

Escuela de Ingeniería		
Obligatorio de: Programación de Redes	Código de materia: 4864	
Fecha: Mayo 2024	Grupo: Todos	Hoja 1 de 4

OBJETIVOS

Poner en práctica los conocimientos adquiridos en la tercera parte del curso:

- Manejo de MOM, RPC y APIs REST

DESCRIPCIÓN DEL SISTEMA: SEGUNDA UPGRADE AL SISTEMA TRIPORTUNITY

Después del éxito del lanzamiento de la aplicación Triportunity, el cliente le pide nuevas funcionalidades de monitoreo. Además, dado que el sistema de servidor está construido utilizando tecnologías de capa de transporte y esto se vuelve complejo de mantener, se solicita al alumno que agregue la infraestructura necesaria para realizar la administración del sistema de forma remota. **Esto involucra el desarrollo de nuevas aplicaciones y la modificación del servidor existente.**

ARQUITECTURA Y REQUERIMIENTOS FUNCIONALES

El sistema consta de al menos cuatro módulos de software:

- *Servidor.*
- *Cliente.*
- Servidor de Estadísticas
- Servidor Administrativo.

REQUERIMIENTOS FUNCIONALES

- Aplicación **Servidor de Estadísticas.**
 - SERF1. **Recepción de viajes desde el servidor principal.** El servidor de estadísticas debe recibir y almacenar los eventos de viajes generados en el servidor principal.
 - SERF2. **Filtrado de viajes .** El servidor de estadísticas debe ser capaz de filtrar los viajes (Por ejemplos por precio, destino u otras opciones elegidas por ustedes). Deben seleccionar al menos 3 criterios y se deben poder combinar.
 - SERF3. **Recepción de eventos de usuarios desde el servidor principal.** El servidor de estadísticas debe recibir los eventos de inicio de sesión de usuarios generados en el servidor principal.
 - SERF4. **Estadísticas de Usuarios.** El servidor de estadísticas debe ser capaz de mostrar la cantidad de usuarios que iniciaron sesión.
 - SERF5. **Reporte.** El usuario del servidor de estadísticas puede:
 - Pedir generar un reporte con la información resumida de los **siguientes N** viajes (siendo N elección del usuario).
 - Consultar el estado de ese reporte (si está pronto o no)
 - Una vez que el reporte esté pronto puede consultarlo.
 - SERF6. **Acceso remoto.** El servidor de estadísticas debe exponer un servicio que permita acceder a las funciones anteriores de manera remota. (No por consola)

Escuela de Ingeniería		
Obligatorio de: Programación de Redes	Código de materia: 4864	
Fecha: Mayo 2024	Grupo: Todos	Hoja 2 de 4

- Aplicación **Servidor Administrativo**.
 - SARF1. **ABM de Viajes**. Se deberá dar de alta, baja o modificación de viajes (sin necesidad de subir una imagen) tomando en cuenta las restricciones ya existentes para los obligatorios anteriores.
 - SARF2. **Consultar Calificaciones de un viaje**. Se deberá poder consultar las calificaciones de un viaje.
 - SARF3. **Ver próximos N viajes**. El usuario puede pedir imprimir en pantalla los siguientes N viajes (N a elección del usuario) en tiempo real. En el momento que se genera un viaje, se debe imprimir en la consola del usuario administrativo, hasta que se impriman los N viajes.
 - SARF4. **Consola**. La interfaz del servidor administrativo puede ser una aplicación de consola. Debe soportar varios usuarios administrativos simultáneos.
- Aplicación **Cliente y Servidor**.

Se deben seguir cumpliendo los requerimientos funcionales de las entregas anteriores.

ESPECIFICACIÓN DE TECNOLOGÍAS

1. Se debe utilizar gRPC, RabbitMQ y REST API
2. Que aplicación usa que tecnología depende del equipo, pudiéndose distribuir las mismas como el equipo entienda mejor, siendo obligatorio utilizar **las tres**.

DOCUMENTACIÓN

Como documentación se espera que se incluya las modificaciones realizadas al código y la motivación de estas así también como la arquitectura principal (qué módulos o componentes tiene el sistema y en qué tecnologías fueron desarrollados y porque se eligió dicha tecnología).

Se tomarán en cuenta los siguientes puntos para la corrección de la documentación:

- Descripción de la arquitectura.
- Documentación de diseño detallada de cada componente.
- Justificación/motivación de mecanismos de comunicación de los componentes de su solución
- Justificación de las decisiones importantes de diseño e implementación.
- Prolijidad, completitud y organización del documento
- Correcta utilización de diagramas (deben utilizarse para aportar claridad a lo documentando, pueden realizar los diagramas que consideren necesarios o útiles)
- Descripción de archivos de configuración e información relevante sobre la ejecución de la solución
- Supuestos que se hayan tomado

- o Cambios que consideren relevantes sobre requerimientos anteriores
- o Detalle de errores encontrados y posibles mejoras

ENTREGA

Se debe entregar por Gestión un archivo PDF que contenga la documentación y en la primera página el link al repositorio con el código.

El repositorio debe incluir una **colección de Postman** o similar (aclarar en la documentación) para probar las acciones disponibles que expone el servidor de estadísticas.

Deben usar el mismo repositorio que la entrega anterior el cual debe pertenecer a la organización de GitHub "ArqSis-PDR-2024-1".

Al realizar la entrega se debe realizar un release en el repositorio y la rama master (o main) no debe ser afectada luego del hito que corresponde a la entrega del obligatorio.

Deben incluir en el repositorio las aplicaciones compiladas en *release*, con ejecutables para Windows (.exe, .dll, etc.) y todo lo necesario para poder ejecutarlas (archivos de configuración, etc.)

Opcionalmente pueden incluir los archivos Dockerfile y docker-compose.yml, junto a una guía, para ejecutar las aplicaciones utilizando Docker.

Datos de prueba (opcional pero recomendado)

Entregar el sistema con una fuente de datos de prueba propia para facilitar la corrección.

EVALUACIÓN

El criterio para la corrección se basará en los siguientes aspectos:

- Deberán **evitar el uso de la clase Thread** para el desarrollo del obligatorio.
- Para esta entrega **se permitirá** que se utilicen parsers existentes (como puede ser Json) para serializar objetos.
- En la defensa se correrá un plan de testing, donde se evaluará cuantos puntos pasa el sistema.
- Criterios de Diseño.
- Documentación presentada.
- Correcta utilización de archivos de configuración
- Porcentaje de cumplimiento con los requerimientos presentados.
- Funcionamiento de la aplicación.
- Calidad de diseño e implementación.

Escuela de Ingeniería		
Obligatorio de: Programación de Redes	Código de materia: 4864	
Fecha: Mayo 2024	Grupo: Todos	Hoja 4 de 4

DEFENSA

La defensa del trabajo intenta:

- Evaluar el conocimiento general de los integrantes del grupo sobre la solución propuesta. Todos los integrantes deben conocer toda la solución.
- Evaluar el aporte individual al trabajo por parte de cada uno de los integrantes del equipo. Se espera que como mínimo cada uno de los integrantes haya participado en la codificación de al menos alguna parte significativa del obligatorio.

Información importante

Defensas: a coordinar.

Plazo máximo de entrega: 17/06/2024

Puntaje mínimo/máximo: 5/25 puntos.

Los obligatorios se forman como máximo por grupos de 3 estudiantes.

Todas las entregas se realizan en **gestión** (gestion.ort.edu.uy) y hasta las **21.00** hs del día de entrega.