<i>TR-Result</i>	Esta primitiva se utiliza para devolver el resultado de transacción iniciada anteriormente
<i>TR-Abort</i>	Esta primitiva se utiliza para abortar una transacción existente

**Tabla 3: Primitivas de Servicio de Transacción**

A modo de ejemplo, vamos a ver en la Figura 6 la concatenación de Primitivas de Servicio de Sesión y de Transacción para el caso de una petición-respuesta:

---

<sup>14</sup> Estas primitivas pueden ser de cuatro tipos, tal y como se puede ver en la Tabla 2.



**Figura 6: Ejemplo intercambio de primitivas entre capa Sesión y Transacción**

Para finalizar, vamos a detallar un poco más las principales características de este protocolo:

- ✓ **Transferencia de Mensajes.**  
Dentro de este protocolo se distinguen dos tipos de mensajes: mensajes de datos y mensajes de control. Los mensajes de datos transportan únicamente datos de usuario, mientras que los mensajes de control se utilizan para los asentimientos, informes de error, etc. pero sin transportar datos de usuario.
- ✓ **Retransmisión hasta el asentimiento.**  
Esta característica se utiliza para la transferencia fiable de datos desde un proveedor WTP a otro, en caso que haya pérdida de paquetes. A modo de comentario, dejar claro que para reducir lo máximo posible el número de paquetes que se transmiten, este protocolo utiliza asentimiento explícito siempre que sea posible.
- ✓ **Asentimiento de usuario.**  
El Asentimiento de Usuario permite al usuario de este protocolo, confirmar cada mensaje recibido por el proveedor WTP.
- ✓ **Información en el Último Asentimiento.**  
Se permite, así pues, enviar información en el último, y únicamente en el último, asentimiento de una transacción. De esta forma, se puede enviar, por ejemplo, información del rendimiento proporcionado por el sistema durante la transacción realizada, etc.
- ✓ **Concatenación y Separación.**

---

Podemos definir concatenación como el proceso de transmitir múltiples Unidades de Datos del Protocolo (PDU<sup>15</sup>) de WTP en una Unidad de Datos del Servicio (SDU<sup>16</sup>) de la red portadora.

Por el contrario, separación es el proceso de separar múltiples PDUs de un único SDU (esto es, el proceso inverso al anterior).

El objetivo de estos sistemas es proveer eficiencia en la transmisión inalámbrica, al requerirse un menor número de transmisiones.

- ✓ Transacciones Asíncronas.  
Para un correcto funcionamiento del protocolo, múltiples transacciones deben ser procesadas de forma asíncrona, debe ser capaz de iniciar múltiples transacciones antes que reciba la respuesta a la primera transacción.
- ✓ Identificador de la Transacción  
Cada transacción está identificada de forma única por los pares de direcciones de los sockets (Dirección fuente, puerto fuente, dirección destino y puerto destino) y por el Identificador de Transacción (TID<sup>17</sup>), el cual se incrementa para cada una de las transacciones iniciadas. Este número es de 16 bits, utilizándose el bit de mayor orden para indicar la dirección.
- ✓ Segmentación y re-ensamblado. (opcional)  
Si la longitud del mensaje supera la Unidad Máxima de Transferencia (MTU<sup>18</sup>), el mensaje puede ser segmentado por el WTP y enviado en múltiples paquetes. Cuando esta operación se realiza, estos paquetes pueden ser enviados y asentidos en grupos. De esta forma, el emisor puede realizar control de flujo cambiando el tamaño de los grupos de mensajes dependiendo de las características de la red.

## **La Capa Inalámbrica de Seguridad de Transporte**

La *Capa Inalámbrica de Seguridad de Transporte* (en adelante WTLS), constituye una capa modular, que depende del nivel de seguridad requerido por una determinada aplicación. Esta capa proporciona a las capas de nivel superior

---

<sup>15</sup> *Protocol Data Unit*

<sup>16</sup> *Service Data Unit*

<sup>17</sup> *Transaction Identifier*

<sup>18</sup> *Maximum Transfer Unit*

de WAP de una interfaz de servicio de transporte seguro, que lo resguarde de una interfaz de transporte inferior.

El principal objetivo de esta capa es proporcionar privacidad, integridad de datos y autenticación entre dos aplicaciones que se comuniquen. Adicionalmente, la WTLS proporciona una interfaz para el manejo de conexiones seguras.

