FATE SISTEMAS · SYAF

SYAF API y MCP
Manual de uso

Guía completa para consultar los operarios, frentes, montes, parcelas, máquinas, tipos de madera y partes diarios de tu empresa desde tus propios sistemas o desde un agente de inteligencia artificial.

`https://syafmcp-g6dxa5akfmetd4e3.brazilsouth-01.azurewebsites.net`

1. Qué es el servicio
2. Conceptos: clave, créditos y vigencia
3. Cómo empezar
4. Primera llamada
5. Operaciones
6. Parámetros y reglas
7. Recorrer y sincronizar datos
8. La respuesta
9. Campos de cada registro
10. Créditos: qué cuesta
11. Errores y límites de uso
12. Ejemplos de código
13. Colección de Postman
14. Conectar un agente de IA por MCP
15. Estado del servicio
16. Seguridad y privacidad
17. Soporte
18. Historial de versiones

Versión del servicio: **1.1.0** · Manual actualizado al 26/11/2026.

Las claves que aparecen son de ejemplo (`fap_EJEMPLO…`) y los datos de las respuestas son ficticios.

1 Qué es el servicio

SYAF API y MCP es un servicio de FATE Sistemas que abre los datos de tu empresa en SYAF para que otros programas los puedan leer.
Tu sistema de gestión, tu planilla, tu tablero de control o un asistente de IA pueden preguntar qué máquinas tiene un frente, qué parcelas tiene un monte o qué se hizo en los partes diarios de la semana, y reciben la respuesta al instante.

> **Es de solo lectura.**
> Consulta datos, pero nunca crea, modifica ni borra nada en SYAF.

Dos formas de entrar

API HTTP

Para integraciones

Rutas web clásicas (`/api/...`) para conectar un sistema de gestión, una planilla o cualquier programa que sepa hacer una llamada HTTP con `curl` , Postman o cualquier lenguaje.

SERVIDOR MCP

Para agentes de IA

Conexión directa (`/mcp`) para asistentes como Claude Code o Codex, que consultan tus datos cuando les hacés una pregunta en lenguaje natural.

Las dos formas usan la misma clave, ofrecen las mismas operaciones, dan exactamente los mismos datos y descuentan del mismo saldo de créditos.

Qué podés consultar

Personal y frentes

Los operarios con su cargo, categoría y frente actual.
Los frentes de trabajo con su encargado.

Montes y máquinas

Los montes con su superficie, departamento y región, y sus parcelas.
Las máquinas con su tipo, frente, horas y kilometraje.
Los tipos de madera.

Partes diarios

Los partes de un período, con horas, combustible y producción.
Cada parte trae sus productos, paradas y recargas.

Son solo datos operativos.
No se devuelven sueldos, jornales, comisiones, retenciones, liquidaciones, observaciones ni fotos.

Dirección del servicio

```
https://syafmcp-g6dxa5akfmetd4e3.brazilsouth-01.azurewebsites.net          # documentación interac
tiva (Swagger)
https://syafmcp-g6dxa5akfmetd4e3.brazilsouth-01.azurewebsites.net/api/...  # API HTTP
https://syafmcp-g6dxa5akfmetd4e3.brazilsouth-01.azurewebsites.net/mcp      # servidor MCP
```

2 Conceptos: clave, créditos y vigencia

La clave

- Empieza con `fap_` y tiene 47 caracteres.
- Está ligada a **una sola empresa**: todas las consultas responden con los datos de esa empresa, y no hay forma de elegir otra.
- Se entrega una sola vez.
  FATE no guarda la clave en claro: si la perdés, se genera una nueva y la anterior deja de funcionar en el acto.
- Al cambiar la clave se conservan el plan, los créditos y la vigencia.

El plan y los créditos

El plan te da una cantidad de **créditos por mes**.
Cada consulta cuesta un crédito, aunque no encuentre nada.
Consultar el saldo y el catálogo de operaciones es gratis.

- El mes es el mes calendario de Uruguay.
  El primer día de cada mes los créditos vuelven a estar completos.
- Los créditos no son acumulables: lo que no se usa en el mes no pasa al mes siguiente.
- Además del plan, FATE puede cargarte **créditos extra**.
  Se usan solo cuando se terminaron los del mes, y no vencen.
- Lo que todavía podés consultar en el mes es la suma de los créditos del mes que te quedan más los extra.

Ejemplo de un mes

Con un plan «Estándar» de 1.000 créditos por mes:

Momento	Créditos del mes	Extra	Qué pasa
1.º del mes	1.000 de 1.000	0	El plan se renueva completo.
Día 20, después de 1.000 consultas	0	0	Las consultas responden `SIN_CREDITOS` y no cobran.
Día 20, FATE carga 250 extra	0	250	Las consultas vuelven a funcionar en el acto, con la misma clave.
Día 30, después de 100 consultas más	0	150	Los extra se van descontando.
1.º del mes siguiente	1.000 de 1.000	150	El plan se renueva; los 150 extra siguen ahí.

La vigencia

La clave tiene una fecha de vigencia, un año por defecto.
La ves en la operación de saldo (`vigenteHasta`).

- Con **renovación automática**, al vencer se renueva sola por otro año, sin que tengas que hacer nada.
- Sin renovación automática, al vencer toda llamada responde `401 CLAVE_VENCIDA` con la fecha en que venció, y no cobra.
  Para seguir, pedí la renovación a FATE Sistemas: la clave sigue siendo la misma.

3 Cómo empezar

1. **Pedí tu clave a FATE Sistemas.**
   Sin ella no se puede continuar.
   Te la entregan una sola vez, por el canal que acuerden.
2. **Guardala en un lugar seguro**, como un gestor de contraseñas o una variable de entorno del servidor que la va a usar.
3. **Probala en el navegador** con la documentación interactiva (sección 4).
4. **Conectá tu sistema o tu agente** por la API HTTP (secciones 5 a 12) o por MCP (sección 14).

> **Cuidá la clave como una contraseña.**
>
> - No la pongas en la URL: queda en historiales y registros de servidores intermedios.
> - No la escribas en código que subas a un repositorio.
> - No la uses desde una página web en el navegador: cualquiera que abra la página la vería.
>   Las consultas tienen que salir de tu servidor, no del navegador de tus usuarios.
> - Si la perdés o sospechás que alguien la vio, pedí una nueva a FATE Sistemas: la anterior deja de funcionar.

Cómo mandar la clave

En cada llamada, en uno de estos dos headers (son equivalentes):

```
X-Api-Key: fap_EJEMPLO00000000000000000000000000000000000000
Authorization: Bearer fap_EJEMPLO00000000000000000000000000000000000000
```

Nunca en la URL ni en el cuerpo de la llamada.

