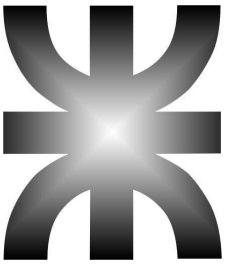


Trabajo Práctico Integrador

Etapa #1 - HTML

TUP - Laboratorio de Computación II

Introducción

El presente documento se presenta como Trabajo Práctico Integrador para todos los alumnos de Laboratorio de Computación II, donde se construirá una Página/App Web aplicando los conceptos que se dan durante el cursado sobre **HTML, CSS y JavaScript**.

Este trabajo tiene como objetivo que el alumno construya una aplicación que sirva para visualizar el estado del clima en diferentes ciudades del mundo.

En las diferentes etapas se guiará al alumno en el desarrollo de la solución, básicamente las **etapas** serán:

- **Etapa #1 - HTML**: Se trabajará principalmente con código HTML, agregando elementos y enlazando las diferentes páginas.
- **Etapa #2 - CSS**: En esta etapa se centrará en el agregado de estilo a las páginas para lograr una correcta visualización de lo que sería la maquetación de la App.
- **Etapa #3 - JS**: La Web se finalizará en esta etapa donde se realizará el desarrollo completo de la lógica JavaScript para darle vida y correcto funcionamiento según los requerimientos.

Importante: cada una de las etapas tiene una entrega obligatoria que deberá ser aprobada para poder avanzar con la siguiente.

Al ser un **TP Individual** que se irá avanzando en etapas, es requerido que cada alumno cree un repositorio en GitHub y vaya subiendo todo el código a medida que va avanzando con el desarrollo.

Etapa #1 - HTML

En esta primera etapa, nos centraremos en crear la estructura básica de cada una de las pantallas de la Web. Se entregará a continuación un código fuente base para que el alumno comience a trabajar sobre el mismo siguiendo los pasos detallados a continuación.

Recomendación: Subir los cambios al repositorio a medida que se va avanzando con el desarrollo o diferentes pasos.

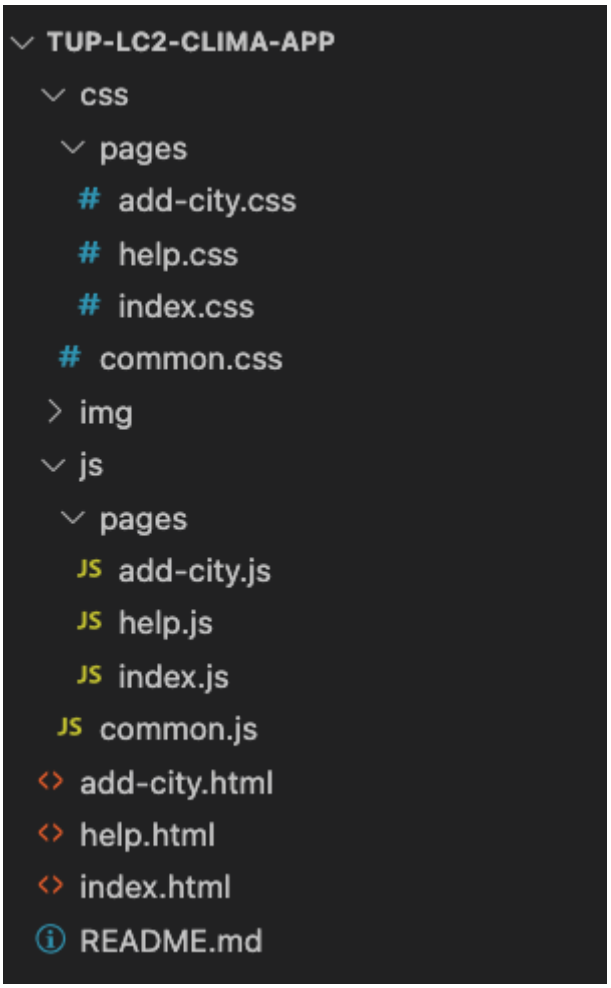
Paso #1: Crear repositorio

Crear un repositorio vacío en GitHub y clonarlo en su local para comenzar a trabajar.

Se recomienda colocar como nombre de repositorio: **tup-lc2-clima-app.**

Paso #2: Definir estructura de carpetas y archivos

Abrir el proyecto en el editor de código a utilizar y crear la estructura de carpetas y archivos de la Aplicación tal como se visualiza a continuación:



A continuación, se explica la finalidad de cada carpeta/archivo:

