SD	**Sistemas Distribuidos**
21/22	Práctica no guiada: Sockets, Eventos, Colas y modularidad
	Gestor de colas en parques temáticos **“Fun with Queues”**

Preámbulo

El objetivo de esta práctica es que los alumnos extiendan y afiancen sus conocimientos sobre el uso de la tecnología de comunicación básica sockets, así como streaming de eventos y colas, estudiadas durante las sesiones de teoría y práctica.

Los sockets son la base de las comunicaciones entre computadores y suponen el punto de enlace básico entre nuestra aplicación e Internet, utilizando muy pocos recursos de red. Como contrapartida, las aplicaciones que utilizan sockets como mecanismo de comunicación deben interpretar los mensajes que intercambian entre sí, es decir, deben establecer un protocolo o acuerdo de comunicación entre ellos. Esta necesidad implica un fuerte acoplamiento entre las aplicaciones distribuidas que utilizan sockets como mecanismo de comunicación, ya que todas las partes deben conocer de antemano la estructura de los mensajes que serán intercambiados y codificarlo en su programación.

Esta práctica establece el marco inicial de un desarrollo que se ampliará durante el cuatrimestre y que permitirá al alumno crear una aplicación similar a las que se desarrollan hoy en día en los entornos empresariales, poniendo el foco en el uso e integración de distintos paradigmas de comunicación susceptibles de ser utilizados.

Especificación

El objetivo de la práctica a desarrollar es construir un sistema distribuido que monitorice y gestione los tiempos de espera de las colas en las atracciones de un parque temático.

La solución será capaz de controlar la localización de los visitantes del parque y los tiempos de espera en las atracciones, informando de los mismos a los visitantes los cuales, ante tiempos excesivamente largos, tomarán la decisión de dirigirse a otra atracción.

Imagen 1: App para la visualización de colas de Disney Orlando

Descripción funcional

La solución completa deberá contener los siguientes elementos dando respuesta a los requerimientos que se exponen dentro de cada uno de ellos.

Mapa

El parque temático será representado mediante un mapa en el que se podrá ver tanto las atracciones como la posición de los visitantes que están circulando por el parque.

Dicho mapa se representará mediante una matriz de 20x20 donde cada elemento de la matriz podrá contener uno de los siguientes valores:

1- Un número entero [0, n]: Indica el tiempo en minutos de una atracción localizada en esa posición.
2- Un visitante: Representado por una cualquier identificador que desee el alumno (un carácter, color, imagen,…) que se indica cuando el visitante se registra en el parque.

El mapa se cargará de una base de datos al arrancar la aplicación central. El número de atracciones y su posición estarán igualmente indicadas en dicha base de datos.

El mapa representa una geometría esférica de manera que las posiciones más al este (límite derecha del mapa) están conectadas con las situadas al oeste y viceversa, de la misma forma que las situadas en el límite norte (zona superior del mapa) conectan con el lado sur y viceversa.

El aforo del parque estará limitado mediante un parámetro de configuración. Caso que el aforo se haya superado los nuevos visitantes no podrán entrar al parque.

********** FUN WITH QUEUES RESORT ACTIVITY MAP *********

ID	Nombre	Pos.	Destino
#	Mike	12,15	2,2
#	Nemo	14,5	15,12
#	Luca	5,3	1,18

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1																				
2		30										12								
3								14												
4																				
5																				
6									10								90			
7																				
8				20																
9													38							
10																			120	
11																				
12															26					
13									64										5	
14																				
15					96															
16																				
17															20					
18	8				23															
19																			45	
20																				

Figura 1: Ejemplo de representación del mapa del parque.

Visitantes

Los visitantes del parque disponen de una aplicación (que simularemos mediante una aplicación de escritorio) la cual emitirá periódicamente (cada segundo) su posición al sistema central de control del parque.

Al entrar al parque y registrarse en la aplicación elegirán una atracción al azar de entre las que el tiempo de espera sea menor de 60 minutos y dirigirán sus pasos hacia ella.

El tiempo de espera en las atracciones cambia dinámicamente en tiempo real por lo que, si mientras se dirige a una atracción, este tiempo supera los 60 minutos, el visitante deberá reconducir sus pasos a otra atracción que tenga menos afluencia.

