

Requisitos entrega API Express.js

- I. La API deberá entregarse antes del **Viernes 27 de Septiembre**
- II. El código deberá ser subido a github, de manera pública o privada y estar correctamente documentado a través del archivo README.md. En caso de publicarse de manera privada deberá ser compartido con el docente a través de la casilla: sganan81@gmail.com
- III. No deben figurar en el código: claves de acceso, contraseñas, tokens, etc. Estos valores solo deben ser utilizados en su archivo **.env** y ser ignorado al momento de inicializar el repositorio(.gitignore). Igualmente pueden indicar como deben crearse las constantes en el archivo **.env** subiendo otro de ejemplo con sus valores en blanco (sample.env). Ver repositorio de referencia [aquí](#)
- IV. La API deberá funcionar en su entorno local en un puerto determinado (Ej. http://localhost:3000) y además deberá ser subida a [Render](#) configurando las variables de entorno a través del dashboard que provee dicha plataforma.
- V. Cada endpoint/request deberá devolver los status code correspondientes (200, 401, 404, 500, etc.) y manejar el JSON de respuesta de manera uniforme en todo el proyecto.
Ej. para las respuestas 2xx: {status:'ok', data : [...]}
Ej. para las respuestas 4xx: {status:'error', msg: 'Bad Request'}
Ej. para las respuestas 5xx: {status:'error', msg: 'Error inesperado al obtener la información'}
- VI. El proyecto se debe desarrollar de manera grupal hasta un máximo de 4 personas.
 - Queda a cargo de cada alumno la búsqueda de una API pública o que requiera autenticación mediante token/api_key. Ej's:
 1. <https://mockapi.com>
 2. <https://developer.themoviedb.org/docs/getting-started>
 3. <https://openweathermap.org/api>
 4. <https://developer.spotify.com/documentation/web-api>
 5. <https://unsplash.com/developers>
 6. https://www.mockaroo.com/mock_apis

VII. Cada Alumno del grupo deberá cubrir los siguientes endpoints/ request.

1. Listado de registros en formato json que muestre 50 o más registros. Ej: /api/v1/empleados/
 2. Visualización de un registro en particular (param id) Ej: /api/v1/empleados/235/
 3. Agregar query params al listado del punto “1” para filtrar la información por algún campo relevante. (Ej.: category, lang, page, year, order, etc.) Ej: /api/v1/empleados?page=1&lang=es
- Aclaraciones:
 - Cada alumno deberá cubrir los puntos del 1 al 3 en su entrega individual dentro del proyecto.
 - Cada una de las rutas de acceso deberán estar documentadas en el README.md para que el docente pueda verificar por cada integrante del grupo los endpoints/request entregados.

VIII. Requerimientos globales.

1. Configurar linter y husky dentro del proyecto.
2. Archivo .gitignore para elementos no incluidos en el repo
3. Verificar el correcto uso de dependencias productivas y de desarrollo.
4. Establecer scripts para el inicio de la aplicación y manejo de errores (linter) Ej.: npm run start
5. Manejar el dominio de nuestra API dentro del archivo .env en caso de utilizar una api externa.

- IX.** La versión final de la api deberá estar sobre la rama “main” pero necesitarán utilizar branch’s por separado para el trabajo grupal. Ej. (dev, develop, branch-nombre-alumno, branch-legajo, etc.)
- Cada alumno deberá subir sus cambios utilizando la herramienta git, ya sea mediante la creación de una Pull Request o simplemente mergeando su Branch con dev o main.

En caso de NO encontrar participación de algún alumno dentro del repositorio, la entrega individual del participante se dará por desaprobada (NO la grupal)

X. Material a compartir con el docente:

1. Repositorio en GitHub (Documentar en README)
2. Url de Render
3. Claves de acceso/token/api_key en caso de que sea necesario.