

Documentación del Backend: Hospital Tracker

Descripción General

Este backend provee servicios REST para la gestión y monitoreo de hospitales, empleados y reportes municipales/estatales. Está construido con **Node.js** y **Express**.

Tecnologías Utilizadas

- Node.js
- Express
- dotenv
- cors
- cookie-parser

Instalación

1. Clona el repositorio.
2. Instala las dependencias:

```
npm install
```
3. Crea un archivo `.env` con tus variables de entorno necesarias (por ejemplo, PORT, credenciales de base de datos, etc).

Configuración de CORS

Solo se permiten solicitudes desde los siguientes orígenes: - `http://localhost:5173` (desarrollo) - `https://geoapphospital-b0yr.onrender.com` (backend producción) - `https://mystifying-carson.192-99-212-154.plesk.page` (frontend temporal)

Si la solicitud proviene de otro origen, será rechazada.

Estructura de Rutas Principales

Ruta base	Descripción
<code>/api/superadmin</code>	Funciones para superadministrador
<code>/api/auth</code>	Autenticación y login

Ruta base	Descripción
/api/estadoadmin	Funciones para administrador estatal
/api/municipioadmin	Funciones para administrador municipal
/api/hospitaladmin	Funciones para administrador de hospital
/api/groups	Gestión de grupos
/api/employees	Gestión de empleados
/api/mobile	Endpoints para app móvil
/api/email	Envío de correos electrónicos
/api/reportes	Generación de reportes
/api/dashboards	Datos para dashboards

Ejecución

Para iniciar el servidor:

```
npm start
```

o

```
node server.js
```

El servidor correrá en el puerto definido en `.env` o en el 4000 por defecto.

Notas

- El backend utiliza middlewares para parseo de JSON, manejo de cookies y control de CORS.
- Cada grupo de rutas está modularizado en la carpeta `/routes`.

Endpoints `/api/employees`

Endpoints GET

- **`/api/employees/grupos-by-hospital?id_hospital=...`**
Devuelve la lista de grupos (`id_group`, `nombre_grupo`) asociados a un hospital.
- **`/api/employees/get-empleados`**
Devuelve todos los empleados con sus datos completos (nombre, CURP, hospital, grupo, rol, etc.).
- **`/api/employees/get-empleados-by-groups`**
Devuelve todos los grupos con la lista de empleados asignados a cada uno.

- **/api/employees/monitoreo**
Devuelve la última ubicación registrada por cada usuario (latitud, longitud, fecha_hora, dentro_geocerca, etc.).

Endpoints POST

- **/api/employees/create-empleado**
Crea un nuevo empleado usando IDs.

Body esperado:

```
{
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "ap_materno": "López",
  "CURP": "CURP123",
  "correo_electronico": "juan@ejemplo.com",
  "telefono": "1234567890",
  "user": "juanp",
  "pass": "123456",
  "role_name": "empleado",
  "id_estado": 1,
  "id_municipio": 2,
  "id_hospital": 3,
  "id_grupo": 4
}
```

- **/api/employees/delete-employee/:id_user**
Elimina completamente a un empleado de todas las tablas relacionadas.

Parámetro (URL): id_user

- **/api/employees/create-empleado-nombres**
Crea un empleado usando los nombres de estado, municipio, hospital y grupo.

Body esperado:

```
{
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "ap_materno": "López",
  "CURP": "CURP123",
  "correo_electronico": "juan@ejemplo.com",
  "telefono": "1234567890",
  "user": "juanp",
  "pass": "123456",
  "role_name": "empleado",
  "estado": "Nuevo León",
  "municipio": "Monterrey",
}
```

```
"hospital": "Hospital General",
"grupo": "Grupo A"
}
```

Endpoints PUT

- **/api/employees/update-employee**
Actualiza la información de un empleado.

Body esperado:

```
{
  "id_user": 2,
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "ap_materno": "López",
  "curp_user": "CURP123",
  "id_estado": 1,
  "id_municipio": 2,
  "id_hospital": 3,
  "id_group": 4
}
```

Endpoints POST

- **/api/email/send-credentials**
Envía por correo electrónico (usando Resend) las credenciales de acceso de un empleado con un correo HTML personalizado.

Body esperado:

```
{
  "correo_electronico": "empleado@ejemplo.com",
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "user": "usuario123",
  "pass": "contraseña123"
}
```

Respuesta exitosa (200):

```
{
  "success": true,
  "data": { /* respuesta de Resend */ }
}
```

Errores posibles:

- 400 si no se envía correo_electronico.

- 500 si ocurre algún error al enviar el correo.

Endpoints GET

- **/api/dashboards/estatal/metricas**
Devuelve métricas generales del estado (empleados, grupos, hospitales, etc.).

Query params: id_estado, fecha_inicio, fecha_fin

Ejemplo de respuesta:

```
{  
  "total_empleados": 120,  
  "total_grupos": 15,  
  "total_hospitales": 8  
}
```

- **/api/dashboards/estatal/entradas-salidas**
Devuelve datos de entradas y salidas de empleados en el estado.
- **/api/dashboards/estatal/eventos-geocerca**
Devuelve eventos de geocerca registrados en el estado.
- **/api/dashboards/estatal/ranking-hospitales**
Devuelve un ranking de hospitales del estado según actividad o métricas.
- **/api/dashboards/estatal/horas-municipio**
Devuelve el total de horas trabajadas por municipio en el estado.
- **/api/dashboards/estatal/distribucion-municipal**
Devuelve la distribución de empleados o actividad por municipio.
- **/api/dashboards/estatal/distribucion-municipal-completa**
Devuelve la distribución completa de actividad por municipio.
- **/api/dashboards/estatal/municipio-detalle**
Devuelve el detalle de un municipio específico del estado.
- **/api/dashboards/municipios-by-estado/:id_estado**
Devuelve la lista de municipios de un estado.

