

REBRICA DEL PROYECTO FINAL DE CURSO

Sección	:	Única -CURS-000385
Curso	:	Docker & Docker Compose for Developer
Instructor	:	Aristedes Novoa Arbildo
Horario	:	Martes y Jueves de 9:00pm a 11:00pm
Alumno	:	

I. CONSIDERACIONES GENERALES

- El desarrollo de los casos de aplicación es personal, sin embargo, está permitido colaborar con sus compañeros de clase o la asesoría de cualquier otro profesional con experiencia en Docker y Docker Compose, Podman u otras tecnologías relacionadas.
- Está permitido usar libros, tutoriales, scripts y presentaciones para revisar y repasar conceptos y casos a reutilizar.
- Reutilizar proyectos previos desarrollados en clase o descargados de internet, incluyendo imágenes, scripts, configuraciones y código fuente de aplicaciones, APIs o servicios; siempre que no violen la propiedad de derecho del autor. Mantener los créditos del autor según corresponda (nombre, datos de contacto, repositorio).

II. ACTIVIDADES A REALIZAR

- Seleccionar el **código fuente** de Aplicaciones, APIS REST o servicios disponibles en línea o desarrollos por usted mismo en **cualquier lenguaje** de programación (Java, Net, Node.js, Python, Php, Go, Angular, React.js, Vue.js, etc).
- Crear un repositorio en github, gitlab o cualquier otra plataforma de gestión de código fuente para versionar el código, comandos, scripts y configuraciones.
- Los sustentos de los casos deben desarrollarse utilizando pantallazos, código fuente, archivos de configuración, diagramas o presentaciones según corresponda.
- El examen, debe incluir los siguientes casos, considerando entre los más importantes los siguientes (cada componente vale 2 puntos):

#	Componente	Consideraciones	Cumple ¹		Comentarios de revisión
			Si	No	
1	Docker Sheet Personalizado	Crear un Docker Cheat Sheet(Hoja de ayuda, ppt, txt, md, etc.) personalizado de Docker y Docker Compose con los principales comandos utilizados en el desarrollo de sus casos , incluyendo aquellos que estime relevantes para utilizarlo en su trabajo u otros escenarios.	2		
2	Contenedores simples	Crear y gestionar contenedores en base a imágenes disponibles en repositorios públicos (Docker Hub, quay.io). Aplicar los comandos de creación , inspección, logs , renombrar y eliminación .	0		
3	Conectando tres Contenedores	Seleccionar imágenes que permitan crear contenedores e intégralos utilizando sus IPs internos (usar inspect). Sugerimos utilizar imágenes de bases de datos y GUIs, stacks tecnológicos como Kafka/RabbitMQ, ELK/EFK, etc. Los casos deben usar imágenes diferentes a los desarrollados en clase.	2		
4	Redes y conectividad	Crear redes y gestionar redes personalizadas. Aplicar los comandos de creación, inspección, renombrar y eliminación. Conectar/desconectar contenedores ya creados, y también utilizar redes al momento de su creación.	2		
5	Creación de Imágenes	Crear imágenes de aplicaciones , APIs o servicios desarrollados en algún lenguaje de programación (Java , Net, Node.js , Python, Php, Go, etc, Angular , React.js). Utilizar Dockerfile y Dockerignore personalizados, se deberán crear contenedores de dichas imágenes y verificar su funcionamiento.			

6	Registro de Imágenes	Publicar sus imágenes creadas por usted en dos o más repositorios públicos y privados. Utilizar Docker Hub y una nube específica (AWS, Azure, Google, IBM Cloud, Oracle Cloud, u otras). También se deberá realizar la descarga de imágenes, crear contenedores y verificar su funcionamiento.	1		
7	Docker Compose	Crear un stack de servicios relacionados, considerando como mínimo cuatro imágenes distintas creadas por terceros y usted mismo; debe incluir cinco contenedores(uno de ellos debe exponerse en diferentes puertos).	0		
8	Docker Compose con Build	Crear un stack de servicios relacionados, considerando la creación de imágenes en caliente (uso de Dockerfile y context). Deber realizarse cualquier cambio en el código fuente de la aplicación, API o servicios y generar los contenedores para verificar los cambios.			
9	Secrets y Volumen	Crear y utilizar secrets en línea o en archivos y utilizarlos en Docker Compose o en la creación de contenedores en línea. También debe crear y utilizar volúmenes y demostrar la persistencia de datos posterior a la eliminación de los contenedores; crear nuevos contenedores y reutilizar dichos volúmenes.			
10	Otros	Otros comandos, aplicaciones o aplicaciones de Docker o Docker Compose como parte de su investigación y complemento a su proceso de entrenamiento.	2		

III. PRESENTACION

El alumno deberá realizar la presentación en vivo de sus casos con la implementación de las funcionalidades solicitadas de acuerdo a la fecha coordinada y programada por Galaxy Training. En caso que usted haya concluido con el desarrollo de su examen antes de la fecha programada puede coordinar con Galaxy Training para programar una fecha de sustentación, de igual manera si tendría algún inconveniente o retraso por temas de trabajo u otros.