KIOSKO DE ASISTENCIA

API para servidores

Especificación de la API **/kiosko/v1** que tu backend debe implementar para que las tablets marquen asistencia con tu marca, tus datos y tus reglas.



API v1 · app 0.19

Noviembre 2026

CONTENIDO

Índice

01 Visión general

02 Convenciones

03 Resumen de rutas

04 Marca: GET /info y GET /config

05 Sesión: POST /login y POST /refresh

06 Padrón: GET /padron

07 Marcar asistencia: POST /marcar

08 Voz: POST /voz (opcional)

09 Actualizaciones: GET /version (opcional)

10 Diagnóstico y demo (opcionales)

11 Seguridad y buenas prácticas

12 Servidor de referencia listo para usar

13 Lista de verificación

01

Visión general

La tablet no depende de ningún proveedor: habla solo con **tu servidor**, mediante una API pequeña y estable. Si tu backend implementa estas rutas, cualquier tablet con Kiosko de Asistencia funciona con tu colegio.



Cómo trabaja la tablet

1. **Configuración:** el auxiliar escribe la URL base de tu servidor. La tablet llama a `GET /kiosko/v1/info` para validarla y mostrar tu nombre, logo y color.
2. **Inicio de sesión:** `POST /login` con la cuenta de la tablet. Guarda el token cifrado.
3. **Padrón:** al iniciar y luego cada 15 minutos descarga `GET /padron` (alumnos y personal, con fotos). Con eso muestra nombre y foto **al instante**, sin esperar a la red.
4. **Marcar:** cada lectura se guarda primero en una cola local con la **hora exacta** y se envía con `POST /marcar`. Tu servidor decide si es entrada, tardanza, salida, etc. Sin internet, la cola espera y se envía después, en orden.
5. **Extras opcionales:** voz (`/voz`), actualizaciones remotas (`/version`), diagnóstico y reseteo para demostraciones.

> **Tú defines las reglas**
> La tablet no calcula tardanzas ni horarios: solo muestra y dice lo que tu servidor responde. Así puedes aplicar horarios por nivel, por aula, feriados o turnos sin tocar la app.

02

Convenciones

TEMA	REGLA
URL base	La que el auxiliar escribe en la tablet, por ejemplo `https://api.tucolegio.com`. La tablet agrega `/kiosko/v1` a cada ruta: `https://api.tucolegio.com/kiosko/v1/padron`.
HTTPS obligatorio	La app **no acepta HTTP sin cifrar**. Usa un certificado válido (Let's Encrypt es gratuito).
Formato	JSON en UTF-8. Cabecera `Content-Type: application/json`.
Autenticación	`Authorization: Bearer <token>` en todas las rutas salvo `/info`, `/login` y `/refresh`.
Token vencido	Responde `401`. La tablet llama una vez a `/refresh` y reintenta. Si también falla, pide iniciar sesión otra vez.
Errores	Códigos HTTP normales con cuerpo `{"mensaje": "…"}`. Un `4xx` en `/marcar` descarta esa marca (no se reintenta); un `5xx` o un corte de red la dejan en cola para reintentar.
Rutas opcionales	Si no implementas una ruta opcional, responde `404` o `501`: la tablet lo entiende y desactiva esa función.
Fechas	ISO 8601 en UTC, por ejemplo `2026-11-03T12:58:04.123Z`.
URLs de imágenes	Absolutas (`https://…`) o relativas a la raíz de tu dominio (`/fotos/123.jpg`).
Tiempos	La tablet espera hasta 8 s para conectar y 12 s para leer la respuesta. Responde `/marcar` en menos de 1 s para una buena experiencia.

03

Resumen de rutas

MÉTODO Y RUTA	AUTH	OBLIGATORIA	PARA QUÉ
GET `/info`	No	Sí	Validar el servidor y traer la marca
POST `/login`	No	Sí	Iniciar sesión
POST `/refresh`	No	Recomendada	Renovar el token sin pedir la clave
GET `/config`	Sí	Sí	Datos del colegio de la sesión y marca
GET `/padron`	Sí	Sí	Alumnos y personal
POST `/marcar`	Sí	Sí	Registrar una lectura
POST `/voz`	Sí	Opcional	Texto a voz (mp3)
GET `/version`	Sí	Opcional	Actualización remota del APK
POST `/diagnostico`	Sí	Opcional	Recibir eventos técnicos
POST `/reset-dia`	Sí	Opcional	Borrar las marcas de hoy (demostraciones)

04

Marca: /info y /config

GET `/kiosko/v1/info` — sin token

Lo llama la tablet cuando el auxiliar escribe la URL. Si responde `200` con este formato, la URL es válida.