- **css**: carpeta donde se deben almacenar todos archivos CSS.
    - **common.css**: en este archivo se debe colocar el código CSS/estilos de la App que son transversales a todas las pantallas o bien que afectan a más de una pantalla.
    - **pages**: carpeta donde se debe crear un archivo CSS por cada página de nuestra aplicación.
        - **[page].css**: En este caso, cada archivo contendrá código CSS con el estilo de la página en cuestión
    - **img**: carpeta donde se colocarán todos los recursos gráficos, imágenes de la App.
- **js**: carpeta donde se deben almacenar todos los archivos JS.
    - **common.js**: en este archivo se debe colocar el código JS, scripts de la App que son transversales a todas las pantallas o bien que afectan a más de una pantalla.
    - **pages**: carpeta donde se debe crear un archivo JS por cada página de nuestra aplicación.
        - **[page].js**: En este caso, cada archivo contendrá código JavaScript con la lógica de esa pantalla puntual.
- **index.html**: Por convención, este documento HTML es la página inicial o de entrada de todo sitio Web, en nuestro caso la utilizaremos para la pantalla “Consultar Clima”.
- **add-city.html**: Documento correspondiente a la pantalla “Agregar Ciudad”.
- **help.html**: Documento correspondiente a la pantalla de “Ayuda” o “Contacto”.
- **README.md**: este archivo no será visualizado por el usuario final y no es obligatorio utilizarlo. El mismo es utilizado para dar detalles sobre el proyecto y será visualizado en el repositorio cuando se ingrese desde GitHub. En el siguiente link podrás cómo escribir un README para darle formato.

Paso #3: Estructura HTML

En este paso, vamos a definir la estructura de nuestro documento HTML. La cual se deberá aplicar a cada página de nuestro sitio para mantener una misma línea visual, obviamente, variando el contenido en sí de la pantalla.

index.html:

```
<!DOCTYPE html>
<html lang="es">

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>TUP LC2 - TP Integrador</title>
   <!-- Estilos CSS -->
   <link rel="stylesheet" href="css/common.css">
</head>

<body>
   <header>
       <h1>
           <span>Clima App</span>
       </h1>
       <nav>
           <ul>
               <li class="current-page"><a href="#">Mi App</a></li>
               <li><a href="/add-city.html">Ciudad</a></li>
               <li><a href="/help.html">Ayuda</a></li>
           </ul>
       </nav>
   </header>
   <!-- Contenido principal de la página -->
   <main>
      <h2>Mi primera App!</h2>
         <section id="form-city-list">
            <!-- Sección donde se debe colocar el formulario, con el select y el
botón de consultar -->
         </section>
         <section id="section-weather-result">
               <!-- Sección donde se debe mostrar el clima de la ciudad -->
         </section>
   </main>

   <footer>
       <p>
           Desarrollado con ♥ - LC2
       </p>
   </footer>
</body>

</html>
```

Pegar el código entregado en el archivo indicado, formatear el mismo (Ctrl + Shift + F) para una correcta visualización. Analizar el código en cuestión, leer los comentarios (<!-- -->) y visualizar en el navegador el resultado obtenido.

Consideración: Con estructura nos referimos a los componentes esenciales de la pantalla los cuales se deberían mostrar en cada una de las pantallas de nuestra aplicación para mantener una misma línea visual. En nuestro caso, el **header** se debería mantener en todas las

pantallas ajustando el comportamiento de los links para cada caso puntual. El **footer** es otro componente que se puede repetir en todas las pantallas. Lo mismo aplica para el contenido principal **main**, con el título dentro de **h2** y cada una de las secciones (**section**) del contenido principal.

Paso #4: Estilos comunes

Para que la aplicación se visualice de mejor manera en el navegador, se entrega el siguiente código CSS que se deberá colocar dentro del archivo common.css. Asegurarse de que el archivo CSS está linkeado correctamente al index.html.

css/common.css

```
* {
   box-sizing: border-box;
   margin: 0;
   padding: 0;
   font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans
Unicode', Geneva, Verdana, sans-serif;
}

:root {
   /* Declaro variables CSS con colores principales */
   --color-theme: #00594c;
   --color-active: #5df2d6;
   --color-back-active: #04957d;
   --color-gradient: linear-gradient(to right, var(--color-theme),
var(--color-back-active));
}

body {
   height: 100%;
   width: 100%;
}

ul {
   list-style: none;
   padding: 0;
   margin: 0;
}

button,
a {
   user-select: none;
   -webkit-user-select: none;
}

header {
   position: fixed;
   top: 0;
   width: 100%;
   background: var(--color-theme);
}

nav>ul {
   display: flex;
   justify-content: space-between;
   justify-content: center;
}

nav li {
```

```
    flex: 0 0 auto;
}

nav li a {
    display: block;
    padding: 16px 12px;
    text-align: center;
    text-decoration: none;
    font-size: 1em;
    font-weight: bold;
    color: var(--color-active);
}

nav li.current-page a {
    color: var(--color-theme);
    background-color: var(--color-active);
    pointer-events: none;
}

main {
    margin-top: 120px;
    overflow: auto;
    padding: 16px;
    padding-bottom: 60px;
    height: calc(100% - 120px);
}

h1 {
    text-align: center;
    margin: 0;
    color: white;
}

h1>img {
    width: 70px;
    vertical-align: middle;
}

h2 {
    color: var(--color-theme);
    margin: 16px 0;
    text-align: center;
}

h3 {
    color: var(--color-theme);
    margin: 8px 0;
}

main section {
    display: flex;
    justify-content: space-around;
}

footer {
    position: fixed;
    bottom: 0px;
    background: var(--color-theme);
    width: 100%;
    text-align: center;
    padding-top: 10px;
    padding-bottom: 10px;
}

footer p {
    font-size: small;
    color: white;
}
```

