

**Bobadilla, Manuel**

---

**Sistema voluntario UCC: sistema  
informático para la gestión de los  
voluntariados del VUCC y sus  
actividades**

**Tesis para la obtención del título de grado de  
Ingeniero Civil**

Directores:

Carreño, Ignacio Luciano

Porrini, Federico

Documento disponible para su consulta y descarga en Biblioteca Digital - Producción Académica, repositorio institucional de la Universidad Católica de Córdoba, gestionado por el Sistema de Bibliotecas de la UCC.



[Esta obra está bajo una licencia de Creative Commons Reconocimiento-No Comercial-Sin Obra Derivada 4.0 Internacional.](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Facultad de Ingeniería

Carrera de Ingeniería de Sistemas



Documento de Trabajo Final

## **Sistema Voluntariado UCC**

Sistema informático para la gestión de los voluntariados  
del VUCC y sus actividades

**Autor**

Manuel Bobadilla

**Tutores**

Ing. Ignacio Luciano Carreño

Ing. Federico Porrini

Córdoba, Argentina

Año 2024

# Índice

<b>Índice</b> .....	<b>2</b>
<b>Resumen</b> .....	<b>6</b>
<b>Abstract</b> .....	<b>7</b>
<b>Glosario</b> .....	<b>8</b>
<b>Problemática</b> .....	<b>10</b>
<b>Objetivos</b> .....	<b>11</b>
Objetivos generales.....	11
Objetivos específicos.....	11
<b>Enfoque RSU</b> .....	<b>12</b>
<b>Marco teórico</b> .....	<b>14</b>
Plataforma de implementación.....	14
Aplicación web.....	14
Aplicación móvil.....	14
Aplicación de escritorio.....	15
Arquitecturas de implementación.....	17
Modelo-Vista-Controlador MVC.....	17
Modelo-Vista-Presentador MVP.....	18
Modelo-Vista-VistaModelo MVVM.....	19
Arquitectura en capas.....	19
Arquitectura de microservicios.....	21
Arquitectura monolítica.....	22
Arquitectura de una sola página SPA.....	22
Renderización.....	24
Renderización del lado del servidor SSR.....	24
Renderización estática SSG.....	24
Renderización del lado del cliente CSR.....	25
Frontend.....	26
Electron.....	26
Flutter.....	27
Django.....	28
Angular.....	29
React.....	31
Backend.....	34
ExpressJs - Nodejs.....	34
Flask - Python.....	35
Spring Boot - Java.....	36
Django - Python.....	38
Base de datos.....	41
MySQL.....	41
PostgreSQL.....	42
Maria DB.....	43

Comparativa Rendimiento MySQL - PostgreSQL - MariaDB.....	44
MS SQL Server.....	46
MongoDB.....	47
<b>CMS.....</b>	<b>49</b>
WordPress.....	49
Wagtail.....	50
OpenCMS.....	51
Strapi.....	52
<b>Despliegue.....</b>	<b>54</b>
Render.....	54
Heroku.....	55
Railway app.....	56
<b>Propuesta de solución.....</b>	<b>57</b>
<b>Alcance funcional.....</b>	<b>60</b>
Funcionalidades de contenidos.....	61
Funcionalidades de reportes.....	61
Funcionalidades de actividades.....	61
Funcionalidades de voluntarios.....	62
Funcionalidades de vehículos.....	62
Funcionalidades de coordinadores.....	62
Funcionalidades de directivos.....	63
<b>Solución generada.....</b>	<b>64</b>
Actividades.....	64
Asistencias.....	66
Flex.....	67
Home.....	68
Streams.....	68
Site settings.....	70
Usuarios.....	71
Voluntariados.....	72
Vehículos.....	74
Reinicio.....	75
<b>Módulos importados.....</b>	<b>77</b>
Django All-Auth.....	77
psycpg2-binary.....	77
Whitenoise.....	77
dj-database-url.....	77
Gunicorn.....	78
python-decouple.....	78
<b>Diagrama de base de datos.....</b>	<b>79</b>
<b>Usuarios del sistema.....</b>	<b>80</b>
Usuario anónimo.....	80
Voluntario no validado.....	80
Voluntario validado.....	81

Coordinador voluntariado.....	81
Directivo VUCC.....	82
<b>Pantallas del sistema.....</b>	<b>84</b>
Home page.....	84
Usuario anónimo.....	84
Usuario voluntario, validado y no validado.....	85
Coordinador.....	86
Directivo VUCC.....	87
Voluntariados.....	88
Usuario anónimo - Voluntario (tanto no validado como validado).....	88
Coordinador.....	89
Directivo VUCC.....	90
Asistencias voluntarios.....	91
Coordinadores - Directivos del VUCC.....	91
Vehículos.....	92
Voluntarios validados - Coordinadores - Directivos VUCC.....	92
Asistencias.....	93
Voluntarios validados y no validados - Coordinadores - Directivos VUCC.....	93
Boletines.....	94
Voluntarios no validados.....	94
Voluntarios validados - Coordinadores - Directivos VUCC.....	94
Donaciones.....	96
Todos los usuarios.....	96
Preguntas frecuentes.....	97
Todos los usuarios.....	97
Login.....	98
Usuario anónimo.....	98
Registro.....	99
Usuario anónimo.....	99
Actividades.....	100
Usuario anónimo.....	100
Voluntario no validado.....	101
Voluntario validado.....	101
Coordinador - Directivo del VUCC.....	102
Inscriptos actividad.....	104
Coordinador - Directivo VUCC.....	104
Tomar asistencia.....	105
Coordinadores - Directivos VUCC.....	105
Historial asistencia por día.....	106
Coordinadores - Directivos del VUCC.....	106
Historial asistencia por voluntario.....	107
Coordinadores - Directivos VUCC.....	107
Inscripciones voluntariado.....	108
Coordinadores - Directivos VUCC.....	108