RESPUESTA 200

```
{
  "nombre": "Colegio San Martín",          // se muestra en la cabecera y en el login
  "logo":   "https://api.tucolegio.com/logo.png", // PNG/JPG cuadrado, 512×512 recomendado
  "color":  "#2E5BBA",                     // color principal (#RRGGBB)
  "api":    1
}
```

GET `/kiosko/v1/config` — con token

Lo llama la tablet en cada sincronización. Permite que una misma instalación sirva a varios colegios (según el usuario que inició sesión) y que cambies la marca sin tocar la tablet.

RESPUESTA 200

```
{
  "colegio": 1,              // identificador del colegio de esta sesión
  "demo":    false,          // true: muestra «Resetear día» y el botón de Wi-Fi (demostraciones)
  "nombre":  "Colegio San Martín",
  "logo":    "https://api.tucolegio.com/logo.png",
  "color":   "#2E5BBA"
}
```

05

Sesión: /login y /refresh

POST `/kiosko/v1/login` — sin token

CUERPO

```
{
  "usuario": "auxiliar.puerta1",
  "clave":   "••••••"
}
```

RESPUESTA 200

```
{
  "token":         "eyJhbGciOi…",
  "refreshToken":  "c8f1…",
  "nombreUsuario": "Auxiliar Puerta 1"
}
```

Clave incorrecta: `401 {"mensaje": "Usuario o clave incorrectos"}`. La tablet **no guarda la contraseña**, solo los tokens (cifrados con el almacén de claves de Android).

POST `/kiosko/v1/refresh` — sin token

CUERPO

```
{ "refreshToken": "c8f1…" }
```

RESPUESTA 200

```
{ "token": "eyJ…", "refreshToken": "9a2e…" }
```

Si devuelves un `refreshToken` nuevo, la tablet lo reemplaza. Si no lo envías, conserva el anterior.

Cuentas para tablets

Crea una cuenta exclusiva para cada tablet o puerta, con permiso **solo de asistencia**. Las tablets están en lugares públicos; nunca uses la cuenta de un director o administrador.

06

Padrón: /padron

GET `/kiosko/v1/padron` con token

Todas las personas que pueden marcar en este colegio: alumnos **y** personal. Incluye a quienes aún no tienen tarjeta (el auxiliar los encuentra en la búsqueda manual).

RESPUESTA 200

```
{
  "personas": [
    {
      "id":        "27983",                // único dentro de su tipo (texto o
número)
      "tipo":      "alumno",               // "alumno" o "personal"
      "nombre":    "AGUILAR NÚÑEZ MARCO",  // como quieras mostrarlo
      "aula":      "Secundaria 5° A",      // para el personal: cargo o "Personal"
      "documento": "70358176",             // DNI u otro (búsqueda manual)
      "tarjeta":   "0003710427",           // código del lector, o null si no tiene
      "foto":      "https://api.tucolegio.com/fotos/27983.jpg"
    }
  ]
}
```

Recomendaciones

- **Fotos:** JPG cuadradas de 512 a 768 px, con la cara centrada (unos 40-70 KB). La tablet las descarga una vez y las guarda; si cambias una foto, cambia también su URL para que la tablet la vuelva a bajar.
- **Tamaño:** un colegio de 1 500 personas responde unos 300 KB. Comprime con gzip.
- **Tarjetas:** guarda el código tal como lo escribe el lector. La tablet busca automáticamente en decimal, sin ceros a la izquierda, en hexadecimal y con los bytes invertidos, y envía a `/marcar` el código del padrón.