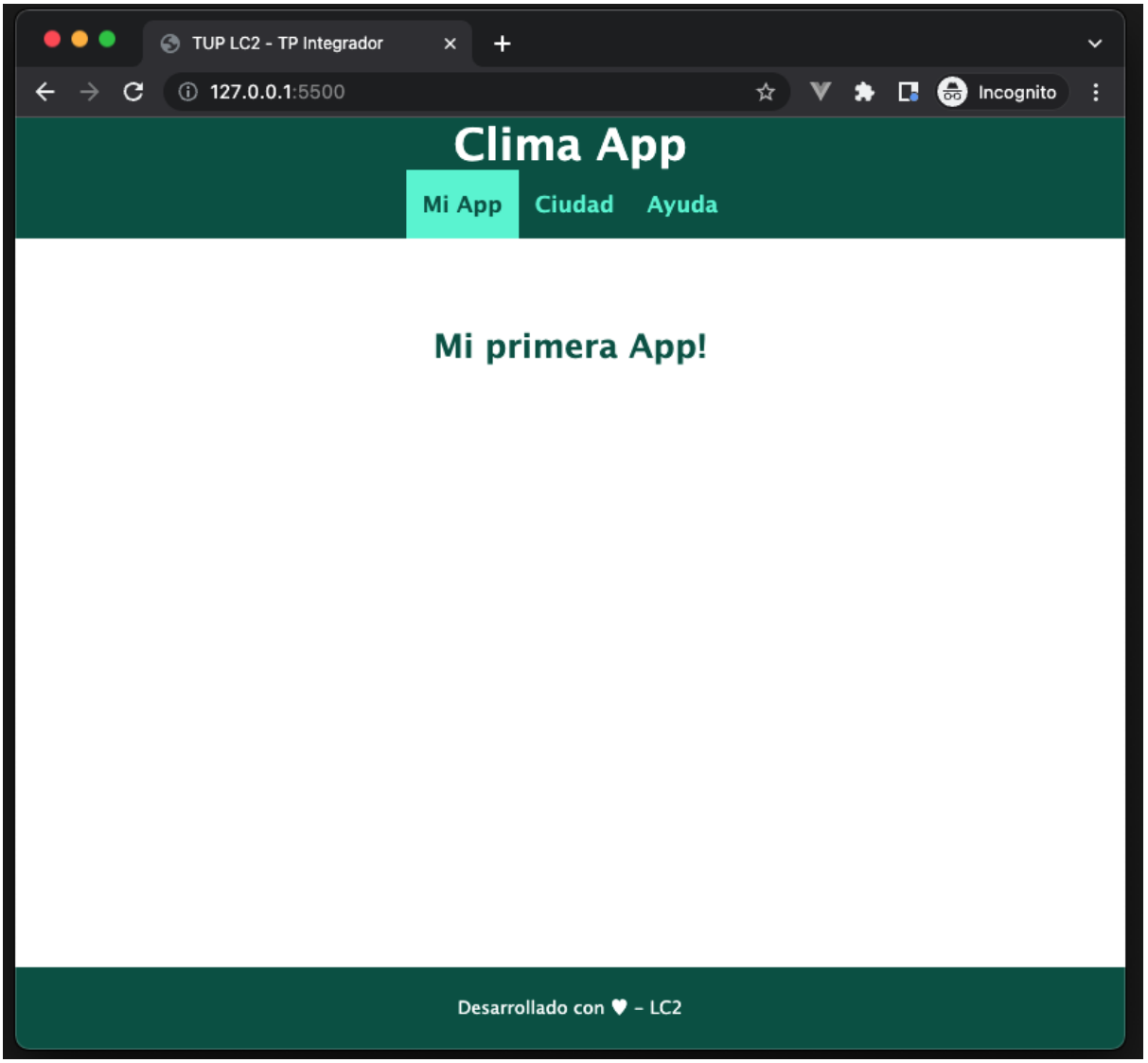
```
main p {
   margin-bottom: 16px;
}

/* Media Query: Estilos a aplicar sólo en pantallas mayores a 400px */
@media (min-width: 400px) {
   main {
       padding-left: 10%;
       padding-right: 10%;
   }
   footer {
       padding-top: 20px;
       padding-bottom: 20px;
   }
}
```

Luego de pegar el código, recordar formatear el mismo para una correcta indentación/visualización.