Ejemplo de respuesta:

```
[  
  { "id_municipio": 1, "nombre_municipio": "Municipio A" },  
  { "id_municipio": 2, "nombre_municipio": "Municipio B" }  
]
```

- **/api/dashboards/nacional/estadisticas-estados**
Devuelve estadísticas generales de todos los estados.
- **/api/dashboards/nacional/totales**
Devuelve totales nacionales (empleados, hospitales, etc.).

Ejemplo de respuesta:

```
{
  "total_empleados": 2000,
  "total_hospitales": 150,
  "total_grupos": 300
}
```

- **/api/dashboards/nacional/ranking-estados**
Devuelve un ranking nacional de estados según actividad o métricas.

Endpoints POST

- **/api/dashboards/grupo**
Devuelve métricas y datos de un grupo específico para dashboards.

Body esperado:

```
{
  "id_grupo": 1,
  "fecha_inicio": "2024-01-01",
  "fecha_fin": "2024-01-31"
}
```

- **/api/dashboards/municipio**
Devuelve métricas y datos de un municipio específico para dashboards.

Body esperado:

```
{
  "id_municipio": 1,
  "fecha_inicio": "2024-01-01",
  "fecha_fin": "2024-01-31"
}
```

Endpoints GET

- **/api/estadoadmin/hospitals-by-user/:id_user**
Devuelve los hospitales del estado asignado al usuario.
- **/api/estadoadmin/hospitals-name-by-user/:id_user**
Devuelve solo los nombres de hospitales del estado asignado al usuario.

- **/api/estadoadmin/hospitaladmin**
Devuelve la lista de administradores de hospital del estado.
- **/api/estadoadmin/groups-by-user/:id_user**
Devuelve los grupos y municipios del estado asignado al usuario.
- **/api/estadoadmin/employees-by-user/:id_user**
Devuelve los empleados y municipios del estado asignado al usuario.
- **/api/estadoadmin/stats-by-user/:id_user**
Devuelve totales de hospitales, grupos y empleados del estado asignado al usuario.

Endpoints POST

- **/api/estadoadmin/create-hospitaladmin**
Crea un nuevo administrador de hospital.

Body:

```
{
  "nombre": "Pedro",
  "ap_paterno": "Gómez",
  "ap_materno": "Ruiz",
  "CURP": "CURP456",
  "user": "pedrog",
  "pass": "123456",
  "role_name": "hospitaladmin",
  "hospital": "Hospital General",
  "id_estado": 1
}
```

Endpoints /api/estadoadmin

Endpoints GET

- **/api/estadoadmin/grupos-by-hospital?id_hospital=...**
Devuelve la lista de grupos (id_group, nombre_grupo) asociados a un hospital.
- **/api/estadoadmin/get-empleados**
Devuelve todos los empleados con sus datos completos (nombre, CURP, hospital, grupo, rol, etc.).
- **/api/estadoadmin/get-empleados-by-groups**
Devuelve todos los grupos con la lista de empleados asignados a cada uno.

- **/api/estadoadmin/monitoreo**
Devuelve la última ubicación registrada por cada usuario (latitud, longitud, fecha_hora, dentro_geocerca, etc.).

Endpoints POST

- **/api/estadoadmin/create-empleado**
Crea un nuevo empleado usando IDs.

Body esperado:

```
{
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "ap_materno": "López",
  "CURP": "CURP123",
  "correo_electronico": "juan@ejemplo.com",
  "telefono": "1234567890",
  "user": "juanp",
  "pass": "123456",
  "role_name": "empleado",
  "id_estado": 1,
  "id_municipio": 2,
  "id_hospital": 3,
  "id_grupo": 4
}
```

Inserta al nuevo empleado en user_data, user_credentials, user_roles y group_users.

- **/api/estadoadmin/delete-employee/:id_user**
Elimina completamente a un empleado de todas las tablas relacionadas: registro_ubicaciones, group_users, user_credentials, user_roles, y user_data.

Parámetro (URL): id_user

- **/api/estadoadmin/create-empleado-nombres**
Crea un empleado usando los nombres de estado, municipio, hospital y grupo.

Body esperado:

```
{
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "ap_materno": "López",
  "CURP": "CURP123",
  "correo_electronico": "juan@ejemplo.com",
  "telefono": "1234567890",
  "user": "juanp",
}
```



```
"pass": "123456",  
"role_name": "empleado",  
"estado": "Nuevo León",  
"municipio": "Monterrey",  
"hospital": "Hospital General",  
"grupo": "Grupo A"  
}
```

El endpoint se encarga de buscar los IDs correspondientes y luego registrar al empleado igual que /create-empleado.

Endpoints PUT

- **/api/estadoadmin/update-employee**
Actualiza la información de un empleado.

Body esperado:

```
{  
  "id_user": 2,  
  "nombre": "Juan",  
  "ap_paterno": "Pérez",  
  "ap_materno": "López",  
  "curp_user": "CURP123",  
  "id_estado": 1,  
  "id_municipio": 2,  
  "id_hospital": 3,  
  "id_group": 4  
}
```

Actualiza la información de un empleado en user_data y también actualiza su relación en group_users.

Endpoints POST

- **/api/email/send-credentials**
Envía por correo electrónico (usando Resend) las credenciales de acceso de un empleado con un correo HTML personalizado.

