

## PROYECTO II – Emotion ANALYSIS in SOA

Instituto Tecnológico de Costa Rica  
Área de Ingeniería en Computadores  
Primer Semestre 2022  
Valor 20%



### OBJETIVO GENERAL

- Diseñar y construir una aplicación utilizando los principios básicos del diseño orientado a servicios, utilizando la arquitectura orientada a eventos.

### OBJETIVOS ESPECÍFICOS

- Diseñar la arquitectura de software necesaria orientada a servicios
- Implementar la aplicación orientada a servicios en un ambiente utilizando contenedores

### ENTREGA:

- Minikube instalado y con hello-world: Jueves 5 de Mayo
- Entrega del primer diseño: 10 Mayo
- Defensa: 17 de mayo, 2022

### DESCRIPCIÓN DEL PROBLEMA

El departamento de recursos humanos de su empresa, está interesado en construir un sistema de software que le permita analizar los sentimientos de los empleados de la empresa al recibir buenas noticias. La idea es eventualmente utilizar la herramienta para comunicar mejor los resultados mensuales de ganancias de la compañía.

Al final de cada día laboral, 10 imágenes de empleados deben ser analizadas, y el resumen del análisis debe ser agregado a un archivo local común, el cual se comparte entre los servicios, y es donde se guardan todos los estados.

El sistema debe correr en forma de contenedores, conteniendo un servicio cada uno (al menos cuatro servicios deben existir en la aplicación, por lo que se aconseja revisar los principios SOLID) y los mensajes entre ellos deben ser asíncronos, se aconseja utilizar un contenedor corriendo como Service Broker.

El sistema puede manejarse como contenedores individuales corriendo en docker en la computadora del desarrollador. Sin embargo, se aconseja utilizar algún orquestador de contenedores como Minikube o una versión local de Kubernetes.

El servicio que identifica la emoción o sentimiento de las imágenes puede ser tanto un servicio local corriendo en un contenedor, como un servicio de la nube utilizado *as a service*

### EN SU ROL DE ARQUITECTO

1. Diseñe la arquitectura orientada a servicios siguiendo los principios SOLID y documente la justificación de cada uno de ellos en su diseño.
2. Documente la funcionalidad de cada uno de los servicios involucrados.
3. Diseñe la arquitectura orientada a eventos y cree el diagrama de componentes.
4. Discuta y defienda su diseño para el 10 de mayo.

### EN SU ROL DE DESARROLLADOR Y OPERACIONES

- ☐ Diseñe y construya el código necesario para cada uno de los servicios.
- ☐ Integre el código en contenedores.
- ☐ Utilice los principios de código limpio. Puede revisar su código con Lint y asegurarse todos los puntos en calidad del código.
- ☐ Implemente y despliegue el código de manera local.
- ☐ Se permite el uso de imágenes de docker públicas en internet.

### ASPECTOS OPERATIVOS Y EVALUACIÓN

1. El proyecto es en grupos de trabajo de máximo 3 personas.
2. La fecha de entrega es según lo que se indique en el cronograma del curso
3. Rúbrica:

ROL DE ARQUITECTO DE SOFTWARE	
Documentación de proceso de diseño	10
Justificación de las decisiones de diseño	10
Diseño inicial de la arquitectura (Arquitectura PRESCRIPTIVA)	15
Diseño final de la arquitectura (Arquitectura DESCRIPTIVA)	15
ROL DE DESARROLLADOR	
Calidad del código	10
Integración del sistema	10

Funcionalidad	20
Service Broker y/o Servicios de mensajería	10
	100

### OTRA REGLAS

1. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
2. Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la primera cita de revisión oficial.
3. La revisión de la documentación será realizada por parte del profesor, no durante la defensa del proyecto.
4. Cada grupo tendrá como máximo 30 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
5. Durante la revisión únicamente podrán participar los estudiantes, asistentes, otros profesores y el coordinador del área.
6. Los estudiantes deberán mostrar un avance del proyecto por semana, esto se llevará a cabo durante la clase a modo "Standup"

### Referencias:

Minikube: <https://minikube.sigs.k8s.io/docs/start/>

RabbitMQ: [https://hub.docker.com/\\_/rabbitmq](https://hub.docker.com/_/rabbitmq) (por si se elige la arquitectura con un broker)

### Referencias no oficiales:

<https://github.com/johnpaulada/emotion-detection-docker>