

Documentación de la API de Tranvías de A Coruña

Esta documentación detalla los endpoints disponibles para consultar información del servicio de autobuses urbanos de A Coruña (Compañía de Tranvías de La Coruña). Se incluye la URL de cada endpoint, su propósito, parámetros, ejemplos de uso y fragmentos de la respuesta JSON. Los endpoints se organizan por tipo de recurso o funcionalidad: **datos estáticos** (información general del sistema), **paradas** (consultas de tiempos de llegada en tiempo real) y **líneas** (información y seguimiento en tiempo real por línea, incluyendo horarios).

Datos estáticos del sistema (Líneas, paradas, tarifas, etc.)

Endpoint: `https://itranvias.com/queryitr_v3.php?dato=20160101T000000_g1_0_20160101T000000&func=7` ¹

- **Descripción:** Proporciona datos estáticos completos del sistema de autobuses urbanos de A Coruña. En un solo JSON se obtiene la lista de **líneas**, **paradas**, **enlaces de transbordo** permitidos y **tarifas** del servicio ¹. Este endpoint corresponde a la función `func=7` de la API de iTranvías, que devuelve todas las novedades del sistema dado un parámetro de fecha inicial apropiado.
- **Parámetros:**
 - `func`: **Requerido.** Debe ser `7` para obtener los datos estáticos generales.
 - `dato`: **Requerido.** Cadena compuesta que indica la fecha y otros parámetros. Para obtener *todos* los datos desde el inicio, se utiliza el valor `20160101T000000_g1_0_20160101T000000` (como en el ejemplo arriba). En esta cadena:
 - `20160101T000000` representa una fecha/hora inicial (1 de enero de 2016 a las 00:00) anterior a cualquier dato, para así recibir **toda** la información.
 - `_g1_` indica el idioma Gallego. Se puede cambiar por `_es_` para Español o `_en_` para Inglés ². (Por ejemplo: `20160101T000000_es_0_20160101T000000` obtendría las descripciones en español).
 - El `0` siguiente corresponde a `idnovedad=0` (sin novedad previa).
 - La última parte repite la fecha inicial (`fechanovedad`).No se requieren otros parámetros. El método de solicitud es GET.

Ejemplo de uso:

```
GET https://itranvias.com/queryitr_v3.php?
dato=20160101T000000_es_0_20160101T000000&func=7
```

(En este ejemplo se solicita la información estática en idioma español.)

Ejemplo de respuesta (JSON abreviado):

```

{
  "resultado": "OK",
  "fecha_peticion": "20230921235412",
  "peticion": "itranvias?
fecha=20160101T000000&plataforma=www&lenguaje=gl&idnovedad=0&fechanovedad=20160101T000000
",
  "tamaño": 72829,
  "iTranvias": {
    "actualizacion": {
      "fecha": "20230832T140825",
      "lineas": [
        {
          "id": 100,
          "lin_comer": "1",
          "nombre_orig": "Abente y Lago",
          "nombre_dest": "Castrillón",
          "color": "982135",
          "rutas": [
            {
              "ruta": 10000,
              "nombre_orig": "",
              "nombre_dest": "",
              "paradas": [523, 180, 1, 2, 3, ...]
            },
            ...
          ]
        },
        ...
      ],
      "paradas": [
        {
          "id": 1,
          "nombre": "Porta Real",
          "posx": -8.39585,
          "posy": 43.370115,
          "enlaces": [100, 1900, 200, 800, ...]
        },
        ...
      ],
      "enlaces": [ ... ],
      "tarifas": [
        {
          "tarifa": "Tarifa ordinaria",
          "precio": 1.3
        },
        ...
      ]
    }
  },
  "Origen": "Web_Beta"
}

```

En la estructura JSON devuelta:

- **lineas** : Lista de todas las líneas de autobús. Cada entrada incluye:
 - **id** : Identificador numérico de la línea.
 - **lin_comer** : Nombre comercial o número de línea (por ejemplo, "1" para la línea 1).
 - **nombre_orig** : Nombre de la cabecera de origen de la línea.
 - **nombre_dest** : Nombre de la cabecera de destino de la línea.
 - **color** : Color representativo de la línea en formato hexadecimal (ej. #RRGGBB).
 - **rutas** : Listado de rutas o variantes que tiene la línea. Cada ruta incluye:
 - **ruta** : Identificador de la ruta (normalmente el primer dígito(s) indican la línea, e.g. 10000 para la ruta principal de la línea 1).
 - **nombre_orig** y **nombre_dest** : Origen y destino específicos de esa ruta (a menudo vacíos si coinciden con la línea principal).
 - **paradas** : Lista de IDs de paradas por las que pasa esa ruta, en orden.
 - **paradas** : Lista de todas las paradas del sistema. Cada parada incluye:
 - **id** : Identificador numérico de la parada.
 - **nombre** : Nombre de la parada.
 - **posx** , **posy** : Coordenadas de la ubicación de la parada (longitud y latitud en WGS84).
 - **enlaces** : Lista de identificadores de líneas que pasan por esta parada (es decir, qué líneas tienen parada aquí).
 - **enlaces** : Información sobre transbordos permitidos y su coste. (La documentación original marca esta sección "En proceso", por lo que no detalla su estructura exacta. Probablemente incluye las reglas de enlace entre líneas y los tiempos o costos asociados.) ³
 - **tarifas** : Lista de tarifas disponibles en el sistema, cada entrada con:
 - **tarifa** : Nombre de la tarifa (por ejemplo, "Tarifa ordinaria").
 - **precio** : Precio en euros de dicha tarifa ⁴ ⁵ .

Este endpoint de datos estáticos es útil para obtener la configuración base del sistema (mapa de líneas, paradas y tarifas) y normalmente se llamaría con poca frecuencia (p.ej., al iniciar una aplicación) dado su tamaño de respuesta considerable ⁶ .

Paradas (Próximos buses en una parada - Tiempo real)

Endpoint: `https://itranvias.com/queryitr_v3.php?func=0&dato=<ID_parada>` ⁷

- **Descripción:** Devuelve los próximos autobuses en llegar a una parada específica, con información en tiempo real. Este endpoint corresponde a **func=0** e informa de las líneas que tienen autobuses próximamente en la parada indicada y los detalles de cada autobús entrante ⁷ . Es el equivalente a la funcionalidad de "próximos buses" en la web/app de iTranvías.
- **Parámetros:**
 - **func** : **Requerido.** Debe ser **0** para consultar próximos buses en una parada.
 - **dato** : **Requerido.** Identificador numérico de la parada a consultar (por ejemplo, 10, 100, 203, etc.).
No requiere parámetros adicionales. (El idioma de la respuesta no se especifica, típicamente devuelve números de línea y datos numéricos que no dependen del lenguaje.)
- **Ejemplo de uso:**

GET https://itranvias.com/queryitr_v3.php?func=0&dato=100

(Consulta de los próximos autobuses para la parada con ID 100.)

• Ejemplo de respuesta (JSON):

```
{
  "resultado": "OK",
  "fecha_peticion": "20230922152056",
  "peticion": "jbuses?parada=100&version=2",
  "tamaño": 700,
  "buses": {
    "parada": 100,
    "lineas": [
      {
        "linea": 500,
        "buses": [
          {
            "bus": 385,
            "tiempo": "<1",
            "distancia": "186",
            "estado": 1,
            "ult_parada": 203
          },
          ...
        ]
      },
      ...
    ]
  }
}
```

En este JSON, la clave principal es `buses`, que contiene:

- `parada`: El ID de la parada consultada (en el ejemplo, 100).
- `lineas`: Lista de líneas **activas** en esa parada *que tienen autobuses en camino*. Cada objeto de línea incluye:
 - `linea`: El identificador de la línea que se acerca (ej. 500 corresponde a cierta línea, probablemente línea 5 si sigue la codificación).
 - `buses`: Lista de autobuses próximos de esa línea, con cada objeto conteniendo:
 - `bus`: Identificador del autobús (vehículo).
 - `tiempo`: Tiempo estimado para que llegue a la parada, en **minutos**. Puede ser un número (ej. "5") o la cadena "<1" cuando falta menos de un minuto ⁸ ⁹.
 - `distancia`: Distancia aproximada del bus a la parada, en **metros** ⁸.
 - `estado`: Estado del bus en tránsito, donde:
 - **0**: El bus está actualmente detenido en *alguna* parada (posiblemente en esta misma).
 - **1**: El bus está en movimiento hacia la parada ¹⁰.
 - `ult_parada`: ID de la última parada por la que pasó el bus antes de dirigirse a la parada consultada ¹¹.