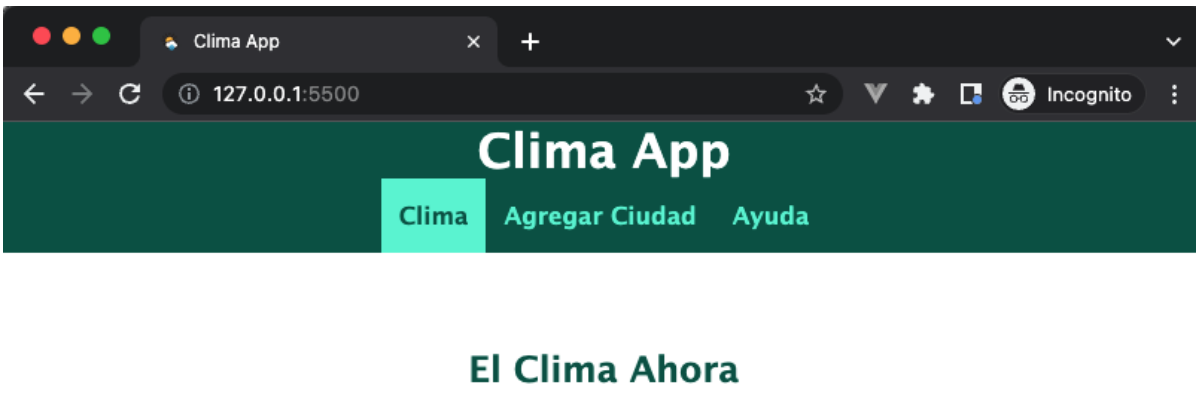
Consideración: en esta etapa del TP no nos vamos a centrar en los estilos CSS pero considere analizar el código que acaba de pegar y ajustar el valor de las propiedades para ver cómo responden en el navegador (luego deje el código tal cual lo entregamos).

Por último, abrir el archivo **index.html** en el navegador donde se deberá encontrar con el siguiente resultado:



Paso #5: Ajustar Header y Título

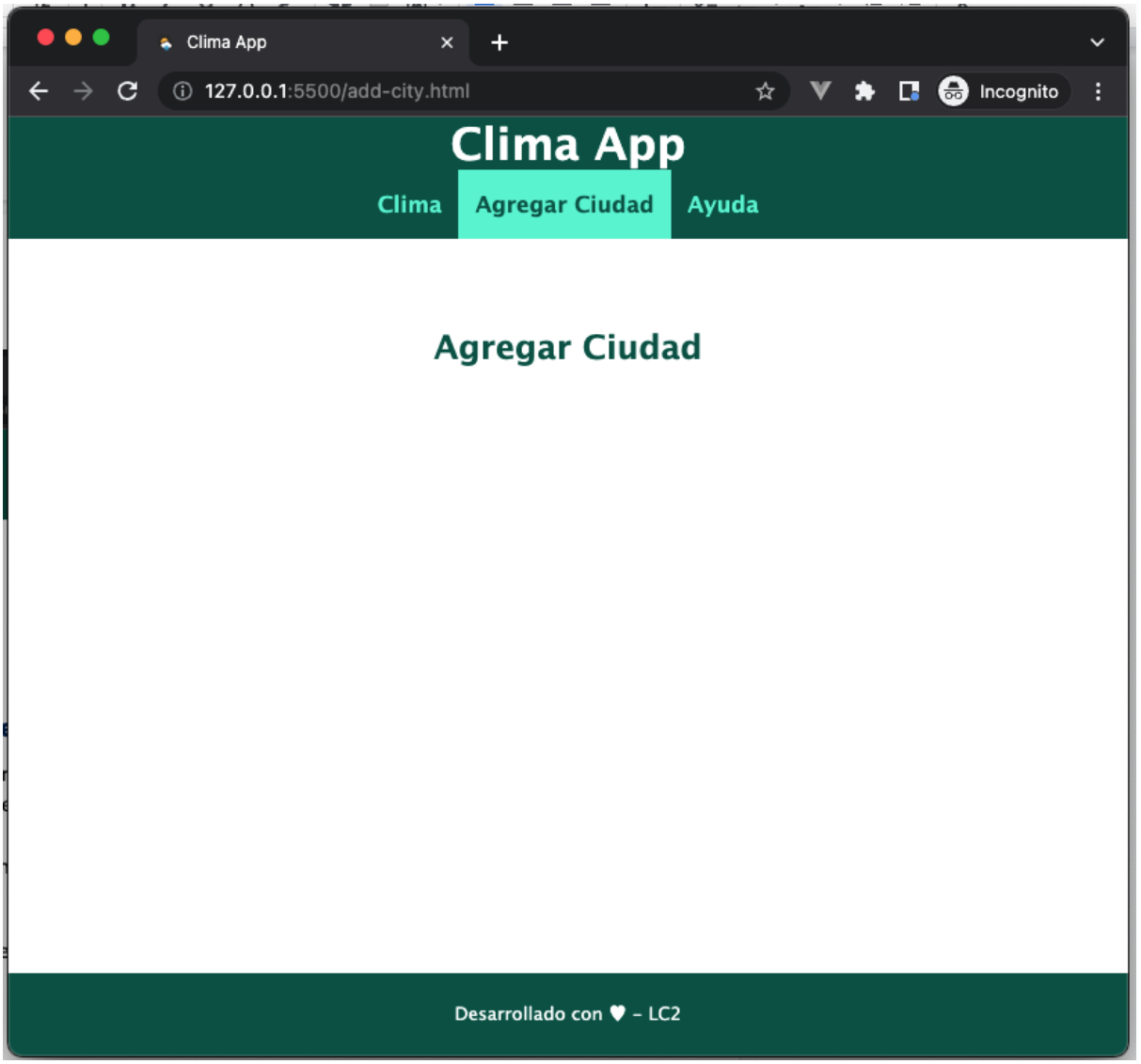
Ajustar el contenido del Head (title e ícono/favicon), el Header y el Título (etiqueta h2) del Documento index.html para que se visualice de la siguiente manera:

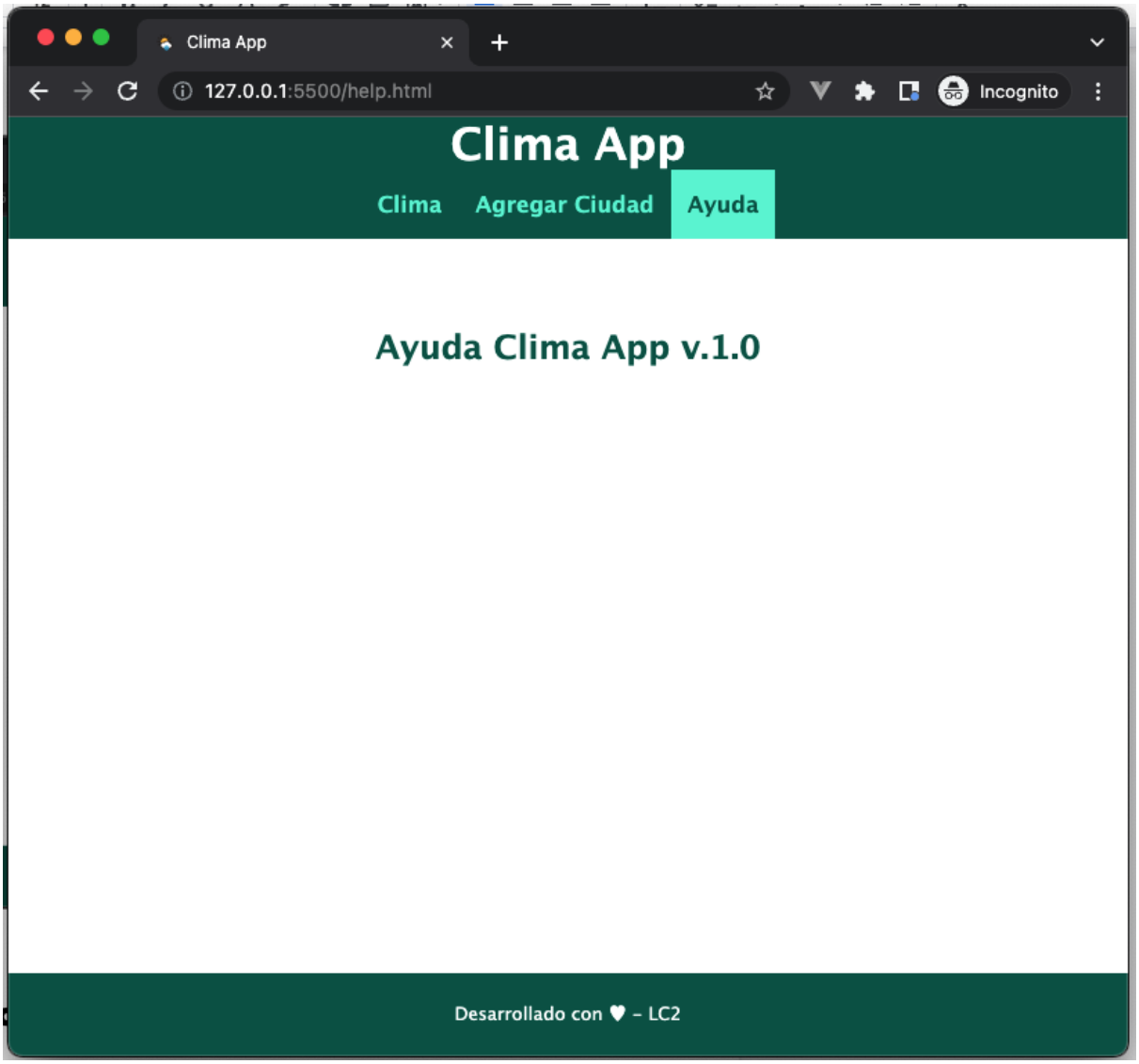


Paso #6: Replicar estructura a otra páginas

Copiar el código HTML generado en el **index.html** hasta el momento y pegarlo en los otros dos documentos. Ajustar el Header para que la opción resaltada se corresponda con la pantalla visualizada (aplicar clase "**current-page**" según corresponda). Ajustar también el título (**h2**) y por, el momento, dejar sólo un **section** vacío debajo del título en estas dos pantallas/documentos (**add-city.html** y **help.html**).