07

Marcar asistencia: /marcar

POST `/kiosko/v1/marcar` — con token · idempotente

CABECERAS

```
Authorization: Bearer eyJ…
Idempotency-Key: 6f1c2a…  // UUID por lectura
```

CUERPO

```
{
  "tarjeta": "0003710427",
  "momento": "2026-11-03T12:58:04.123Z"
}
```

RESPUESTA 200

```
{
  "resultado": "TARDANZA",
  "nombre":    "AGUILAR NÚÑEZ MARCO",
  "nombreVoz": "Aguilar Marco",
  "foto":      "https://…/27983.jpg",
  "aula":      "Secundaria 5° A",
  "tipo":      "alumno",
  "mensaje":   "Entrada 07:58",
  "detalle":   null
}
```

Reglas que debes cumplir

- **Usa `momento`, no la hora del servidor.** Es la hora en que se pasó la tarjeta. Si la tablet estuvo sin internet, la marca llega minutos u horas después y la tardanza debe calcularse con `momento`. Rechaza (`400`) momentos del futuro o de más de 7 días atrás.
- **Idempotencia.** Si llega dos veces la misma `Idempotency-Key` (la tablet reintenta cuando se corta la red a mitad de la respuesta), devuelve la misma respuesta **sin registrar dos veces**. Guarda las claves al menos 24 horas.
- **Tarjeta desconocida:** responde `200` con `"resultado": "NO_REGISTRADA"`, no un `404`.
- **Personal:** el mismo endpoint sirve para trabajadores; responde `"tipo": "personal"`.

Valores de «resultado»

RESULTADO	CUÁNDO	LA TABLET MUESTRA · LA VOZ DICE
ENTRADA	Primera marca del día, a tiempo	Verde «Entrada» · «{nombreVoz} registró entrada»
TARDANZA	Primera marca, después de la hora límite	Ámbar «Tardanza» · «… registró tardanza»
SALIDA	Marca de salida	Azul «Salida» · «… registró salida»
YA_MARCO	Volvió a pasar la tarjeta poco después de entrar	Azul «Ya registró su entrada»

YA_SALIO	Ya había marcado salida	Azul «Ya registró su salida»
SIN_ENTRADA	Intenta salir sin haber entrado	Ámbar «Sin entrada previa»
SIN_HORARIO	Hoy no le corresponde asistir	Ámbar «Hoy no tiene horario»
NO_REGISTRADA	La tarjeta no es de nadie	Rojo «Tarjeta no registrada · Acércate al auxiliar»
INHABILITADA	La persona tiene la asistencia bloqueada	Rojo «Asistencia inhabilitada»
NO_HABILITADO	Colegio o periodo no habilitado	Rojo «Colegio no habilitado»
ERROR_PERSONAL	Otro motivo (explícalo en `detalle`)	Rojo «No se pudo registrar» + `detalle`

Campos de la respuesta

`nombre`, `foto` y `aula` se usan si la persona no estaba en el padrón local (por ejemplo, alguien agregado hace minutos). `nombreVoz` es lo que dice la voz: corto y natural, «Apellido Nombre». `mensaje` y `detalle` son textos libres para el registro y la pantalla.

08

Voz: /voz opcional

POST `/kiosko/v1/voz` — con token

CUERPO

```
{ "texto": "Aguilar Marco registró
entrada" }
```

RESPUESTA 200

```
{ "url": "https://…/voz/9f2a.mp3" }
```

Genera el audio con el servicio de texto a voz que prefieras (Google Cloud, Azure, Amazon Polly, OpenAI, ElevenLabs…) y devuelve la URL de un **MP3**. Guárdalo con el hash del texto para no generarlo dos veces.

- La tablet **guarda cada frase**: la segunda vez suena al instante y también sin internet.
- Al conectarse, la tablet pide frases fijas («Asistencia guardada», «Tarjeta no registrada», «Wi-Fi apagado»…) para tenerlas disponibles sin red.
- Si no la implementas (`404` / `501`), la tablet usa pitidos: uno corto al leer y otro grave para los errores.

09

Actualizaciones: /version opcional

GET `/kiosko/v1/version` — con token

```
{
  "versionCode": 40,                        // entero; la tablet actualiza si es
mayor que el instalado
  "versionName": "0.20.0",
  "apk":         "https://…/kiosko-0.20.0.apk",
  "sha256":      "8a7b792ac37a01e2…",       // del archivo APK; la tablet lo
verifica
  "notas":       "Mejoras de velocidad."
}
```

En Configuración la tablet muestra «Actualizar a 0.20.0». Al confirmar se descarga, se verifica el SHA-256 y se instala sola, sin salir del kiosko. Publica **siempre el APK oficial** que te entregamos: Android solo acepta actualizaciones firmadas con la misma firma que la versión instalada.