Body esperado:

```
{  
  "correo_electronico": "empleado@ejemplo.com",  
  "nombre": "Juan",  
  "ap_paterno": "Pérez",  
  "user": "usuario123",  
  "pass": "contraseña123"  
}
```

Respuesta exitosa (200):

```
{
  "success": true,
  "data": { /* respuesta de Resend */ }
}
```

Errores posibles:

- 400 si no se envía correo_electronico.
- 500 si ocurre algún error al enviar el correo.

Endpoints GET

- **/api/dashboards/estatal/metricas**
Devuelve métricas generales del estado (empleados, grupos, hospitales, etc.).

Query params: id_estado, fecha_inicio, fecha_fin

Ejemplo de respuesta:

```
{
  "total_empleados": 120,
  "total_grupos": 15,
  "total_hospitales": 8
}
```

- **/api/dashboards/estatal/entradas-salidas**
Devuelve datos de entradas y salidas de empleados en el estado.
- **/api/dashboards/estatal/eventos-geocerca**
Devuelve eventos de geocerca registrados en el estado.
- **/api/dashboards/estatal/ranking-hospitales**
Devuelve un ranking de hospitales del estado según actividad o métricas.
- **/api/dashboards/estatal/horas-municipio**
Devuelve el total de horas trabajadas por municipio en el estado.
- **/api/dashboards/estatal/distribucion-municipal**
Devuelve la distribución de empleados o actividad por municipio.
- **/api/dashboards/estatal/distribucion-municipal-completa**
Devuelve la distribución completa de actividad por municipio.
- **/api/dashboards/estatal/municipio-detalle**
Devuelve el detalle de un municipio específico del estado.

- **/api/dashboards/municipios-by-estado/:id_estado**
Devuelve la lista de municipios de un estado.

Ejemplo de respuesta:

```
[  
  { "id_municipio": 1, "nombre_municipio": "Municipio A" },  
  { "id_municipio": 2, "nombre_municipio": "Municipio B" }  
]
```

- **/api/dashboards/nacional/estadisticas-estados**
Devuelve estadísticas generales de todos los estados.
- **/api/dashboards/nacional/totales**
Devuelve totales nacionales (empleados, hospitales, etc.).

Ejemplo de respuesta:

```
{  
  "total_empleados": 2000,  
  "total_hospitales": 150,  
  "total_grupos": 300  
}
```

- **/api/dashboards/nacional/ranking-estados**
Devuelve un ranking nacional de estados según actividad o métricas.

Endpoints POST

- **/api/dashboards/grupo**
Devuelve métricas y datos de un grupo específico para dashboards.

Body esperado:

```
{  
  "id_grupo": 1,  
  "fecha_inicio": "2024-01-01",  
  "fecha_fin": "2024-01-31"  
}
```

- **/api/dashboards/municipio**
Devuelve métricas y datos de un municipio específico para dashboards.

Body esperado:

```
{  
  "id_municipio": 1,  
  "fecha_inicio": "2024-01-01",  
  "fecha_fin": "2024-01-31"  
}
```

Endpoints GET

- **/api/estadoadmin/hospitals-by-user/:id_user**
Devuelve los hospitales del estado asignado al usuario.
- **/api/estadoadmin/hospitals-name-by-user/:id_user**
Devuelve solo los nombres de hospitales del estado asignado al usuario.
- **/api/estadoadmin/hospitaladmin**
Devuelve la lista de administradores de hospital del estado.
- **/api/estadoadmin/groups-by-user/:id_user**
Devuelve los grupos y municipios del estado asignado al usuario.
- **/api/estadoadmin/employees-by-user/:id_user**
Devuelve los empleados y municipios del estado asignado al usuario.
- **/api/estadoadmin/stats-by-user/:id_user**
Devuelve totales de hospitales, grupos y empleados del estado asignado al usuario.

Endpoints POST

- **/api/estadoadmin/create-hospitaladmin**
Crea un nuevo administrador de hospital.

Body:

```
{
  "nombre": "Pedro",
  "ap_paterno": "Gómez",
  "ap_materno": "Ruiz",
  "CURP": "CURP456",
  "user": "pedrog",
  "pass": "123456",
  "role_name": "hospitaladmin",
  "hospital": "Hospital General",
  "id_estado": 1
}
```

Endpoints /api/estadoadmin

Endpoints GET

- **/api/estadoadmin/grupos-by-hospital?id_hospital=...**
Devuelve la lista de grupos (id_group, nombre_grupo) asociados a un hospital.

- **/api/estadoadmin/get-empleados**
Devuelve todos los empleados con sus datos completos (nombre, CURP, hospital, grupo, rol, etc.).
- **/api/estadoadmin/get-empleados-by-groups**
Devuelve todos los grupos con la lista de empleados asignados a cada uno.
- **/api/estadoadmin/monitoreo**
Devuelve la última ubicación registrada por cada usuario (latitud, longitud, fecha_hora, dentro_geocerca, etc.).

Endpoints POST

- **/api/estadoadmin/create-empleado**
Crea un nuevo empleado usando IDs.

Body esperado:

```
{
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "ap_materno": "López",
  "CURP": "CURP123",
  "correo_electronico": "juan@ejemplo.com",
  "telefono": "1234567890",
  "user": "juanp",
  "pass": "123456",
  "role_name": "empleado",
  "id_estado": 1,
  "id_municipio": 2,
  "id_hospital": 3,
  "id_grupo": 4
}
```

Inserta al nuevo empleado en user_data, user_credentials, user_rols y group_users.