Simularemos la geolocalización del visitante mediante las coordenadas de la matriz que representa el mapa.

Los visitantes podrán moverse en cualquier dirección: N,S,W,E,NW,NE,SW,SE. Cada segundo el visitante dará un paso en dirección a la atracción elegida.

Una vez que el visitante ha llegado a una atracción deberá esperar un tiempo simulando el tiempo que tarda en completar la atracción y luego volverá a elegir otra atracción al azar dirigiendo sus pasos hacia ella.

Las funcionalidades que entregará la aplicación del visitante son:

1- **Registro del visitante:** antes de poder entrar al parque los visitantes deben registrarse en la aplicación. Un visitante no podrá empezar a usar el resto de las funcionalidades de la aplicación hasta que no se haya registrado.

2- Visualización del mapa y actividad del parque: una vez registrados los visitantes podrán entrar al parque y ver toda la actividad del parque incluidos los movimientos del resto de los visitantes. De esta forma en caso de que el tiempo de espera en una atracción sea superior a 60 minutos, el visitante cambiará su rumbo a otra atracción. Dado que el aforo del parque está limitado, si tras haberse registrado en la aplicación, un visitante intenta entrar al parque con el aforo superado, el sistema no le permitirá el acceso al mismo debiendo esperar a que otro visitante salga del parque.

3- **Enviar al módulo central los movimientos del visitante:** Los visitantes pueden moverse líbremente en cualquier dirección: N,S,W,E,NW,NE,SW,SE. Para simular el movimiento del visitante, la aplicación decidirá cada segundo el siguiente movimiento que aproxime al visitante a la atracción destino y lo enviará al módulo central.

Tiempos de espera

El parque dispone de una serie de sensores en las colas de las atracciones que miden la longitud de estas, contando el número de personas que acceden y salen de la atracción.

Estos sensores envían dicho dato a un módulo externo al sistema central (denominado WaitingTimeServer) cada vez que hay una alteración (un nuevo contaje) en dicho sensor. Con

dicho dato y siendo conocido tanto el número de personas que entran en cada ciclo en la atracción, así como el tiempo de cada ciclo, una sencilla lógica en el módulo WaitingTimeServer es capaz predecir los tiempos de espera en cada atracción que serán leídos por el sistema central.

Funcionamiento de la operación del parque

En el momento que el parque abre sus puertas, el sistema central que lo gobierna entrará en funcionamiento realizando las siguientes acciones:

1- Permitirá que los visitantes se registren y accedan al parque hasta alcanzar el aforo completo.
2- Verificará que el visitante está registrado antes de permitirle su acceso y de supervisar sus movimientos.
3- Accederá al módulo de tiempo de espera periódicamente (cada X segundos) para obtener el tiempo de espera de cada atracción.
4- Obtendrá los movimientos de todos los visitantes que hayan entrado al parque.
5- Retransmitirá el mapa completo a las aplicaciones de todos los visitantes y a los distintos canales donde se pueda visualizar el mapa.

Los visitantes irán entrando y saliendo del parque indiscriminadamente.

IMPORTANTE: El funcionamiento de la solución será totalmente autonómo, es decir, una vez arranquen los distintos módulos, todo sucederá de forma automática y no habrá más interacción con la misma salvo para encender o apagar módulos (ej.: sacar o introducir a un visitante del parque) . Todo se visualizará en las pantallas de los visitantes y del sistema central.

Diseño técnico

Se propone al alumno implementar un sistema distribuido compuesto al menos de los siguientes módulos y con la siguiente arquitectura:

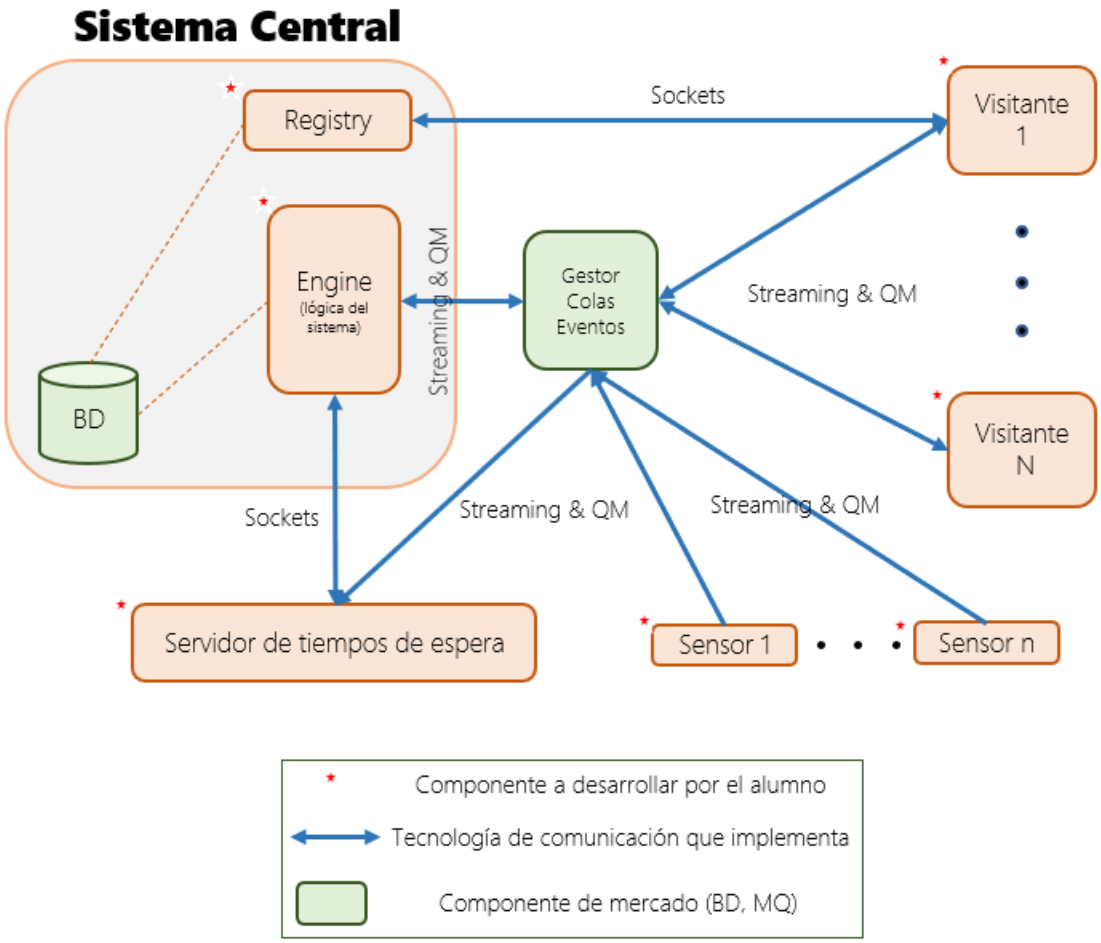

Figura 1: Esquema conceptual de la arquitectura, interconexiones entre los componentes y tipos de interfaces.

Los componentes software que el alumno ha de desarrollar son los siguientes:

- Sistema central: Módulos centrales de la solución
  - Registry: Módulo para la gestión de cuentas de visitantes
  - Engine: Módulo que implementa la lógica de gestión del parque.

- Visitante: Aplicación del visitante.
- Servidor de tiempos de espera: Aplicación que devuelve el tiempo de espera que existe en una determinada atracción.
- Sensor: Dispositivo (simulado por una aplicación) que cuenta y envía el número de visitantes de cada atracción.

Los componentes pueden ser desarrollados en el lenguaje de preferencia del alumno: Java, C/C++, . NET, Python, etc. asumiendo el alumno la responsabilidad de su conocimiento y forma de desplegarlo en el laboratorio.

A continuación, se especifica más detalladamente cada componente.

Sistema central

Contiene todos los módulos que implementan la estructura principal del juego.

Engine:

Se trata de la aplicación que implementará la lógica fundamental del parque. El nombre de la aplicación será **obligatoriamente "FWQ_Engine".** Para su ejecución recibirá por la línea de parámetros los siguientes argumentos:

- IP y puerto del broker/Bootstrap-server del gestor de colas
- Número máximo de visitantes
- IP y puerto del FWQ_WatingTimeServer

Al arrancar la aplicación irá recibiendo los eventos de movimientos del gestor de colas y se conectará al servidor de tiempos de espera cada X segundos para actualizar los tiempos de espera de las atracciones.