4 Primera llamada

Desde el navegador (Swagger)

1. Abrí la dirección del servicio en el navegador: vas a ver la documentación interactiva con cada operación, sus parámetros y sus respuestas.
2. Tocá **Authorize**, pegá tu clave y confirmá.

3. En cada operación, tocá **Try it out**, completá los parámetros y tocá **Execute**.

La clave queda solo en esa pestaña y se borra al cerrarla o recargarla.
Cada consulta que ejecutes desde ahí cuesta créditos igual que desde cualquier otro programa.
El documento OpenAPI está en `/swagger/v1/swagger.json`, por si tu herramienta genera clientes a partir de él.

Desde la terminal

Para no dejar la clave en el historial de la terminal, cargala en una variable sin que se vea en pantalla (pegala y apretá Enter):

```
read -s SYAF_API_KEY && export SYAF_API_KEY
HOST=https://syafmcp-g6dxa5akfmetd4e3.brazilsouth-01.azurewebsites.net
```

Comprobá que el servicio está en línea (no pide clave):

```
curl "$HOST/health"
# ok
```

Consultá tu saldo (no cuesta créditos):

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/cuenta/saldo"
```

Si la respuesta empieza con `"ok":true`, la clave funciona.
Si dice `CLAVE_INVALIDA`, revisá que la copiaste completa.

Tu primera consulta

Los frentes de trabajo activos de tu empresa (cuesta un crédito):

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/frentes"
```

Los ejemplos de este manual usan las variables `HOST` y `SYAF_API_KEY` de esta sección.

5 Operaciones

Todas las rutas son `GET`.
Cada operación existe por HTTP y como herramienta MCP, con los mismos parámetros.
Cada entidad tiene dos operaciones: `listar`, que la recorre por páginas, y `porId`, que trae un registro.

Operación	Ruta HTTP	Herramienta MCP	Costo
Listar operarios	`/api/operarios`	`operarios_listar`	1 crédito
Operario por id	`/api/operarios/{id}`	`operarios_por_id`	1 crédito
Listar frentes	`/api/frentes`	`frentes_listar`	1 crédito
Frente por id	`/api/frentes/{id}`	`frentes_por_id`	1 crédito
Listar montes	`/api/montes`	`montes_listar`	1 crédito
Monte por id	`/api/montes/{id}`	`montes_por_id`	1 crédito
Listar parcelas	`/api/parcelas`	`parcelas_listar`	1 crédito
Parcela por id	`/api/parcelas/{id}`	`parcelas_por_id`	1 crédito
Listar máquinas	`/api/maquinas`	`maquinas_listar`	1 crédito
Máquina por id	`/api/maquinas/{id}`	`maquinas_por_id`	1 crédito
Listar tipos de madera	`/api/tiposMadera`	`tipos_madera_listar`	1 crédito
Tipo de madera por id	`/api/tiposMadera/{id}`	`tipos_madera_por_id`	1 crédito
Listar partes diarios	`/api/partesDiarios`	`partes_diarios_listar`	1 crédito
Parte diario por id	`/api/partesDiarios/{id}`	`partes_diarios_por_id`	1 crédito
Saldo de créditos	`/api/cuenta/saldo`	`cuenta_saldo`	Gratis
Catálogo de operaciones	`/api/catalogo`	`catalogo`	Gratis

En las operaciones por id, el `id` va en la ruta por HTTP y como argumento en la herramienta MCP.
Devuelven el registro aunque esté inactivo: mirá el campo `activo`.
Si el id no existe, `resultado` es `vacio`, no viene `datos` y cuesta un crédito igual.

Operarios

Listar operarios

`GET /api/operarios · MCP operarios_listar`

Los empleados de tu empresa por páginas, ordenados por id, con su cargo, categoría y frente actual. Sirve para recorrer el maestro completo o sincronizarlo con otro sistema (sección 7).

Parámetro	Tipo	Valores	Si no lo mandás
`desdeId`	entero	0 o más; trae los ids mayores a este valor	`0`
`cantidad`	entero	1 a 200	`50`
`estado`	opción	`activos`, `inactivos` o `todos`	`activos`
`idFrente`	entero	0 o más; solo los operarios de ese frente. Con `0`, los que no tienen frente	Sin filtro
`modificadosDesde`	fecha	`AAAA-MM-DD` o con hora	Sin filtro de fecha

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/operarios?desdeId=10&cantidad=2&idFrente=1"
```

Respuesta: una lista de operarios en `datos` (campos en la sección 9) y, si la página vino llena, `siguienteDesdeId`.

Operario por id

`GET /api/operarios/{id} · MCP operarios_por_id`

El operario con ese id, con su cargo, categoría y frente actual, o ninguno.

Parámetro	Tipo	Valores	Si no lo mandás
`id`	entero	1 o más	Obligatorio

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/operarios/12"
```

Frentes

Listar frentes

`GET /api/frentes · MCP frentes_listar`

Los frentes de trabajo por páginas, ordenados por id, con el nombre de su encargado.
Parámetros: `desdeId`, `cantidad`, `estado` y `modificadosDesde`, iguales que en listar operarios.

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/frentes?desdeId=2&cantidad=2"
```

Frente por id

`GET /api/frentes/{id}` · MCP `frentes_por_id`

El frente con ese id, con su encargado, o ninguno.
Parámetro: `id`, entero de 1 o más, obligatorio.

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/frentes/4"
```

Montes

Listar montes

`GET /api/montes` · MCP `montes_listar`

Los montes (predios) por páginas, ordenados por id, con su superficie, departamento y región.
Parámetros: `desdeId`, `cantidad`, `estado` y `modificadosDesde`, iguales que en listar operarios.

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/montes?cantidad=2"
```

Monte por id

`GET /api/montes/{id}` · MCP `montes_por_id`

El monte con ese id, con su departamento y región, o ninguno.
Parámetro: `id`, entero de 1 o más, obligatorio.

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/montes/1"
```

Parcelas

Listar parcelas

`GET /api/parcelas` · MCP `parcelas_listar`

Las parcelas de los montes por páginas, ordenadas por id, con su monte, superficie, ubicación y plantación.
Parámetros: `desdeId`, `cantidad`, `estado` y `modificadosDesde`, iguales que en listar operarios, y además:

Parámetro	Tipo	Valores	Si no lo mandás
`idMonte`	entero	0 o más; solo las parcelas de ese monte. Con `0`, las que no tienen monte	Sin filtro

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/parcelas?cantidad=2&idMonte=2"
```

Parcela por id

`GET /api/parcelas/{id}` · MCP `parcelas_por_id`

La parcela con ese id, con su monte, superficie y ubicación, o ninguna.
Parámetro: `id`, entero de 1 o más, obligatorio.

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/parcelas/7"
```

Máquinas

Listar máquinas

`GET /api/maquinas` · MCP `maquinas_listar`

Las máquinas por páginas, ordenadas por id, con su tipo, frente actual, horas y kilometraje.
Parámetros: `desdeId`, `cantidad`, `estado` y `modificadosDesde`, iguales que en listar operarios, y además:

Parámetro	Tipo	Valores	Si no lo mandás
`idFrente`	entero	0 o más; solo las máquinas de ese frente. Con `0`, las que no tienen frente	Sin filtro

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/maquinas?cantidad=2&idFrente=1"
```

Máquina por id

`GET /api/maquinas/{id}` · MCP `maquinas_por_id`

La máquina con ese id, con su tipo y frente actual, o ninguna.
Parámetro: `id`, entero de 1 o más, obligatorio.

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/maquinas/1"
```

Tipos de madera

Listar tipos de madera

`GET /api/tiposMadera` · MCP `tipos_madera_listar`

Los tipos de madera por páginas, ordenados por id.
Parámetros: `desdeId`, `cantidad`, `estado` y `modificadosDesde`, iguales que en listar operarios.