Inscripciones coordinadores voluntariado.....	109
Directivo VUCC.....	109
Panel administración.....	110
Directivos VUCC.....	110
Páginas.....	111
Directivos VUCC.....	111
Imágenes.....	113
Directivos VUCC.....	113
Voluntarios.....	114
Directivos VUCC.....	114
Voluntariados.....	116
Directivos VUCC.....	116
Usuarios.....	118
Directivos VUCC.....	118
Contactos redes sociales.....	121
Directivos VUCC.....	121
<b>Conclusión.....</b>	<b>122</b>
<b>Bibliografía.....</b>	<b>123</b>

## Resumen

En el ámbito de la Universidad Católica de Córdoba se encuentra el VUCC (Voluntariado Universidad Católica de Córdoba), es el área encargada de coordinar y gestionar el desarrollo de las actividades por parte de los diversos equipos de voluntariados que se encuentran en el marco de la universidad.

Sin un sitio web propio, poseen poca difusión de las propuestas, tanto de voluntariados, como de las actividades disponibles y de los contactos para la participación en los mismos, lo que disminuye el alcance efectivo de dichas propuestas dentro de la comunidad universitaria.

Además, las actividades administrativas desarrolladas dentro de los voluntariados se realizan de forma manual, descentralizada y autorregulada. Lo que provoca una alta carga de trabajo en las tareas operativas dentro de los voluntariados, informalidad y propicia la generación de errores. Disminuyendo también la transparencia de los procesos, tanto para los voluntarios como para los coordinadores y directivos del área de voluntariados.

Ante esto, se desarrolla un sistema web capaz de hacer frente a las necesidades operativas del área de voluntariados UCC, dando soporte a los procesos claves desarrollados en dicha área. También permitiendo la presentación y exposición de las diversas propuestas disponibles e informaciones relevantes.

El sistema propuesto puede desglosarse en dos partes. Por un lado, referida a la exposición de información, haciendo frente a la problemática de la difusión de las propuestas existentes. Para esto, se implementó una solución CMS (sistema de gestión de contenidos), donde se pone foco en la fácil manipulación de la información expuesta en la página, permitiendo que usuarios no técnicos puedan manipular y actualizar los contenidos del sitio web de una manera fácil, intuitiva y eficiente. Y en segunda parte, el conjunto de soluciones para hacer frente a los procesos de negocios desarrollados en los voluntariados, permitiendo la formalización y automatización de los mismos, centralizando la información y facilitando el acceso a los datos. Aumentando la eficiencia en el desarrollo de las actividades de los diversos voluntariados, facilitando la obtención de diversos datos clave y aliviando la carga en la realización de las tareas rutinarias que se llevan a cabo.

## **Abstract**

In the context of the Universidad Católica de Córdoba, the VUCC (Voluntariado Universidad Católica de Córdoba) is the area responsible for coordinating and managing the development of activities by various volunteerings associated with the university.

Without its own website, there is limited dissemination of proposals, both for volunteering opportunities and available activities, along with contact information for participation, reducing the effective reach of these proposals within the university community.

Furthermore, administrative activities within the volunteerings are carried out manually, in a decentralized and self-regulated manner. This leads to a high workload in operational tasks, informality, and the likelihood of errors. It also reduces transparency in processes for both volunteers and the coordinators and directors of the volunteer area.

In response to these challenges, a web system is developed to address the operational needs of the UCC volunteering area, providing support for the key processes in the area. It also enables the presentation and exhibition of various available proposals and relevant information.

The proposed system can be divided into two parts. Firstly, focusing on information exposure to tackle the issue of disseminating existing proposals, a CMS (Content Management System) solution is implemented. This emphasizes easy manipulation of the information displayed on the page, allowing non-technical users to handle and update website content easily, intuitively, and efficiently. Secondly, a set of solutions addresses the business processes within the volunteerings, allowing formalization and automation. This centralizes information and facilitates data access, increasing efficiency in the development of activities within the various volunteerings, facilitating the acquisition of diverse key data, and easing the workload in the execution of routine tasks that are carried out.

## Glosario

VUCC: Área de gestión de los equipos de voluntariado en el ámbito de la Universidad Católica de Córdoba.

Cliente: Se refiere a un dispositivo o software que solicita servicios, recursos o información desde otro dispositivo o programa llamado servidor.

Servidor: Es un componente que proporciona servicios, recursos o información a otros dispositivos o programas llamados clientes. Los servidores están diseñados para atender las solicitudes de los clientes.

