

PROTOCOLO TREN-TIERRA: ESPECIFICACIONES

(Versión 1.2b)

Madrid, Agosto 2004

ÍNDICE

ÍNDICE.....	1
1. INTRODUCCIÓN.....	3
2. DESCRIPCIÓN A NIVEL DE CONEXIÓN	4
2.1. INICIALIZACIÓN	4
2.2. MENSAJES DE ACTUALIZACIÓN	4
2.3. DESCONEXIÓN	5
3. DESCRIPCIÓN A NIVEL DE PROTOCOLO	6
4. MENSAJES	7
4.1. MENSAJES DESDE EL GESTOR TREN-TIERRA AL OTS	7
4.1.1. REGISTRO	7
4.2. MENSAJES DESDE EL OTS AL GESTOR TREN-TIERRA.....	7
4.2.1. VIDA	7
4.2.2. CONFIRMACIONREGISTRO	8
4.2.3. DESCRIPCIONLINEA	8
4.2.4. CHAPAMATRICULA	8
4.2.5. MATRICULAPOSICION.....	9
4.2.6. PREVISIONTIEMPO	10
ANEXO I: Fichero de especificación XDR	12

CONTROL DE CAMBIOS

Versión	1.0	Fecha	25-Jun-2.004
Descripción:	Versión inicial del documento		
Versión	1.1	Fecha	30-Jun-2.004
Descripción:	Corrección andenes en MatriculaPosicion. Campo Linea.		
Versión	1.2	Fecha	21-Jul-2.004
Descripción:	Lista de andenes.Registro con array de líneas.Confirmación de registro. Mensaje de vida		
Versión	1.2a	Fecha	22-Jul-2.004
Descripción:	Corrección de las erratas que impedían la compilación del fichero .x		
Versión	1.2b	Fecha	23-Jul-2.004
Descripción:	Aclaración del orden en que seenvían los andenes al principio de la co- municacion		

Los cambios se representan en Colores: NEGRO corresponde a versiones anteriores
 AZUL corresponde al texto modificado o añadido
 ROJO corresponde a textos críticos (en general añadidos)

1. INTRODUCCIÓN

Este documento pretende exponer el protocolo a seguir en la comunicación entre el sistema OTS (OBJECT TRACKING SYSTEM) y el gestor de Tren-Tierra.

El OTS es el sistema que realiza el seguimiento de la posición de los trenes dentro de las líneas, utilizando información del CTC, TETRA y Balizas.

El gestor de Tren-Tierra es el sistema que controla y supervisa la comunicación wifi con los sistemas embarcados.

El gestor Tren-Tierra requiere de cierta información sobre la posición de los trenes para que la comunicación wifi con los sistemas embarcados sea fluida, información que le será suministrada por medio de una conexión con el OTS usando el protocolo descrito en este documento.

El protocolo está diseñado para proporcionar la siguiente funcionalidad:

- Registro de un gestor Tren-Tierra para recibir información de una o más líneas
- Confirmación de registro de esas líneas por parte del servidor y volcado de los andenes de los que están compuestos dichas líneas.
- Volcado inicial de la base de datos de los trenes de esas líneas, la cual se compone por la relación de chapas-matrículas y el último evento de posición para cada uno de los trenes.
- Mensajes de actualización de cambio en una relación chapa-matrícula.
- Mensajes de actualización de evento de posición.
- Mensajes de previsión de tiempo.
- Mensajes de vida del servidor

2. DESCRIPCIÓN A NIVEL DE CONEXIÓN

La comunicación entre el OTS y el gestor Tren-Tierra se realiza mediante sockets. El OTS actúa de servidor (espera conexiones) y el gestor de Tren-Tierra actúa como cliente (abre una conexión contra el OTS).

El sistema OTS espera conexiones en el puerto 8010.

A continuación describo la manera de proceder la comunicación entre el OTS y el gestor Tren-Tierra.