- **/api/estadoadmin/delete-employee/:id_user**
Elimina completamente a un empleado de todas las tablas relacionadas: registro_ubicaciones, group_users, user_credentials, user_rols, y user_data.

Parámetro (URL): id_user

- **/api/estadoadmin/create-empleado-nombres**
Crea un empleado usando los nombres de estado, municipio, hospital y grupo.

Body esperado:

```
{
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "ap_materno": "López",
  "CURP": "CURP123",
  "correo_electronico": "juan@ejemplo.com",
  "telefono": "1234567890",
  "user": "juanp",
  "pass": "123456",
  "role_name": "empleado",
  "estado": "Nuevo León",
  "municipio": "Monterrey",
  "hospital": "Hospital General",
  "grupo": "Grupo A"
}
```

El endpoint se encarga de buscar los IDs correspondientes y luego registrar al empleado igual que /create-empleado.

Endpoints PUT

- **/api/estadoadmin/update-employee**
Actualiza la información de un empleado.

Body esperado:

```
{
  "id_user": 2,
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "ap_materno": "López",
  "curp_user": "CURP123",
  "id_estado": 1,
  "id_municipio": 2,
  "id_hospital": 3,
  "id_group": 4
}
```

Actualiza la información de un empleado en user_data y también actualiza su relación en group_users.

Endpoints POST

- **/api/email/send-credentials**
Envía por correo electrónico (usando Resend) las credenciales de acceso de un empleado con un correo HTML personalizado.

Body esperado:

```
{
  "correo_electronico": "empleado@ejemplo.com",
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "user": "usuario123",
  "pass": "contraseña123"
}
```

Respuesta exitosa (200):

```
{
  "success": true,
  "data": { /* respuesta de Resend */ }
}
```

Errores posibles:

- 400 si no se envía correo_electronico.
- 500 si ocurre algún error al enviar el correo.

Endpoints GET

- **/api/dashboards/estatal/metricas**
Devuelve métricas generales del estado (empleados, grupos, hospitales, etc.).

Query params: id_estado, fecha_inicio, fecha_fin

Ejemplo de respuesta:

```
{
  "total_empleados": 120,
  "total_grupos": 15,
  "total_hospitales": 8
}
```

- **/api/dashboards/estatal/entradas-salidas**
Devuelve datos de entradas y salidas de empleados en el estado.
- **/api/dashboards/estatal/eventos-geocerca**
Devuelve eventos de geocerca registrados en el estado.
- **/api/dashboards/estatal/ranking-hospitales**
Devuelve un ranking de hospitales del estado según actividad o métricas.
- **/api/dashboards/estatal/horas-municipio**
Devuelve el total de horas trabajadas por municipio en el estado.
- **/api/dashboards/estatal/distribucion-municipal**
Devuelve la distribución de empleados o actividad por municipio.

- **/api/dashboards/estatal/distribucion-municipal-completa**
Devuelve la distribución completa de actividad por municipio.
- **/api/dashboards/estatal/municipio-detalle**
Devuelve el detalle de un municipio específico del estado.
- **/api/dashboards/municipios-by-estado/:id_estado**
Devuelve la lista de municipios de un estado.

Ejemplo de respuesta:

```
[
  { "id_municipio": 1, "nombre_municipio": "Municipio A" },
  { "id_municipio": 2, "nombre_municipio": "Municipio B" }
]
```

- **/api/dashboards/nacional/estadisticas-estados**
Devuelve estadísticas generales de todos los estados.
- **/api/dashboards/nacional/totales**
Devuelve totales nacionales (empleados, hospitales, etc.).

Ejemplo de respuesta:

```
{
  "total_empleados": 2000,
  "total_hospitales": 150,
  "total_grupos": 300
}
```

- **/api/dashboards/nacional/ranking-estados**
Devuelve un ranking nacional de estados según actividad o métricas.

Endpoints POST

- **/api/dashboards/grupo**
Devuelve métricas y datos de un grupo específico para dashboards.

Body esperado:

```
{
  "id_grupo": 1,
  "fecha_inicio": "2024-01-01",
  "fecha_fin": "2024-01-31"
}
```

- **/api/dashboards/municipio**
Devuelve métricas y datos de un municipio específico para dashboards.

Body esperado:


```
{
  "id_municipio": 1,
  "fecha_inicio": "2024-01-01",
  "fecha_fin": "2024-01-31"
}
```

Endpoints GET

- **/api/estadoadmin/hospitals-by-user/:id_user**
Devuelve los hospitales del estado asignado al usuario.
- **/api/estadoadmin/hospitals-name-by-user/:id_user**
Devuelve solo los nombres de hospitales del estado asignado al usuario.
- **/api/estadoadmin/hospitaladmin**
Devuelve la lista de administradores de hospital del estado.
- **/api/estadoadmin/groups-by-user/:id_user**
Devuelve los grupos y municipios del estado asignado al usuario.
- **/api/estadoadmin/employees-by-user/:id_user**
Devuelve los empleados y municipios del estado asignado al usuario.
- **/api/estadoadmin/stats-by-user/:id_user**
Devuelve totales de hospitales, grupos y empleados del estado asignado al usuario.

Endpoints POST

- **/api/estadoadmin/create-hospitaladmin**
Crea un nuevo administrador de hospital.