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/tiposMadera?cantidad=2"
```

Tipo de madera por id

`GET /api/tiposMadera/{id}` · MCP `tipos_madera_por_id`

El tipo de madera con ese id, o ninguno.
Parámetro: `id`, entero de 1 o más, obligatorio.

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/tiposMadera/1"
```

Partes diarios

Listar partes diarios

`GET /api/partesDiarios` · MCP `partes_diarios_listar`

Los partes diarios cuya fecha cae en un período, por páginas, ordenados por id.
Cada parte trae adentro sus productos, paradas y recargas.

Parámetro	Tipo	Valores	Si no lo mandás
`desde`	fecha	`AAAA-MM-DD`; primer día del período. Se compara con la fecha del parte	Obligatorio
`hasta`	fecha	`AAAA-MM-DD`; último día del período, incluido. El período no puede superar los 31 días	Obligatorio
`desdeId`	entero	0 o más; trae los ids mayores a este valor	`0`
`cantidad`	entero	1 a 200	`50`
`estado`	opción	`activos`, `inactivos` o `todos`	`activos`
`idFrente`	entero	0 o más; solo los partes de ese frente	Sin filtro
`idMaquina`	entero	0 o más; solo los partes de esa máquina	Sin filtro
`idOperario`	entero	0 o más; solo los partes donde esa persona figura como operario o como maquinista	Sin filtro
`idMonte`	entero	0 o más; solo los partes de ese monte	Sin filtro

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/partesDiarios?desde=2026-11-16&hasta=2026-11-22"
```

Un período de más de 31 días, o con `hasta` anterior a `desde`, responde `400 PARAMETRO_INVALIDO` y no cuesta créditos.
Para traer un período más largo, partilo en tramos de hasta 31 días.
Esta operación no tiene `modificadosDesde`.

Parte diario por id

`GET /api/partesDiarios/{id}` · MCP `partes_diarios_por_id`

El parte diario con ese id, con sus productos, paradas y recargas, o ninguno.
Parámetro: `id`, entero de 1 o más, obligatorio.

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/partesDiarios/4790"
```

```
"datos": {
  "id": 4790, "fecha": "2026-11-19T00:00:00", "turno": "D",
  "maquinista": "JUAN PÉREZ GÓMEZ", "maquina": "FORWARDER 1", "frente": "FRENTE 1",
  "actividad": "EXTRACCION", "monte": "LOS PINOS", "parcela": "PARCELA NORTE",
  "hsOperario": 9.5, "hsMaquina": 8.2, "combustibleConsumido": 95, "m3Total": 184.5, ...
  "productos": [ { "idProducto": 1, "producto": "Rolos Export", "largo": 7.2, "volumen": 120,
"carros": 10, ... } ],
  "paradas":   [ { "idMotivo": 1, "motivo": "Descanso", "minutos": 30, "horas": 0.5, ... } ],
  "recargas":  [ { "idRecarga": 1, "recarga": "Gasoil", "cantidad": 95 } ]
}
```

Cuenta

Saldo de créditos

`GET /api/cuenta/saldo` · MCP `cuenta_saldo` · gratis

El plan, los créditos del mes y la vigencia de la clave.
No tiene parámetros.

```
"datos": {
  "plan": "Estándar", "mensuales": 1000, "usadosMes": 45, "extra": 0,
  "mes": "2026-11", "vigenteHasta": "2027-11-24", "renovacionAutomatica": false,
  "disponiblesMes": 955
}
```

Catálogo de operaciones

`GET /api/catalogo` · MCP `catalogo` · gratis

La lista de operaciones con sus parámetros, tipos, límites, valores por defecto, costo y descripción, tal como las valida el servicio.
Sirve para que un programa o un agente descubra qué puede consultar sin leer este manual.
No tiene parámetros.

```
{ "nombre": "frentes.listar", "descripcion": "Lista frentes de trabajo por lotes ordenados po
r id, con su encargado.", "costo": 1,
  "parametros": [
    { "nombre": "cantidad", "tipo": "entero", "requerido": false, "min": 1, "max": 200, "defa
ult": "50", "descripcion": "…" },
    { "nombre": "estado", "tipo": "opcion", "requerido": false, "min": 0, "max": 0, "defaul
t": "activos", "descripcion": "…",
      "opciones": ["activos", "inactivos", "todos"] }
  ] }
```

6 Parámetros y reglas

Todos los parámetros

Parámetro	Tipo	Valores admitidos	Si no lo mandás	Dónde
`id`	entero	1 o más	Obligatorio	todas las operaciones por id
`desdeId`	entero	0 o más	`0`	todos los listados
`cantidad`	entero	1 a 200	`50`	todos los listados
`estado`	opción	`activos`, `inactivos`, `todos`	`activos`	todos los listados
`modificadosDesde`	fecha	ver formatos de fecha	Sin filtro	listar operarios, frentes, montes, parcelas, máquinas y tipos de madera
`idFrente`	entero	0 o más	Sin filtro	listar operarios, máquinas y partes diarios
`idMonte`	entero	0 o más	Sin filtro	listar parcelas y partes diarios
`idMaquina`	entero	0 o más	Sin filtro	listar partes diarios
`idOperario`	entero	0 o más	Sin filtro	listar partes diarios
`desde`	fecha	ver formatos de fecha	Obligatorio	listar partes diarios
`hasta`	fecha	ver formatos de fecha; hasta 31 días contando `desde`	Obligatorio	listar partes diarios

Formatos de fecha

Formato	Ejemplo
Solo fecha	`2026-11-23`
Fecha y hora	`2026-11-23T14:30:00`
Con fracciones de segundo	`2026-11-23T14:30:00.25`
Con zona horaria	`2026-11-23T14:30:00-03:00` o `2026-11-23T17:30:00Z`

Las fechas de SYAF se guardan sin zona horaria, en la hora de la base de tu empresa, y se comparan con lo que mandás tal cual.
Si le ponés zona (`Z` o `-03:00`), primero se pasa a UTC.
Lo más simple es no mandar zona.
En `desde` y `hasta` mandá solo la fecha (`AAAA-MM-DD`): el período se cuenta por días.
El año mínimo es 1753.