Framework: Es un conjunto estructurado de herramientas, bibliotecas y convenciones que proporciona una base para el desarrollo de software. Su propósito es facilitar y agilizar el proceso de creación de aplicaciones al ofrecer un marco de trabajo común y predefinido.

Tiempo hasta el primer byte (TTFB): Se considera el tiempo que transcurre entre el click en un vínculo y la primera parte del contenido que llega. Es una métrica que mide el tiempo que transcurre entre la solicitud de un recurso y el momento en que comienza a llegar el primer byte de una respuesta (Jeremy Wagner y Barry Pollard, s. f., sección "TTFB").

First Contentful Paint (FCP): Es el momento en que se hace visible el contenido solicitado (cuerpo del artículo, etcétera), mide el tiempo que transcurre desde que el usuario navegó a la página por primera vez hasta que se renderiza en la pantalla cualquier parte del contenido de la página (Philip Walton, s. f., sección "FCP").

Interaction to Next Paint (INP): Se trata de una métrica representativa que evalúa si una página responde de forma coherente y rápida a las entradas del usuario. Cuando una interacción hace que una página no responda, se considerará una mala experiencia del usuario. El INP observa la latencia de todas las interacciones que un usuario realiza con la página y muestra un solo valor que corresponde a todas (o casi todas) las interacciones. Un INP bajo significa que la página pudo responder rápidamente a todas (o a la gran mayoría) de las interacciones del usuario de manera constante (Jeremy Wagner, s. f., sección "INP").

Tiempo de bloqueo total (TBT): Es una métrica proxy para INP, que calcula la cantidad de tiempo que se bloqueó el subproceso principal durante la carga de la página, mide la cantidad de tiempo total después de la primera pintura con contenido (FCP) durante la cual el subproceso principal se bloqueó por el tiempo suficiente para evitar la respuesta de la entrada. El subproceso principal se considera "bloqueado" cada vez que hay una tarea que se ejecuta en el subproceso principal durante más de 50 milisegundos (Philip Walton, s. f., sección "TBT").

Polyfill: Es un fragmento de código (generalmente JavaScript en la Web) que se utiliza para proporcionar una funcionalidad moderna en navegadores antiguos que no lo admiten de forma nativa.

Backend: Se refiere a la parte del sistema que se encarga del procesamiento y almacenamiento de datos, así como de la lógica de negocio

Frontend: Es la parte de la aplicación o sistema que interactúa directamente con los usuarios finales. Es la interfaz de usuario visible y la experiencia que tienen los usuarios al interactuar con una aplicación, sitio web o cualquier otro software.

AOT: AOT, de sus siglas Ahead-of-Time, se refiere a un enfoque de compilación en el que el código fuente de un programa se traduce en código ejecutable de manera anticipada, antes de que el programa se ejecute. Esto es contrario al enfoque JIT (Just-In-Time)

JIT: JIT, de sus siglas Just-in-Time, es un enfoque de compilación en el que el código fuente de un programa se traduce a código ejecutable durante el tiempo de ejecución.

Motor de plantillas: Un motor de plantillas es una herramienta que permite incrustar datos dinámicos en archivos de plantillas estáticas, produciendo así contenido HTML dinámico para ser enviado al navegador

ORM: Un ORM, de sus siglas Object-Relational Mapping es una herramienta o framework que facilita la interacción entre un sistema de gestión de bases de datos relacional (como MySQL, PostgreSQL o SQLite) y un lenguaje de programación orientado a objetos.

## Problemática

La carencia del VUCC de un sitio web propio dificulta la difusión de las diversas propuestas existentes, tanto en cuanto a los voluntariados disponibles, como también las actividades de dichos voluntariados, informaciones relevantes y datos de contacto de los directivos del VUCC y de los miembros coordinadores de los equipos de voluntariado.

Además de la carencia de un medio de exposición de información previamente descrito, tampoco se cuenta con un sistema para hacer soporte a los procesos claves que se desarrollan dentro del VUCC, lo que lleva a que, tanto coordinadores, como directivos, destinen una gran cantidad de tiempo en la realización de actividades operativas de forma manual, desperdiciando esfuerzos que podrían destinarse en la realización de actividades más enriquecedoras en cuanto a su impacto social, objetivo principal del VUCC. Esta realización de las actividades de forma manual también propicia la generación de errores a la hora de realizar las mismas. Por otra parte, como resultado de las características mencionadas anteriormente, sobre el modo en el que se llevan a cabo los procedimientos operativos en los voluntariados, la información derivada de estos procesos, como por ejemplo, los registros de asistencia de los voluntarios, no se encuentra consolidada en un lugar de fácil acceso, donde la información se encuentre disponible para ser expuesta de una forma adecuada.

En cuanto a los voluntarios que desarrollan las actividades, surgen inconvenientes en cuanto a la capacidad de relevar sus registros de asistencias generados a lo largo del tiempo, producto de la descentralización e informalidad de dichos registros.