2.1. INICIALIZACIÓN

1. El gestor Tren-Tierra abre una conexión contra el puerto 8010 del OTS
2. El gestor Tren-Tierra manda un mensaje de registro (ver 4.1.1 en pág. 7) con la lista de líneas a supervisar junto con un texto que le identifique (p.ej. “Gestor Tren-Tierra/IG - PuMa”), además del intervalo de tiempo deseado entre mensajes de vida del servidor.
3. El OTS manda un mensaje de confirmación de registro (ver 4.2.2 en pág. 8) con las líneas en las que ha quedado registrado
4. El OTS manda un mensaje de descripción de línea (ver 4.2.3 en pág. 8) para cada línea en que ha quedado registrado
5. El OTS manda todas las relaciones chapa-matrícula que haya en las líneas para las que se haya registrado el gestor Tren-Tierra (ver 4.2.4 en pág. 8).
6. El OTS manda la última posición conocida para todos los trenes en las líneas para las que se haya registrado (los trenes van identificados por su matrícula de cabecera, ver 4.2.5 en pág 9).

2.2. MENSAJES DE ACTUALIZACIÓN

Según vayan cambiando las asociaciones entre chapas-matriculas de los trenes de las líneas para las que el gestor Tren-Tierra se haya registrado, se mandarán mensajes del tipo:

- ChapaMatricula (ver 4.2.4 pág. 8)

Según se detecte que un tren ha pasado por una posición de las definidas en los eventos de posición en las líneas para las que se haya registrado, se mandarán mensajes del tipo:

- MatriculaPosicion (ver 4.2.5 en pág. 9)

Según un tren salga de una estación o cambie la previsión de andén de destino, se mandarán mensajes del tipo:

- PrevisionTiempo (ver 4.2.6 en pág. 10)

Cada n segundos (siendo n el número intervalo_vida que se da en el mensaje de confirmación de registro), se mandará un mensaje del tipo:

- Vida (ver 4.2.1 en pág. 7)

2.3. DESCONEXIÓN

En el caso de detectar mensajes erróneos cualquiera de las dos partes (OTS o gestor Tren-Tierra), se cerrará la conexión.

También se cerrará la conexión por parte del gestor Tren-Tierra en el momento en que no esté interesado en seguir recibiendo información de alguna de las líneas en las que se ha registrado. En caso de querer dejar de recibir información de algunas líneas pero dejando otras, habrá de desconectarse y volverse a conectar para empezar de nuevo el proceso de registro con las líneas de las que quiere tener información.

3. DESCRIPCIÓN A NIVEL DE PROTOCOLO

La transmisión de los mensajes se realiza usando XDR¹.

Los mensajes se delimitan según `xdr*_rec()` de UNIX. Esto es:

- Los mensajes se dividen para la transmisión en partes menores que el MTU.
- Cada parte se precede de un número entero de 4 bytes en big-endian, en el que el MSB (bit alto) indica en caso de estar a 1 que con esta parte se completa un mensaje, y los 31 bits bajos indican el tamaño en bytes de la parte (sin contar a este número entero del que se le ha precedido).

Para la descripción a nivel de bytes del envío de cada mensaje, véase el fichero de especificación `xdr` adjunto (véase pág. 12) y el estándar XDR citado arriba.

¹Descrito en <http://www.ietf.org/rfc/rfc1832.txt>

4. MENSAJES

4.1. MENSAJES DESDE EL GESTOR TREN-TIERRA AL OTS

4.1.1. Registro

Funciones

- Identificar al gestor Tren-Tierra en el OTS
- Informar al OTS de las líneas de las que quiere información

Cuándo se manda

- Al conectarse el gestor Tren-Tierra al OTS.

Descripción de su contenido

- *identificador_cliente*: cadena que identifica de forma unívoca a esta instancia del gestor Tren-Tierra (p.ej. "Gestor Tren-Tierra/IG - PuMa").
- *lineas*: array de enteros enumerando las líneas que se quieren supervisar (p.ej. 8,12)
- *intervalo_vida*: entero especificando el número de segundos que quiere que transcurran entre dos mensajes de vida del servidor (p.ej. 30)

Notas

- Al conectarse se mandará un sólo mensaje de registro, especificando todas las líneas de las que quiere ser informado. El servidor devolverá como contestación un mensaje ConfirmacionRegistro.
- Dicho mensaje sólo se enviará una vez, y será al inicio de la comunicación.

