

PROYECTO DE SOFTWARE 2021 (SI305)
AGOSTO 2021

INFORME FINAL

Autores:
Pereyra Nehuen (15926/1)
Pereyra Iyael (16440/6)

Índice

1. Introducción	2
2. Especificación resumida	2
3. Palabras Claves	2
4. Marco teórico	2
4.1. Entornos virtuales (virtualenv, conda, etc.)	2
4.2. Flask	2
4.3. Vue.js	3
5. Problemas y soluciones surgidas durante el desarrollo	3
6. Consideraciones éticas sobre el desarrollo	4
6.1. Seguridad por Iyael Pereyra	4
6.2. Finalidad y calidad de los datos por Nehuen Pereyra	4
7. Conclusiones	5
8. Bibliografía y referencias	6
9. Anexo 1: guía de usuario	7
10. Anexo 2: guía para el desarrollador	17

1. Introducción

Este informe corresponde al sistema de centros de ayuda y turnos realizado para la cátedra de proyecto de software en el año 2020.

2. Especificación resumida

El sistema creado busca ayudar a la comunidad a enfrentar la situación de pandemia por la que se está atravesando en el año 2020. El sistema busca ayudar brindando información relevante sobre los diversos centros de ayuda (centros de donación de ropa, de comida, etc.) ubicados en la provincia de Buenos Aires. Para ello, este permite que una persona con acceso a internet pueda acceder a un sitio web y visualizar centros de ayuda cercanos, en los cuales se realizan donaciones de ropa, comida, plasma o de sangre. Esto ayuda tanto a organizaciones, como personas individuales a poder obtener de una forma rápida y centralizada la información relevante. Entre la información de los centros de ayuda que se brindan, se encuentra el nombre, localización geográfica, horario de apertura y cierre, teléfono y correo electrónico de contacto, entre otros. Además de poder visualizarlos, se puede sacar un turno, para realizar una donación respetando el protocolo que posee el centro de ayuda. Se tiene en cuenta la accesibilidad al sitio web para que pueda ser accedido sin problemas por cualquier persona y que sea responsive.

3. Palabras Claves

Python, Flask, VueJs, Centro de ayuda y Turnos.

4. Marco teórico

4.1. Entornos virtuales (virtualenv, conda, etc.)

Cuando se desarrollan varios proyectos en una misma computadora, es posible que las dependencias de un proyecto (librerías, frameworks, intérprete, etc.) generen problemas sobre las dependencias de otro proyecto. Esto puede suceder debido a que diferentes proyectos utilizan una misma dependencia (por ejemplo, una librería) pero con diferente versión. Gracias al uso de los entornos virtuales, lo que se logra es que cada proyecto posea su propio entorno virtual el cual tenga todas las dependencias que este necesita para su correcto funcionamiento y se encuentren aislado del resto del sistema, evitando de esta manera posibles problemas. Dando un ejemplo de un módulo de Python que permite crear entornos virtuales se encuentra “virtualenv”. Como se explica en su documentación oficial: “virtualenv es una herramienta para crear entornos aislados de Python.” (<https://virtualenv.pypa.io/en/latest/>). Del mismo sitio, se observa que está licenciado bajo la licencia MIT. Según su github (<https://github.com/pypa/virtualenv>), este se encuentra actualmente en la versión 20.6.0, cuenta con 53 contribuidores y posee 167 releases.

4.2. Flask

Según Wikipedia: “Flask es un framework minimalista escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código” (<https://es.wikipedia.org/wiki/Flask>). Este framework fue desarrollado por Armin Ronacher, el cual es un programador austriaco de software de código abierto. En la página oficial¹ podemos ver que depende del motor de plantillas jinja y

del kit de herramientas Werkzeug WSGI. Este framework tiene una licencia BSD (Berkeley Software Distribution) según el proyecto en github (<https://github.com/pallets/flask/>) la cual permite el uso del código fuente en software no libre. De la misma fuente, vemos que es utilizado en más de 787 mil proyectos y posee más de 600 contribuidores.

4.3. Vue.js

Como se describe en el proyecto github: “Vue.js es un marco de JavaScript progresivo que se puede adoptar de forma incremental para crear interfaces de usuario en la web” (<https://github.com/vuejs/vue>). Este framework fue creado por Evan You, el cual es un desarrollador de software independiente como se autodefine en su página (<https://evanyou.me>) además de tener más de 388 contribuidores en github. En la página oficial² destaca estas tres características:

- **Accesible:** por la facilidad de poder comenzar a desarrollar teniendo un poco de conocimiento de html, css y javascript.
- **Versátil:** destaca la facilidad de escalar el sistema de ser una librería a un framework completo con todas las funciones.
- **Performancia:** tamaño de la librería sumamente pequeño, con un virtual DOM sumamente rápido y esfuerzos mínimos de optimización.

Este framework es: “un proyecto de código abierto con licencia del MIT” como se describe en github.

