

Trabajo Práctico – Refugio de animales

**UNGS - Entrega Demo Informe - Ingenieria de Software -
Comisión 1 - Grupo 3.**

Participantes: Matías Nissero (matiasnissero@gmail.com),
Nazareno Avalos (codigorap13@gmail.com), Emiliano Aguilar
(1999kamerun@gmail.com), Leandro Aguirre
(leann.aguirre94@gmail.com).

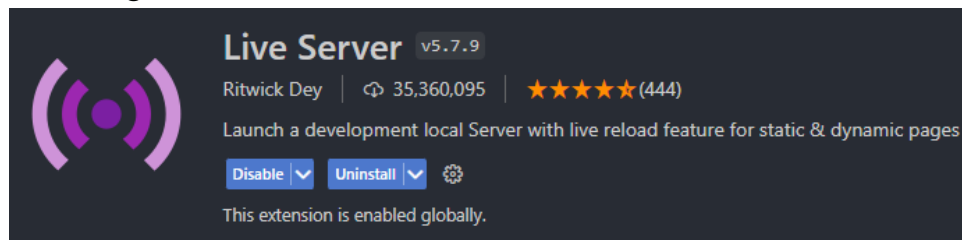
Profesores: Andres Rojas, Daniel Alvarez.

Link repositorio GitHub:

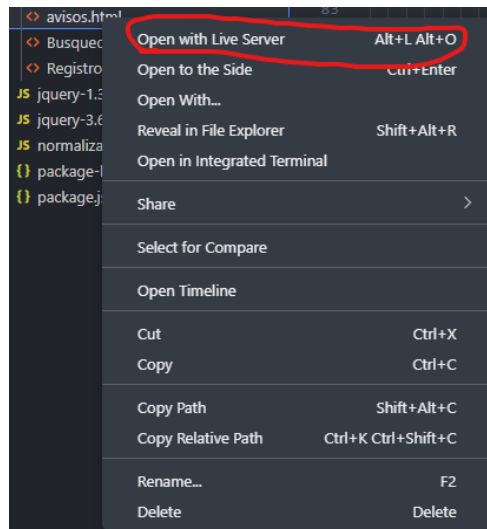
<https://github.com/nissero/Refugio-de-Animales>

Requisitos de ejecución:

1. Abrir visual Studio Code
2. Descargar e instalar la extensión “Live server”



- 3.
3. Agregar el proyecto a ejecutar en Visual Studio Code
4. Buscar en la carpeta Pages, el html a ejecutar
5. Darle click derecho y “Open with Live Server” para que corra la pagina web en tu navegador, se va a abrir automaticamente.



- 6.

Informe de código fuente

En este informe se detallan los roles y las funciones que ejecutan los distintos archivos HTML y Javascript que componen el código fuente del proyecto del portal web para un refugio de animales, a continuación se listan y describen cada uno de ellos.

aviso.html: Este documento html se compone de los avisos de animales publicados en el sitio, que se encuentran categorizados por tipo de aviso, si son de adopción o de perdido/encontrado, además también se clasifican por el tipo de animal, perros, gatos, aves, etc. Los avisos pueden buscarse fácilmente utilizando un filtro que te muestra él o los avisos deseados según cada categoría o combinación de categorías que selecciones.

Busqueda_organizacion.html: En él podemos buscar las organizaciones que están registradas en el portal web, algunas de ellas pueden tener la categoría de “destacada” ya que han contribuido con una donación al sitio, por otro lado, se incluye un filtro de búsqueda para buscar una o varias organizaciones según el tipo de servicio que ofrecen. Además se visualiza un mapa donde se muestra la ubicación de las organizaciones.

Registro_organizaciones.html: En esta sección del portal los representantes de las organizaciones podrán enviar una solicitud de registro al coordinador del sitio, para esto se muestra un formulario de inscripción donde se solicita nombre de la organización, una pequeña descripción, servicios que ofrece, dirección, entre otros. Para facilitar la correcta localización de la organización en su solicitud de registro, se incluye una función de normalización de direcciones del AMBA, al tipear se van mostrando posibles direcciones, al completar la calle y la altura se muestran todas las direcciones de distintos partidos/localidades que coincidan con los datos ingresados.

style.css y sytleBusOrg.css: Estos son archivos de hojas de estilo CSS que poseen los comandos para darle el formato visual a las distintas secciones del sitio web.

avisos.json y organizaciones.json: Poseen la información en formato JSON de los avisos y las organizaciones que están publicados en el portal.

agregarOrg.js: Mediante una función la cual captura el evento click del botón “enviar” captura y analiza los datos ingresados en el formulario.

Si están completos, los agrega a un array json y dispara una alert que fue correcta la solicitud junto con sus datos, si no esta completo brinda un alert que el formulario no esta completo.

app.js: Aquí se encuentran las funciones que leen el archivo json de los avisos, los procesa y los muestra en la sección “Avisos” del portal, además se encarga de seleccionar y mostrar tipos de avisos específicos según los filtros de búsqueda que hayan sido aplicados. También cuenta con los objetos de los avisos, entre ellos están:

- Avisos: Es un objeto abstracto, padre de los objetos avisoAdopcion y avisoPerdidoEncontrado, tiene como atributo titulo(titulo del aviso), perfilAnimal(informacion extra del animal), telefono(método de contacto), tipoAviso (tipo de aviso), img (imagen del aviso).