En consecuencia, a nivel organizacional, puede observarse una ineficiencia en el desarrollo de las actividades operativas, lo que genera una carga de trabajo innecesaria desde el punto de vista del impacto generado por los voluntariados. También se generan inconvenientes producto de este desarrollo manual y auto-gestionado de las actividades, en donde la información generada debe ser reprocesada para ser utilizada por parte de los directivos del VUCC en sus tareas de gestión, lo cual, también conlleva una pérdida de tiempos y esfuerzos. Los voluntarios se ven afectados al no poder consultar por sí mismos el estado de sus registros de asistencias, inscripción a los voluntariados y actividades.

Ante esta problemática se plantea el desarrollo de un sistema web capaz de dar respuesta a dichos inconvenientes de forma eficaz y eficiente.

## Objetivos

Para solventar las problemáticas planteadas, se propone la implementación de un sistema web en base a los siguientes objetivos.

### Objetivos generales

- Mejorar la experiencia por parte de los voluntarios que desarrollan actividades en el VUCC.
- Mejorar la capacidad de difusión de las propuestas e información relevante por parte del VUCC.
- Alivianar la carga de trabajo en actividades operativas para los coordinadores de los voluntariados y directivos del VUCC.

### Objetivos específicos

- Desarrollar un sistema web cuyos contenidos sean autogestionables por parte del VUCC.
- Implementar en el sistema web los procesos operativos de los voluntariados para la automatización de los mismos.
- Centralizar la información operativa del VUCC.
- Exponer la información de forma adecuada y oportuna a quien corresponda.

## Enfoque RSU

Como parte del proyecto, se tuvo en consideración el impacto social del mismo y su vinculación con la política de RSU que asume la Universidad Católica de Córdoba, en esta sección se realiza dicho análisis.

Secretaría de Proyección y Responsabilidad Social Universitaria. (s.f.). "La RSU es aquella habilidad y efectividad de la Universidad para responder a las necesidades de transformación de la sociedad donde está inmersa, mediante el ejercicio de sus funciones sustantivas: docencia, investigación, extensión y gestión interna." Recuperado de <https://www2.ucc.edu.ar/proyeccion/secretaria-de-proyeccion-y-responsabilidad-social-universitaria/>.

La RSU tiene cinco impactos.

- Educativo
- Cognoscitivo y epistemológico
- Social
- Organizacional o interno
- Ambiental

El trabajo final desarrollado se relaciona principalmente con el cuarto impacto (organizacional o interno), debido a que se aumenta la eficiencia, y optimiza los procesos internos desarrollados dentro del marco de los voluntariados UCC, brindando una herramienta que, entre otras cosas, automatiza, formaliza y facilita las actividades organizacionales y de gestión propia de cada equipo de voluntariado. Además de, a través de esta automatización y formalización, disminuir la posibilidad de errores y confusiones producto de la ejecución manual.

Este sistema permitirá realizar de forma automática las listas de asistencia a las actividades; asistirá en la creación de los registros de asistencias; soportará la alta-baja-modificación de voluntarios, con su información relevante para el VUCC, en el sistema; permitirá gestionar la inscripción de los voluntarios con los diversos voluntariados; permitirá la generación automática, a demanda, de reportes de asistencia según distintos criterios de agregación, para cubrir la necesidad de información; enfocada en la fácil gestión de los voluntariados y sus actividades en el sistema. Todas estas actividades se desarrollaron teniendo en mente que sean simples y rápidas para el usuario final, aportando

una herramienta que facilite el desarrollo de las tareas operativas y rutinarias que se llevan a cabo en los voluntariados y por parte de la dirección del VUCC, disminuyendo la carga de trabajo existente en la realización de las mismas.

Para esto, se trabajó a la par con un equipo de voluntariado (Voluntariado Barrio el Milagro) y la actual directora del VUCC (Graciela Franchini). Mostrando y validando los progresos realizados, siendo sus devoluciones de vital aporte para refinar y ajustar el sistema. A la vez, en cada reunión, obtener nuevos requerimientos y reglas de negocio que se desconocían previamente, logrando desarrollar una solución que cubra sus necesidades y sea producto de un diálogo y participación del beneficiario final durante el proceso de desarrollo.

## Marco teórico

### Plataforma de implementación

#### **Aplicación web**

Una aplicación web es un programa informático que se ejecuta en un navegador web y permite a los usuarios realizar diversas tareas o acceder a funciones específicas a través de una interfaz de usuario interactiva. Al acceder a través de un navegador web, no es necesario instalar una aplicación adicional, en caso de poseer algún navegador previamente, pero es necesario contar con una conexión a internet para utilizar la aplicación.

En este esquema de aplicación, los datos y la lógica de negocio quedan almacenados en el servidor donde se aloje el sistema. Esto proporciona una serie de beneficios clave, como la accesibilidad desde cualquier lugar con conexión a internet, reduciendo la dependencia del dispositivo del usuario y ofreciendo una experiencia consistente en diferentes plataformas. Pero, por contraparte, sin una conexión a internet no se puede hacer uso del sistema. Además, al ejecutarse en el navegador, las aplicaciones web permiten actualizaciones simultáneas para todos los usuarios, sin requerir intervención manual.