5. Problemas y soluciones surgidas durante el desarrollo

Durante el desarrollo, nos encontramos con problemas habituales al trabajar con nuevas tecnologías, en el cual recurrimos a la documentación para solucionar los problemas, esto de forma general. De forma particular, tuvimos los siguientes problemas, que debimos utilizar un tiempo mayor para solucionarlos:

- **Conectar el ORM (Asignación objeto-relacional) a la base de datos:** La conexión del ORM a la base de datos, que en nuestro caso utilizamos SQLAlchemy, nos trajo problemas en un principio, viendo guías de conexión por la web y leyendo la documentación, logramos realizar la conexión.
- **Guardado del archivo correspondiente al protocolo de vista de un centro de ayuda en el servidor:** Cuando implementamos la funcionalidad de almacenar el archivo correspondiente al protocolo de vista, de un centro de ayuda, nos surgió un problema cuando se tenía que almacenar el archivo. Lo que queríamos hacer era que desde el código Python, se creará un directorio propio para cada centro de ayuda y, dentro de este directorio, se almacene el archivo correspondiente al protocolo de vista. El problema venía en que estábamos generando directorios desde el código y si bien de forma local nos funcionaba, cuando subimos el proyecto al servidor provisto por la cátedra, esto no funcionaba (por cuestiones de seguridad no se permitía realizar la operación de crear directorios). Como solución a este problema, lo que hicimos fue cambiar la manera en la cual se almacena el archivo. Al final, el archivo se almacena en una carpeta precreada, no teniendo así que crear una nueva carpeta desde el código de Python.

6. Consideraciones éticas sobre el desarrollo

6.1. Seguridad por Iyael Pereyra

El sistema desarrollado cuenta con una buena seguridad tanto para los usuarios finales como para los administradores del sitio.

Al momento de desarrollar el sistema, se pensó en la seguridad del mismo como una parte fundamental que debía ser hecha de la mejor manera posible. Por ello, varias decisiones de diseño fueron basadas en la seguridad.

Una buena parte de la seguridad está provista por las tecnologías que se decidió utilizar. Estas tecnologías proveen seguridad en sus ámbitos, siendo ampliamente utilizadas por la comunidad y constantemente actualizadas para afrontar los riesgos actuales que van surgiendo. Otra parte de la seguridad fue hecha a partir de buenas prácticas de desarrollo y una buena configuración del sistema.

El sistema utiliza el orm SQL Alchemy para manejar todas las cuestiones relacionadas a la base de datos. Esta tecnología provee protección ante ataques de Inyección SQL (se encarga de escapar todos los datos que entran a la base de datos para que de esta manera no se pueda inyectar código).

El sistema almacena la información del usuario completamente en el servidor, como datos de sesión (la única información almacenada en el lado del cliente es la cookie de sesión para que el servidor pueda identificar al usuario).

Al desarrollar el sistema se tuvo en cuenta la criticidad de mantener segura la información de los usuarios, por lo que se aplicó un buen algoritmo de hash a las contraseñas de los usuarios (de esta forma, aunque un atacante logre acceder a la base de datos, no va a poder de forma simple obtener las contraseñas de los usuarios).

Todas las rutas del sistema están protegidas por verificación de permisos. Es decir, un usuario solo puede acceder a aquellas rutas en las cuales tenga los permisos correspondientes.

El sistema utiliza el motor de plantillas Jinja, el cual provee protección contra ataques de Inyección de Plantillas (Template Injection) y ataques de Secuencia de Comandos en Sitios Cruzados (XSS, Cross-Site Scripting). Esta protección se debe gracias a que Jinja escapa todos los datos dinámicos, no dejando así la posibilidad de inyectar código.

El sistema cuenta con buena seguridad en todos sus formularios. Todos los datos ingresados en los formularios son validados de forma adecuada en el servidor, antes de realizar cualquier acción en base a ellos. Además, gracias al uso de la tecnología Flask Form, el sistema cuenta con protección contra ataques de Falsificación de Petición en Sitios Cruzados (Cross-Site Request Forgery), gracias a la utilización de un token para validar cada formulario.

En resumen, la seguridad de un sistema informático es crítica tanto para los usuarios que lo utilizan como para los administradores. Por ello al momento de desarrollar un sistema, la seguridad no se debe ver como algo adicional, sino más bien como algo fundamental que el sistema debe poseer para su buen funcionamiento. La ausencia de éste puede comprometer a sus usuarios y al sitio entero.

6.2. Finalidad y calidad de los datos por Nehuen Pereyra

Se analiza la finalidad y calidad de los datos en el proyecto realizado en la cátedra, pero primero definimos los dos términos. La finalidad de los datos está relacionada con la recopilación de datos en una aplicación, en donde los datos recopilados, deben ser utilizados con el finalidad que originó la recolección. Estos datos no pueden ser utilizados para otros medios como por ejemplo: vender a terceros los datos recopilados, realizar fraude electrónico o telefónico, campañas publicitarias, obtener patrones de comportamiento por parte del cliente, entre otras finalidades. Mientras que la calidad de los datos, se refiere a los datos personales que se recopilan y almacenan deben ser veraces

y no excesivos, en base a la finalidad que tienen en el sistema. Estos datos no pueden ser obtenidos de forma implícita (sin el consentimiento de la persona que carga sus datos) o de forma ilegal. Por último, estos datos una vez cumplida su funcionalidad, deben ser eliminados.

Se realizará un análisis de los 3 tipos de datos más importantes, en la aplicación que son: los datos del usuario, centros de ayuda y turnos.

