



## Prueba Técnica – Desarrollador Backend .NET

Bienvenido a la prueba de desarrollador Backend de Lisit, en esta evaluación tomaremos en cuenta tu manejo con .NET 6, patrones de diseño y arquitectónicos. Cabe mencionar que la legibilidad de tu código, la capacidad de iterar y refactorizar el mismo en tu solución serán aspectos de alto impacto en tu evaluación. Mucha suerte.

La empresa **The Last Bug SPA** nos ha pedido diseñar una API basado en microservicios para poder gestionar usuarios y comunidades en las que residen para evaluar futuras ayudas sociales. Los Endpoint de creación y edición de datos deben estar sometidos a validación de seguridad con rol de administrador, así mismo, la solución deberá prestar servicios a aplicaciones internas del estado las cuales no estarán sometidas a validación de seguridad.

La solución deberá proveer los siguientes servicios (puedes agruparlos como desees):

- Servicios de autenticación: debe contener mínimo los endpoints para login y registrar usuarios.
- Servicios de localización: considerando la estructura País, región y comuna se debe permitir un CRUD básico para cada estructura. (País puede tener varias regiones, y una región puede tener varias comunas)
  - Los GETS son de libre consumo para cualquier sistema
  - Los POST, PUTS, DELETE estarán sometidos a validación de rol, solo administrador.
- Servicios de ayudas sociales: Están asignados por comuna y solo a los residentes de dichas comunas
  - A una persona no se le puede asignar más de una vez con el mismo servicio social el mismo año.
  - El administrador puede ver personas y los servicios asignados, le puede asignar alguna ayuda social.
  - Una persona puede obtener sus ayudas sociales asignados por año y el último vigente.
  - El administrador puede obtener las ayudas sociales asignadas a un usuario.
  - El administrador puede crear nuevas ayudas sociales para las comunas o regiones. Si se crea en una región se asigna a todas las comunas de esta.
- Logging: Debe registrar cada acción de un usuario, quien hizo que y a qué hora. Un administrador puede rescatar esta información cuando desee escogiendo el día.

Acotación:

- Para seguridad puedes utilizar la plataforma Identity-Server
- Usar external logins (Google, Facebook, Twitter, etc.) es bien valorado.
- Puede utilizar cualquiera de estas tecnologías EF, Dapper o ADO.NET.
- Auto documentar la API con Swagger es bien valorado.
- Las propiedades de navegación y una vista de modelos de datos normalizados e inicializados, respetando las FK y restricciones de tamaño es altamente calificado.
- Libertad creativa es bien valorado, si deseas agregar algo que sientas que aporta, ¡¡Genial!! :D

**Para su evaluación enviar el repositorio público donde se encuentre el código (GitHub), con instrucciones de montado o bien publicado en cualquier hosting y el modelo de base de datos generado.**