La escalabilidad se ve mejorada gracias a la arquitectura de servidor-cliente, lo que facilita la gestión de un mayor número de usuarios y datos. La menor dependencia del almacenamiento local y la compatibilidad entre plataformas son características que hacen que las aplicaciones web sean flexibles y fáciles de mantener. Asimismo, la rápida implementación de actualizaciones y correcciones, así como la integración con servicios en la nube, contribuyen a una gestión eficiente y segura del sistema.

#### **Aplicación móvil**

Una aplicación móvil es un programa diseñado para ejecutarse en dispositivos móviles, como teléfonos inteligentes. Este tipo de aplicaciones deben descargarse e instalarse directamente en el dispositivo del usuario desde una tienda de aplicaciones específica, como la App Store para dispositivos iOS o Google Play para dispositivos Android.

En este modelo de aplicación, los datos y la lógica de negocio pueden residir tanto en el dispositivo del usuario como en servidores remotos. Esta dualidad permite que algunas funciones se realicen incluso sin conexión a internet, mejorando la experiencia del usuario en entornos sin conectividad. Sin embargo, algunas características pueden depender de una conexión activa para acceder a ciertas funcionalidades o sincronizar datos.

Las aplicaciones móviles destacan al ofrecer una experiencia más centrada y adaptada a las capacidades específicas de los dispositivos móviles, esto producto a su capacidad de integración con cámaras, sensores, funciones de geolocalización, entre otros. Esto permite un mayor grado de personalización y optimización para el tamaño de pantalla y la interfaz táctil, proporcionando una experiencia de usuario más intuitiva y eficiente.

Pueden funcionar de manera más fluida en condiciones de baja conectividad, aprovechando al máximo las capacidades del dispositivo y ofreciendo un rendimiento óptimo.

### **Aplicación de escritorio**

Una aplicación de escritorio es un programa informático diseñado para ejecutarse en computadoras personales, ya sea en sistemas operativos Windows, macOS o Linux. Las aplicaciones de escritorio se instalan directamente en el dispositivo del usuario y generalmente no requieren una conexión a internet para su funcionamiento, aunque algunas funciones pueden depender de la conectividad, al igual que la sincronización con datos que se encuentren alojados en el servidor remoto.

Tanto los datos como la lógica de negocio se almacenan y ejecutan localmente en el dispositivo del usuario. Esto permite un acceso rápido a las funciones, incluso sin una conexión activa a internet. Sin embargo, las actualizaciones y mejoras suelen requerir intervención manual por parte del usuario, ya que no se implementan de manera automática como en el caso de las aplicaciones web.

Este tipo de aplicaciones ofrecen un alto grado de rendimiento y funcionalidad, aprovechando los recursos locales de la computadora, como la

capacidad de procesamiento y almacenamiento. Estas aplicaciones son ideales para tareas que requieren un alto rendimiento gráfico, procesamiento intensivo o interacciones complejas. Por contraparte, suelen estar más limitadas en términos de accesibilidad desde diferentes ubicaciones

## Arquitecturas de implementación

### **Modelo-Vista-Controlador MVC**

La arquitectura Modelo-Vista-Controlador (MVC) es un enfoque organizativo que separa las preocupaciones relacionadas con la gestión de datos, la lógica de negocio y la presentación de la interfaz de usuario. Aunque comúnmente asociada con aplicaciones web, la implementación de MVC puede extenderse a aplicaciones móviles y de escritorio.

La arquitectura MVC organiza la aplicación de la siguiente forma.

- El modelo representa la lógica de negocio y la gestión de datos. Contiene la lógica para acceder, actualizar y manipular los datos de la aplicación. En un entorno no web, los datos pueden almacenarse localmente en el dispositivo del usuario o en un servidor.
- La vista se encarga de la presentación de la interfaz de usuario. En una implementación no web, podría ser una interfaz gráfica de usuario (GUI), una interfaz de línea de comandos (CLI) o cualquier otro medio de presentación. La vista muestra los datos al usuario y recoge la entrada del mismo.
- El controlador actúa como intermediario entre el modelo y la vista. Maneja las solicitudes del usuario, actualiza el modelo según sea necesario y actualiza la vista para reflejar cualquier cambio en los datos. En una aplicación no web, el controlador puede responder a eventos del usuario o incluso a comandos desde una interfaz de línea de comandos.

A diferencia de una aplicación web, en un entorno no web, la conexión entre estas partes de MVC puede ser más directa, sin depender de solicitudes HTTP o navegadores web. Sin embargo, los principios fundamentales de separación de responsabilidades y modularidad siguen siendo válidos.

Los beneficios de implementar MVC en este contexto incluyen la facilidad de mantenimiento, la escalabilidad y la reutilización de código.

### **Modelo-Vista-Presentador MVP**

La arquitectura Modelo-Vista-Presentador es un enfoque organizativo que comparte similitudes con MVC, pero con algunas diferencias en la forma en que se gestionan las interacciones del usuario. Al igual que MVC, MVP también puede ser aplicada en una variedad de contextos, incluyendo aplicaciones web, móviles y de escritorio.

La arquitectura MVP organiza la aplicación de la siguiente manera.