Reglas

- Los nombres de parámetro distinguen mayúsculas: `desdeid` no es `desdeId` .

- Los valores de `estado` también: van en minúsculas, tal como están escritos.
- Un parámetro que la operación no conoce, uno repetido, uno obligatorio que falta o un valor fuera de rango responden `400 PARAMETRO_INVALIDO`.
  El mensaje dice cuál, y la llamada no cuesta créditos.
- Las columnas que vuelven y el orden son fijos por operación: no se eligen.
  Todos los listados vienen ordenados por id.
- `estado` existe solo en los listados.
  Las operaciones por id devuelven el registro aunque esté inactivo.
- `idFrente=0` en operarios y máquinas trae los que no tienen frente, e `idMonte=0` en parcelas trae las que no tienen monte.
  En la respuesta vienen con el id en `0` y el nombre vacío.
- Los filtros se combinan: se devuelven solo los registros que cumplen todos.
- Una página trae como máximo 200 filas.
- No hay búsqueda por texto: para encontrar un registro por nombre, recorré el listado y filtrá en tu sistema.

7 Recorrer y sincronizar datos

Recorrer un maestro completo

Vale para operarios, frentes, montes, parcelas, máquinas y tipos de madera.

1. Llamá al listado con `desdeId=0` y `cantidad=200`.
2. Si la respuesta trae `siguienteDesdeId`, llamá de nuevo con `desdeId` igual a ese valor.
3. Cuando la respuesta ya no trae `siguienteDesdeId`, terminaste.

Cada página cuesta un crédito: con `cantidad=200` recorrés 200 registros por crédito.
Si la última página vino justo llena, la siguiente llamada vuelve `vacio`: también cuesta un crédito y es la señal de que terminaste.
Para incluir las bajas, agregá `estado=todos`.

Traer solo los cambios con `modificadosDesde`

Después de la primera recorrida completa, no hace falta volver a bajar todo.
Los listados de los seis maestros aceptan `modificadosDesde`: devuelven solo los registros dados de alta, de baja o modificados desde esa fecha.

1. Anotá la fecha y hora en que empezás la sincronización.
2. Llamá al listado con `estado=todos` y `modificadosDesde` igual al comienzo de la sincronización anterior.
   Si vienen varias páginas, seguí con `siguienteDesdeId` y los mismos filtros.
3. Guardá la fecha que anotaste en el paso 1: es el `modificadosDesde` de la próxima vez.

```
curl -H "X-Api-Key: $SYAF_API_KEY" "$HOST/api/frentes?estado=todos&modificadosDesde=2026-11-23"
```

- Cada registro trae `fechaUpd`, su último cambio conocido, cuando SYAF lo tiene.
- Un registro dado de baja llega con `activo: false`: desactivalo en tu sistema.
- La llamada cuesta un crédito aunque no haya cambios (`"resultado": "vacio"`).
- Volver a leer un registro no hace daño: ante la duda, pedí desde un rato antes.

> **Mandá siempre `estado=todos` junto con `modificadosDesde`.**
> Con el valor por defecto (`activos`) no te enterás de las bajas.

> **Hacé cada tanto una recorrida completa.**
> La fecha de cambio depende de que SYAF la registre al guardar.
> Para no depender de eso, recorré el maestro entero sin `modificadosDesde` cada cierto tiempo, por ejemplo una vez por semana.

Traer los partes diarios

1. Llamá a `partesDiarios` con `desde`, `hasta` y `cantidad=200`.
2. Si la respuesta trae `siguienteDesdeId`, llamá de nuevo con las mismas fechas y ese `desdeId`.
3. Para un período de más de 31 días, repetí lo anterior por tramos de hasta 31 días.

Los partes no tienen `modificadosDesde`: para enterarte de correcciones, volvé a pedir el período que te interesa.
Con `estado=todos` vienen también los partes dados de baja, con `activo: false`.

Receta recomendada para mantener un sistema al día

Momento	Qué hacer	Costo
Carga inicial	Anotá la fecha y hora, y recorré cada maestro con `cantidad=200` y `estado=todos`.	1 crédito cada 200 registros
Sincronizaciones	Llamá a cada listado con `estado=todos` y `modificadosDesde` igual al comienzo de la sincronización anterior.	1 crédito por página; 1 si no hubo cambios
Cada cierto tiempo	Recorrida completa de cada maestro, sin `modificadosDesde`.	1 crédito cada 200 registros
Partes diarios	Pedí el día o la semana que cerró, con `desde` y `hasta`.	1 crédito cada 200 partes

Cuántos créditos se usan

Uso	Créditos aproximados
Bajar un maestro de 1.000 registros	5 o 6
Traer los cambios de los seis maestros, una vez por día, sin novedades	6 por día, unos 180 por mes
Traer los partes de una semana con 300 partes	2
Consultar un registro por id	1

Son cuentas a partir de la regla de un crédito por llamada: tu uso real depende de cuántos registros tengas y de cada cuánto consultes.

8 La respuesta

Todas las respuestas, por HTTP y por MCP, vienen en el mismo sobre JSON.
Los nombres de los campos van en camelCase.
Las propiedades que no aplican no aparecen: nunca vienen con `null`.

Ejemplo completo


```
{
  "ok": true,
  "operacion": "operarios.listar",
  "resultado": "ok",
  "cantidad": 1,
  "datos": [
    {
      "id": 12,
      "nroEmpleado": 112,
      "nombre": "JUAN",
      "apellidos": "PÉREZ GÓMEZ",
      "cedula": "12345678",
      "idCargo": 6,
      "cargo": "OPERADOR DE COSECHADORA",
      "idCategoria": 2,
      "categoria": "OFICIAL",
      "idFrente": 1,
      "frente": "FRENTE 1",
      "telefono": "",
      "celular": "099000000",
      "mail": "juan.perez@ejemplo.com",
      "fechaIngreso": "2023-03-01T00:00:00",
      "activo": true,
      "fechaAlta": "2023-02-27T10:20:00",
      "fechaUpd": "2026-09-14T08:05:00"
    }
  ],
  "siguienteDesdeId": 12,
  "consultadoEn": "2026-11-24T13:01:11.5481234Z",
  "creditos": {
    "consumidos": 1,
    "mes": "2026-11",
    "plan": "Estándar",
    "mensuales": 1000,
    "usadosMes": 47,
    "disponiblesMes": 953,
    "extra": 0
  },
  "requestId": "7c41e09b2d5f4a8e9b6c3d2e1f0a0011"
}
```


El ejemplo pidió `cantidad=1`: como la página vino llena, trae `siguienteDesdeId`.
El operario no tiene `fechaEgreso` ni `fechaBaja` porque sigue en la empresa: las fechas sin dato no aparecen.