Body:

```
{
  "nombre": "Pedro",
  "ap_paterno": "Gómez",
  "ap_materno": "Ruiz",
  "CURP": "CURP456",
  "user": "pedrog",
  "pass": "123456",
  "role_name": "hospitaladmin",
  "hospital": "Hospital General",
  "id_estado": 1
}
```

Endpoints `/api/estadoadmin`

Endpoints GET

- **`/api/estadoadmin/grupos-by-hospital?id_hospital=...`**
Devuelve la lista de grupos (`id_group`, `nombre_grupo`) asociados a un hospital.
- **`/api/estadoadmin/get-empleados`**
Devuelve todos los empleados con sus datos completos (`nombre`, `CURP`, `hospital`, `grupo`, `rol`, etc.).
- **`/api/estadoadmin/get-empleados-by-groups`**
Devuelve todos los grupos con la lista de empleados asignados a cada uno.
- **`/api/estadoadmin/monitoreo`**
Devuelve la última ubicación registrada por cada usuario (`latitud`, `longitud`, `fecha_hora`, `dentro_geocerca`, etc.).

Endpoints POST

- **`/api/estadoadmin/create-empleado`**
Crea un nuevo empleado usando IDs.

Body esperado:

```
{
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "ap_materno": "López",
  "CURP": "CURP123",
  "correo_electronico": "juan@ejemplo.com",
  "telefono": "1234567890",
  "user": "juanp",
  "pass": "123456",
  "role_name": "empleado",
  "id_estado": 1,
  "id_municipio": 2,
  "id_hospital": 3,
  "id_grupo": 4
}
```

Inserta al nuevo empleado en `user_data`, `user_credentials`, `user_roles` y `group_users`.

- **`/api/estadoadmin/delete-employee/:id_user`**
Elimina completamente a un empleado de todas las tablas relacionadas: `registro_ubicaciones`, `group_users`, `user_credentials`, `user_roles`, y `user_data`.

Parámetro (URL): `id_user`

- **/api/estadoadmin/create-empleado-nombres**
Crea un empleado usando los nombres de estado, municipio, hospital y grupo.

Body esperado:

```
{
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "ap_materno": "López",
  "CURP": "CURP123",
  "correo_electronico": "juan@ejemplo.com",
  "telefono": "1234567890",
  "user": "juanp",
  "pass": "123456",
  "role_name": "empleado",
  "estado": "Nuevo León",
  "municipio": "Monterrey",
  "hospital": "Hospital General",
  "grupo": "Grupo A"
}
```

El endpoint se encarga de buscar los IDs correspondientes y luego registrar al empleado igual que /create-empleado.

Endpoints PUT

- **/api/estadoadmin/update-employee**
Actualiza la información de un empleado.

Body esperado:

```
{
  "id_user": 2,
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "ap_materno": "López",
  "curp_user": "CURP123",
  "id_estado": 1,
  "id_municipio": 2,
  "id_hospital": 3,
  "id_group": 4
}
```

Actualiza la información de un empleado en user_data y también actualiza su relación en group_users.

Endpoints POST

- **/api/email/send-credentials**

Envía por correo electrónico (usando Resend) las credenciales de acceso de un empleado con un correo HTML personalizado.

Body esperado:

```
{
  "correo_electronico": "empleado@ejemplo.com",
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "user": "usuario123",
  "pass": "contraseña123"
}
```

Respuesta exitosa (200):

```
{
  "success": true,
  "data": { /* respuesta de Resend */ }
}
```

Errores posibles:

- 400 si no se envía correo_electronico.
- 500 si ocurre algún error al enviar el correo.

Endpoints GET

- **/api/dashboards/estatal/metricas**

Devuelve métricas generales del estado (empleados, grupos, hospitales, etc.).

Query params: id_estado, fecha_inicio, fecha_fin

Ejemplo de respuesta:

```
{
  "total_empleados": 120,
  "total_grupos": 15,
  "total_hospitales": 8
}
```

- **/api/dashboards/estatal/entradas-salidas**

Devuelve datos de entradas y salidas de empleados en el estado.

- **/api/dashboards/estatal/eventos-geocerca**

Devuelve eventos de geocerca registrados en el estado.

- **/api/dashboards/estatal/ranking-hospitales**
Devuelve un ranking de hospitales del estado según actividad o métricas.
- **/api/dashboards/estatal/horas-municipio**
Devuelve el total de horas trabajadas por municipio en el estado.
- **/api/dashboards/estatal/distribucion-municipal**
Devuelve la distribución de empleados o actividad por municipio.
- **/api/dashboards/estatal/distribucion-municipal-completa**
Devuelve la distribución completa de actividad por municipio.
- **/api/dashboards/estatal/municipio-detalle**
Devuelve el detalle de un municipio específico del estado.
- **/api/dashboards/municipios-by-estado/:id_estado**
Devuelve la lista de municipios de un estado.

Ejemplo de respuesta:

```
[
  { "id_municipio": 1, "nombre_municipio": "Municipio A" },
  { "id_municipio": 2, "nombre_municipio": "Municipio B" }
]
```

- **/api/dashboards/nacional/estadisticas-estados**
Devuelve estadísticas generales de todos los estados.
- **/api/dashboards/nacional/totales**
Devuelve totales nacionales (empleados, hospitales, etc.).

Ejemplo de respuesta:

```
{
  "total_empleados": 2000,
  "total_hospitales": 150,
  "total_grupos": 300
}
```

- **/api/dashboards/nacional/ranking-estados**
Devuelve un ranking nacional de estados según actividad o métricas.

Endpoints POST

- **/api/dashboards/grupo**
Devuelve métricas y datos de un grupo específico para dashboards.

Body esperado:

```
{
  "id_grupo": 1,
  "fecha_inicio": "2024-01-01",
  "fecha_fin": "2024-01-31"
}
```

- **/api/dashboards/municipio**
Devuelve métricas y datos de un municipio específico para dashboards.