- Al igual que en MVC, el modelo en MVP representa la lógica de negocio y la gestión de datos. Es responsable de acceder, actualizar y manipular los datos de la aplicación. Estos datos pueden residir localmente en el dispositivo del usuario o en un servidor remoto.
- La vista en MVP es similar a la vista en MVC y se encarga de la presentación de la interfaz de usuario. La diferencia radica en que la vista en MVP es más pasiva y tiene un conocimiento mínimo del modelo. La vista se limita a mostrar los datos proporcionados por el presentador y a recoger eventos de usuario.
- El presentador en MVP actúa como un intermediario entre el modelo y la vista. A diferencia del controlador en MVC, el presentador en MVP tiene una mayor responsabilidad en la gestión de la lógica de presentación y la coordinación de la interacción del usuario. El presentador actualiza la Vista con los datos del modelo y maneja las acciones del usuario, actualizando el modelo según sea necesario.

En comparación con MVC, MVP busca minimizar la lógica en la vista, evitando que esta tenga un conocimiento profundo del modelo.

MVP promueve la separación de responsabilidades, lo que facilita la realización de pruebas y el mantenimiento del código. La arquitectura MVP también favorece la reutilización de código y la escalabilidad en aplicaciones.

### **Modelo-Vista-VistaModelo MVVM**

La arquitectura Modelo-Vista-VistaModelo es un enfoque organizativo especialmente útil en el desarrollo de aplicaciones con interfaces de usuario ricas y complejas, como aplicaciones móviles y de escritorio, aunque también puede utilizarse en aplicaciones web. Se compone por las siguientes partes.

Similar a las arquitecturas anteriores, el modelo en MVVM representa la lógica de negocio y la gestión de datos. Se encarga de acceder, actualizar y manipular los datos de la aplicación.

La vista en MVVM es responsable de la presentación de la interfaz de usuario, a diferencia de MVC y MVP, la vista en MVVM tiene un conocimiento limitado de la lógica de negocio y se centra principalmente en la presentación de datos.

El componente distintivo de MVVM es la "VistaModelo". Este actúa como un intermediario entre el modelo y la vista. Su función principal es proporcionar a la vista los datos necesarios para su presentación y gestionar las interacciones del usuario. A través de enlaces de datos (data binding), la "VistaModelo" notifica a la vista sobre los cambios en los datos y actualiza automáticamente la interfaz de usuario.

En comparación con MVC y MVP, MVVM busca desacoplar aún más la vista del modelo y facilitar la gestión de la interfaz de usuario. El uso de enlaces de datos permite una conexión más directa entre los datos y la interfaz, eliminando la necesidad de que la vista actualice manualmente los cambios.

### **Arquitectura en capas**

La arquitectura en capas es comúnmente utilizada en el desarrollo de software, independientemente de la plataforma (web, móvil, escritorio). Esta arquitectura organiza la aplicación en capas lógicas, cada una con responsabilidades específicas, facilitando la modularidad, la reutilización de código y el mantenimiento del sistema.

Dependiendo del autor, y las necesidades específicas del sistema, varían levemente la cantidad de capas y denominación de las mismas. Generalmente, se habla de tres capas principales (presentación, lógica de aplicación y acceso a

datos), pero algunas descripciones también incluyen una cuarta capa de infraestructura o servicios comunes.

- La capa de presentación se encarga de la interfaz de usuario y la interacción con el usuario final. En una aplicación web, esta capa sería la interfaz gráfica de usuario implementada generalmente con HTML, CSS y JavaScript. En otras plataformas, puede ser la interfaz de usuario de una aplicación móvil o de escritorio.
- La capa de lógica de aplicación contiene la lógica central de la aplicación y las reglas de negocio. Aquí se procesan las solicitudes del usuario, se realizan operaciones y se gestionan los flujos de trabajo.
- La capa de acceso a datos se encarga de interactuar con el almacenamiento de datos, ya sea una base de datos, servicios web o cualquier otro sistema de persistencia. Aquí se realizan operaciones de lectura y escritura en la base de datos.
- La capa de infraestructura proporciona servicios comunes y utilidades necesarios para el funcionamiento de la aplicación. Puede incluir componentes como la gestión de la seguridad, servicios de registro, manejo de excepciones y otros aspectos transversales a la aplicación.

Entre las características que se destacan de este tipo de arquitectura se encuentra la modularidad, debido a que la aplicación se divide en módulos o capas claramente definidos, lo que facilita el desarrollo y la evolución del sistema; reutilización de código, cada capa puede ser desarrollada y probada de manera independiente, permitiendo la reutilización de componentes en diferentes partes de la aplicación; mantenimiento, ya que los cambios pueden realizarse de manera más sencilla al tener áreas específicas de la aplicación separadas por responsabilidades; la arquitectura en capas permite escalar cada capa por separado, lo que facilita la gestión de un mayor número de usuarios y datos, permitiendo un escalado de la aplicación ajustado a la necesidad de cada capa; también cada capa puede implementarse utilizando tecnologías específicas que mejor se adapten a sus necesidades, siempre y cuando se respeten las interfaces definidas.