- **Datos de usuario:** La aplicación privada posee un sistema de logueo, por lo tanto se requiere tener un usuario para poder acceder. Existen dos tipos de usuario: administrador y operador de centro de ayuda. Los usuarios con rol administrador son los encargados de cargar los datos, por lo tanto la principal finalidad de la recolección de estos datos, es la de poder acceder al sistema y asignarle un rol a un usuario. Los datos recopilados son los siguientes: nombre, apellido, correo electrónico, nombre de usuario, contraseña y rol. Si analizamos la calidad de los datos, estos no son excesivos y son los mínimos para poder identificar un usuario con la persona que utilizara esa cuenta. Los datos son cargados de forma explícita. Estos datos son eliminados, cuando un administrador da de baja a un usuario.
- **Centros de ayuda:** Tanto en la aplicación privada como en la pública, se muestra un listado de los centros de ayuda y se puede acceder a su detalle. La carga de los datos, se realiza también en ambas aplicaciones, pero al realizarla desde la aplicación privada por un usuario, el centro de ayuda se puede cargar en estado “publicado” y “aceptado” por lo tanto es visible en los listados de ambas aplicaciones. En cambio, si una persona carga un centro de ayuda desde la aplicación pública, este se carga en estado “pendiente”, en donde un administrador u operador de centro de ayuda, debe aceptarlo y ponerlo publicado para que sea visible en los listados. Los datos recopilados son los siguientes: nombre del centro de ayuda, dirección, teléfono, hora de apertura, hora de cierre, tipo de centro, municipio, sitio web, correo electrónico, protocolo y localización geográfica. Si analizamos la calidad de los datos para este caso, podemos ver que hay datos que identifican al centro de ayuda específico mientras que otros brindan información de contacto. Se puede ver que los datos recopilados no son excesivos, ya que los datos que identifican al centro de ayuda son mínimos y solo hay 3 datos de contacto. Estos datos son eliminados, cuando un usuario con rol de administrador elimina un centro de ayuda.
- **Turnos:** En la aplicación privada, es solo posible ver el listado de turnos con sus detalles. En cambio en la aplicación pública, se puede ver la cantidad de turnos solicitados en la semana actual, pero es un dato disociado ya que los datos personales, que posee un turno se procesaron, de manera tal que la información mostrada en la aplicación pública, no pueda ser asociada a una persona específica. La carga de datos, se puede realizar desde ambas aplicaciones. Los datos recopilados son los siguientes: nombre, apellido, email, teléfono donante (opcional), fecha del turno y horario del turno. Si analizamos la calidad de los datos, vemos que los datos recopilados no son excesivos ya que permiten identificar a la persona que saca el turno y los datos mínimos del turno solicitado que es la fecha y la hora. Estos datos son eliminados, cuando un usuario con rol de administrador elimina un turno.

7. Conclusiones

Si bien el sistema costó mucho tiempo y dedicación de desarrollo, los conocimientos y habilidades adquiridas nos brindaron mucho saber sobre cómo realizar un sistema.

Antes de habernos embarcado en la realización de este proyecto, teníamos un conocimiento básico sobre cómo podría ser realizar una aplicación real. Pero una vez finalizada la materia y terminado

el trabajo, pudimos conectar varios conceptos que nos hizo tener una buena idea de cómo trabaja un sistema real.

Además nos permitió utilizar lo aprendido para otras materias, como es el caso de “Tecnologías de Producción de Software”, en el cual se nos asignó un cliente, que fue el Jefe de informática de la Facultad de Medicina de la UNLP. En donde desarrollamos un sistema de cátedras y oficinas utilizando Flask y las buenas prácticas aprendidas, al realizar este sistema de centros de ayuda y turnos.

8. Bibliografía y referencias

- 1.<https://flask.palletsprojects.com/en/2.0.x/.Flask>. Consultado el 20 de julio 2021.
- 2.<https://es.vuejs.org/index.html>. *Vue.js*. Consultado el 20 de julio 2021.

9. Anexo 1: guía de usuario

Comenzaremos la guía de usuario por la aplicación privada. Si accedemos a la aplicación privada por el siguiente link: https://admin-grupo20.proyecto2020.linti.unlp.edu.ar/iniciar_sesion nos mostrara la pantalla de inicio de sesión de la figura 1, donde podemos iniciar sesión con un usuario o utilizando un login alternativo, usando una cuenta de gmail (el usuario debe estar previamente registrado en el sistema).

