

Guía de corrección

| | |
|----------------------------|----------------------------|
| Nombre de estudiantes: | |
| Nombre profesor prácticas: | |
| Turno: | |
| Nota final: | A rellenar por el profesor |

***** **LEER DETENIDAMENTE** *****

Para la corrección de las prácticas, estos son los apartados y valoraciones que se tendrán en cuenta. El estudiante ***deberá entregar este documento en papel*** cumplimentado el día de la corrección, a falta de que el profesor valide los apartados desarrollados. **Para poder realizar la evaluación, el día de la corrección, el estudiante deberá desplegar en los laboratorios el escenario indicado en la especificación de la práctica.** Este escenario debe ser puesto en marcha delante del profesor lanzando cada uno de los procesos y componentes necesarios, coincidiendo con los pasos detallados en la memoria que cada estudiante debe entregar.

| Concepto a evaluar | A | P |
|---|---|---|
| Despliegue, modularidad y escalabilidad (2 puntos) . | | |
| El sistema se despliega correctamente según se determina en la especificación de la práctica y se ha comentado en clase y sin necesidad de usar los entornos de compilación para su corrección. | | |
| Es posible desplegar tantas instancias del mismo módulo como se requiera en distintas máquinas o en la misma a criterio del profesor. | | |
| Funcionamiento base (hasta 2 puntos) | | |
| Se pueden parametrizar distintos aspectos de la solución en cada uno de los módulos evitando que dichos parámetros se encuentren definidos de forma fija en el código. Así será posible indicar el API_KEY del OpenWeather en un archivo (o menú a tal propósito). De la misma forma la ciudad de la que se solicita el clima podrá ser cambiada a voluntad en tiempo de ejecución en el módulo CTC para validar los requerimientos expresados en el enunciado. | | |
| La aplicación no falla durante el transcurso normal de su ejecución mostrando todas las figuras encontradas en el archivo json. | | |
| El módulo Front se puede invocar tantas veces como se desee simultáneamente desde cualquier navegador de manera que el estado global del servicio (mapa con localizaciones, clima, movimiento de los taxis, clientes, mensajes de estado, errores, ...) pueda ser visualizable desde varias pantallas a la vez. | | |
| El API REST expone la información de la auditoría de seguridad y se muestra a través de un Front Web en un navegador web (se valorará la actualización del cliente web en tiempo real) | | |

| Seguridad (hasta 2 puntos) | | |
|--|--|--|
| Se ha incorporado la seguridad entre el Registry y los taxis con la protección adecuada del canal. El API expuesto por el Registry estará disponible para ser consumido desde cualquier aplicación externa que se autentique correctamente (Taxi, Postman,...) | | |
| Se ha incorporado la seguridad adecuada entre la central y los taxis y la autenticación con tokens caducables. Las claves de cifrado no residen en el código. | | |
| Se ha incorporado el cifrado de mensajes que se publican y consumen en los Topics mediante cifrado simétrico . Las claves de cifrado no residen en el código. Las claves simétricas son distintas para cada taxi. Un cambio en una clave simétrica de solo en uno de los extremos provocará una incidencia de comunicación del taxi afectado con la central. El error se mostrará en todos los interfaces oportunos (taxi, front, central). El sistema se recuperará cuando las claves vuelvan a ser correctas. | | |
| El sistema dispone del registro de auditoría solicitado en el documento de especificación. | | |
| Resiliencia (hasta 2 puntos) | | |
| Cualquier fallo en cualquier componente solo invalida el servicio proporcionado por ese componente. El resto de los componentes del sistema pueden seguir con su operativa normal salvo en lo que se vea afectado por el componente caído. Así, por ejemplo, si el Front o el API_Central no funciona, la operación del servicio puede continuar sin problemas. | | |
| El sistema se recupera de forma correcta cuando se restituye el servicio de cualquier componente caído debiendo reiniciarse el mínimo número de módulos el sistema. Se mostrarán los mensajes de restauración del sistema en los distintos módulos. | | |
| Se capturan los errores derivados de cualquier caída y se muestran al usuario de forma controlada en todos los módulos afectados incluido el Front. Ej.: Si el CTC deja de estar disponible por cualquier problema, la central lo detectará mostrando un error lo más específico posible. Entre otros, algunos ejemplos no exhaustivos de errores podrían ser: Visualizables en Central y Front: "Imposible conectar con Taxi X. Mensajes no comprensibles". " Taxi X no disponible. Taxi fuera de servicio." "Imposible acceder al clima. Conexión con Openweather no disponible". Etc ... Visualizables en Taxi: "Imposible conectar con Central" "Registry no responde" "Mensajes con central no comprensibles" Etc ... | | |
| General (hasta 2 puntos) | | |
| Entrega de la memoria (Informe de desarrollo, detalle de despliegue, resultados, formato y corrección). | | |
| El estudiante es capaz de explicar el código fuente y responder a las preguntas que el profesor haga a tal propósito. (Este requerimiento es indispensable para poder ser evaluado) | | |
| Diseño, confort, interfaces, acabado profesional. | | |
| Otros aspectos reseñables por el alumno. | | |

Nota: Es obligatorio desplegar todo el escenario, no se corregirán componentes independientes. LA PRÁCTICA NO ES EVALUABLE si no se logra desplegar el escenario mínimo en, al menos, 3 computadoras distintas con todos los componentes conectados e

interactuando adecuadamente. Las posibilidades de despliegue válidas serán similares a las ya expresadas para la primera parte de la práctica.

Las opciones de despliegue y su correspondiente valoración será el siguiente:

1- En máquinas del laboratorio. Opción mejor valorada.

2- En una red con portátiles propios: En caso de que la opción anterior no funcione se acepta la creación de una red con portátiles pero siempre con el criterio de despliegue distribuido. Sería menos valorado (a criterio del profesor puede restar algún punto) pero no impide su presentación ni buena evaluación.

3- En un portátil con máquinas virtuales (o dockerizando) en diferentes redes dentro de un solo equipo.

4- En un portátil en modo local host: Será evaluable, pero con una penalización importante (a criterio del profesor).

En todos los casos se puede y será valorado el uso dockers realizando diferentes contenedores con las distintas aplicaciones separadas para distribuirlas en las máquinas.

Importante: Un despliegue incorrecto puede penalizar el resto de los apartados de la corrección.