

Entrega 5: Deploy y consumo de la API

II-2018

Esta entrega busca conectar la aplicación que desarrollaron en PHP con la API desarrollada por ustedes en la entrega anterior. Además deberán agregar ciertas funcionalidades a su aplicación. En concreto, deberán:

- Montar la API en un servidor.
- Consumir los JSON de la API en la aplicación web en PHP.
- Hacer un dashboard para las tiendas.

Detalles Académicos

Para esta entrega cada encargado de grupo recibirá los datos de un servidor ubicado en el DCC. En este servidor deben hacer el traspaso a un entorno de producción de la API.

Introducción

Para finalizar la aplicación de la NebStore, lo único que queda es juntar ambas componentes. Por un lado, tenemos la tienda y el sistema de pagos, y por el otro el sistema de mensajería. Don N., dueño de la NebStore, está muy ansioso por la finalización del proyecto, ya que le permitirá a la gente gastar sus NebCoins.

Deploy de la Web API

En esta entrega deben consumir la API desde tu aplicación en PHP, por lo que deberás subirla a un servidor para poder accederla desde cualquier computador. Esto significa que la API desarrollada en su entrega anterior debe ser **subida a producción**¹ en el servidor

¹Para hacer deploy, puedes revisar el siguiente tutorial <https://www.digitalocean.com/community/tutorials/how-to-serve-flask-applications-with-uswgi-and-nginx-on-ubuntu-18-04>. De todas formas, hay muchas formas de hacer esto y queda a tu criterio la manera de proceder.

destinado para eso. Esta API **no** debe estar en el servidor `bases.ing.puc.cl`, sino en el nuevo que te entregamos para esta entrega.

Consumir la API desde la aplicación PHP

Una vez subida la API al servidor, debe ser posible consumirla desde la aplicación en PHP. **Una vez que un usuario inicia sesión en la aplicación en PHP**, debe ser capaz de:

- Revisar todos sus mensajes recibidos.
- Revisar todos sus mensajes enviados.
- Enviar un mensaje a un usuario en particular.
- Buscar mensajes por texto utilizando las funcionalidades de la búsqueda por texto de la entrega anterior.

Además el usuario debe ser capaz de ingresar un rango entre dos fechas. Para cada mensaje emitido por el usuario entre esas dos fechas, se debe visualizar en un mapa los lugares en que ha estado ese usuario (según la geolocalización de los mensajes). Para esto puede usar cualquier librería que permita manejar mapas en entornos web. Un ejemplo es **Leaflet**².

Importante: Toda esta parte del trabajo se realiza en `bases.ing.puc.cl`. Las rutas implementadas en la aplicación de php quedan a criterio tuyo. Cada una de estas funcionalidades no deberían estar todas en una sola página, para que no quede demasiado colapsada. No es necesario implementar autenticación del lado de la API.

Implementar un dashboard

Debe ser posible tener una página en la que se puede ingresar una tienda en particular para poder ver estadísticas sobre ella. Se espera que se pueda ver:

- La lista de los clientes ordenada según los que más dinero han gastado en esa tienda.
- Un gráfico que muestre la cantidad de dinero recibido por la empresa en cada mes.
- Alguna otra estadística que usted encuentre interesante, y que pueda ser mostrada con un gráfico.

Para esto puede usar cualquier librería que permita manejar gráficos en entornos web. Un ejemplo es **C3.js**³.

²<http://leafletjs.com/>

³<https://c3js.org/>

Detalles adicionales

Para facilitar la corrección se solicita indicar a los ayudantes en un **readme** toda la información necesaria para comprender de manera más rápida cualquier aspecto de la entrega. Es **obligación** que el **readme** contenga la información para correr la aplicación y probar cada una de las funcionalidades. Los ayudantes se reservan el derecho a descontar décimas en una entrega en la que se haya dificultado la corrección. **La entrega es para el jueves 29 de noviembre, a las 23:59**