## **Arquitectura de microservicios**

La arquitectura de microservicios descompone una aplicación en servicios pequeños, independientes y autosuficientes, conocidos como microservicios. Cada microservicio se centra en una tarea o función específica y se comunica con otros microservicios a través de interfaces bien definidas. A diferencia de las aplicaciones monolíticas, donde todas las funcionalidades y componentes del sistema están integrados en un solo código base y se ejecutan como una única entidad, los microservicios fragmentan la aplicación en diversos microservicios independientes.

Cada microservicio es una unidad autónoma que gestiona su propia lógica de negocio, base de datos e incluso interfaz de usuario de ser necesario. Estos servicios pueden ser desarrollados, implementados y escalados de forma independiente, lo que facilita la evolución continua y la entrega rápida de nuevas funcionalidades.

La comunicación entre microservicios se realiza a menudo a través de protocolos ligeros como HTTP/REST o mensajería. Esto permite una integración eficiente entre los servicios y facilita la construcción de sistemas distribuidos.

Descentraliza la aplicación, lo que significa que cada microservicio puede ser implementado y escalado de manera independiente según la carga de trabajo específica que maneje. Esto mejora la escalabilidad y la resistencia del sistema.

Cada microservicio puede ser desarrollado utilizando tecnologías diferentes, lo que otorga la posibilidad de elegir las herramientas y tecnologías más adecuadas para cada función específica.

Se facilita la escalabilidad y la tolerancia a fallos, ya que un problema en un microservicio no debería afectar a toda la aplicación. La recuperación de errores puede manejarse a nivel de microservicio.

Los beneficios de este tipo de arquitectura son una mayor flexibilidad, agilidad en el desarrollo, fácil mantenimiento, y la capacidad de escalar y desplegar servicios de forma independiente. Sin embargo, hay que tener en cuenta que también introduce desafíos como la gestión de la complejidad y la necesidad de soluciones para coordinar la comunicación entre servicios.

### **Arquitectura monolítica**

La arquitectura monolítica es un enfoque tradicional en el diseño de software donde una aplicación se desarrolla, implementa y ejecuta como una entidad única e integrada. A diferencia de las arquitecturas distribuidas o basadas en microservicios, la arquitectura monolítica consolida todas las funciones y componentes de la aplicación en un solo código base.

Las características más relevantes de la arquitectura monolítica incluyen su integración centralizada, en este enfoque, todos los módulos y componentes de la aplicación están consolidados en un solo programa, permitiendo que la aplicación se ejecute como una unidad cohesiva. La lógica de negocio, interfaz de usuario y capa de datos están interconectadas y se gestionan de manera centralizada en un único entorno de desarrollo. Se utiliza un conjunto homogéneo de tecnologías y lenguajes de programación para toda la aplicación.

Esta centralización también se refleja en el proceso de desarrollo, donde cualquier cambio, adición o actualización afecta al conjunto completo de la aplicación, la implementación de la aplicación monolítica se realiza como una única unidad, lo que implica que cualquier cambio realizado requiere la implementación completa de la aplicación, además, este proceso puede resultar en períodos de inactividad durante la implementación de nuevas versiones.

La escalabilidad en esta arquitectura implica escalar toda la aplicación, incluso si solo una parte específica de la misma experimenta una mayor demanda, esto puede limitar la eficiencia en términos de recursos.

A pesar de las desventajas que se mencionaron previamente en comparación con enfoques más modernos, esta arquitectura puede ser apropiada para aplicaciones pequeñas o proyectos donde la complejidad no es un factor crítico. Su simplicidad y facilidad de desarrollo pueden ser beneficiosas en los contextos apropiados.

### **Arquitectura de una sola página SPA**

La arquitectura de única página (SPA, por sus siglas en inglés, Single Page Application) es un enfoque moderno de desarrollo de aplicaciones web que difiere de las aplicaciones web tradicionales en varios aspectos. En una SPA, la interfaz de

usuario se carga una sola vez en el navegador y, a partir de ese momento, las interacciones del usuario no requieren recargar la página completa.

En una SPA, la carga inicial de la aplicación se realiza de forma única, descargando todos los recursos necesarios, como HTML, CSS y JavaScript, al inicio. Posteriormente, las interacciones del usuario se gestionan mediante solicitudes asíncronas al servidor, actualizando solo las partes específicas de la página que requieren cambios, esto proporciona una experiencia de usuario más fluida y rápida, ya que las transiciones entre secciones de la aplicación se realizan de manera instantánea sin recargar la página, creando una sensación de interactividad continua, similar a las aplicaciones de escritorio. En una SPA, tanto los datos como parte de la lógica de negocio se transfieren al cliente, permitiendo que la aplicación responda de manera ágil a las acciones del usuario sin depender constantemente del servidor para obtener nuevas páginas.

Si bien las SPAs ofrecen ventajas en términos de experiencia de usuario y rendimiento, también pueden requerir un conjunto diferente de habilidades y consideraciones en comparación con las aplicaciones web tradicionales, producto de la utilización de peticiones asíncronas y trasladando parte de la lógica de negocios y gestión de datos en el cliente.