Body esperado:

```
{
  "id_municipio": 1,
  "fecha_inicio": "2024-01-01",
  "fecha_fin": "2024-01-31"
}
```

Endpoints GET

- **/api/estadoadmin/hospitals-by-user/:id_user**
Devuelve los hospitales del estado asignado al usuario.
- **/api/estadoadmin/hospitals-name-by-user/:id_user**
Devuelve solo los nombres de hospitales del estado asignado al usuario.
- **/api/estadoadmin/hospitaladmin**
Devuelve la lista de administradores de hospital del estado.
- **/api/estadoadmin/groups-by-user/:id_user**
Devuelve los grupos y municipios del estado asignado al usuario.
- **/api/estadoadmin/employees-by-user/:id_user**
Devuelve los empleados y municipios del estado asignado al usuario.
- **/api/estadoadmin/stats-by-user/:id_user**
Devuelve totales de hospitales, grupos y empleados del estado asignado al usuario.

Endpoints POST

- **/api/estadoadmin/create-hospitaladmin**
Crea un nuevo administrador de hospital.

Body:

```
{
  "nombre": "Pedro",
  "ap_paterno": "Gómez",
}
```

```
"ap_materno": "Ruiz",
"CURP": "CURP456",
"user": "pedrog",
"pass": "123456",
"role_name": "hospitaladmin",
"hospital": "Hospital General",
"id_estado": 1
}
```

Endpoints /api/estadoadmin

Endpoints GET

- **/api/estadoadmin/grupos-by-hospital?id_hospital=...**
Devuelve la lista de grupos (id_group, nombre_grupo) asociados a un hospital.
- **/api/estadoadmin/get-empleados**
Devuelve todos los empleados con sus datos completos (nombre, CURP, hospital, grupo, rol, etc.).
- **/api/estadoadmin/get-empleados-by-groups**
Devuelve todos los grupos con la lista de empleados asignados a cada uno.
- **/api/estadoadmin/monitoreo**
Devuelve la última ubicación registrada por cada usuario (latitud, longitud, fecha_hora, dentro_geocerca, etc.).

Endpoints POST

- **/api/estadoadmin/create-empleado**
Crea un nuevo empleado usando IDs.

Body esperado:

```
{
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "ap_materno": "López",
  "CURP": "CURP123",
  "correo_electronico": "juan@ejemplo.com",
  "telefono": "1234567890",
  "user": "juanp",
  "pass": "123456",
  "role_name": "empleado",
  "id_estado": 1,
  "id_municipio": 2,
  "id_hospital": 3,
}
```

```
"id_grupo": 4
}
```

Inserta al nuevo empleado en user_data, user_credentials, user_roles y group_users.

- **/api/estadoadmin/delete-employee/:id_user**
Elimina completamente a un empleado de todas las tablas relacionadas: registro_ubicaciones, group_users, user_credentials, user_roles, y user_data.

Parámetro (URL): id_user

- **/api/estadoadmin/create-empleado-nombres**
Crea un empleado usando los nombres de estado, municipio, hospital y grupo.

Body esperado:

```
{
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "ap_materno": "López",
  "CURP": "CURP123",
  "correo_electronico": "juan@ejemplo.com",
  "telefono": "1234567890",
  "user": "juanp",
  "pass": "123456",
  "role_name": "empleado",
  "estado": "Nuevo León",
  "municipio": "Monterrey",
  "hospital": "Hospital General",
  "grupo": "Grupo A"
}
```

El endpoint se encarga de buscar los IDs correspondientes y luego registrar al empleado igual que /create-empleado.

Endpoints PUT

- **/api/estadoadmin/update-employee**
Actualiza la información de un empleado.

Body esperado:

```
{
  "id_user": 2,
  "nombre": "Juan",
  "ap_paterno": "Pérez",
  "ap_materno": "López",
  "curp_user": "CURP123",
}
```



```
"id_estado": 1,  
"id_municipio": 2,  
"id_hospital": 3,  
"id_group": 4  
}
```

Actualiza la información de un empleado en user_data y también actualiza su relación en group_users.

Endpoints POST

- **/api/email/send-credentials**

Envía por correo electrónico (usando Resend) las credenciales de acceso de un empleado con un correo HTML personalizado.

Body esperado:

```
{  
  "correo_electronico": "empleado@ejemplo.com",  
  "nombre": "Juan",  
  "ap_paterno": "Pérez",  
  "user": "usuario123",  
  "pass": "contraseña123"  
}
```

Respuesta exitosa (200):

```
{  
  "success": true,  
  "data": { /* respuesta de Resend */ }  
}
```

Errores posibles:

- 400 si no se envía correo_electronico.
- 500 si ocurre algún error al enviar el correo.

Endpoints GET

- **/api/dashboards/estatal/metricas**

Devuelve métricas generales del estado (empleados, grupos, hospitales, etc.).

Query params: id_estado, fecha_inicio, fecha_fin

Ejemplo de respuesta:

```
{  
  "total_empleados": 120,  
}
```

```
"total_grupos": 15,  
"total_hospitales": 8  
}
```

- **/api/dashboards/estatal/entradas-salidas**
Devuelve datos de entradas y salidas de empleados en el estado.
- **/api/dashboards/estatal/eventos-geocerca**
Devuelve eventos de geocerca registrados en el estado.
- **/api/dashboards/estatal/ranking-hospitales**
Devuelve un ranking de hospitales del estado según actividad o métricas.
- **/api/dashboards/estatal/horas-municipio**
Devuelve el total de horas trabajadas por municipio en el estado.
- **/api/dashboards/estatal/distribucion-municipal**
Devuelve la distribución de empleados o actividad por municipio.
- **/api/dashboards/estatal/distribucion-municipal-completa**
Devuelve la distribución completa de actividad por municipio.
- **/api/dashboards/estatal/municipio-detalle**
Devuelve el detalle de un municipio específico del estado.
- **/api/dashboards/municipios-by-estado/:id_estado**
Devuelve la lista de municipios de un estado.