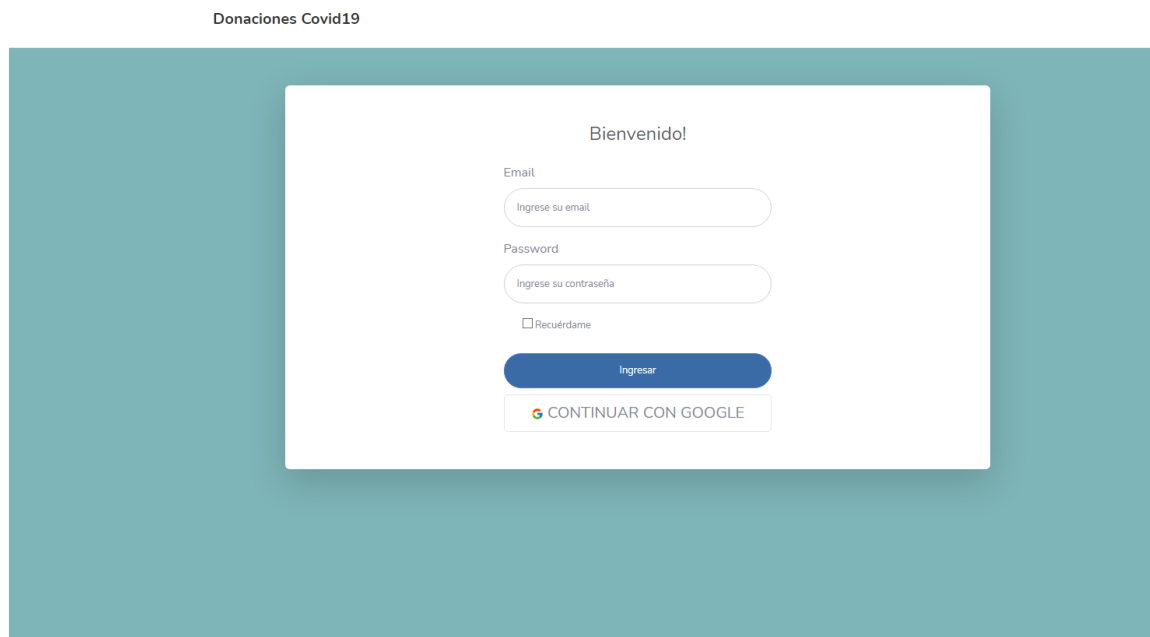


Figura 1: Inicio de sesión aplicación privada.

Para iniciar sesión como administrador, ingresamos con el siguiente usuario:

- **Usuario:** admin@admin.com
- **Contraseña:** admin123

Nos redirigirá al listado de usuarios (ya que ingresamos con un usuario con rol administrador) se puede observar en la figura 2 que se puede acceder a: agregar, modificar, borrar, ver el detalle y filtrar usuarios en esta pantalla. En el icono lateral izquierdo se puede acceder al perfil y cerrar la sesión. Además en el panel lateral, se tiene acceso a la página principal de centros de ayuda, turnos y configuración. En la figura 3 se puede ver la pantalla para registrar un usuario, mientras en la figura 4 se puede ver para modificar un usuario.

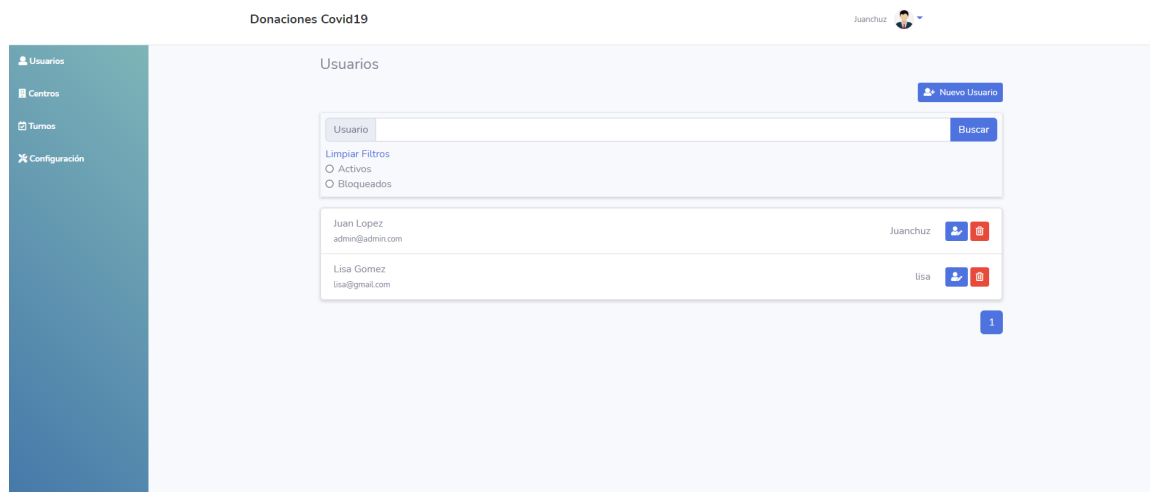



Figura 2: Página principal de usuarios.

The screenshot shows the 'Registrar Usuario' form in the 'Donaciones Covid19' application. The sidebar is the same as in Figure 2. The main content area is titled 'Registrar Usuario' and contains a form with the following fields: 'Nombre' (Name), 'Apellido' (Last Name), 'Correo Electrónico' (Email), 'Nombre de Usuario' (Username), 'Contraseña' (Password), and 'Repetir Contraseña' (Repeat Password). Below these fields is a section for 'Administrador Operador' with a text area. At the bottom of the form is a large blue button labeled 'Enviar'.

Figura 3: Registro de un usuario.

Donaciones Covid19 Juanchuz 

- Usuarios
- Centros
- Turnos
- Configuración

Información de Cuenta

Nombre

Apellido

Correo Electrónico

Nombre de usuario


Contraseña

Roles
☒ Administrador
☐ Operador

[Enviar](#)

Figura 4: Modificar un usuario.

Si accedemos a la pantalla principal de centros de ayuda, podemos ver la pantalla de la figura 5. Se muestra el listado de centros de ayuda (con sus respectivos estados) y se puede acceder a: agregar, modificar, borrar, ver el detalle, ir a la sección de turnos y filtrar centros de ayuda. En la figura 6 se muestra la pantalla para agregar un centro de ayuda, mientras en la figura 7 para modificar un centro. En la figura 8 se puede ver la sección de turnos, de un centro de ayuda particular. En esta sección se puede: agregar, modificar y filtrar los turnos. En la figura 9 se puede ver cuando accedemos a solicitar un turno, mientras en la figura 10 cuando modificamos un turno.

