

EXAMEN TÉCNICO

Especificación: se detallan las siguientes especificaciones:

1. Un menú cuenta con una lista de ítems (al menos 1), y una lista de submenús (en donde cada elemento de la lista es un menú, que puede estar vacía). Cada menú posee además una descripción y un flag que dice si este menú está activo o no.
2. Un ítem de 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 ítem de menú "Milanesa napo" 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.
3. Un restaurante puede ofrecer diferentes menús, por ejemplo, "Menú del día" o "Menú de verano".

Descripción:

Esta sección consiste en desarrollar la parte back de un sistema cuyo objetivo es manejar los menús de un restaurante. Para ello, se deberá definir una arquitectura de la forma que crea conveniente.

Se detallan los siguientes requerimientos:

1. La información va a ser consumida a través de servicios REST.
2. Se debe poder listar los menús a consumir por el front-end.
3. Se debe proveer una funcionalidad para que, dado un menú, se pueda devolver los ítems del mismo agrupados por precio. Tener en cuenta que en un futuro, también se pueden agrupar por ranking.
4. Se debe realizar una función que dado un menú, devuelva la suma de los precios de todos sus ítems (incluyendo submenús). Esta información no va a ser consumida por el front-end, es lógica pura de negocio.
5. Se debe realizar una función que dado un menú, devuelva la cantidad de submenús que están activos. Esta información no va a ser consumida por el front-end, es lógica pura de negocio.
6. Si ahora se pide agregar una nueva lógica que dado un menú, se muestre en consola los nombres de todos los submenús, que cambios haría?
7. Se requiere loguear el tiempo de ejecución de cada método, o de un conjunto de ellos.
8. 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.
9. Crear las entidades que considere necesarias.

Asuma que no existe recursión infinita en cada menú.

Tecnologías / Buenas prácticas:

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

- Uso de interfaces
- Spring (inyección de dependencias)
- Maven
- Patrones de diseño.

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.