Al igual que hemos hecho en los protocolos anteriores, en la Tabla 4 vamos a ver las primitivas de servicio<sup>19</sup> que sustentan la comunicación entre dos capas situadas en dos equipos distintos:

Nombre de la primitiva	Descripción
<i>SEC-Unitdata</i>	Esta primitiva se utiliza para intercambiar datos de usuario entre los dos participantes. Sólo puede ser invocada cuando existe previamente una conexión segura entre las direcciones de transporte de los dos participantes.
<i>SEC-Create</i>	Esta primitiva se utiliza para iniciar el establecimiento de una conexión segura.
<i>SEC-Exchange</i>	Esta primitiva se utiliza en la creación de una conexión segura si el servidor desea utilizar autenticación de clave pública o intercambio de claves con el cliente.
<i>SEC-Commit</i>	Esta primitiva se inicia cuando el <i>handshake</i> <sup>20</sup> se completa y cualquiera de los equipos participantes solicita cambiar a un nuevo estado de conexión negociado.
<i>SEC-Terminate</i>	Esta primitiva se utiliza para finalizar la conexión.
<i>SEC-Exception</i>	Esta primitiva se utiliza para informar al otro extremo sobre las alertas de nivel de aviso.
<i>SEC-Create-Request</i>	Esta primitiva se utiliza por el servidor para solicitar al cliente que inicie un nuevo <i>handshake</i> .

**Tabla 4: Primitivas de Servicio de Capa de Seguridad**

<sup>19</sup> Estas primitivas pueden ser de cuatro tipos, tal y como se puede ver en la Tabla 2.

<sup>20</sup> Término utilizado para denominar el intercambio de primitivas entre cliente y servidor con el objetivo de establecer una sesión segura. Posteriormente veremos este intercambio de primitivas.

Hemos hablado anteriormente del proceso de establecimiento de una sesión segura o *handshake*. En la Figura 7 podemos ver este intercambio de primitivas:

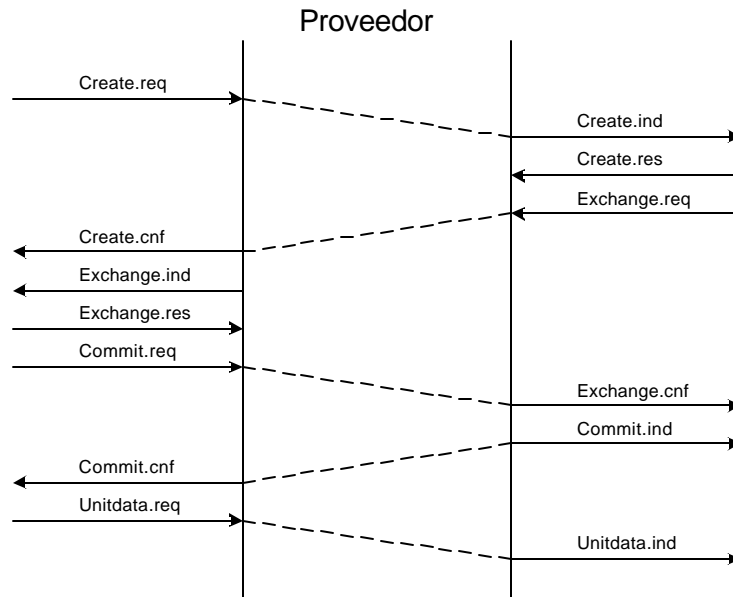


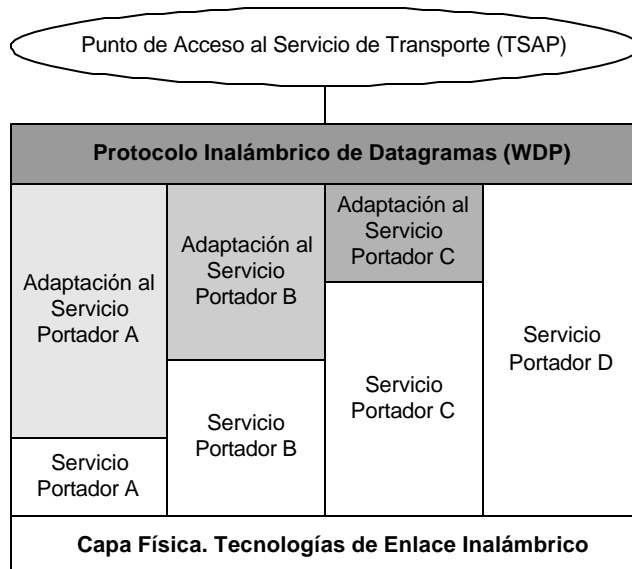
Figura 7: Secuencia de Primitivas para el establecimiento de una sesión segura

## El Protocolo Inalámbrico de Datagramas

El *Protocolo Inalámbrico de Datagramas* (en adelante WDP<sup>21</sup>) ofrece un servicio consistente al protocolo (Seguridad, Transacción y Sesión) de la capa superior de WAP, comunicándose de forma transparente sobre uno de los servicios portadores disponibles.