10

Diagnóstico y demostraciones opcionales

POST `/kiosko/v1/diagnostico` — con token

```
{
  "dispositivo": "70da1d640a071f5c",
  "version": "0.19.1",
  "eventos": [ { "t": "2026-11-03T20:07:26Z", "tipo": "lectura", "datos": { "codigo":
"0003710427", "encontrado": true } } ]
}
```

Eventos técnicos en lotes de hasta 400 (lecturas, lector conectado o desconectado, red, errores, rendimiento). Guárdalos para dar soporte remoto. Responde `200`.

POST `/kiosko/v1/reset-dia` — con token

Solo si `/config` devuelve `"demo": true`. Borra las marcas de hoy de ese colegio para repetir una demostración desde cero. En colegios reales responde `403`.

11

Seguridad y buenas prácticas

- **HTTPS** con certificado válido. Sin él la tablet no se conecta.
- Tokens con vencimiento (por ejemplo 12 h) y `refreshToken` de larga duración que puedas revocar.
- Limita los intentos de `/login` por IP y por usuario.
- Cada cuenta de tablet solo ve el padrón **de su colegio**. Filtra siempre por el colegio de la sesión.
- Valida `momento` (no futuro, no más de 7 días) y la `Idempotency-Key`.
- Las fotos de menores son datos personales: sirve las URLs solo a sesiones válidas o con nombres imposibles de adivinar, y cumple la ley de protección de datos de tu país.

12

Servidor de referencia listo para usar

Incluimos un servidor completo en un solo archivo, **sin dependencias** (Node.js 18 o superior). Puedes usarlo tal cual en un colegio pequeño o como guía para tu propio backend.

```
# 1. Descomprime servidor-ejemplo.zip y entra a la carpeta
ADMIN_CLAVE=una-clave-larga PORT=8080 node servidor.js

# 2. Abre el panel en el navegador
http://localhost:8080/admin
```

Qué trae el panel

PESTAÑA	PARA QUÉ
Marca	Nombre, logo, color, hora de tardanza, minutos entre entrada y salida, modo demostración.
Alumnos y personal	Pegar el padrón desde Excel (una línea por persona: `tipo;nombre;aula;documento;tarjeta;foto`).
Marcas del día	Ver entradas, salidas y tardanzas por fecha y descargarlas en CSV.
Usuarios de la tablet	Crear cuentas o cambiar claves.
Actualizar tablets	Subir un APK nuevo y publicarlo; las tablets lo instalan desde Configuración.

Publicarlo en internet con HTTPS

La forma más simple es **Caddy**, que obtiene el certificado automáticamente:

```
# Caddyfile
api.tucolegio.com {
    reverse_proxy 127.0.0.1:8080
}
```

Con nginx, usa `proxy_pass http://127.0.0.1:8080;`, `proxy_set_header X-Forwarded-Proto $scheme;` y un certificado de Let's Encrypt (certbot). Mantén el proceso activo con `pm2` o `systemd`.

> **Demostración en línea**
>
> Hay un servidor de demostración en `https://kiosko.kole.pe` (usuario `demo`, clave `demo1234`) para probar una tablet sin instalar nada.

13

Lista de verificación

Prueba tu implementación con `curl` antes de conectar una tablet:

```
B=https://api.tucolegio.com/kiosko/v1
curl $B/info
T=$(curl -s -X POST $B/login -H 'Content-Type: application/json' \
      -d '{"usuario":"tablet1","clave":"secreta"}' | jq -r .token)
curl $B/config -H "Authorization: Bearer $T"
curl $B/padron -H "Authorization: Bearer $T" | jq '.personas | length'
curl -X POST $B/marcar -H "Authorization: Bearer $T" -H 'Content-Type: application/json' \
     -H 'Idempotency-Key: prueba-1' -d '{"tarjeta":"0003710427","momento":"2026-11-
03T12:00:00Z"}'
# Repite la última línea: debe devolver lo mismo y NO registrar dos veces
```

- ☐ `/info` responde sin token y con `nombre`, `logo` y `color`.
- ☐ `/login` con clave errada devuelve `401`.
- ☐ `/padron` incluye alumnos y personal, con `tipo` correcto.
- ☐ `/marcar` con una tarjeta desconocida devuelve `200` y `NO_REGISTRADA`.
- ☐ La tardanza se calcula con `momento`: una marca de las 07:58 enviada a las 09:00 es puntual.
- ☐ La misma `Idempotency-Key` no duplica la marca.
- ☐ Un token vencido devuelve `401` y `/refresh` entrega uno nuevo.
- ☐ Todo responde por HTTPS con un certificado válido.