Nota: Solo se listan las líneas que tienen buses próximos en ese momento. Si una línea pasa por la parada pero ya no tiene más servicios en el día o ningún bus próximo, no aparecerá en la respuesta ¹².

Líneas (Información y seguimiento por línea en tiempo real)

Esta sección cubre varios endpoints para obtener información específica de una línea de autobús, incluyendo datos generales de la línea, posiciones de sus autobuses en ruta, posiciones geográficas (GPS) de los autobuses, listado de paradas de la línea y el trazado completo de la ruta, así como las horas de salida programadas para esa línea.

Información básica de una línea

Endpoint: `https://itranvias.com/queryitr_v3.php?func=1&dato=<ID_línea>` ¹³

- **Descripción:** Devuelve información básica sobre una línea específica. Esto incluye los datos descriptivos de la línea (similares a los obtenidos en los datos estáticos generales). Corresponde a `func=1`. En la práctica, la respuesta contiene una lista con la información de la(s) línea(s) solicitada(s) ¹⁴. Si se solicita una línea por ID, se obtendrá esa línea; en algunos casos este endpoint puede devolver todas las líneas (si *dato* no se especifica o se usa cierto valor especial), pero típicamente se emplea con un ID de línea específico. *(La documentación sugiere que devuelve una lista de líneas con la misma estructura descrita en la sección de datos estáticos)* ¹⁴.

- **Parámetros:**

- `func`: **Requerido.** Debe ser `1` para consultar información de línea.
- `dato`: **Requerido.** Identificador de la línea que se quiere consultar. Es un número que identifica la línea (por ejemplo, 100 para línea 1, 200 para línea 2, 500 para línea 5, etc., siguiendo la codificación utilizada en los datos estáticos).

- **Ejemplo de uso:**

```
GET https://itranvias.com/queryitr_v3.php?func=1&dato=500
```

(Consulta de la información de la línea con ID 500, que corresponde a la línea 5 en la numeración comercial.)

- **Ejemplo de respuesta (JSON):**

```
{
  "resultado": "OK",
  "fecha_peticion": "20230922222756",
  "peticion": "jlineas?",
  "tamaño": 4644,
  "lineas": [
    {
      "id": "100",
      "nom_comer": "1",
```

```

        "color_linea": "982135",
        "orig_linea": "Abente y Lago",
        "dest_linea": "Castrillón",
        "dest_ida": "Pza. Pablo Iglesias",
        "dest_vuelta": "Abente y Lago"
    },
    ...
],
"Origen": "Web_Beta"
}

```

En la respuesta, bajo la clave `lineas` se proporciona una lista (en el ejemplo, aparece la línea 100 que corresponde a la línea 1 comercial):

- `id`: Identificador de la línea (como cadena).
- `nom_comer`: Nombre comercial o número de la línea.
- `color_linea`: Código de color hexadecimal asociado a la línea.
- `orig_linea`: Nombre de la cabecera de origen de la línea (inicio del recorrido).
- `dest_linea`: Nombre de la cabecera de destino de la línea (fin del recorrido).
- `dest_ida`: Destino de la línea en sentido de ida (nombre de la última parada en el sentido de ida).
- `dest_vuelta`: Destino de la línea en sentido de vuelta (nombre de la última parada en el sentido de regreso).

Nota: Este endpoint puede ser útil para obtener la información de una línea específica sin tener que descargar todo el dataset estático. La estructura de la respuesta es similar a la del bloque de `lineas` de los datos estáticos ¹⁴.