Este protocolo ofrece servicios a los protocolos superiores del estilo a direccionamiento por número de puerto, segmentación y re-ensamblado opcional y detección de errores opcional, de forma que se permite a las aplicaciones de usuario funcionar de forma transparente sobre distintos servicios portadores disponibles. Para ello, se plantea una arquitectura de protocolo como el que se muestra en la Figura 8:

<sup>21</sup> *Wireless Datagram Protocol*



**Figura 8: Arquitectura del Protocolo Inalámbrico de Datagramas**

Al igual que hemos hecho en los protocolos anteriores, en la Tabla 5 vamos a ver las primitivas de servicio<sup>22</sup> que se utilizan en este protocolo:

Nombre de la primitiva	Descripción
<i>T-DUnitdata</i>	Esta primitiva es la utilizada para transmitir datos como datagramas. No requiere que exista una conexión para establecerse.
<i>T-DError</i>	Esta primitiva se utiliza para proporcionar información a la capa superior cuando ocurre un error que pueda influenciar en el servicio requerido.

**Tabla 5: Primitivas de Servicio de la Capa de Datagramas**

Por último, vamos a ver la arquitectura de este protocolo dentro de la arquitectura global de WAP, para el caso de utilizarse GSM como servicio portador, que es el protocolo que más nos puede interesar por su amplia implantación en los sistemas de comunicaciones móviles telefónicas existentes hoy en día.

<sup>22</sup> Estas primitivas pueden ser de cuatro tipos, tal y como se puede ver en la Tabla 2.

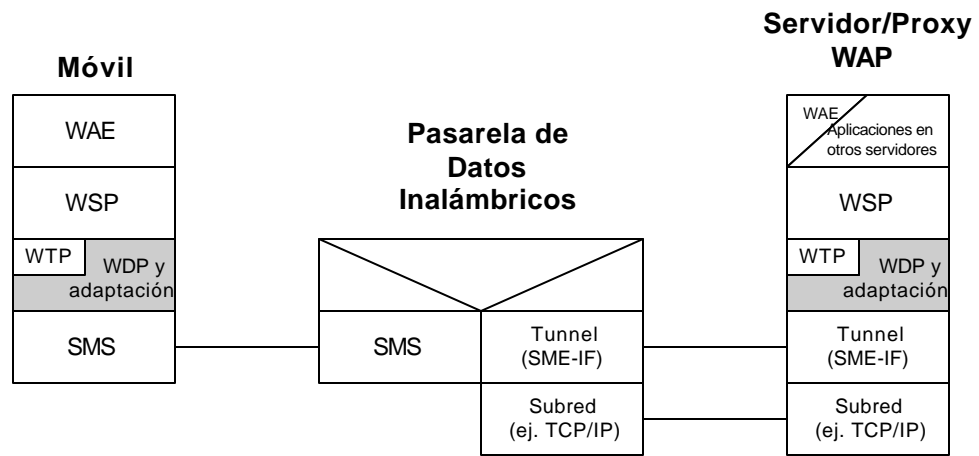


Figura 9: WDP sobre GSM SMS

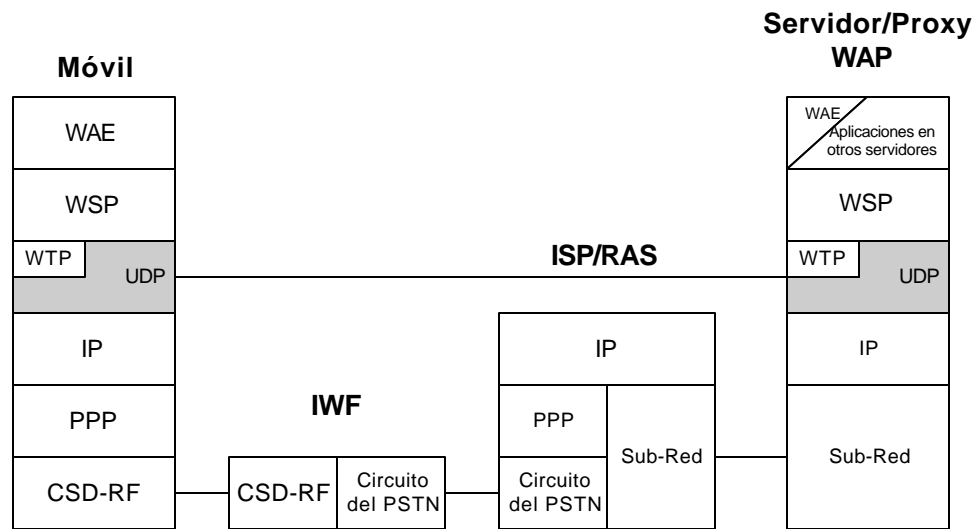
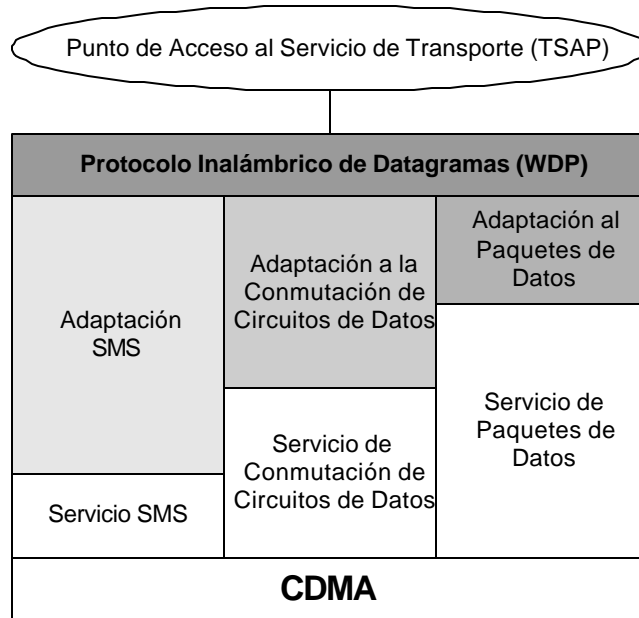