4.2. MENSAJES DESDE EL OTS AL GESTOR TREN-TIERRA

4.2.1. Vida

Funciones

- Tener constancia de que el servidor sigue vivo y la conexión activa

Cuándo se manda

- Cada n segundos, siendo n el número *intervalo_vida* del mensaje de ConfirmacionRegistro.

Descripción de su contenido

- *instante*: instante (time_t) en el que se ha generado el mensaje de vida (p.ej. 1088145367).

4.2.2. ConfirmacionRegistro

Funciones

- Confirmación de la petición de registro
- Informar al Gestor TrenTierra de las líneas de las que se le van a informar
- Identificar al servidor y la versión del protocolo

Cuándo se manda

- Como respuesta al mensaje de registro del Gestor TrenTierra.

Descripción de su contenido

- *versión_protocolo*: cadena que identifica el número de versión del protocolo usado por el servidor (p.ej. "1.2").
- *identificador_servidor*: cadena que identifica al servidor (p.ej. "OTS/SICO - PuMa").
- *identificador_cliente*: cadena que identificación que envió el Gestor TrenTierra (p.ej. "Gestor Tren-Tierra/IG - PuMa").
- *lineas*: array de enteros enumerando las líneas de las que informará al Gestor Tren-Tierra (p.ej. 8,12).
- *intervalo_vida*: entero especificando el número de segundos que transcurrirán entre dos mensajes de vida del OTS (p.ej. 30).

4.2.3. DescripcionLinea

Funciones

- Informar de los andenes que pertenecen a una línea

Cuándo se manda

- Después del mensade de confirmación de registro y antes de volcar las relaciones chapa-matrícula iniciales.

Descripción de su contenido

- *linea*: entero que indica la línea de la que se está informando (p.ej. 11).
- *andenesvia1*: array de cadenas de los andenes de la vía 1 (p.ej. PLAZA_ELIPTICA_1, ABRANTES_1, PAN_BENDITO_1)
- *andenesvia2*: array de cadenas de los andenes de la vía 2 (p.ej. PAN_BENDITO_2, ABRANTES_2, PLAZA_ELIPTICA_2)

Notas

- Se manda un mensaje de DescripciónLínea para cada línea en la que se ha quedado registrado.
- Los andenes se mandan en el orden normal de circulación para esa vía.

4.2.4. ChapaMatricula

Funciones

- Informar de una relación chapa-matrículas

Cuándo se manda

- Al conectarse el gestor Tren-Tierra al OTS, después de la etapa de registro, se manda para cada uno de los trenes de las líneas en las que el gestor Tren-Tierra se ha registrado.
- Cuando aparece una nueva chapa
- Cuando un tren cambia de sentido (para que la relación de matrículas siempre empiece por la matrícula de cabecera)
- Cuando una chapa desaparece
- Cuando la composición asociada a una chapa cambia

Descripción de su contenido

- *instante_identificacion*: instante (time_t) en el que se ha tenido conocimiento de esta relación chapa-matrículas (p.ej. 1088145363).
- *chapa*: cadena que identifica la chapa (p.ej. "N19").
- *matriculas*: array de cadenas que identifican las matrículas asociadas (p.ej. {"M5005", "M5006", "M5008"}), dando así la composición del tren.
- *linea*: entero que identifica la línea a la que se refiere el mensaje (p.ej. 8).

Notas

- La lista de matrículas se dará siempre en orden, empezando por la matrícula del coche de cabecera.
- Cuando una chapa desaparece, se manda un mensaje ChapaMatricula con una lista de matrículas nula (vacía).