Ejemplo de respuesta:

```
[  
  { "id_municipio": 1, "nombre_municipio": "Municipio A" },  
  { "id_municipio": 2, "nombre_municipio": "Municipio B" }  
]
```

- **/api/dashboards/nacional/estadisticas-estados**
Devuelve estadísticas generales de todos los estados.
- **/api/dashboards/nacional/totales**
Devuelve totales nacionales (empleados, hospitales, etc.).

Ejemplo de respuesta:

```
{  
  "total_empleados": 2000,  
  "total_hospitales": 150,  
  "total_grupos": 300  
}
```

- **/api/dashboards/nacional/ranking-estados**
Devuelve un ranking nacional de estados según actividad o métricas.

Endpoints POST

- **/api/dashboards/grupo**
Devuelve métricas y datos de un grupo específico para dashboards.

Body esperado:

```
{  
  "id_grupo": 1,  
  "fecha_inicio": "2024-01-01",  
  "fecha_fin": "2024-01-31"  
}
```

- **/api/dashboards/municipio**
Devuelve métricas y datos de un municipio específico para dashboards.

Body esperado:

```
{  
  "id_municipio": 1,  
  "fecha_inicio": "2024-01-01",  
  "fecha_fin": "2024-01-31"  
}
```

Endpoints para app móvil (/api/empleados y /api/ubicaciones)

Endpoints POST

- **/api/empleados/login**
Inicia sesión de empleado desde la app móvil.

Body esperado:

```
{  
  "user": "usuario",  
  "pass": "contraseña"  
}
```

Respuesta exitosa:

```
{  
  "success": true,  
  "token": "jwt_token",  
  "user": {  
    "id_user": 1,  
    "nombre": "Juan",  
    "role": "empleado"  
  }  
}
```

```
}  
}
```

Errores posibles:

- 401 si las credenciales son incorrectas.
- **/api/ubicaciones**
Registra la ubicación del empleado.

Body esperado:

```
{  
  "id_user": 1,  
  "latitud": 25.6,  
  "longitud": -100.3,  
  "fecha_hora": "2024-07-19T10:00:00Z",  
  "dentro_geocerca": true  
}
```

Respuesta exitosa:

```
{ "success": true }
```

- **/api/empleados/cambiar-password**
Permite al empleado cambiar su contraseña desde la app móvil.

Body esperado:

```
{  
  "id_user": 1,  
  "old_pass": "anterior",  
  "new_pass": "nueva"  
}
```

Respuesta exitosa:

```
{ "success": true, "message": "Contraseña actualizada" }
```

Errores posibles:

- 400 si la contraseña anterior es incorrecta.

Endpoints `/api/municipioadmin`

Endpoints GET

- **/api/municipios-by-estado/:id_estado**
Devuelve la lista de municipios de un estado específico.

Parámetro (URL): id_estado

Ejemplo de respuesta:

```
[
  { "id_municipio": 1, "nombre_municipio": "Municipio A" },
  { "id_municipio": 2, "nombre_municipio": "Municipio B" }
]
```

- **/api/municipios-by-estado-hospital/:id_estado**
Devuelve la lista de municipios de un estado que tienen hospitales registrados.

Parámetro (URL): id_estado

Ejemplo de respuesta:

```
[
  { "id_municipio": 1, "nombre_municipio": "Municipio A" },
  { "id_municipio": 3, "nombre_municipio": "Municipio C" }
]
```

- **/api/hospitals-by-user/:id_user**
Devuelve los hospitales del municipio asignado al usuario.

Parámetro (URL): id_user

Ejemplo de respuesta:

```
[
  { "id_hospital": 1, "nombre_hospital": "Hospital Municipal 1" },
  { "id_hospital": 2, "nombre_hospital": "Hospital Municipal 2" }
]
```

- **/api/stats-by-user/:id_user**
Devuelve totales de hospitales, grupos y empleados del municipio asignado al usuario.

Parámetro (URL): id_user

Ejemplo de respuesta:

```
{
  "total_hospitales": 3,
  "total_grupos": 5,
  "total_empleados": 40
}
```

- **/api/empleados-by-user/:id_user**
Devuelve los empleados y grupos del municipio asignado al usuario.

Parámetro (URL): id_user

Ejemplo de respuesta:

```
[
  {
    "id_user": 1,
    "nombre": "Juan",
    "grupo": "Grupo A"
  },
  {
    "id_user": 2,
    "nombre": "Ana",
    "grupo": "Grupo B"
  }
]
```

- **/api/grupos-by-user/:id_user**
Devuelve los grupos del municipio asignado al usuario.

Parámetro (URL): id_user

Ejemplo de respuesta:

```
[
  { "id_group": 1, "nombre_grupo": "Grupo A" },
  { "id_group": 2, "nombre_grupo": "Grupo B" }
]
```

Endpoints POST

- **/api/create-municipioadmin**
Crea un nuevo administrador municipal.

Body esperado:

```
{
  "nombre": "Carlos",
  "ap_paterno": "Ramírez",
  "ap_materno": "López",
  "CURP": "CURP789",
  "user": "carlosr",
  "pass": "123456",
  "role_name": "municipioadmin",
  "municipio": "Municipio A",
  "id_estado": 1
}
```

Respuesta exitosa:

```
{ "success": true, "message": "Administrador municipal creado" }
```

Errores posibles:

- 400 si faltan datos obligatorios o el usuario ya existe.