Sobre de una consulta

Campo	Qué es
`ok`	`true` si la consulta se hizo.
`operacion`	Nombre de la operación, por ejemplo `operarios.listar`.
`resultado`	`ok` si trajo datos, `vacio` si no encontró nada. `vacio` no es un error y cuesta un crédito.
`cantidad`	Cantidad de registros en `datos`. En `catalogo` vale `1` aunque `datos` traiga todas las operaciones.
`datos`	Un objeto en `saldo` y en las consultas por id; una lista en los listados y en `catalogo`. Con resultado `vacio`, no aparece (objeto) o es `[]` (lista).
`siguienteDesdeId`	Solo en los listados, y solo cuando vinieron tantas filas como pediste en `cantidad`: es el `desdeId` de la página siguiente.
`consultadoEn`	Fecha y hora de la consulta, en UTC.
`creditos`	Estado de tus créditos después de esta llamada.
`requestId`	Identificador de la llamada. Si algo falla, mandalo a soporte.

Bloque `creditos`

Campo	Qué es
`consumidos`	Créditos que costó esta llamada: `1` en una consulta, `0` en saldo y catálogo.
`mes`	Mes al que se imputó, `AAAA-MM`, en hora de Uruguay.
`plan`	Nombre de tu plan.
`mensuales`	Créditos que da el plan por mes.
`usadosMes`	Créditos consumidos en el mes, contando esta llamada. Incluye los que se tomaron del extra.
`disponiblesMes`	Créditos del plan que te quedan este mes: `mensuales` menos `usadosMes`, nunca menos de cero.
`extra`	Créditos extra que te quedan. Se usan cuando `disponiblesMes` llega a cero.

Lo que todavía podés consultar este mes es `disponiblesMes` más `extra`.
Como cada respuesta trae este bloque, no hace falta llamar a `cuenta/saldo` después de cada consulta.

Sobre de un error

```
{
  "ok": false,
  "error": {
    "codigo": "SIN_CREDITOS",
    "mensaje": "No quedan créditos disponibles este mes.\nContactá a FATE Sistemas para habilitar
más."
  },
  "requestId": "7c41e09b2d5f4a8e9b6c3d2e1f0a0011"
}
```

`error.codigo` es estable y es lo que conviene mirar desde un programa.
`error.mensaje` es para mostrar a una persona y puede cambiar de redacción.

9 Campos de cada registro

Operario

Campo	Tipo	Qué es
`id`	entero	Id interno del operario.
`nroEmpleado`	entero	Número de empleado.
`nombre`, `apellidos`, `cedula`	texto	Datos personales.
`idCargo`, `cargo`	entero, texto	Cargo laboral (id y nombre).
`idCategoria`, `categoria`	entero, texto	Categoría del empleado (id y nombre).
`idFrente`, `frente`	entero, texto	Frente de trabajo actual (id y nombre). `0` y `""` si no tiene.
`telefono`, `celular`, `mail`	texto	Contacto.
`fechaIngreso`, `fechaEgreso`	fecha	Ingreso y egreso de la empresa.
`activo`	booleano	`true` si el operario está activo.
`fechaAlta`, `fechaBaja`, `fechaUpd`	fecha	Alta, baja y última modificación del registro.

Frente

Campo	Tipo	Qué es
`id`	entero	Id interno del frente.
`nombre`	texto	Nombre del frente.
`idEncargado`, `encargado`	entero, texto	Operario encargado del frente: su id y su nombre y apellidos. `0` y `""` si no tiene.
`idClienteSector`	entero	Id del cliente o sector al que trabaja el frente.
`esTaller`	booleano	`true` si el frente es un taller.

`idCentroCosto`, `idDeposito`	entero	Centro de costo y depósito asociados.
`erpCode`	texto	Código del frente en tu sistema de gestión, si lo cargaste.
`activo`	booleano	`true` si el frente está activo.
`fechaAlta`, `fechaBaja`	fecha	Alta y baja del registro.
`fechaUpd`	fecha	Último cambio conocido.

Monte

Campo	Tipo	Qué es
`id`	entero	Id interno del monte.
`nombre`, `abrev`	texto	Nombre y abreviatura.
`hectareas`	número	Superficie en hectáreas, redondeada a dos decimales.
`idDepartamento`, `departamento`	entero, texto	Departamento (id y nombre).
`idRegion`, `region`	entero, texto	Región (id y nombre).
`direccion`, `contacto`, `telefono`	texto	Ubicación y contacto del monte.
`visible`	booleano	Marca `visible` de SYAF.
`activo`	booleano	`true` si el monte está activo.
`fechaAlta`, `fechaBaja`	fecha	Alta y baja del registro.
`fechaUpd`	fecha	Último cambio conocido.

Parcela

Campo	Tipo	Qué es
`id`	entero	Id interno de la parcela.
`idMonte`, `monte`	entero, texto	Monte al que pertenece (id y nombre). `0` y `""` si no tiene.
`nroParcela`, `descripcion`	entero, texto	Número y descripción de la parcela dentro del monte.
`hectareas`, `hectareasParcela`	número	Superficie en hectáreas, redondeada a dos decimales.
`volArbol`	número	Volumen por árbol, redondeado a dos decimales.
`latitud`, `longitud`	número	Ubicación en grados decimales, sin redondear. No aparecen cuando la parcela no tiene coordenadas cargadas.
`fechaPlantacion`	fecha	Fecha de plantación, si está cargada.
`materialGenetico`	texto	Material genético plantado.

`visible`	booleano	Marca `visible` de SYAF.
`activo`	booleano	`true` si la parcela está activa.
`fechaUpd`	fecha	Último cambio conocido.

Máquina

Campo	Tipo	Qué es
`id`	entero	Id interno de la máquina.
`nroMaquina`, `descripcion`, `alias`	entero, texto	Número, descripción y alias de la máquina.
`matricula`, `marca`, `modelo`, `serie`, `nroMotor`	texto	Identificación del vehículo.
`idTipo`, `tipo`	entero, texto	Tipo de vehículo (id y nombre), por ejemplo `FORWARDER` o `CAMIONETA`.
`idFrente`, `frente`	entero, texto	Frente de trabajo actual (id y nombre). `0` y `""` si no tiene.
`kmActual`, `hsActual`	número	Kilometraje y horas de uso actuales, con dos decimales.
`cisterna`	booleano	`true` si la máquina es una cisterna.
`erpCode`	texto	Código de la máquina en tu sistema de gestión, si lo cargaste.
`activo`	booleano	`true` si la máquina está activa.
`fechaAlta`, `fechaBaja`	fecha	Alta y baja del registro.
`fechaUpd`	fecha	Último cambio conocido.