Donaciones Covid19 Juanchuz 

- Usuarios
- Centros
- Turnos
- Configuración

Centros de Ayuda

[+ Nuevo Centro](#)

[Buscar](#)

[Limpiar Filtros](#)
☐ Pendientes
☐ Rechazados
☐ Aceptados














Centro Batata Centro de Alimentos Municipio Almirante Brown 14.00hs - 16.00hs	  
Centro Cafe Centro de Alimentos Municipio Almirante Brown 14.00hs - 16.00hs	  
Centro Coco Centro de Alimentos Municipio Almirante Brown 14.00hs - 15.00hs	  
Centro el Arroz Centro de Alimentos Municipio Merlo	  

Figura 5: Página principal de centros de ayuda.

Donaciones Covid19 Juanchoz 

Usuarios
 Centros
 Turnos
 Configuración

Crear Centro de Ayuda

Nombre

Dirección Teléfono

Hora de Apertura Hora de Cierre

Tipo de Centro

Municipio

Sitio Web

Correo Electrónico

Protocolo de Vista

☒ Publicado
☐ Pendiente
☐ Aceptado


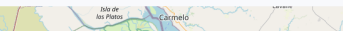

 

Figura 6: Registro de un centro de ayuda.

Donaciones Covid19 Juanchoz 

Usuarios
 Centros
 Turnos
 Configuración

Actualizar Centro de Ayuda

Nombre

Dirección Teléfono

Hora de Apertura Hora de Cierre

Tipo de Centro

Municipio

Sitio Web

Correo Electrónico

Protocolo de Vista

☐ Publicado
☒ Rechazado
☐ Aceptado
☐ Eliminar Protocolo de Vista








 

Figura 7: Modificar un centro de ayuda.

Donaciones Covid19 Juanchoz 

-  Usuarios
-  Centros
-  Turnos
-  Configuración

Centro Cafe

Dirección: Av 200
Telefono: +54 294 410-2020

Turnos + Solicitar turno

Email

Elija alguno de los siguientes emails ▾


Limpiar Filtros
Buscar





10/02/2021, 15:30
centro 1 laura
maru@gmail.com
+54 221 458-6587

+

1

Figura 8: Listado de turnos para un centros de ayuda particular.

Donaciones Covid19 Juanchoz 

-  Usuarios
-  Centros
-  Turnos
-  Configuración

Solicitar turno del Centro Cafe

Dirección: Av 200
Telefono: +54 294 410-2020

Correo Electrónico


Correo Electrónico

Telefono del donante

telefono donante


Fecha

dd/mm/aaaa



Horario

--:--



Guardar

Figura 9: Solicitar un turno para un centro de ayuda particular.

The screenshot shows the 'Donaciones Covid19' web application. On the left is a vertical sidebar with a blue gradient background containing four menu items: 'Usuarios', 'Centros', 'Turnos', and 'Configuración'. The main content area is titled 'Editar turno del Centro Cafe' and includes the address 'Dirección: Av 200' and phone number 'Telefono: +54 294 410-2020'. Below this is a form with four input fields: 'Correo Electrónico' (containing 'maru@gmail.com'), 'Telefono del donante' (containing '+54 221 458-6587'), 'Fecha' (containing '10/02/2021'), and 'Horario' (containing '15:30'). A blue 'Guardar' button is at the bottom of the form. The top right of the page shows the user 'Juanchuz' with a profile icon.

Figura 10: Modificar un turno para un centro de ayuda particular.

Si accedemos a la pantalla principal de turnos, podemos ver la pantalla de la figura 11. Donde se ve el listado de turnos de hoy y los siguientes dos días. Se pueden modificar y filtrar los turnos.

The screenshot shows the 'Donaciones Covid19' web application with the 'Turnos' section selected in the sidebar. The main content area is titled 'Turnos' and features a search bar with a 'Centro' dropdown and a 'Buscar' button. Below the search bar is an 'Email' dropdown menu with the text 'Elija alguno de los siguientes emails'. A 'Limpiar Filtros' link is also present. The list of shifts shows one entry for 'Centro Cafe' with details: '03/08/2021, 13:00', 'centro@2021.com', and '+54 221 458-2156'. Each entry has a settings icon (gear) to its right. A blue box with the number '1' is at the bottom right of the list. The top right of the page shows the user 'Juanchuz' with a profile icon.

Figura 11: Listado de turnos.

Por último, en la barra lateral podemos acceder a las configuraciones, se puede ver en la figura 12. Donde se pueden configurar los parámetros que ahí aparecen.