Endpoints /api/reportes

Endpoints POST

- **/api/reportes/empleado**
Genera un reporte de registros de ubicación de un empleado en un rango de fechas.

Body esperado:

```
{
  "id_user": 1,
  "fecha_inicio": "2024-07-01",
  "fecha_fin": "2024-07-19"
}
```

Respuesta exitosa:

```
[
  {
    "latitud": 25.6,
    "longitud": -100.3,
    "fecha_hora": "2024-07-19T10:00:00Z",
    "dentro_geocerca": true
  },
  {
    "latitud": 25.7,
    "longitud": -100.4,
    "fecha_hora": "2024-07-18T09:30:00Z",
    "dentro_geocerca": false
  }
]
```

Errores posibles:

- 400 si faltan datos obligatorios o el usuario no existe.

Endpoints /api/superadmin

Endpoints GET

- **/api/superadmin/hospitals**
Devuelve la lista de todos los hospitales registrados.

Ejemplo de respuesta:

```
[
  { "id_hospital": 1, "nombre_hospital": "Hospital General" },
  { "id_hospital": 2, "nombre_hospital": "Hospital Municipal" }
]
```

- **/api/superadmin/estados**
Devuelve la lista de todos los estados registrados.

Ejemplo de respuesta:

```
[
  { "id_estado": 1, "nombre_estado": "Nuevo León" },
  { "id_estado": 2, "nombre_estado": "Jalisco" }
]
```

- **/api/superadmin/estadoadmins**
Devuelve la lista de todos los administradores estatales.

Ejemplo de respuesta:

```
[
  { "id_user": 1, "nombre": "Pedro", "estado": "Nuevo León" },
  { "id_user": 2, "nombre": "Ana", "estado": "Jalisco" }
]
```

- **/api/superadmin/totaladmins**
Devuelve el total de administradores estatales registrados.

Ejemplo de respuesta:

```
{ "total_admins": 5 }
```

- **/api/superadmin/superadmin-hospital-work**
Devuelve la lista de hospitales donde el superadmin tiene asignación de trabajo.

Ejemplo de respuesta:

```
[
  { "id_hospital": 1, "nombre_hospital": "Hospital General" }
]
```

- **/api/superadmin/superadmin-hospital-ubi/:id_user**
Devuelve la ubicación del hospital asignado al superadmin.

Parámetro (URL): id_user

Ejemplo de respuesta:

```
{
  "id_hospital": 1,
```



```
"nombre_hospital": "Hospital General",  
"latitud": 25.6,  
"longitud": -100.3  
}
```

- **/api/superadmin/superadmin-hospitals-by-municipio**
Devuelve la lista de hospitales de un municipio y estado específicos.

Query params: id_estado, id_municipio

Ejemplo de respuesta:

```
[  
  { "id_hospital": 1, "nombre_hospital": "Hospital Municipal 1" }  
]
```

- **/api/superadmin/hospitales-by-municipio**
Devuelve la lista de hospitales de un municipio específico.

Query param: id_municipio

Ejemplo de respuesta:

```
[  
  { "id_hospital": 2, "nombre_hospital": "Hospital Municipal 2" }  
]
```

Endpoints POST

- **/api/superadmin/create-admin**
Crea un nuevo administrador estatal.

Body esperado:

```
{  
  "nombre": "Pedro",  
  "ap_paterno": "Gómez",  
  "ap_materno": "Ruiz",  
  "CURP": "CURP456",  
  "user": "pedrog",  
  "pass": "123456",  
  "role_name": "estadoadmin",  
  "estado": "Nuevo León"  
}
```

Respuesta exitosa:

```
{ "success": true, "message": "Administrador estatal creado" }
```

Errores posibles:

- 400 si faltan datos obligatorios o el usuario ya existe.
- **/api/superadmin/delete-admin/:id_user**
Elimina un administrador estatal.

Parámetro (URL): id_user

Respuesta exitosa:

```
{ "success": true, "message": "Administrador eliminado" }
```

Errores posibles:

- 400 si el usuario no existe.
- **/api/superadmin/create-superadmin**
Crea un nuevo superadministrador.

Body esperado:

```
{
  "nombre": "Carlos",
  "ap_paterno": "Ramírez",
  "ap_materno": "López",
  "CURP": "CURP789",
  "user": "carlosr",
  "pass": "123456",
  "role_name": "superadmin"
}
```

Respuesta exitosa:

```
{ "success": true, "message": "Superadministrador creado" }
```

Errores posibles:

- 400 si faltan datos obligatorios o el usuario ya existe.

Endpoints PUT

- **/api/superadmin/update-admins**
Actualiza la información de un administrador estatal.

Body esperado:

```
{
  "id_user": 2,
  "nombre": "Pedro",
  "ap_paterno": "Gómez",
  "ap_materno": "Ruiz",
  "CURP": "CURP456",
  "estado": "Nuevo León"
}
```

Respuesta exitosa:

```
{ "success": true, "message": "Administrador actualizado" }
```

Errores posibles:

- 400 si faltan datos obligatorios o el usuario no existe.
- **/api/superadmin/superadmin-hospital**
Asigna un hospital a un superadmin.

Body esperado:

```
{  
  "id_user": 1,  
  "id_hospital": 2  
}
```

Respuesta exitosa:

```
{ "success": true, "message": "Hospital asignado" }
```

Errores posibles:

- 400 si faltan datos obligatorios o el usuario/hospital no existe.