Tipo de madera

Campo	Tipo	Qué es
`id`	entero	Id interno del tipo de madera.
`nombre`	texto	Nombre, por ejemplo `GLOBULUS` o `GRANDIS`.
`activo`	booleano	`true` si está activo.
`fechaAlta`, `fechaBaja`	fecha	Alta y baja del registro.
`fechaUpd`	fecha	Último cambio conocido.

Parte diario

Campo	Tipo	Qué es
`id`	entero	Id interno del parte.

`fecha`	fecha	Día del parte: es la fecha que filtran `desde` y `hasta`.
`turno`	texto	Turno, tal como lo carga la app.
`fechaHoraIni`, `fechaHoraFin`	fecha	Comienzo y fin de la jornada.
`idMaquinista`, `maquinista`	entero, texto	Quien manejó la máquina.
`idOperario`, `operario`	entero, texto	Empleado al que corresponde el parte.
`idMaquina`, `maquina`	entero, texto	Máquina (su alias).
`idFrente`, `frente`	entero, texto	Frente de trabajo.
`idActividad`, `actividad`	entero, texto	Actividad.
`idMonte`, `monte`, `idParcela`, `parcela`	entero, texto	Dónde se trabajó.
`hsOperario`	número	Horas del operario.
`hsMaquina`, `hsMaquinaIni`, `hsMaquinaFin`, `hsMotor`	número	Horas de máquina y lecturas de horómetro.
`combustibleConsumido`	número	Combustible consumido.
`arboles`, `hectareas`, `volArbol`, `cantPlantas`	número	Producción cargada en la cabecera.
`m3Inicial`, `m3Final`, `m3Total`, `distancia`	número	Volumen y distancia.
`nroRegistroPapel`	entero	Número del parte en papel, si se cargó.
`activo`, `fechaBaja`, `fechaUpd`	booleano, fecha	Estado, baja y último cambio del parte.
`productos`	lista	Lo producido: ver abajo.
`paradas`	lista	Las paradas de la jornada: ver abajo.
`recargas`	lista	Las recargas: ver abajo.

Detalle del parte

Lista	Campos de cada elemento
`productos`	`idProducto`, `producto` (id y nombre), `largo`, `trozas`, `volumen`, `carros`, `camiones`, `viajes`, `arboles` e `idParcela`.
`paradas`	`idMotivo`, `motivo` (id y nombre), `fechaIni`, `fechaFin`, `minutos` y `horas`.
`recargas`	`idRecarga`, `recarga` (id y nombre) y `cantidad`.

Un parte sin productos, paradas o recargas trae esa lista vacía.
Del parte salen las horas, los consumos y la producción.
No salen la liquidación, la aprobación, las observaciones ni las fotos.

Saldo (`cuenta/saldo`)

`datos` trae `plan` , `mensuales` , `usadosMes` , `extra` , `mes` y `disponiblesMes` , con el mismo significado que en el bloque `creditos` .
Además trae la vigencia de la clave:

Campo	Qué es
`vigenteHasta`	Último día en que vale la clave (`AAAA-MM-DD`).
`renovacionAutomatica`	`true` si al vencer se renueva sola por otro año.

Formato de los valores

- Los textos vienen sin espacios a los costados y con 500 caracteres como máximo.
- Las cantidades son números con decimales.
- Las fechas vienen sin zona horaria, como las guarda tu base.
- Un texto sin valor viene como `""` y un número sin valor como `0` .
- Una fecha que SYAF no tiene cargada no aparece en la respuesta, por ejemplo `fechaEgreso` de un operario que sigue en la empresa.
  SYAF marca esas fechas vacías con un valor anterior a 1901 o con el 1980-01-01 00:00, y el servicio las omite.

No se devuelven sueldos, jornales, comisiones, retenciones, precios de liquidación, observaciones ni fotos.

10 Créditos: qué cuesta

Situación	¿Cuesta?
Consulta que trae datos (`ok`)	1 crédito
Consulta que no encuentra nada (`vacio`)	1 crédito
`cuenta/saldo` y `catalogo`	No
Clave inválida o vencida, parámetro inválido, límite de llamadas o sin créditos	No
La base de tu empresa no respondió, o error interno del servicio	Se cobra y se devuelve en la misma llamada
Mensajes de protocolo de MCP (conexión, lista de herramientas)	No
`/health` , `/version` y `/postman`	No (no piden clave)

Un parte diario cuesta lo mismo que cualquier otro registro: su detalle de productos, paradas y recargas viene en la misma llamada.

> **Un reintento es una llamada nueva y cuesta otro crédito.**
> Si tu programa reintenta automáticamente, que lo haga solo ante `429` , `500` o `503` , que no cobran.
> Nunca reintentes una respuesta `ok` o `vacio` .

Cuando se terminan los créditos

- Cada consulta responde `402 SIN_CREDITOS` y no se cobra.
- El saldo y el catálogo siguen funcionando.
- FATE Sistemas recibe un aviso automático la primera vez en el mes que te quedás sin créditos.
- Para seguir antes del mes siguiente, pedí créditos extra a FATE Sistemas.
  Quedan disponibles en cuanto se cargan, sin cambiar la clave.
- También podés pedir un cambio de plan: la clave sigue siendo la misma.

Para no gastar de más

- Usá `cantidad=200` en los listados: cada página cuesta lo mismo, traiga 1 o 200 registros.
- Sincronizá con `modificadosDesde` en lugar de recorrer el maestro completo cada vez.
- Usá los filtros (`idFrente`, `idMonte`, `idMaquina`, `idOperario`) para traer solo lo que necesitás.
- Guardá en tu sistema lo que ya trajiste: los maestros cambian poco.
- Leé el bloque `creditos` de cada respuesta en lugar de llamar al saldo.

11 Errores y límites de uso

Códigos de error

Código	HTTP	Cuándo	¿Cobra?	Qué hacer
`CLAVE_INVALIDA`	401	Falta la clave, no existe, fue dada de baja o no está habilitada para SYAF.	No	Revisar el header y que la clave esté completa. Si sigue, consultar a FATE Sistemas.
`CLAVE_VENCIDA`	401	La clave pasó su fecha de vigencia y no tiene renovación automática.	No	Pedir la renovación a FATE Sistemas.
`PARAMETRO_INVALIDO`	400	Parámetro desconocido, repetido, faltante o fuera de rango, o período de más de 31 días. El mensaje dice cuál.	No	Corregir la llamada.
`LIMITE_LLAMADAS`	429	Más de 60 llamadas por minuto o más de 2 a la vez con la misma clave.	No	Esperar lo que indica el header `Retry-After` (60 segundos) y reintentar.
`SIN_CREDITOS`	402	No quedan créditos del mes ni extra.	No	Pedir créditos extra o esperar al mes siguiente.
`EMPRESA_NO_DISPONIBLE`	503	No se pudo consultar la base de tu empresa.	Se devuelve	Reintentar en unos minutos. Si persiste, mandar el `requestId` a soporte.
`ERROR_INTERNO`	500	Cualquier otra falla del servicio.	Se devuelve	Mandar el `requestId` a soporte.