Posición de autobuses en el recorrido de una línea

Endpoint: `https://itranvias.com/queryitr_v3.php?func=2&dato=<ID_línea>` ¹⁵

- **Descripción:** Proporciona el estado actual de los autobuses que operan en una línea dada, posicionados en su recorrido. Es decir, indica en qué tramo o parada del recorrido se encuentra cada autobús de la línea. Este endpoint (`func=2`) devuelve, para la línea consultada, la lista de paradas de cada **sentido** de la línea y los buses que se encuentran próximos a cada parada en ese momento ¹⁶. Sirve para visualizar la ubicación relativa de los autobuses a lo largo de la ruta (por orden de paradas).

- **Parámetros:**

- `func`: **Requerido.** Debe ser `2` para consultar posiciones de buses en la línea.
- `dato`: **Requerido.** Identificador de la línea a consultar (por ejemplo, 500 para línea 5).

- **Ejemplo de uso:**

```
GET https://itranvias.com/queryitr_v3.php?func=2&dato=500
```

(Consulta de los buses en ruta para la línea con ID 500.)

- **Ejemplo de respuesta (JSON abreviado):**

```

{
  "resultado": "OK",
  "fecha_peticion": "20230922224449",
  "peticion": "jparadas?linea=500&version=2",
  "tamaño": 315,
  "mls": 0,
  "paradas": [
    {
      "sentido": 0,
      "paradas": [
        {
          "parada": 184,
          "buses": [
            {
              "bus": 357,
              "estado": 1,
              "distancia": 0.288
            },
            ...
          ]
        },
        ...
      ]
    },
    {
      "sentido": 1,
      "paradas": [
        {
          "parada": 570,
          "buses": [
            {
              "bus": 342,
              "estado": 0,
              "distancia": 0.000
            }
          ]
        },
        ...
      ]
    }
  ],
  "Origen": "Web_Beta"
}

```

En la estructura devuelta, la clave `paradas` organiza la información por sentidos de la línea:

- Se listan dos objetos principales bajo `paradas`, uno por cada sentido de la línea (`"sentido": 0` para el sentido de ida, `"sentido": 1` para la vuelta).
- Cada uno contiene una lista de **paradas** (clave `"paradas"` dentro de cada sentido), en secuencia de recorrido. Para cada parada de ese sentido se indica:

- **parada** : El ID de la parada.
- **buses** : Una lista de autobuses que están actualmente cerca o en esa parada. Si no hay autobuses inmediatos en una parada, puede que la lista esté vacía o esa parada no aparezca en la respuesta (solo se listan paradas donde hay algún bus cercano/en ella). Cada objeto de la lista de **buses** tiene:
 - **bus** : ID del autobús.
 - **estado** : Estado del bus respecto a la parada:
 - **0**: El autobús está *en la parada* indicada.
 - **1**: El autobús está *en movimiento* (entre paradas).
 - **17**: El autobús se está incorporando a la ruta, está en un ramal/extensión, o fuera del itinerario habitual de ida/vuelta ¹⁷.
 - **distancia** : Representa la posición del bus a lo largo del recorrido, expresada como una fracción (valor entre 0.0 y 1.0). Un valor **0** significa que el bus está en la primera parada de ese sentido, **1** indica la última parada, y valores intermedios representan la fracción de distancia recorrida. *Nota*: Esta fracción **no** es la distancia real en metros, sino una proporción calculada asumiendo la distancia total del recorrido como 1.0 (las paradas intermedias se espacian equidistantemente en este cálculo) ¹⁸.

Este endpoint permite, por ejemplo, dibujar la línea en una interfaz y marcar aproximadamente dónde están los buses (por orden de paradas). No proporciona coordenadas geográficas, solo la posición relativa en la secuencia de paradas.