4.2.5. MatriculaPosicion

Funciones

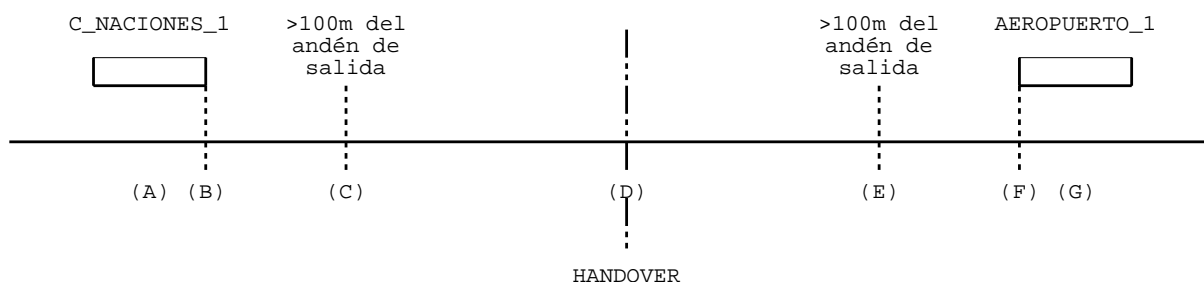
- Informar de que un tren ha pasado por una posición.
- Informar de la última posición conocida de cada tren en el volcado inicial de la base de datos.

Cuándo se manda

- Al conectarse el gestor Tren-Tierra al OTS, después de la etapa de registro, se manda para cada uno de los trenes de las líneas en las que el gestor Tren-Tierra se ha registrado.
- Cuando un tren pasa por uno de los puntos antes descritos.

Descripción de su contenido

- *instante_identificacion*: instante (time_t) en el que se ha pasado por dicha posición.(p.ej. 1088145421).
- *matricula_cabecera*: cadena que identifica las matrícula de cabecera del tren que ha pasado por dicha posición (p.ej. "M5005").
- *posicion*: entero representando el tipo de posición por la que ha pasado (p.ej. 1). Los tipos de posición son los siguientes (se indica asimismo si rellena los campos de anden_salida y/o anden_llegada):



Punto	Posición	Tipo	anden_salida	anden_llegada
(A)	POS_SALIDA	1	X	
(B)	POS_INTERESTACION	2	X	
(C)	POS_FIN_CAMARA_SALIDA	3	X	
(D)	POS_EVENTO_HANDOVER	4	X	X
(E)	POS_COMIENZO_CAMARA_LLEGADA	5		X
(F)	POS_APROXIMACION	6		X
(G)	POS_ESTACIONAMIENTO	7		X

- *linea*: entero que identifica la línea a la que se refiere el mensaje (p.ej. 8).
- *anden_salida*: cadena que identifica el andén de salida (p.ej. "C_NACIONES_1").
- *anden_llegada*: cadena que identifica el andén de salida (p.ej. "AEROPUERTO_1").

Notas

- Los puntos se definen en los extremos de los circuitos de vía. Esto hace que los puntos sean aproximados, ya que sólo se detecta cuándo el tren cambia de circuito de vía.

4.2.6. PrevisionTiempo

Funciones

- Informar de la previsión del tiempo que va a necesitar un tren en ir desde un andén hasta el siguiente.

Cuándo se manda

- Cada vez que un tren sale de una estación.
- Cuando el destino previsible del tren cambia.

Descripción de su contenido

- *instante_prevision_llegada*: instante (time_t) en el que se estima que el tren llegará al siguiente andén (p.ej. 1088147481).
- *matricula_cabecera*: cadena que identifica las matrícula de cabecera del tren del que se hace la previsión (p.ej. "M5005").
- *linea*: entero que identifica la línea a la que se refiere el mensaje (p.ej. 8).
- *anden_salida*: cadena que identifica el andén de salida (p.ej. "C_NACIONES_1").
- *anden_llegada*: cadena que identifica el andén de salida (p.ej. "AEROPUERTO_1").

Notas

- Las previsiones del andén de llegada se hacen según las posiciones de las agujas e informaciones de dispositivos similares disponibles en el momento de salida del andén. Estos

pueden cambiar durante el recorrido en interestación, cambiando así el destino previsible del tren. En caso de cambiar el destino previsible, se mandará un nuevo mensaje de PrevisionTiempo.