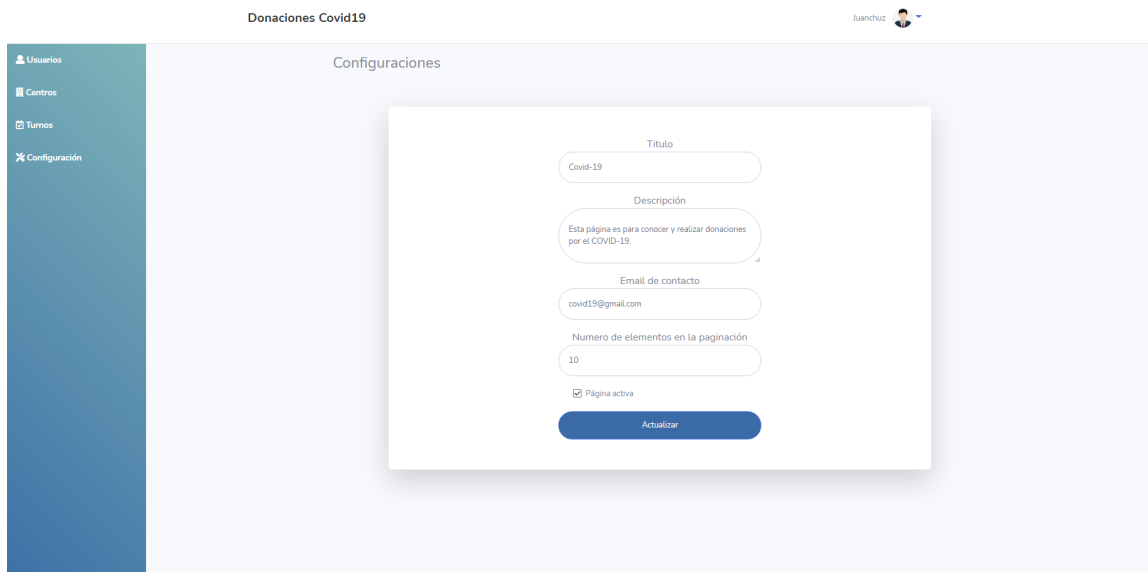


Figura 12: Configuraciones de la aplicación.

Si accedemos al perfil de usuario, podemos ver como se ve en la figura 13.

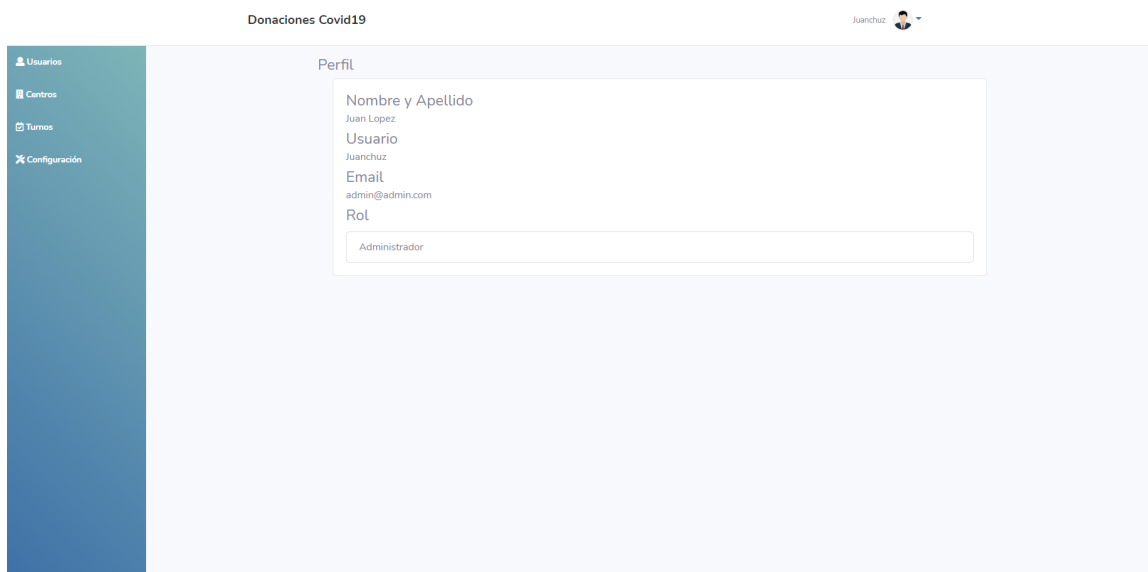


Figura 13: Perfil del usuario.

Si por último en la aplicación privada, accedemos con la cuenta de un usuario con rol de operador de centro de ayuda, podemos ver la figura 14. Donde se redirige a la página principal de los centros de ayuda además de poseer varias funciones y accesos limitados.