Posición geográfica de los buses de una línea (coordenadas GPS)

Endpoint: `https://itranvias.com/queryitr_v3.php?func=99&mostrar=B&dato=<ID_línea>`

¹⁹

- **Descripción:** Proporciona las **coordenadas geográficas** (longitud/latitud) en tiempo real de los autobuses de una línea determinada. Es útil para **mostrar los buses** moviéndose en un mapa. Este endpoint usa **func=99** con el parámetro **mostrar=B** (B de *buses*), para indicar que se quiere la ubicación de autobuses ¹⁹.
- **Parámetros:**
 - **func** : **Requerido.** Debe ser **99** para acceder a funcionalidades avanzadas (mapas en este caso).
 - **mostrar** : **Requerido.** Debe ser **B** (literal mayúscula) para solicitar posiciones de buses (B de *buses*).
 - **dato** : **Requerido.** Identificador de la línea a consultar (p.ej., 600, 500, etc.).

• Ejemplo de uso:

```
GET https://itranvias.com/queryitr_v3.php?func=99&mostrar=B&dato=600
```

(Consulta de coordenadas de los buses activos de la línea con ID 600.)

• Ejemplo de respuesta (JSON):

```
{
  "resultado": "OK",
  "fecha_peticion": "20230922231324",
  "peticion": "jmapas?linea=600&mostrar=b",
}
```



```

    "tamaño": 199,
    "mapas": [
      {
        "buses": [
          {
            "sentido": 0,
            "buses": [
              {
                "bus": 342,
                "posx": -8.413613,
                "posy": 43.362973
              },
              ...
            ]
          },
          {
            "sentido": 1,
            "buses": [
              {
                "bus": 357,
                "posx": -8.409210,
                "posy": 43.369842
              },
              ...
            ]
          }
        ],
        ...
      },
      ...
    ],
    "Origen": "Web_Beta"
  }

```

En esta respuesta, bajo la clave `mapas` se entrega un objeto (dentro de una lista) que contiene:

- `buses` : Una lista de dos elementos, uno por cada sentido (`sentido: 0` e `sentido: 1`).
- Para cada sentido, la lista interna `buses` proporciona la posición de los autobuses que están actualmente en ese sentido de la línea:
- `bus` : Identificador del autobús.
- `posx` : Coordenada X (longitud) de la posición actual del autobús (WGS84).
- `posy` : Coordenada Y (latitud) de la posición actual del autobús.

Si una línea no tiene buses en uno de los sentidos (por ejemplo, ningún bus en sentido de vuelta en ese momento), es posible que el objeto de ese sentido aparezca vacío o no aparezca. Normalmente, se listarán solo los sentidos donde haya autobuses en circulación.

Posición de las paradas de una línea (coordenadas de paradas)

Endpoint: `https://itranvias.com/queryitr_v3.php?func=99&mostrar=P&dato=<ID_línea>`

20

- **Descripción:** Devuelve las coordenadas geográficas de **todas las paradas** que conforman el recorrido de una línea dada. Útil para trazar las paradas en un mapa para esa línea. Usa `func=99` con `mostrar=P` (*P de paradas*).

- **Parámetros:**

- `func`: **Requerido.** `99` para funciones de mapa.
- `mostrar`: **Requerido.** `P` para obtener posiciones de paradas (*P de paradas*).
- `dato`: **Requerido.** ID de la línea a consultar.

- **Ejemplo de uso:**

```
GET https://itranvias.com/queryitr_v3.php?func=99&mostrar=P&dato=600
```

- **Ejemplo de respuesta (JSON abreviado):**

```
{
  "resultado": "OK",
  "fecha_peticion": "20230922231358",
  "peticion": "jmapas?linea=600&mostrar=p",
  "tamaño": 9540,
  "mapas": [
    {
      "paradas": [
        {
          "sentido": "0",
          "paradas": [
            {
              "id": "570",
              "parada": "Rda. de Monte Alto, 52",
              "posx": -8.40657,
              "posy": 43.380598
            },
            ...
          ]
        },
        {
          "sentido": "1",
          "paradas": [
            {
              "id": "184",
              "parada": "Avda. de Navarra (Praza dos
Av\u00e9dos)",
              "posx": -8.429150,
              "posy": 43.365486
            }
          ]
        }
      ]
    }
  ]
}
```

```

    },
    ...
  ],
  }
]
},
],
"Origen": "Web_Beta"
}