Debería encontrarse con el siguiente resultado para las páginas **Agregar Ciudad** y **Ayuda** respecticamente:

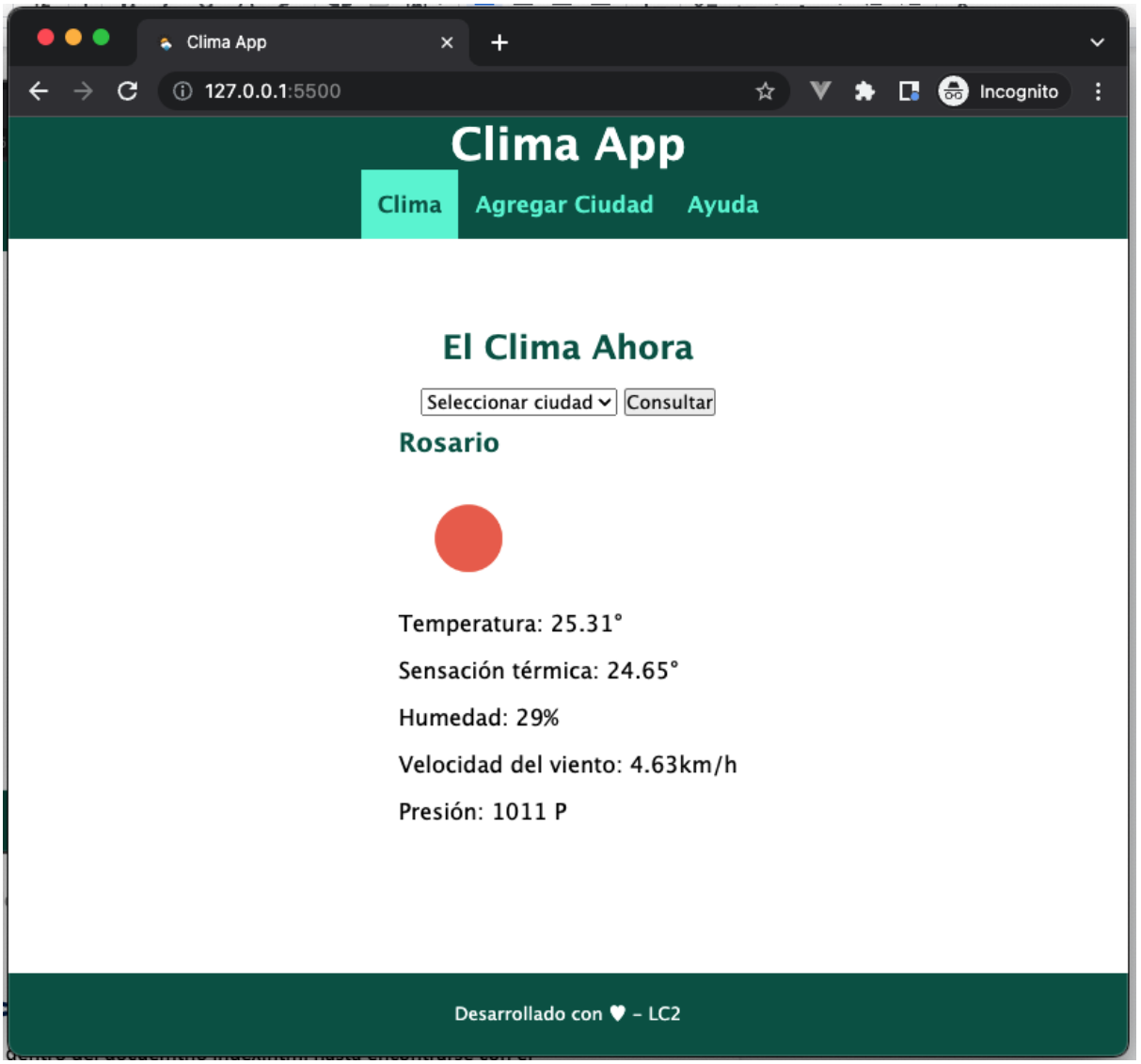




Importante: Asegurarse de que la navegación funcione correctamente en cada pantalla hacia las otras dos.

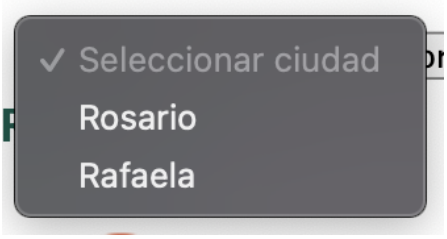
Paso #6: Agregar contenido en index.html

Agregar elementos HTML dentro del documento index.html hasta encontrarse con el siguiente resultado:



Consideraciones:

- Dentro del **section#form-city-list** agregar un formulario (**form**) con un input de selección (**select**) y un botón de consultar (**button**).
- Dentro de las opciones del Select colocar las siguientes opciones:



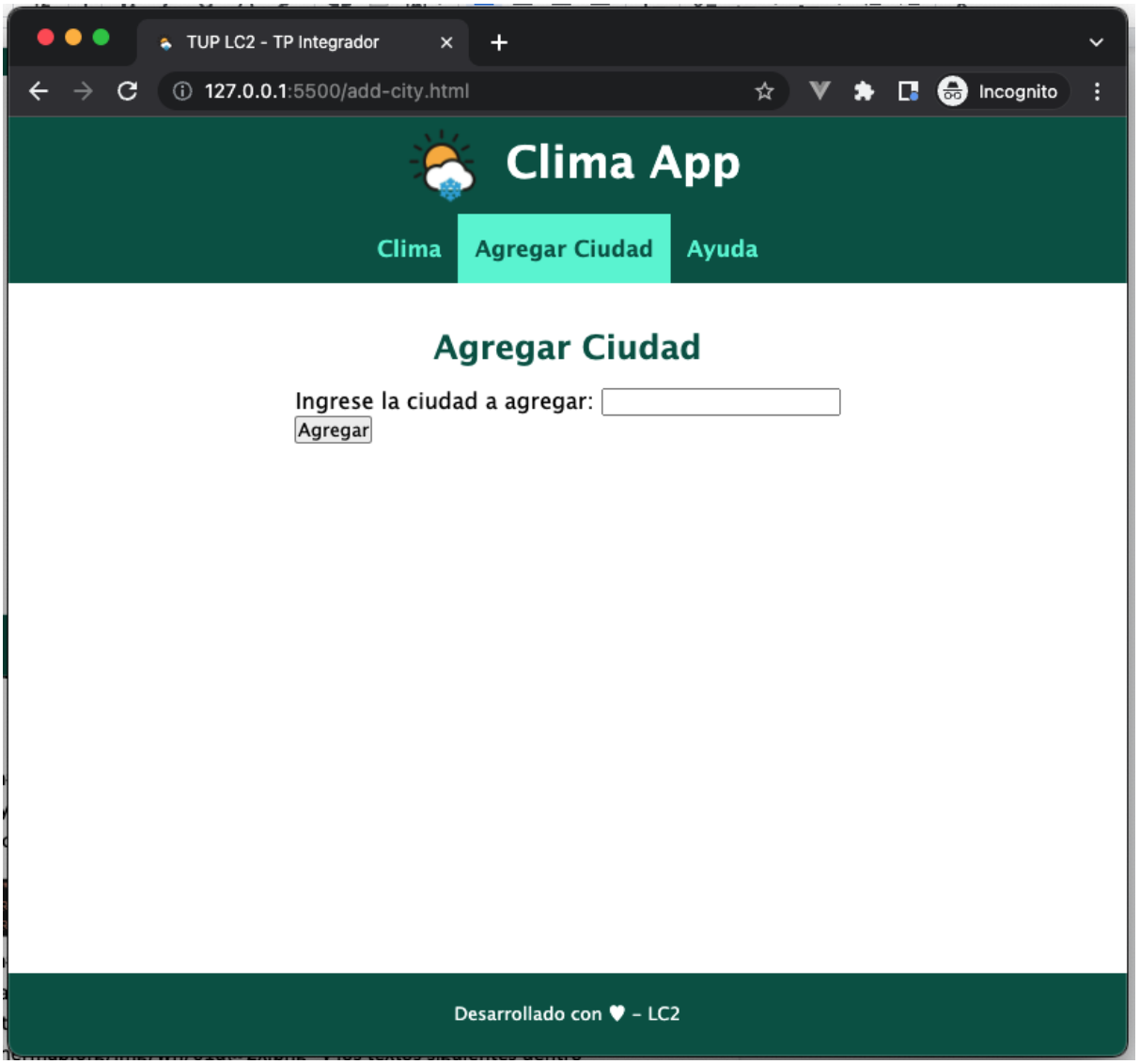
  - “Seleccionar ciudad” - value=”” - disabled - selected.
  - “Rosario” - value=”ROSARIO”

    - “Rafaela” - value=”RAFAELA”

      Consideración: Estas opciones son únicamente ilustrativas. Cuando avancemos con JS el select comenzará vacío y cargaremos las opciones programáticamente desde Javascript.
- Dentro del **section#section-weather-result** agregar un **div** con la clase “**card**”. Dentro del **div.card** agregar el nombre de la ciudad (**h3**), una imagen (**img**) que haga referencia (absoluta) a esta imagen “http://openweathermap.org/img/wn/01d@2x.png” y los textos siguientes dentro de un cada uno.

Paso #7: Agregar contenido en add-city.html

En el documento add-city.html, debajo del título de la pantalla (**h2**), en el **section** sin clase ni identificador agregar un formulario (**form**) con elementos para obtener como resultado lo siguiente:



Paso #8: Agregar contenido en help.html

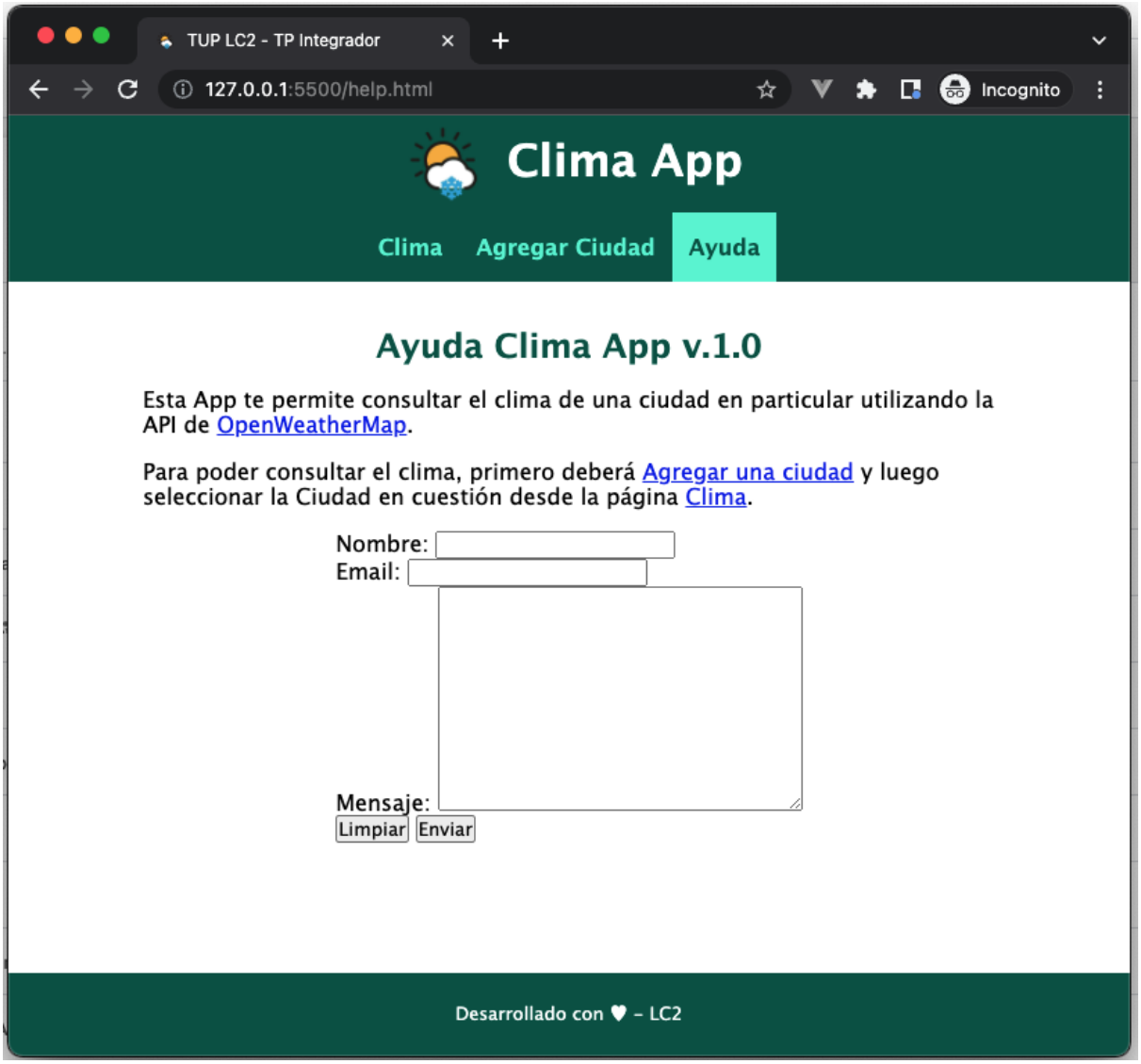
En el documento help.html, debajo del título de la pantalla (**h2**), agregar dos párrafos (**p**) con contenido para que se visualice de la siguiente manera:

Consideraciones: El primer link debe llevar a la siguiente página "https://openweathermap.org/api" y abrirse en una nueva pestaña (target="_blank"), mientras que los otros dos deben llevar a las respectivas páginas de nuestra App dentro de la misma pestaña.

Por último luego del segundo **p** agregado anteriormente, agregar un section que contenga un form con inputs y botones tal como se indica a continuación:

- Nombre: input de tipo texto
- Email: input de tipo email
- Mensaje: textarea
- Botón de Limpiar
- Botón de Enviar

El resultado para esta pantalla debería ser el siguiente:


TUP LC2 - TP Integrador
127.0.0.1:5500/help.html
Incognito
Clima App
Clima
Agregar Ciudad
Ayuda
Ayuda Clima App v.1.0
Esta App te permite consultar el clima de una ciudad en particular utilizando la API de OpenWeatherMap.
Para poder consultar el clima, primero deberá Agregar una ciudad y luego seleccionar la Ciudad en cuestión desde la página Clima.
Nombre:
Email:
Mensaje:
Limpiar
Enviar
Desarrollado con ♥ - LC2

Resultado final

Si, ahora tu página del Clima no se ve muy elegante pero no te pero no te preocupes, de eso nos encargaremos en la Etapa #2 - CSS, por ahora lo importante es que te centres en el código HTML y coloques los elementos para que las páginas se visualicen tal como te lo hemos indicado.

Entrega

La fecha límite para la entrega es el **29/04/22**, la cual se deberá realizar a través de email a ambos profesores (fabricio.garcia@frro.utn.edu.ar y praissiguie@frro.utn.edu.ar).

En el asunto del correo se deberá colocar: **TP Integrador Clima - Etapa 1 - [Nombre Apellido]**.

Ejemplo: **TP Integrador Clima - Etapa 1 - Juan Perez**.

En el cuerpo del mensaje, se debe indicar:

- URL del repositorio.
- Consideraciones: cualquier consideración relacionada al desarrollo y/o resolución del problema. Si te pareció muy fácil o complicado. Que mejorarías para las futuras etapas.