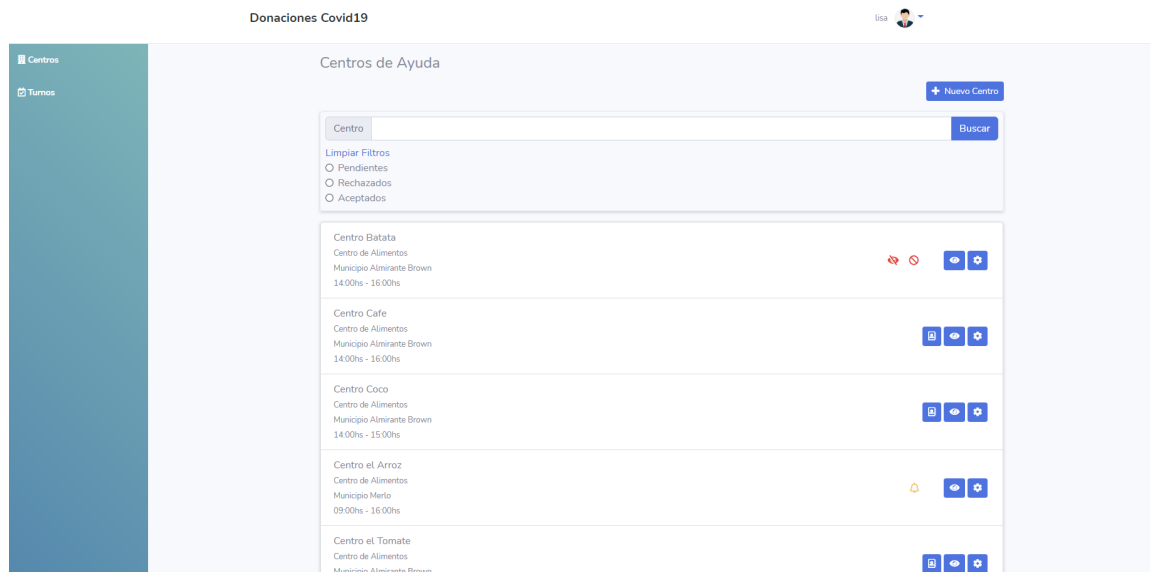


Figura 14: Vista usuario rol operador.

Ahora analizaremos la aplicación pública. Si accedemos al **link** veremos la pantalla que se muestra en la figura 15. Donde se puede acceder a las funciones de: buscar un centro de ayuda, solicitar un turno, solicitar centro, estadísticas e iniciar sesión en la aplicación privada. En la figura 16 se puede ver como luce el mapa, para buscar centros de ayuda. En la figura 17 se puede ver, el formulario para solicitar un turno en un centro particular. En la figura 18 se puede ver para solicitar un centro de ayuda. Por último, en la figura 19 se pueden ver las estadísticas.

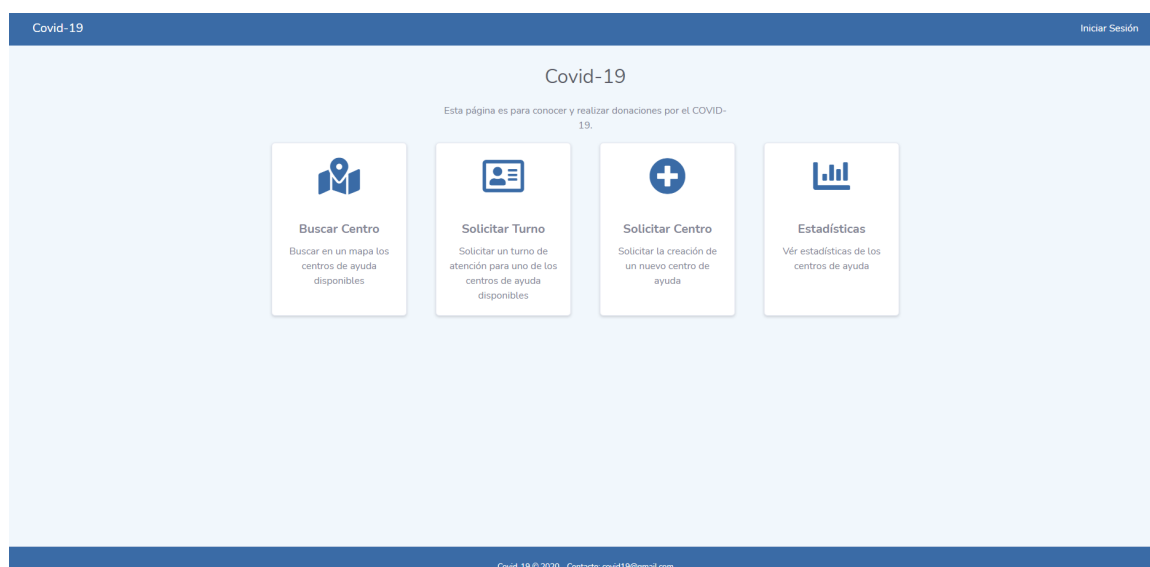


Figura 15: Pantalla principal aplicación publica.

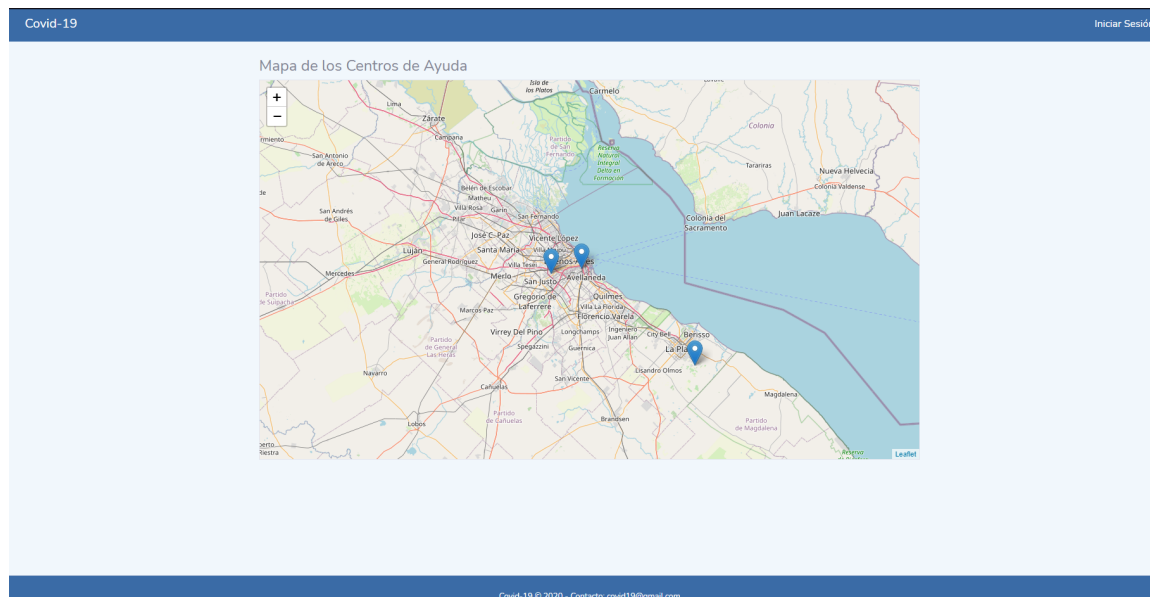


Figura 16: Mapa para buscar centros de ayuda.

