



## TERCER PROYECTO

11-29 de octubre de 2021

### Indicaciones generales

- Este es un proyecto **individual**. Podrá discutir con sus compañer@s sobre posibles soluciones al problema planteado, pero no debe intercambiar líneas de código ni instrucciones específicas.
- El proyecto se publica el lunes 11 de octubre de 2021 a las 7:00 a.m. y debe entregarse a más tardar a las 11:59 p.m. del viernes 29 de octubre de 2021.
- En **e-aulas** puede acceder a las diapositivas, los enunciados de los talleres y a la sección correspondiente a este proyecto.
- La actividad en **e-aulas** se cerrará a la hora en punto acordada. La solución de la actividad debe ser subida antes de esta hora. El material entregado a través de **e-aulas** será calificado tal como está. Si ningún tipo de material es entregado por este medio, la nota de la evaluación será 0.0.

**Se aconseja subir a e-aulas versiones parciales de la solución a la actividad.**

- **Cualquier incumplimiento de lo anterior conlleva la anulación del examen.**
- Las respuestas deben estar totalmente justificadas.
- **Entrega:** 1 archivo **‘.zip’** con la solución al proyecto y el soporte visible en servicios desplegados en la nube. En este archivo incluya una carpeta por punto con los soportes de cada punto. Suba su archivo comprimido a **e-aulas**.

**Importante:** no use acentos ni deje espacios en los nombres de los archivos que cree.

1. [30 ptos.] (**Contenido**) Ud. ha decidido continuar con el desarrollo de su proyecto anterior. Con las lecciones aprendidas del proyecto 2, Ud. ahora busca hacer su portal más profesional. Para esto debe concentrar parte de su esfuerzo en dos ítems:
  - a) Mensaje/contenido: su página debe dejar de contener solamente los tableros en Dash, para ahora tener una explicación de los datos considerados, los tableros desplegados y observaciones derivadas de los mismos. Recuerde que Ud. busca que su tablero sea una fuente de información para personas del público general sobre un tema que puede ser relativamente técnico. Sea claro/a en sus explicaciones.
  - b) Diseño: así como su tablero debe contar con un mensaje y contenido claros, también es importante que sea agradable para el lector. Mejore su diseño pensando en entregar una buena experiencia de usuario. Su proyecto lo compartiremos con otras personas para que den su opinión.
  - c) Cree una instancia en AWS EC2 y despliegue allí su tablero. Asegúrese de que su aplicación sea accesible a través del puerto 80 (HTTP).

### Entregables:

- a) Por Slack: dirección IPv4 y DNS público de la instancia que aloja la página en ejecución.
- b) Por e-aulas: archivos .py y otros usados para la página. Pantallazos de la configuración de todos los servicios usados en AWS. Arquitectura de la aplicación.



2. [40 ptos.] (**Despliegue multi-capa**) En esta oportunidad queremos extraer los datos de la fuente y almacenarlos en una base de datos relacional. Para esto cree una base de datos en RDS y almacene allí la información extraída de la fuente. Deberá crear las tablas y registros necesarios. Modifique su aplicación en Dash para comunicarse con la base de datos y extraer de allí la información **estrictamente** necesaria para desplegar cada tablero. Es decir, NO realice consultas que incluyan tablas, atributos o registros diferentes a los que se muestran en cada tablero. Utilice el dominio y servicio DNS del proyecto anterior. **Entregables:**
- a) Por Slack: dirección IPv4 de la instancia que aloja el servicio web, URL del dominio, endpoint de la base de datos RDS.
  - b) Por e-aulas: pantallazos de la configuración de todos los servicios usados en AWS. Arquitectura de la aplicación.
  - c) Por e-aulas: modelo relacional de la base de datos en RDS. Scripts (.sql) de creación del modelo de datos (DDL).
  - d) Por e-aulas: script de extracción de datos de la fuente y carga en la base de datos (ETL).
  - e) Por e-aulas: archivos fuente de la aplicación e instrucciones de ejecución.
3. [30 ptos.] (**Contenedores**) En vez de usar una instancia EC2 y un servicio RDS, Ud. ha preferido montar su aplicación multi-capa en contenedores Docker. Utilice Docker Compose para definir los servicios web y de base de datos de la aplicación. Utilice Docker también para la configuración inicial de los contenedores y los servicios. Lance los contenedores en una instancia EC2 y exponga el servicio web por el puerto 80. Use imágenes de base para crear sus contenedores. **Entregables:**
- a) Por Slack: dirección IPv4 de la instancia que aloja el contenedor con el servicio web.
  - b) Por e-aulas: pantallazos de la configuración de todos los servicios usados en AWS. Arquitectura de la aplicación.
  - c) Por e-aulas: archivos de Docker y otros necesarios para configurar y lanzar los servicios web y de base de datos.
  - d) Por e-aulas: URL del contenedor publicado en DockerHub, instrucciones para descargar y ejecutar el contenedor en un host.