Ejemplos

```
{ "ok": false,
  "error": { "codigo": "PARAMETRO_INVALIDO", "mensaje": "El parámetro cantidad debe ser un entero
entre 1 y 200" },
  "requestId": "7c41e09b2d5f4a8e9b6c3d2e1f0a0013" }
```

```
{ "ok": false,
  "error": { "codigo": "PARAMETRO_INVALIDO", "mensaje": "El período entre desde y hasta no puede s
uperar los 31 días" },
  "requestId": "7c41e09b2d5f4a8e9b6c3d2e1f0a0015" }
```

```
{ "ok": false,
  "error": { "codigo": "LIMITE_LLAMADAS", "mensaje": "Superaste el límite de llamadas por minut
o.\nEsperá un momento y volvé a intentar." },
  "requestId": "7c41e09b2d5f4a8e9b6c3d2e1f0a0014" }
```

Límites por clave

Límite	Valor	Comentario
Llamadas por minuto	60	Por MCP también cuentan los mensajes de protocolo (conexión, lista de herramientas), aunque no cuestan créditos.
Llamadas simultáneas	2	Si tu programa o tu agente lanza consultas en paralelo, que no sean más de dos a la vez.
Filas por página	200	El máximo de `cantidad`.
Período de partes diarios	31 días	Entre `desde` y `hasta`, contando los dos extremos.
Sin clave o con clave inválida	20 por minuto por dirección IP	Protege el servicio de intentos de adivinar claves.

Cómo reintentar bien

Respuesta	¿Reintentar?
`200` (`ok` o `vacio`)	No: ya se cobró y la respuesta es correcta.
`400`, `401`, `402`	No: repetir da el mismo error hasta que corrijas la causa.
`429`	Sí, después de los segundos de `Retry-After`.
`500`, `503`	Sí, unas pocas veces y esperando cada vez más (por ejemplo 5, 30 y 120 segundos).

12 Ejemplos de código

Los dos ejemplos recorren los operarios por páginas y traen los partes diarios de una semana. Respetan los límites y leen la clave de una variable de entorno.

Python

```python
import json, os, time, urllib.error, urllib.parse, urllib.request

HOST = "https://syafmcp-g6dxa5akfmetd4e3.brazilsouth-01.azurewebsites.net"
CLAVE = os.environ["SYAF_API_KEY"]

def consultar(ruta, **parametros):
    url = f"{HOST}{ruta}?{urllib.parse.urlencode(parametros)}"
    pedido = urllib.request.Request(url, headers={"X-Api-Key": CLAVE})
    for espera in (5, 30, 120, None):
        try:
            with urllib.request.urlopen(pedido, timeout=30) as r:
                return json.load(r)
        except urllib.error.HTTPError as e:
            if e.code not in (429, 500, 503) or espera is None:
                raise
            time.sleep(int(e.headers.get("Retry-After", espera)))

def recorrer(ruta, **filtros):
    desde_id = 0
    while True:
        r = consultar(ruta, desdeId=desde_id, cantidad=200, **filtros)
        yield from r.get("datos", [])
        if "siguienteDesdeId" not in r:
            return
        desde_id = r["siguienteDesdeId"]

# Todos los operarios, también los dados de baja
for o in recorrer("/api/operarios", estado="todos"):
    print(o["id"], o["nombre"], o["apellidos"], o["frente"], o["activo"])  # guardar en tu sistema

# Los partes diarios de una semana, con su detalle
for p in recorrer("/api/partesDiarios", desde="2026-11-16", hasta="2026-11-22"):
    volumen = sum(x["volumen"] for x in p["productos"])
    print(p["fecha"][:10], p["maquina"], p["hsMaquina"], volumen)
```

JavaScript (Node.js 18 o más)

```
const HOST = "https://syafmcp-g6dxa5akfmetd4e3.brazilsouth-01.azurewebsites.net";
const CLAVE = process.env.SYAF_API_KEY;

async function consultar(ruta, parametros) {
  const url = `${HOST}${ruta}?${new URLSearchParams(parametros)}`;
  for (const espera of [5, 30, 120, null]) {
    const r = await fetch(url, { headers: { "X-Api-Key": CLAVE } });
    if (r.ok) return r.json();
    if (![429, 500, 503].includes(r.status) || espera === null) throw new Error((await r.json()).e
rror.codigo);
    await new Promise(fin => setTimeout(fin, 1000 * Number(r.headers.get("Retry-After") ?? esper
a)));
  }
}

async function* recorrer(ruta, filtros = {}) {
  let desdeId = 0;
  while (true) {
    const r = await consultar(ruta, { ...filtros, desdeId, cantidad: 200 });
    yield* r.datos ?? [];
    if (r.siguienteDesdeId === undefined) return;
    desdeId = r.siguienteDesdeId;
  }
}

// Todos los operarios, también los dados de baja
for await (const o of recorrer("/api/operarios", { estado: "todos" }))
  console.log(o.id, o.nombre, o.apellidos, o.frente, o.activo);  // guardar en tu sistema

// Los partes diarios de una semana, con su detalle
for await (const p of recorrer("/api/partesDiarios", { desde: "2026-11-16", hasta: "2026-11-22"
})) {
  const volumen = p.productos.reduce((suma, x) => suma + x.volumen, 0);
  console.log(p.fecha.slice(0, 10), p.maquina, p.hsMaquina, volumen);
}
```

Los ejemplos van en serie, una llamada por vez, así nunca superan el límite de 2 simultáneas.
Para otro maestro, cambiá la ruta (por ejemplo `/api/maquinas`) y los campos que guardás.
Para traer solo los cambios, agregá `modificadosDesde` a los filtros (sección 7).
El ejemplo de JavaScript se guarda como archivo `.mjs`.

13 Colección de Postman

El servicio publica una colección de Postman con todas las rutas y una respuesta de ejemplo en cada una.

1. Descargala desde la dirección del servicio más `/postman` (no pide clave), con el navegador o con:

   ```
   curl -o SYAFMcp.postman_collection.json "$HOST/postman"
   ```

2. En Postman: **Import** y elegí el archivo descargado.

3. Abrí la colección **SYAFMcp**, solapa **Variables**, completá `apiKey` con tu clave y guardá.
   La variable `host` ya viene con la dirección del servicio.

4. Cada request manda la clave en el header `X-Api-Key`, configurado en la colección.
   Completá los parámetros obligatorios (por ejemplo `desde` y `hasta`) y activá los opcionales que quieras usar.