```

La respuesta, bajo `mapas`, contiene un objeto con la clave `paradas`, que a su vez es una lista de dos elementos (uno por cada sentido de la línea):

- Cada sentido (identificado por `"sentido": "0"` o `"1"`) incluye una lista `paradas` de todas las paradas en ese sentido del recorrido. Cada parada proporciona:
 - `id`: Identificador de la parada (como string en este caso).
 - `parada`: Nombre de la parada.
 - `posx`: Longitud de la ubicación de la parada.
 - `posy`: Latitud de la ubicación de la parada.

Este endpoint permite obtener rápidamente todas las paradas de una línea con sus coordenadas, para por ejemplo dibujarlas en un mapa o listar sus nombres.

Trazado del recorrido de una línea (polilínea de la ruta)

Endpoint: `https://itranvias.com/queryitr_v3.php?func=99&mostrar=R&dato=<ID_línea>`

21

- **Descripción:** Devuelve el *trazado geográfico completo* de la ruta de una línea, típicamente en forma de una serie de coordenadas que describen la polilínea del recorrido. Este endpoint usa `func=99` con `mostrar=R` (*R* de *recorrido*).

- **Parámetros:**

- `func`: **Requerido.** 99 para funciones de mapa.
- `mostrar`: **Requerido.** R para obtener el recorrido de la línea.
- `dato`: **Requerido.** ID de la línea a consultar.

- **Ejemplo de uso:**

```
GET https://itranvias.com/queryitr_v3.php?func=99&mostrar=R&dato=600
```

- **Ejemplo de respuesta (JSON abreviado):**

```

{
  "resultado": "OK",
  "fecha_peticion": "20230922231404",
  "peticion": "jmapas?linea=600&mostrar=r",
  "tamaño": 26731,

```

```

"mapas": [
  {
    "recorridos": [
      {
        "sentido": "0",
        "recorrido": "-8.406574518012148,43.3805987842061,0
-8.406662548136888,43.38047546094802,0 ..."
      },
      {
        "sentido": "1",
        "recorrido": "-8.42915046203125,43.36548635912345,0
-8.428990000000000,43.365300000000000,0 ..."
      }
    ]
  }
],
"Origen": "Web_Beta"
}

```

Bajo la clave `mapas`, el objeto contiene `recorridos` con dos entradas, una por sentido (`"0"` y `"1"`). Cada una tiene:

- `sentido`: `"0"` o `"1"`.
- `recorrido`: Un string con la secuencia de coordenadas que describen el trazado de la línea en ese sentido. Las coordenadas (longitud, latitud) están separadas por espacios, y cada par va seguido de `,0`. Este formato indica posiblemente la ruta punto a punto; el `0` podría ser un separador o un valor de altitud (0 m).

Para usar estos datos en un mapa, se suele dividir el string por espacios para obtener cada par `lon,lat` y trazar la polilínea resultante. Este endpoint permite dibujar el camino exacto que siguen los autobuses de la línea en el mapa.

Horarios de salida de una línea (servicios programados)

Endpoint:

`func=8&dato=<ID_línea>&fecha=<AAAAMMDD>`

`https://itranvias.com/queryitr_v3.php?`

22

- **Descripción:** Proporciona las **horas de salida** programadas de la línea consultada para una fecha específica. Es decir, devuelve los horarios de los servicios (tanto de ida como de vuelta) que tiene una línea en un día dado. Este endpoint corresponde a `func=8` e incluye un parámetro de fecha.
- **Parámetros:**
 - `func`: **Requerido.** `8` para consultar las salidas de la línea.
 - `dato`: **Requerido.** ID de la línea (por ejemplo, 500).
 - `fecha`: **Requerido.** Fecha para la cual se desean los horarios, en formato **AAAAMMDD** (año-mes-día, todo numérico). Por ejemplo, `20230922` para el 22 de septiembre de 2023.
- **Ejemplo de uso:**

```
GET https://itranvias.com/queryitr_v3.php?func=8&dato=500&fecha=20230922
```

(Solicita los horarios de la línea con ID 500 para el día 2023-09-22.)