Anexo I: Fichero de especificación XDR

```
const MAX_LONG_NOMBRE_CHAPA = 5;
const MAX_LONG_NOMBRE_MATRICULA = 6;
const MAX_NRO_COCHES = 18; /* Max 3 unidades, unidad de max 6 coches */
const MAX_LONG_NOMBRE_ANDEN = 30;
const MAX_NRO_LINEAS = 32;
const MAX_NRO_ANDENES = 64;
const MAX_LONG_IDENTIFICADOR = 32;
const MAX_LONG_NUMVERSION = 8;

typedef long Timestamp; /* segundos desde epoch de UNIX (1/01/1970) */
typedef string Chapa<MAX_LONG_NOMBRE_CHAPA>;
typedef string Matricula<MAX_LONG_NOMBRE_MATRICULA>;
typedef long Linea;
typedef string Anden<MAX_LONG_NOMBRE_ANDEN>;
typedef long LapsoSegs;
typedef string IdentificadorCliente<MAX_LONG_IDENTIFICADOR>;
typedef string IdentificadorServidor<MAX_LONG_IDENTIFICADOR>;
typedef string VersionProtocolo<MAX_LONG_NUMVERSION>;

/* Mensajes desde Tren-Tierra a OTS */

struct Registro {
    IdentificadorCliente identificador_cliente;
    Linea lineas<MAX_NRO_LINEAS>;
    LapsoSegs intervalo_vida;
};

enum TipoMensajeTTOTS {
    MSG_REQ_REGISTRO = 1
};

union MensajeTTOTS switch(TipoMensajeTTOTS tipo) {
    case MSG_REQ_REGISTRO:
        Registro registro;
};

/* Mensajes desde OTS a Tren-Tierra */
struct Vida {
    Timestamp instante;
};

struct ConfirmacionRegistro {
    VersionProtocolo version_protocolo;
```

```
    IdentificadorServidor identificador_servidor;
    IdentificadorCliente identificador_cliente;
    Linea lineas<MAX_NRO_LINEAS>;
    LapsoSegs intervalo_vida;
};

struct DescripcionLinea {
    Linea Linea;
    Anden andenesvia1<MAX_NRO_ANDENES>;
    Anden andenesvia2<MAX_NRO_ANDENES>;
};

struct ChapaMatricula {
    Timestamp instante_identificacion;
    Chapa chapa;
    Matricula matriculas<MAX_NRO_COCHES>;
    Linea linea;
};

enum TipoPosicion {
    POS_SALIDA = 1,
    POS_INTERESTACION = 2,
    POS_FIN_CAMARA_SALIDA = 3,
    POS_EVENTO_HANDOVER = 4,
    POS_COMIENZO_CAMARA_LLEGADA =5,
    POS_APROXIMACION = 6,
    POS_ESTACIONAMIENTO = 7
};

struct MatriculaPosicion {
    Timestamp instante_posicion;
    Matricula matricula_cabecera;
    TipoPosicion posicion;
    Linea linea;
    Anden anden_salida;
    Anden anden_llegada;
};

/* El mensaje de previsión de tiempo se manda automáticamente cada vez */
/* que un tren sale de una estación */
struct PrevisionTiempo {
    Timestamp instante_prevision_llegada;
    Matricula matricula_cabecera;
    Linea linea;
    Anden anden_salida;
```

```
        Anden anden_llegada;
};

enum TipoMensajeOTSTT {
    MSG_VIDA = 0,
    MSG_CONFIRMACION_REGISTRO = 1,
    MSG_DESCRIPCION_LINEA = 2,
    MSG_CHAPA_MATRICULA = 3,
    MSG_MATRICULA_POSICION = 4,
    MSG_PREVISION_TIEMPO = 5
};

union MensajeOTSTT switch(TipoMensajeOTSTT tipo) {
    case MSG_VIDA:
        Vida vida;
    case MSG_CONFIRMACION_REGISTRO:
        ConfirmacionRegistro confirmacion_registro;
    case MSG_DESCRIPCION_LINEA:
        DescripcionLinea descripcion_linea;
    case MSG_CHAPA_MATRICULA:
        ChapaMatricula chapa_matricula;
    case MSG_MATRICULA_POSICION:
        MatriculaPosicion matricula_posicion;
    case MSG_PREVISION_TIEMPO:
        PrevisionTiempo prevision_tiempo;
};
```