Preguntas

1. ¿Me contarías sobre algún artículo, webinar, podcast o libro de algún tema de tecnología o desarrollo que leíste recientemente? ¿Por qué te gustó y por qué debería leerlo yo?

   Hace un tiempo vi en YouTube, una grabación de un evento donde Uncle Bob habló sobre Clean Code (Link). Me pareció fascinante como tomaba un código redundante, poco entendible y desordenado, para refactorizarlo en algo mucho más legible, ordenado, modularizado y prolijo. Creo que el video es una buena introducción que nos puede ayudar a mejorar poco a poco nuestro código gracias a las buenas prácticas que enseña.

2. ¿Qué feature de WoowUp te gustó mucho? ¿Por qué?

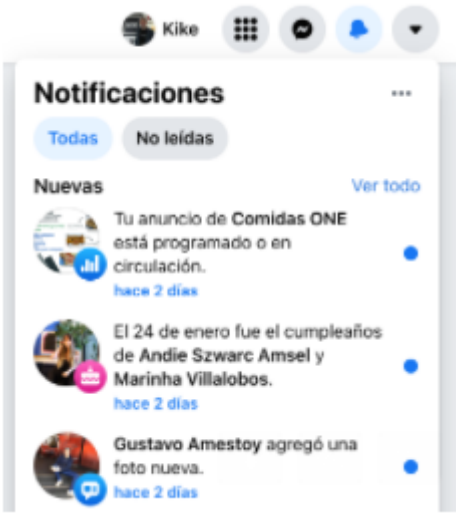
   Todavía no tengo demasiada información sobre las distintas Features de WoowUp, pero me resulta interesante conocer la manera en la que agrupan el comportamiento de los usuarios para luego poder segmentar la información por diferentes categorías.

3. ¿Cuál es tu remuneración pretendida?

   70.000ARS

Ejercicio - Sistema de Alertas

Seguramente conoces la funcionalidad de Notificaciones de Facebook: es esa campanita arriba en el menú donde te muestra las nuevas alertas que el sistema tiene para mostrarte sobre distintos temas: un amigo cumple años, una página que seguís compartió una publicación, un amigo publicó una foto, alguien comentó un posteo tuyo, una sugerencia de amistad, etc.



Facebook no es el único. En general todas las aplicaciones tienen un sistema de alertas o notificaciones. En este ejercicio, te vamos a pedir que hagas una versión muy simplificada.

Se pide programar un sistema para enviar alertas a usuarios que tenga la siguiente funcionalidad:

1. Se pueden registrar usuarios que recibirán alertas.
2. Se pueden registrar temas sobre los cuales se enviarán alertas.
3. Los usuarios pueden optar sobre cuales temas quieren recibir alertas.
4. Se puede enviar una alerta sobre un tema y lo reciben todos los usuarios que han optado recibir alertas de ese tema.
5. Se puede enviar una alerta sobre un tema a un usuario específico, solo lo recibe ese único usuario.
6. Una alerta puede tener una fecha y hora de expiración. Las alertas que tienen expiración, no se muestran al usuario si han expirado.
7. Hay dos tipos de alertas:
    1. Informativas: Solo le llega al usuario si ha optado recibir alertas de ese tema
    2. Urgentes: Siempre le llega al usuario la alerta, sin importar si ha optado recibir alertas de ese tema o no.
8. Un usuario puede marcar una alerta como leída.
9. Se pueden listar todas las alertas no expiradas de un usuario que aún no ha leído, ordenadas por fecha de la más reciente a la más antigua
10. Se pueden listar todas las alertas no expiradas para un tema (de la más reciente a la más antigua). Se muestra para cada alerta si es para todos los usuarios o para uno específico).

Aclaraciones importantes:

- La aplicación se ejecuta desde línea de comando. En ningún caso pedimos que escribas código de front end.
- Debe tener Tests Unitarios.
- No debes hacer ningún tipo de persistencia de datos (base de datos, archivos). Todo debe resolverse con estructuras en memoria.
- Si tenés que hacer algún supuesto sobre algo que no esté claro en el ejercicio, por favor anotalo para que lo tengamos en cuenta al revisar el código.

Al responder el ejercicio, por favor entregá código que funcione y se pueda probar. Se minucioso en los detalles y en la claridad del código que escribas para que al revisor le sea fácil entenderlo.

Cuando revisamos el ejercicio, esto es lo que evaluamos:

1. Solución: ¿El código soluciona correctamente los requisitos?

2. Programación orientada a objetos: ¿Hay un modelo de clases pensado, que modela la realidad del enunciado? ¿Se usa herencia, polimorfismo, patrones de diseño?

3. Legibilidad del código: Miramos la elección de los nombres de las variables, los nombres de los métodos y de las clases. ¿Son lo suficientemente representativos como

para entender su funcionamiento o propósito, si lo leyera un compañero tuyo sin que vos estés?

4. Principio de responsabilidad única: ¿Cada método hace una única cosa? (¿O tienen mucho código y se podría refactorizar en varios métodos más cortitos?). Cada clase, ¿tiene una única responsabilidad?

5. Unit Test: ¿Hay tests que prueben el correcto funcionamiento de los casos principales?

Podés usar cualquier lenguaje con el que te sientas cómodo, creemos que un buen programador puede ser productivo en cualquier lenguaje. Sugerencia: Podés usar https://gist.github.com/ para compartirnos el código.

Ejercicio SQL

Escribir una consulta SQL que traiga todos los clientes que han comprado en total más de 100,000$ en los últimos 12 meses usando las siguientes tablas:

Clientes: ID, Nombre, Apellido

Ventas: Fecha, Sucursal, Numero_factura, Importe, Id_cliente.

Solución en MySQL:

```
select c.id, c.nombre, c.apellido, sum(v.importe) as monto_total
from clientes c join ventas v on c.id = v.id_cliente
where timestampdiff(month, v.fecha, current_date()) <= 12
group by c.id
having monto_total > 100000.00;
```