• **Ejemplo de respuesta (JSON):**

```
{
  "resultado": "OK",
  "fecha_peticion": "20230922232647",
  "peticion": "servicios",
  "tamaño": 668,
  "servicios": [
    {
      "fecha": "20230922",
      "tipo": "Especial",
      "ida": [
        635,
        700,
        725,
        543,
        ...
      ],
      "vuelta": [
        700,
        725,
        743,
        757,
        ...
      ]
    }
  ],
  "Origen": "Web_Beta"
}
```

La respuesta contiene un arreglo `servicios` (puede haber más de un objeto si la fecha abarca distintos tipos de servicio, pero normalmente será uno por tipo de día):

- `fecha`: La fecha consultada (AAAAMMDD).
- `tipo`: Tipo de servicio para ese día (por ejemplo, *Laborable*, *Sábado*, *Festivo* o *Especial*). En el ejemplo, "Especial" indica un horario especial aplicado ese día.
- `ida`: Lista de horas de salida en sentido de ida (desde la cabecera de origen) para ese día.
- `vuelta`: Lista de horas de salida en sentido de vuelta (desde la cabecera de destino) para ese día.

Nota: Las horas de salida están representadas como números que suelen indicar el minuto del día. Es decir, probablemente **635** signifique 10:35 (ya que $10 \text{ horas} * 60 + 35 \text{ min} = 635$ minutos después de medianoche), **700** sería 11:40 ($11 * 60 + 40 = 700$), etc. Para convertirlos a horas legibles, se divide entre 60 para horas y el resto son minutos. La lista `ida` corresponde a los horarios en que un bus parte de la terminal inicial de la línea, y `vuelta` cuando parte de la terminal opuesta (regreso). Si una línea tiene varios tipos de servicios (por ejemplo, horarios especiales), podría haber múltiples objetos en el

array `servicios` con distintos `tipo`. En la mayoría de casos, habrá uno solo (p. ej., "Laborable", "Sábado" o "Festivo").

Leyenda de identificadores: Ten en cuenta que los identificadores de líneas y paradas son numéricos y se mantienen consistentes a través de los endpoints. Por ejemplo, una parada con `id: 184` en los datos estáticos será referenciada con `parada: 184` en las respuestas de posiciones de buses; una línea con `id: 500` corresponde a la misma línea en todos los endpoints de líneas. Se recomienda obtener primero la lista de líneas y paradas del endpoint de datos estáticos para luego utilizar correctamente estos IDs en las consultas de tiempo real.

Frecuencia de actualización: Los datos de *tiempo real* (próximos buses, posiciones de buses) cambian constantemente. Para evitar sobrecargar el servidor, se debe consultar en intervalos razonables (por ejemplo, cada 30 segundos o 1 minuto para posiciones). Los datos estáticos y horarios cambian con muy poca frecuencia (solo cuando hay modificaciones de rutas u horarios en el sistema).

Esta documentación está basada en la ingeniería inversa de la API usada por la web iTranvías ²³. Si bien recoge los principales endpoints conocidos hasta la fecha, podría haber funcionalidades adicionales no documentadas públicamente. Cada endpoint detallado aquí ha sido probado y observado según la información disponible ²⁴. Usando estos endpoints, los desarrolladores pueden construir aplicaciones con información tanto estática como en tiempo real del transporte urbano de A Coruña de forma eficiente.

1 2 3 4 5 6 **General - APIs Transporte Público Galicia**
<https://tpgalicia.github.io/urban/coruna/general/>

7 8 9 10 11 12 **Paradas - APIs Transporte Público Galicia**
<https://tpgalicia.github.io/urban/coruna/paradas/>

13 14 15 16 17 18 19 20 21 22 **Líneas - APIs Transporte Público Galicia**
<https://tpgalicia.github.io/urban/coruna/lineas/>

23 24 **Urbano de A Coruña - APIs Transporte Público Galicia**
<https://tpgalicia.github.io/urban/coruna/>