

## EXAMEN TÉCNICO

### **Parte Front end**

#### Descripción:

Esta sección consiste en desarrollar una pantalla en donde se muestre los menús que contiene un Restaurante. Se detallan las siguientes especificaciones:

1. Un menú cuenta con un nombre, una descripción, un precio con su respectiva moneda, una foto, los días que está disponible, fecha desde y hasta en el cual es válido, horario desde hasta en el cual aplica y un ranking de puntuación numérico que va desde el 1 al 5. Por ejemplo, el menú "cena 1" es válido de Viernes a Domingos de 20:00 hs. a 23:00 hs, desde el 1 de Marzo al 1 de Abril y tiene una puntuación de 5.
2. La lista de menús tiene que contener información básica, como el nombre, el precio, fecha desde y hasta que esté disponible y acciones de borrar y modificar. Al hacer click en el menú, tiene que mostrarme el detalle completo debajo del mismo.
3. Las acciones borrar y modificar solo tienen que mostrar un mensaje que diga "en construcción".
4. Se debe proporcionar una sección de filtros, donde se puedan filtrar los menús por día en el que está disponible, fecha desde y hasta de validez, horario y precio. Los filtros tienen que filtrar realmente la grilla.
5. La pantalla debe contar con un menú, donde además de la solapa "menú", existan las solapas "promociones", "empleados", "reportes". Estas solapas no tienen que tener información alguna, solo son mockups para representar el menú.
6. La pagina debera ser responsive.

#### Tecnologías:

Se deben utilizar las siguientes tecnologías:

- AngularJS
- CSS3
- HTML 5
- Spring web MVC
- Maven

Se pueden agregar todas las cosas que crea necesario, también serán valoradas.

### **Parte back end**

#### Descripción:

Esta sección consiste en desarrollar la parte back de lo pedido en el apartado anterior. Para ello, se deberá definir una arquitectura básica de la forma que crea conveniente.

Se detallan las siguientes especificaciones:

1. La parte back deberá proveer la lista de menús a consumir por el front end.

2. Los datos de menús a devolver deben ser mocks. No hace falta obtenerlo de una base de datos. Poner estos mocks en el lugar que crea conveniente.
3. Crear las entidades que considere necesarias.
4. Se pide también que el back provea una funcionalidad para que se pueda devolver los menús agrupados por precio. Tener en cuenta que en un futuro, también se pueden agrupar por ranking.

#### Tecnologías / Buenas prácticas:

Se valorará el uso de las siguientes tecnologías/ buenas prácticas:

- Uso de interfaces
- Spring (inyección de dependencias)
- Maven

Se pueden agregar todas las cosas que crea necesario, también serán valoradas.

### **Resultado**

#### Entrega:

Se deberá entregar el código fuente del proyecto completo. Se puede entregar en formato zip, SVN, Github, etc. Es indistinto.

#### Evaluación:

Se van a evaluar los siguientes aspectos:

1. Uso de las tecnologías requeridas.
2. Distribución de funcionalidad.
3. Diseño.
4. Completitud.
5. Prolijidad del código.