

## Actividad 6. Aplicación Pública

Fecha de entrega: 13/12 a las 23:59

Puntaje máximo esperado: 40 pts

Puntaje entrega tarde: 30 pts

## Actividad 6. Aplicación Pública

Se deberá desarrollar **otra aplicación** que permita visualizar de manera pública la información de los centros de ayuda social. Básicamente es un portal donde las personas pueden ver la información de los centros de ayuda social registrados en el sistema y alguna información estadística. Además podrán realizar la solicitud de carga de un nuevo centro de ayuda social y obtener un turno para concurrir a un centro de ayuda.

El código de esta nueva aplicación deberá compartir espacio dentro del mismo repositorio de código de la aplicación Flask, la cual deberá desarrollarse utilizando el framework VueJs. Como requisito para el correcto funcionamiento en el servidor se deberá ubicar el código correspondiente a esta aplicación dentro del directorio **web** del proyecto.

### 4.1 Home

En la vista principal de la aplicación deberá contar con una explicación resumida del servicio que brinda el portal, y una breve explicación de las acciones que puede realizar un visitante:

- Ver centro de ayuda social en el mapa.
- Crear una solicitud de carga de un nuevo centro de ayuda social.
- Sección de estadísticas en base a información de los centros de ayuda social.

### 4.2 Mapa de centros ayuda social

En esta sección se deberá visualizar un mapa donde se encuentren los centros de ayuda social registrados y publicados desde la aplicación privada. Toda esta información deberá ser obtenida vía servicio desde la API desarrollada en el punto **1.5.2**. Cada centro de ayuda social se visualizará como un puntero en el mapa y al hacer clic sobre cada uno, se deberá observar la información de contacto del mismo (nombre, dirección, teléfono, hora de apertura - hora de cierre).

**A modo de ejemplo:**

<b>Comedor Casa del Niño Belén</b>
------------------------------------

Dirección: B1906GMG, Calle 5 Bis 1589, B1906GMG Ringuelet, Provincia de Buenos Aires  
Horario: 12:30 - 22:00  
Teléfono: -

**Se debe tener en cuenta que esta funcionalidad se debe realizar con el Framework VueJS.** Se deberá utilizar un componente que sirva para realizar esta implementación.

Librería sugerida: <https://vue2-leaflet.netlify.app/>

### 4.3 Solicitud de carga

En este apartado se brinda la posibilidad a los usuarios que realicen la solicitud de carga en el sistema de nuevos centros de ayuda social. La carga se realizará por medio de un formulario donde se solicita al usuario toda la información necesaria (ver en **1.5.1** datos necesarios para cada centro de ayuda social). Además se visualiza un mapa donde el usuario deberá colocar la ubicación geográfica del centro de ayuda social.

Para evitar el registro masivo de centros de ayuda, el formulario deberá contar con un captcha.

Librería sugerida: <https://www.google.com/recaptcha/about/>

### 4.4 Solicitud de turno

En este apartado se debe permitir crear un turno para un centro en particular en una fecha determinada, utilizando los servicios que nos provee la API de turnos (ver sección **2.2.2**). Dado un centro de ayuda seleccionado, se debe brindar la posibilidad de solicitar un turno.

El usuario debe especificar primero para que fecha espera el turno. La aplicación debe mostrarles los turnos disponibles para ese día y al seleccionar uno y completar su datos de contacto (Nombre, apellido, email y teléfono) puede realizar la solicitud.

### Consideraciones generales:

- El prototipo debe ser desarrollado utilizando Python, JavaScript, HTML5, CSS3 y MySQL, y **respetando el modelo en capas MVC**.
- El código deberá escribirse siguiendo las [guías de estilo de Python](#)
- El código Python deberá ser documentado utilizando [docstrings](#).
- **El uso de [jinja](#) como motor de plantillas es obligatorio para la aplicación privada.**
- Se debe utilizar [Flask](#) como framework de desarrollo web para la aplicación privada.
- Se deberán realizar **validaciones de los datos de entrada** tanto del lado del cliente como del lado del servidor. *Para las validaciones del lado del servidor se deben realizar en un módulo aparte* que reciba los datos de entrada y devuelva el resultado de las validaciones. En caso de fallar el controlador debe retornar la respuesta indicando el error de validación.
- Podrán utilizar librerías que facilitan algunas de las tareas que deben realizar en el trabajo como pueden ser: conexión a servicios externos, librerías de parseo, librerías

## Proyecto de Software 2020 - Trabajo Integrador (TI)

con patrones para buenas prácticas, validaciones de datos, etc. **Pero todos los miembros del equipo deben demostrar en la defensa pleno conocimiento del funcionamiento de estas librerías y una idea de cómo solucionan el problema.**

- Para la interacción con la base de datos se deberá utilizar un ORM que nos permita además tener una capa de abstracción con la BD.
- Para la aplicación pública se debe utilizar el Framework web [VueJS](#) versión 2. Se deberá utilizar la librería [vue-router](#) para implementar el ruteo de la aplicación.
- No pueden utilizar un framework/generador de código para el desarrollo de la API de Instituciones.
- Debe tener en cuenta los conceptos de Semántica Web proporcionada por HTML5 siempre y cuando sea posible con una correcta utilización de las etiquetas del lenguaje.
- El trabajo será evaluado desde el servidor de la cátedra que cada grupo deberá gestionar mediante Git. **NO se aceptarán entregas que no estén realizadas en tiempo y forma en el servidor provisto por la cátedra.**
- Deberán visualizarse los aportes de cada uno/a de los/as integrantes del grupo de trabajo.
- El/la ayudante a cargo evaluará el progreso y la participación de cada integrante mediante las consultas online y el seguimiento mediante GitLab.
- Toda vista (**HTML5** y **CSS3**) debe validar contra las especificaciones de la W3C (<http://validator.w3.org/> y <https://jigsaw.w3.org/css-validator/> respectivamente). En esta oportunidad puede utilizar **Bootstrap** u otro framework similar. En caso de que alguna de las vistas no valide, deberá realizar un breve informe indicando cuales son los errores encontrados.
- Todas las vistas deben cumplir ser web responsive y visualizarse de forma correcta en distintos dispositivos. Al menos deben contemplar 3 resoluciones distintas:
  - res < 360
  - 360 < res < 768
  - res > 768
- La entrega es obligatoria. Todos y todas los/as integrantes deben presentarse a la defensa.
- El sistema no debe ser susceptible a SQL Injection y XSS.
- **Importante:**
  - El proyecto podrá ser realizado de modo individual o en grupos de tres o cuatro integrantes (será responsabilidad de los y las estudiantes la conformación de los equipos de trabajo). Todos y todas los/as estudiantes cumplirán con la totalidad de la consigna, sin excepciones.

### Información del servidor

Tener en cuenta que el servidor de la cátedra utiliza los siguientes servicios y versiones:

- Servidor de Base de Datos: MariaDB 10.1.44
- Intérprete Python: Python 3.6.9
- Servidor web: Apache 2.4.29 (Ubuntu)