- AvisoAdopcion: Extiende de Aviso, es la expresión de un aviso de adopción, entre sus atributos, además de los de su padre (Avisos) contiene cuidados(cuidados extras que tiene el animal).

- avisoPerdidoEncontrado: Extiende de Aviso, es la expresión de un aviso de adopción, entre sus atributos, además de los de su padre (Avisos) contiene dirección (dirección de donde se perdió/encontró el animal) y fecha(fecha de pérdida/encuentro).

Luego, cuando se ejecuta, comienza exportando el json avisos.json, a partir de aquí, muestra los avisos en mostrarAvisos(), previamente chequeando checkboxes del HTML con los que interactúa el usuario, cuando termina de filtrar, guarda los avisos filtrados en el array avisosFiltrados y los añade al doc llamando a añadirAvisosAlDoc() pasándole como parámetro el array con los avisos filtrados, en añadirAvisosAlDoc(), añade mediante un forEach, todos los avisos en el array avisosAMostrar, dependiendo si son de pérdida/búsqueda o de adopción

appOrg.js: Cumple un rol similar al archivo app.js, su tarea es leer el archivo json donde se encuentra alojada la información de las organizaciones registradas en el sitio y mostrarlas en la sección “Buscar organización”, también procesa los filtros de búsqueda y muestra los avisos correspondientes a los filtros aplicados.

mapaDeBusqOrg.js:

Se encarga de mostrar en el mapa a las organizaciones seleccionadas según los filtros de búsqueda aplicados en la sección “Buscar organización”.

Verifica si los checks de refugio, veterinaria y peluquería están marcados, en base a esto se llama la función filtrado que se encarga de guardar en una lista los markers filtrados y una vez hecho esto se usa la función muestraOrganizacionFiltrada() que se encargará de añadir los markers en el evento click del botón “actualizar-organización”. Se hace algo similar con el botón “ver-todo” que se le agrega el evento click llamando a la función mostrarTodasLasOrg que remueve todos los markers filtrados y añade nuevamente todos los markers en general.

metodosMapa.js: Como su nombre lo indica, en este archivo se encuentran todas las funciones relacionadas a los mapas que se encuentran en las distintas secciones del sitio, estas son crear el mapa y mostrar en él la ubicación solicitada según las coordenadas

provistas y agregar los marcadores correspondientes a las ubicaciones registradas.

`crearMapa()` devuelve un mapa de la API leaflet, con las coordenadas x, y de la UNGS.

`removeMarkers(mapa)` remueve todos los markers del mapa pasado como parámetro

`añadirMarkers(arrDeMarkers, mapa)` añade al mapa pasado como parámetro los markers contenidos en el array pasado como parámetro.

normalizadorOrganizacion.js: Su tarea consiste en brindar la funcionalidad de normalización de direcciones del AMBA para el formulario de solicitud de registro de organización en la sección “Registro organizaciones” del sitio, para esto se comunica con la API de normalización de direcciones de la Unidad de Sistemas de Información Geográfica del Gobierno de la Ciudad de Buenos Aires.

se agrega el evento input a “Direccion” que llama a la funcion normalizar, que consiste en agarrar el valor del momento y pasarlo como consulta a la api del usig, una vez hecho esto se hace una peticion a la url para pedir la lista de direcciones relacionadas a ese valor que paso el usuario , con esta lista devuelta por la api se llama a la funcion añadirDirecciones(response) la cual q se encarga de listar una por una cada direccion de la lista en forma de botones , para que el usuario tenga la oportunidad mediante el evento click de poder autocompletar la direccion del input “Direccion” .

Informe lado del Cliente

En este informe se detalla cómo sería la utilización del sitio web por parte del cliente (usuario que ingresa al sitio), en él se describe cómo interactúa el cliente con el portal y las funcionalidades que este le ofrece.

Al ingresar al sitio el cliente se encuentra con la sección “Avisos”, en ella puede visualizar los avisos publicados, además puede buscar avisos específicos utilizando los filtros de búsqueda provistos, estos permiten al cliente visualizar los avisos de su interés, se pueden buscar por tipo de aviso (adopción y perdido/encontrado) y por tipo de animal (perros, gatos, aves, etc.).

En la sección “Buscar organización” el cliente podrá visualizar las organizaciones registradas en el sitio, además contará con la posibilidad de buscarlas por nombre o aplicar filtros de búsqueda según servicios ofrecidos, que lo ayuden a encontrar las organizaciones de su interés, también podrá ver la ubicación de las organizaciones en un mapa interactivo.

Por último se encuentra la sección “Registro organizaciones”, en ella los representantes de las organizaciones podrán enviar una solicitud de registro de su organización al coordinador del sitio, para ello contará con un formulario de registro en el que se le solicitarán los datos correspondientes como nombre, dirección, etc. Para facilitarle al cliente, en este caso el representante, el ingreso correcto de la dirección de su organización se le provee la función de normalización de direcciones que le permite visualizar las distintas opciones posibles a la dirección que él ingresó y al darle click a alguna de ellas el campo de dirección del formulario se autocompleta con la dirección seleccionada, finalmente luego de ingresar todos los datos, si estos son válidos se muestra un mensaje en pantalla informando que su solicitud se ha enviado con éxito y los datos que ha ingresado.