

## GENERALES

Se recomienda leer este documento hasta el final antes de comenzar.

El objetivo de esta asignación es evaluar los conocimientos de un desarrollador en el framework Angular v1. Está diseñado para ser un trabajo individual. El trabajo que se pide que se realice es opcional, sin una fecha límite de entrega, compromisos ni garantías de ninguna de las partes.

La url base de la aplicación es <http://beta.maptasking.com/> A esto agregue el endpoint a llamar, por ejemplo <http://beta.maptasking.com/api/jobs>. Todos los endpoints requieren autenticación por bearer token.

Nos referimos a componente en la aplicación como uno o más componentes de angular (controller, service, etc) responsables de una misión particular dentro de la aplicación. Por ejemplo un componente llamado login puede ser el conjunto de un LoginController y un LoginService, el desarrollador decidirá que componente de angular hace más sentido para realizar el trabajo necesario.

Se considera un plus la utilización del menor número de dependencias posible, como también la capacidad del desarrollador de implementar el menor número de componentes de angular para lograr el trabajo deseado, siempre y cuando se mantenga como primera prioridad la claridad del código y cada cosa en su lugar. También se considera beneficioso usar el menor número de inyectables necesarios para cumplir una función (hint: controllerAs).

Se espera que utilice la versión más actualizada estable de angular ~1.4

## GUÍA

1. Clone el repositorio [será enviado en otro correo]

En el repositorio encontrará el inicio de una aplicación de angular, esta solamente contiene la lógica necesaria para realizar una llamada a un proveedor OAuth2 (POST /token) y recibir un bearer token

2. Complete el componente de autenticación

Para que el API autorice un request deberá agregar al request el header Authorization con el valor **bearer {{token}}** en donde {{token}} es el token recibido durante la autenticación. El token es válido por 15 días y funciona para multiples request.

La autenticación deberá persistir entre cierres de browser. No es necesario encriptar el token.

3. Implemente ui-router

Utilice al menos 1 vista anidada y alguna pantalla con multiples vistas.

4. Cree el componente de tareas (jobs)

Se podrá:

- Listar tareas
- Agregar/editar tareas

Las tareas tienen contactos, se debe poder seleccionar el contacto para una tarea y dar la capacidad al usuario de agregar un nuevo contacto de no existir el deseado en la lista, cosa que debe hacerse durante el flujo de creación/edición de tarea.

Endpoints necesarios:

GET POST PUT api/jobs

GET POST PUT api/contacts

5. Cree el componente de contactos (contacts)

GET POST PUT api/contacts

6. Utilice al menos una directive en algún lugar de la aplicación, en donde más le haga sentido

7. Siga estas directrices de estilo de código:

Utilice una estructura basada en componentes de la aplicación

<https://scotch.io/tutorials/angularjs-best-practices-directory-structure>

Mantenga los archivos .js simples y claros. Como regla de dedo si no se pueden ver todos los componentes declarados en el .js a simple vista sin mover la pantalla, entonces deberían separarse en varios archivos. La cantidad de archivos creados no es una negativa, al contrario, el orden es un plus.

Minimice el riesgo de sobre-escribir funciones/variables globales

Para el nombre de los controllers y services utilice PascalCase, para factories camelCase. Utilice la palabra completa Controller, Service, Factory, etc. Mantenga esta nomenclatura para los nombres de los archivos .js

No utilice ni templates ni scripts inline

Para vistas parciales prenda el nombre del html con \_

8. La simplicidad, estética y experiencia de usuario en el diseño de las pantallas son un plus
9. Suba la aplicación a un repo de GitHub y haga un deploy funcional, no minificado a una dirección web. Comparta los links al correo: [dev@nexsyscomputing.com](mailto:dev@nexsyscomputing.com)
10. Utilice bower para las dependencias de la aplicación, no trackee las dependencias en el repo. Bower install debe ser capaz de reestablecer las dependencias en su totalidad.

## BONUS

Opcionalmente realice:

1. Los jobs tienen una ubicación geográfica, use el módulo Angular Google Maps para mostrar un mapa con dichos puntos.

`ui-gmap-google-map`

2. Conecte los elementos de la lista de jobs con los pines del mapa de tal forma que desde la lista se pueda saber cuál es el pin correspondiente en el mapa, y desde el mapa se pueda saber a qué elemento corresponde el pin (haciendo click, mouseover, infowindow o combinación de las anteriores, como le parezca que sea más amigable al usuario)
3. Muestre la ubicación del usuario y la precisión.

## CRITERIOS A EVALUAR

Sin orden específico:

- Creatividad
- Cumplimiento de los objetivos
- Entendimiento y apego a las instrucciones
- Orden, arquitectura y legibilidad del código (SOLID)
- Interfaz visual y experiencia de usuario
- WOW factor
- Manejo del framework
- Dominio de mejores prácticas angular y javascript