Para mayor sencillez de implementación, ante cada movimiento de cada visitante le responderá con todo el tablero.

Los alumnos podrán decidir la forma de expresar el mapa, aunque se recomienda un array de bytes donde cada elemento de la matriz de bytes es una posición en del mapa.

Registry:

Se trata de la aplicación que implementará el registro de losv isitantes. El nombre de la aplicación será **obligatoriamente "FWQ_Registry".** Para su ejecución recibirá por la línea de parámetros los siguientes argumentos:

- Puerto de escucha

La aplicación permanecerá a la espera hasta recibir una solicitud de registro de un jugador. Los datos que puede guardar de un jugador son:

- Alias/ID
- Nombre
- Password

Bases de Datos

Contendrán el mapa de partida del parque, los datos de los visitantes, sus credenciales de acceso, el número de visitantes que caben en cada ciclo de la atracción y el tiempo de ciclo, así como cualquier otro parámetro de configuración o dato que los alumnos consideren necesario para la resolución de la práctica.

Servirán de recursos compartidos entre los distintos elementos que forman el sistema central del parque allí donde aplique dicha compartición tal y como se muestra en la figura 1.

Los alumnos podrán decidir el motor de BD a implementar recomendando los profesores los siguientes: SQlite, MySQL, SQLServer o MongoDB. Igualmente será posible usar simples ficheros para la implementación.

Visitantes

Aplicación que implementa la funcionalidad del jugador. El nombre de la aplicación será **obligatoriamente "FWQ_Visitor".** Para su ejecución recibirá por la línea de parámetros los siguientes argumentos:

- IP y Puerto del FWQ_Registry
- IP y Puerto del Broker/Bootstrap-server del gestor de colas

La aplicación tendrá las siguientes opciones:

- Crear perfil: Se conectará al módulo FWQ_Registry para crear un nuevo usuario.
- Editar perfil: Se conectará al módulo FWQ_Registry para editar o actualizar el perfil de un usuario existente.
- Entrar al parque: Se solicitará el Alias y el password del usuario. Una vez comprobado el aforo por parte del módulo FWQ_Engine, se le permitirá el acceso al parque. El mapa irá mostrándose en pantalla conforme el visitante se mueve. Las respuestas por parte del servidor podrán ser:
  - El mapa actualizado
  - Parque cerrado
  - Cualquier otro mensaje que el alumno entienda necesario para implementar su práctica.
- Salir del parque: Si un visitante abandona el parque, este dejará de aparecer en el mapa. Si la aplicación del visitante se cierra de forma imprevista tendrá que desaparecer igualmente.

Se podrán instanciar tantos visitantes como se deseen.

Gestor de colas y streaming de eventos

Será implementado mediante la tecnología Apache Kafka.

Se implementarán los topics , productores y consumidores necesarios para resolver los requerimientos propuestos.

Sensores

Dispositivos que simularemos mediante una aplicación de escritorio que se denominará **obligatoriamente "FWQ_Sensor"**. Esta aplicación enviará cada cierto tiempo aleatorio (entre 1 y 3 segundos) al servidor de tiempos, el número de personas que se encuentra en la cola de la atracción en el que dicho sensor está ubicado.

Para su ejecución recibirá por la línea de parámetros los siguientes argumentos:

- IP y Puerto del Broker/Bootstrap-servers del gestor de colas.
- ID de la atracción en la que está ubicado.

Se podrán instanciar tantos sensores como se deseen.

Servidor de tiempos de espera

Aplicación que implementa la funcionalidad necesaria para conocer el tiempo de espera de cada atracción. El nombre de la aplicación será **obligatoriamente "FWQ_WaitingTimeServer".** Para su ejecución recibirá por la línea de parámetros los siguientes argumentos:

- Puerto de escucha
- IP y Puerto del Broker/Bootstrap-servers del gestor de colas

Esta aplicación irá recibiendo de los sensores (a través del gestor de colas) el número de visitantes que hay en las colas de cada atracción.

Por otro lado, permanecerá a la escucha indefinidamente esperando a que la aplicación FWQ_Engine le solicite los tiempos de espera de todas las atracciones.

