Este es el segundo examen parcial del curso *ARSW*, 2015-2. El examen tiene 4 preguntas; otorga un total de 100 puntos. El examen es **individual** y se permite el uso de sus apuntes y material que usted haya elaborado en el curso, así como acceder a la documentación técnica en línea. **Implica anulación:** Tener abierto un cliente de correo electrónico, mensajería o compartir información con sus compañeros. Una vez terminado el ejercicio, suba el código en formato .ZIP al espacio correspondiente en moodle.

Nombre y código: _____

Pregunta	1	2	3	4	**Total**
Puntos	60	10	10	20	100
Puntaje					

Quien-Da-Mas es una plataforma que permite realizar subastas en línea. Por el momento se quiere que los compradores al enterarse de un oferta (subasta) de un producto respondan con la suma de dinero que estan dispuestos a pagar por el mismo. Esta suma no debe ser inferior a la de un precio base (`startPrice`). La plataforma se encarga de procesar y elegir automáticamente la oferta mas favorable económicamente.

La versión actual de la plataforma consta de una herramienta de escritorio a través de la cual los oferentes publican los productos que quieren subastar (proyecto ManejadorOfertas). Por otro lado, los 'compradores' tienen una herramienta a la que se le notifica (bajo un esquema de mensajería de tipo publicador/suscriptor) cuando se haya publicado un nuevo producto.

Por ahora, la herramienta de los compradores no tiene interfaz gráfica, y es básicamente una simulación que en la medida que recibe ofertas, automáticamente realiza una propuesta económica aleatoria para la misma (proyecto QuienDaMasApp `https://github.com/ARSW-ECI/quien-da-mas`), siguiendo un esquema cliente/servidor (a través del servicio remoto `agregarOferta` de ManejadorOfertas).

Tenga en cuenta que para distinguir un 'comprador' de otro, cada cliente genera una identidad aleatoriamente al iniciarse. La versión actual de la plataforma está incompleta, aunque `ManejadorOfertas` está enviando eventos cada vez que se oferta un nuevo producto, las instancias de `QuienDaMasApp` no están haciendo nada con los mismos una vez lo reciben.

En esta oportunidad se utiliza el sistema de mensajería *RabbitMQ* `https://www.rabbitmq.com/tutorials/amqp-concepts.html`.

1. En el código
   (a) (20 puntos) Los compradores registren su oferta una vez que sean notificados de un nuevo producto.
   (b) (20 puntos) El servidor (ManejadorOfertas) muestre en la ventana de 'ofertas aceptadas' la oferta ganadora una vez se hayan recibido las tres primeras propuestas (si la información se muestra en la consola en lugar de la ventana valdrá la mitad del punto).
   (c) (20 puntos) Informar al compradores que se le vendió un producto. Cuando una instancia del 'comprador' (QuienDaMasApp) se entere de que es ganador de la subasta, éste debe imprimir en su pantalla "El comprador XXXXX compro el producto YYYY", donde "XXXXX" y "YYYYY" son los respectivos identificadores.

2. En la hoja de respuestas.
   (a) (10 puntos) Bajo la configuración actual del servicio ofrecido por ManejadorOfertas, indique dos posibles inconsistencias que se podrían presentar, considerando que habrá muchas instancias de QuienDaMasApp accediendo a dicho servicio.

3. En el código
   (a) (10 puntos) Implemente una solución que prevenga las inconsistencias antes mencionadas, evitando (por eficiencia) hacer bloqueos innecesarios.

4. En la hoja de respuestas
   (a) (20 puntos) Realice el diagrama de actividades UML de lo realizado (independientemente de que funcione o no), incluyendo las columnas 'ManejadorOfertas' y 'QuienDaMasApp'.