Covid-19 Iniciar Sesión

Solicitar turno para el Centro Cafe

Los campos con (*) son obligatorios.

Nombre*	Apellido*
<input type="text" value="nombre"/>	<input type="text" value="apellido"/>
Email*	Telefono donante
<input type="text" value="email"/>	<input type="text" value="telefono donante"/>
Fecha del turno a solicitar*	
<input type="text" value="02/08/2021"/>	<input type="text"/>
Horario del turno a solicitar*	
<input type="text" value="Seleccione un horario disponible"/>	
<input type="button" value="Solicitar"/>	

Covid-19 © 2020 - Contacto: covid19@gmail.com

Figura 17: Solicitar turno.

Covid-19 Iniciar Sesión

Solicitar Centro

Los campos con (*) son obligatorios.

Nombre*

Dirección*

Teléfono*

Hora de Apertura*

--:--

Hora de Cierre*

--:--

Tipo de Centro*

Por favor, seleccione uno

Municipio*

Por favor, seleccione uno

Sitio Web

Correo Electrónico

Localización

Figura 18: Solicitar un centro de ayuda.

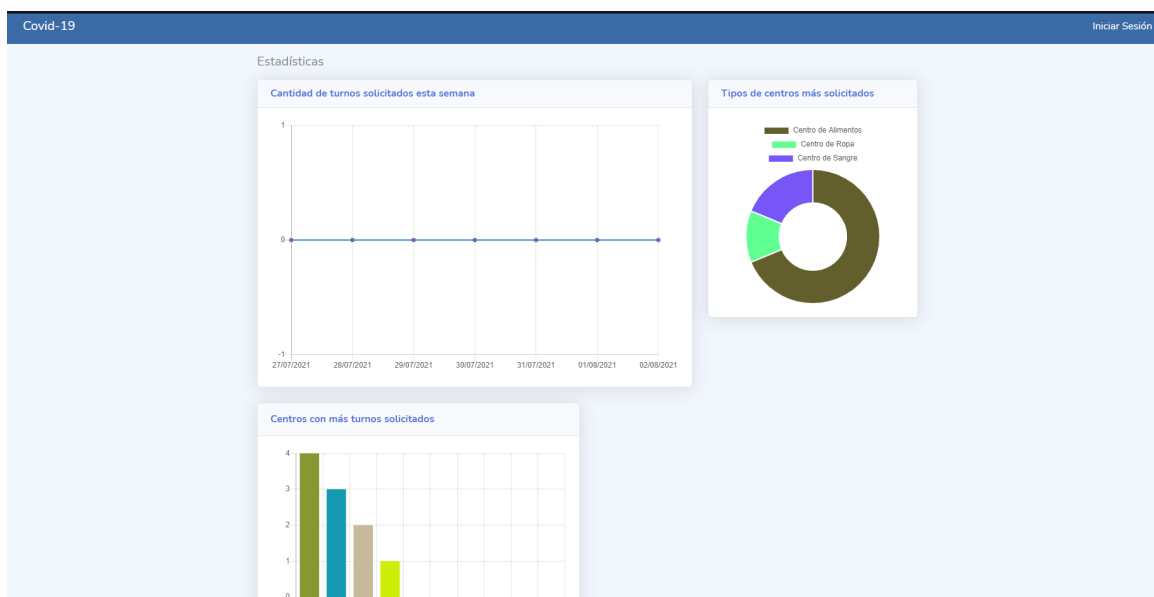


Figura 19: Estadísticas.

10. Anexo 2: guía para el desarrollador

El proyecto realizado consta de 2 partes, la aplicación privada y la aplicación pública. La aplicación privada fue realizada con Python (versión 3.6). Utiliza el framework Flask como interfaz web. Todas las dependencias (librerías / frameworks) que requiere la aplicación privada se encuentran especificadas en el archivo requirements.txt. Se utilizó MariaDB como motor de base de datos. Para su desarrollo, se utilizó la arquitectura Modelo, Vista, Controlador (MVC).

Rutas importantes dentro del directorio del proyecto:

- **Modelos:** app/models.
- **Plantillas de la vista:** app/templates.
- **Contronroladores:** app/resources.
- **Rutas finales del sistema (urls):** config/routes.
- **Auxiliares:** app/helpers.
- **Formularios:** app/helpers/forms.
- **Semillas para la base de datos:** seeds.

Se creó una API para comunicar la aplicación privada con la publica. La documentación de la misma está en el README.md.

La aplicación pública fue realizada con javascript. Se utilizó el framework de Vuejs (en su versión 2.6.11) como interfaz web. Todas las dependencias (librerías) que requiere la aplicación pública, se encuentran especificadas en el archivo package.json.