La aplicación dispondrá de una BD (un fichero si así se desea) con los datos de las atracciones (ID, tiempo de ciclo, número de visitantes en cada ciclo) para poder realizar el cálculo del tiempo de espera.

Aclaraciones finales

Antes incongruencias funcionales o aspectos no considerados en los anteriores apartados que impidan la correcta implementación de la práctica, los alumnos deberán implementar la alternativa que entiendan más conveniente previa consulta con el profesor para evitar un excesivo nivel de complejidad.

Guía mínima de despliegue

Para la correcta evaluación de la práctica es necesario comprobar que la aplicación distribuida solicitada es desplegada en un entorno verdaderamente distribuido. Es por ello que para su prueba es necesario al menos 3 PCs distintos en los que se desplegarán los componentes solicitados. Al menos se ha de desplegar junto con el núcleo y el servidor de clima, 2 jugadores, proporcionando el siguiente escenario:

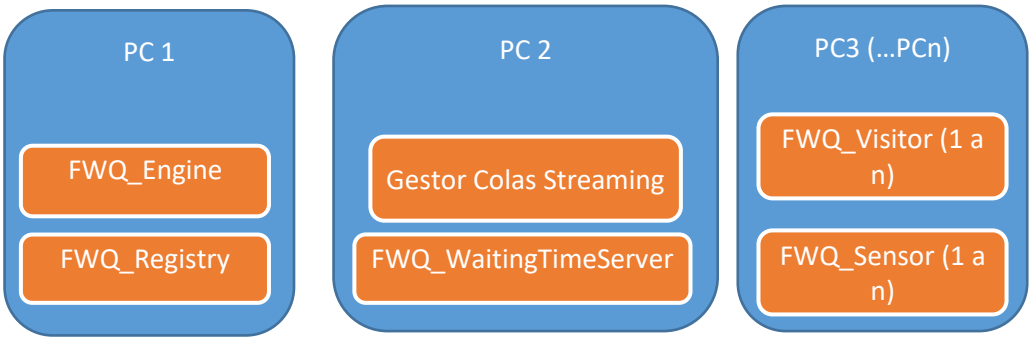

Figura 4. Escenario físico mínimo para el despliegue de la práctica.

Entregables y evaluación

La evaluación de la práctica se realizará en los laboratorios. **Se podrá realizar en grupos de hasta 2 personas repartiéndose el trabajo a su criterio sin perjuicio de que, durante el momento de la corrección, el profesor pueda preguntar a cada alumno del grupo por aspectos de cualquiera de los módulos**. Los alumnos deben desplegar por ellos mismos la práctica que resuelve el enunciado anterior. Deben desplegar un sistema completo, con todos los módulos, todos ellos interconectados entre sí. **Este requisito es indispensable para poder realizar la corrección**. Además, deben poderse evaluar positiva o negativamente todos los apartados que aparecerán en la Guía de corrección que se entregará a tal propósito. Cada uno de los apartados puntúa de forma variable, por tanto, cada apartado no implementado o que no pueda comprobarse su correcto funcionamiento no podrá ser tenido en cuenta y por tanto no puntuara. Los alumnos deberán presentar para la evaluación el documento **"Guía de corrección"** cumplimentado para que el profesor pueda validar los apartados implementados.

Los alumnos deberán entregar, además, mediante la funcionalidad de evaluación del UACloud antes de la fecha establecida a su profesor de prácticas una **memoria de prácticas**, con el código fuente y compilados generados, así como un documento donde se detalle la siguiente

información. El formato es libre, pero debe ser un documento ordenado y debidamente formateado, cuidando la redacción y ortografía.

- Portada con los nombres, apellidos y DNI de los alumnos, año académico y el título de la práctica.
- Un informe donde se indique el nombre de los componentes software desarrollados y una descripción de cada uno de ellos, explicando y enviando además el código fuente de todos ellos.
- El detalle, paso a paso, de una guía de despliegue de la aplicación, que deberá ser la misma que utilice cuando haga la corrección de la práctica.
- Capturas de pantalla que muestren el funcionamiento de las distintas aplicaciones conectadas.

Cada profesor de prácticas podrá solicitar a los alumnos cualquier otra evidencia que el profesor considere adecuada para poder formalizar la evaluación.

La fecha de entrega será en la semana del 25/10/2021.