Figura 10: WDP sobre GSM Canal de Datos de Circuitos Conmutados



**Figura 11: WDP sobre Servicios Portadores CDMA**

---

## Apéndice A: Glosario de Términos

---

API	<i>Application Programming Interface</i> Interfaz de Programación de Aplicación
CDMA	<i>Code Division Multiple Access</i> Acceso Múltiple por División en el Código
CSD	<i>Circuit Switched Data</i> Conmutación de Circuitos de Datos
GSM	<i>Global System for Mobile Communications</i> Sistema Global para Comunicaciones Móviles
IP	<i>Internet Protocol</i> Protocolo de Internet
MAC	<i>Medium Access Control</i> Control de Acceso al Medio
MDG	<i>Mobile Data Gateway</i> Pasarela de Datos Móviles
PPP	<i>Point-to-Point Protocol</i> Protocolo Punto-a-Punto
SAR	<i>Segmentation and Reassembly</i> Segmentación y Re-ensamblado
URI	<i>Universal/Uniform Resource Identifier</i> Identificador Universal/Uniforme de Recursos
WAE	<i>Wireless Application Environment</i> Entorno Inalámbrico de Aplicación
WAP	<i>Wireless Application Protocol</i> Protocolo Inalámbrico de Aplicación
WDP	<i>Wireless Datagram Protocol</i> Protocolo Inalámbrico de Datagramas
WSP	<i>Wireless Session Protocol</i> Protocolo Inalámbrico de Sesión
WTLS	<i>Wireless Transport Layer Security</i> Capa de Seguridad de Transporte Inalámbrico
WTP	<i>Wireless Transaction Protocol</i> Protocolo Inalámbrico de Transacciones

---

## **Bibliografia**

---

- [WAPARCH] “Wireless Application Protocol Architecture Specification”  
WAP Forum, 30-Abril-98  
URL: <http://www.wapforum.com/>
- [WAPWDP] “Wireless Datagram Protocol Specification”  
WAP Forum, 30-Abril-98  
URL: <http://www.wapforum.com/>
- [WAPWTP] “Wireless Transaction Protocol Especification”  
WAP Forum, 30-Abril-98  
URL: <http://www.wapforum.com/>
- [WAPWTLS] “Wireless Transport Layer Security Specification”  
WAP Forum, 30-Abril-98  
URL: <http://www.wapforum.com/>
- [WAPWSP] “Wireless Session Protocol Specification”  
WAP Forum, 30-Abril-98  
URL: <http://www.wapforum.com/>
- [WAPWAEO] “Wireless Application Environment Overview”  
WAP Forum, 30-Abril-98  
URL: <http://www.wapforum.com/>
- [WAPWAES] “Wireless Application Environment Specification”  
WAP Forum, 30-Abril-98  
URL: <http://www.wapforum.com/>
- [WAPWML] “Wireless Markup Language Specification”  
WAP Forum, 30-Abril-98  
URL: <http://www.wapforum.com/>
- [WAPWTAI] “Wireless Telephony Application Interface Specification”  
WAP Forum, 30-Abril-98  
URL: <http://www.wapforum.com/>
- [WAPWTA] “Wireless Telephony Application Specification”  
WAP Forum, 30-Abril-98  
URL: <http://www.wapforum.com/>