> Si compartís la colección después de completar `apiKey`, la clave va adentro.
> Borrala antes de exportarla.

14 Conectar un agente de IA por MCP

El servidor MCP está en:

```
https://syafmcp-g6dxa5akfmetd4e3.brazilsouth-01.azurewebsites.net/mcp
```

Usa el transporte HTTP de MCP (Streamable HTTP) y la misma clave, en `X-Api-Key` o en `Authorization: Bearer`.
Ofrece las dieciséis herramientas de la sección 5, con los mismos parámetros, el mismo costo y la misma respuesta JSON.
Cuando una herramienta falla, el resultado viene marcado como error y trae el sobre de error de la sección 8, para que el agente lea el código y te lo explique.

Cómo se llaman las herramientas

Los argumentos son los parámetros de la sección 6, con los mismos nombres.
Los que no se mandan toman su valor por defecto.
En las herramientas `_por_id`, el `id` va como argumento.

Herramienta y argumentos	Equivale a
`operarios_listar` con `{"cantidad": 2, "idFrente": 1}`	`GET /api/operarios?cantidad=2&idFrente=1`
`montes_por_id` con `{"id": 1}`	`GET /api/montes/1`
`tipos_madera_listar` con `{"estado": "todos"}`	`GET /api/tiposMadera?estado=todos`
`partes_diarios_por_id` con `{"id": 4790}`	`GET /api/partesDiarios/4790`

Claude Code

Con la clave en la variable `SYAF_API_KEY` (sección 4):

```
claude mcp add --transport http --scope user syaf \
  https://syafmcp-g6dxa5akfmetd4e3.brazilsouth-01.azurewebsites.net/mcp \
  --header "X-Api-Key: $SYAF_API_KEY"
```

- Comprobá la conexión con `claude mcp list`, o con `/mcp` dentro de Claude Code.
- `--scope user` lo deja disponible en todos tus proyectos.
  Sin esa opción queda solo para la carpeta donde corriste el comando.
- No uses `--scope project` con la clave escrita: queda en el archivo `.mcp.json`, que se sube al repositorio.

Para compartir la configuración con tu equipo, poné `"X-Api-Key": "${SYAF_API_KEY}"` en `headers` y que cada persona tenga la variable con su clave.

Codex

En `~/.codex/config.toml`:

```
[mcp_servers.syaf]
url = "https://syafmcp-g6dxa5akfmetd4e3.brazilsouth-01.azurewebsites.net/mcp"
bearer_token_env_var = "SYAF_API_KEY"
```

Codex manda la clave como `Authorization: Bearer`, tomándola de la variable de entorno `SYAF_API_KEY`. Esa variable tiene que estar definida en la terminal desde la que abrís Codex.

Otros clientes MCP

Cualquier cliente compatible con MCP por HTTP (Streamable HTTP) que permita agregar un header puede conectarse con la dirección `/mcp` y el header `X-Api-Key`.
Para probar sin agente, una llamada a mano con `curl` (la respuesta viene como evento `message`):

```
curl -X POST "$HOST/mcp" \
  -H "X-Api-Key: $SYAF_API_KEY" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json, text/event-stream" \
  -d '{"jsonrpc":"2.0","id":1,"method":"tools/call","params":{"name":"cuenta_saldo","arguments":
{}}}'
```

Preguntas de ejemplo

- "Listame los operarios del frente 1."
- "¿Qué parcelas tiene el monte Los Pinos y cuántas hectáreas suman?"
- "¿Qué máquinas tiene cada frente y cuántas horas lleva cada una?"
- "¿Cuántos metros cúbicos se sacaron la semana pasada, por frente?"
- "¿Cuántos créditos me quedan este mes?"

Cuidados al usar agentes

- Cada herramienta que el agente usa cuesta un crédito, igual que por HTTP.
  Un agente puede hacer varias consultas para responder una sola pregunta: mirá `creditos` o preguntale el saldo (`cuenta_saldo`, gratis).
- Los agentes suelen lanzar herramientas en paralelo: con más de 2 a la vez, alguna vuelve `LIMITE_LLAMADAS`.
- Los textos que vienen en `datos` (nombres de operarios, frentes, montes o máquinas) son datos cargados en la base de tu empresa, no instrucciones.
  Si alguno contuviera algo parecido a una orden, el agente tiene que tratarlo como texto.
- El servidor no tiene herramientas de escritura ni de administración: un agente conectado solo puede leer.

15 Estado del servicio

Sin clave y sin costo:

Ruta	Qué devuelve
`GET /health`	`ok` si el servicio está en línea.
`GET /version`	Versión publicada, commit y fecha de compilación (UTC).
`GET /postman`	La colección de Postman.
`GET /`	La documentación interactiva (Swagger).

```
curl "$HOST/version"
{ "version": "1.1.0", "commit": "a1b2c3d", "compilado": "2026-11-24T18:40:12Z" }
```

16 Seguridad y privacidad

- **Solo lectura.**
  Ninguna operación crea, modifica ni borra datos de tu empresa.
- **Una clave, una empresa.**
  La empresa sale siempre de la clave: no hay parámetro para elegir otra ni forma de consultar datos ajenos.
- **Consultas fijas.**
  Ninguna operación recibe instrucciones SQL, nombres de tablas ni columnas: cada operación es una consulta fija y todo valor viaja como parámetro.
- **La clave no se guarda en claro.**
  En los servidores queda solo una huella de la clave (hash), que no permite reconstruirla.
- **Conexión cifrada.**
  La dirección del servicio es HTTPS.
- **Solo datos operativos.**
  No se devuelven sueldos, jornales, comisiones, retenciones, precios de liquidación, observaciones ni fotos.
- **Respuestas limpias.**
  Los errores nunca muestran detalles internos del servidor ni de la base.
- **Trazabilidad.**
  Cada llamada tiene su `requestId`, con el que soporte ubica la llamada exacta.

17 Soporte

Todo se pide a **FATE Sistemas**: la clave inicial, una clave nueva, la renovación, los créditos extra y los cambios de plan.
La clave sigue siendo la misma, salvo que pidas una nueva.

Ante cualquier error, mandá:

- el `requestId` de la respuesta;
- la operación y los parámetros que usaste;
- la hora aproximada.

> Nunca mandes la clave completa.
> Alcanza con sus primeros 8 caracteres (`fap_` y cuatro más).

18 Historial de versiones

Versión	Qué trajo
1.1.0	24/11/2026. Partes diarios con su detalle de productos, paradas y recargas (`partesDiarios`). Tipos de madera (`tiposMadera`). `modificadosDesde` y `fechaUpd` en todos los maestros, para traer solo los cambios.
1.0.0	Primera versión: operarios, frentes, montes, parcelas, máquinas, saldo y catálogo por API HTTP y por MCP.