## Renderización

A continuación se plantean las posibles alternativas en cuanto a cómo y dónde debe renderizar la información en el sistema web

### **Renderización del lado del servidor SSR**

La renderización del lado del servidor es aquella en la que el servidor procesa y genera el HTML completo que luego será enviado al cliente para la navegación, evitando así la generación de vistas en el cliente. Por lo general, la renderización del servidor produce un FCP (First Contentful Paint) rápido. La ejecución de la lógica de la página y la renderización en el servidor permiten evitar el envío de mucho JavaScript al cliente. Esto también ayuda a reducir la TBT (Tiempo de bloqueo total) de una página, que también puede reducir el INP (Interaction to Next Paint), ya que el subproceso principal no se bloquea con tanta frecuencia durante la carga de la página. Cuando se bloquea el subproceso principal con menos frecuencia, las interacciones del usuario tendrán más oportunidades de ejecutarse antes, esto tiene sentido, ya que con la renderización del servidor, en realidad solo se envía texto y vínculos al navegador del usuario.

Con el procesamiento en el servidor, es menos probable que los usuarios tengan que esperar a que se ejecute JavaScript en su cliente. Sin embargo, este enfoque tiene una posible desventaja, generar páginas en el servidor requiere tiempo, lo que puede generar un TTFB (Time To First Byte) más alto (Addy Osmani y Jason Miller, s. f., sección "Renderización").

Además, entre las desventajas que presenta este enfoque es la mayor carga y demanda de recursos por parte del servidor, esto debido a las propias características de este esquema de renderización, que carga en mayor medida al servidor, también, puede encontrarse inapropiado en casos donde se requiera un alto grado de interactividad y cambios rápidos de interfaz.

### **Renderización estática SSG**

Con renderizado estático, es necesario generar un archivo HTML por cada página que el usuario pueda acceder de manera anticipada. Tiene la ventaja de ser capaz de responder a una petición de manera muy rápida, producto a que el HTML se encuentra prefabricado, pero, en casos donde no se pueda predecir de antemano la solicitud del cliente, o, cuando el contenido a mostrar tiene una gran cantidad de

contenido dinámico generado por los usuarios, se presentan las limitaciones de este tipo de implementaciones.

Este enfoque ofrece un FCP rápido, además de un INP y TBT bajos, a diferencia de la renderización del servidor, también logra un TTFB rápido y coherente, ya que no es necesario que el HTML de una página se genere de forma dinámica en el servidor (Addy Osmani y Jason Miller, s. f., sección “Renderización”).

### **Renderización del lado del cliente CSR**

El procesamiento del lado del cliente implica renderizar páginas directamente en el navegador con JavaScript. La lógica, la recuperación de datos, las plantillas y el enrutamiento se controlan en el cliente y no en el servidor.

La renderización del cliente puede ser difícil de obtener y de mantener rápida para los dispositivos móviles. Si se realiza un trabajo adecuado, la renderización del cliente puede alcanzar el rendimiento de la renderización pura del servidor, lo que presupone un monto de JavaScript limitado y proporcionar valor en la menor cantidad posible de recorridos de ida y vuelta con el servidor.

La principal desventaja de la renderización del lado del cliente es que la cantidad de JavaScript necesaria tiende a aumentar a medida que crece una aplicación, lo que puede tener efectos negativos en el INP de una página. Esto se vuelve especialmente difícil con la incorporación de nuevas bibliotecas de JavaScript, polyfills y código de terceros, que compiten por la potencia de procesamiento y, a menudo, deben procesarse antes de que se pueda renderizar el contenido de una página, empeorando la experiencia de usuario (Addy Osmani y Jason Miller, s. f., sección “Renderización”).

En contramano con SSR, se le exige menos procesamiento al servidor, producto de la renderización en el cliente, pero, producto de la naturaleza de la solución, es más exigente computacionalmente en los dispositivos del usuario.

## Frontend

En esta sección se contendrán las alternativas de interfaz de usuario de las aplicaciones de escritorio, aplicaciones web y aplicaciones móviles, siendo estas las alternativas de plataformas de implementación descritas en su apartado.

### Electron

Electron es un framework de código abierto que permite a los desarrolladores crear aplicaciones de escritorio multiplataforma utilizando tecnologías web estándar como HTML, CSS y JavaScript.

Electron destaca porque permite a los desarrolladores aprovechar sus habilidades en tecnologías web para construir aplicaciones de escritorio. Puede utilizarse HTML para la estructura, CSS para el diseño y JavaScript para la lógica de la aplicación. Otra de las ventajas principales de Electron es su capacidad para construir aplicaciones que son ejecutables en múltiples plataformas, incluyendo Windows, macOS y Linux, esto facilita el desarrollo de aplicaciones consistentes para diversos sistemas operativos.

Además, al basarse en Node.js, permite a los desarrolladores utilizar JavaScript tanto en el frontend como en el backend de la aplicación, esto facilita la creación de aplicaciones de escritorio al poder trabajar de ambos lados con la misma tecnología.

Sigue una arquitectura de múltiples procesos derivado de Chromium, esto luego repercute en el desarrollador de la aplicación, que controla dos tipos de procesos, el proceso principal y los procesos de renderizado (Electron, s. f., sección "Process Model"). *"Cada aplicación de Electron tiene un único proceso principal, que actúa como el punto de entrada a la aplicación. (...) El propósito primario del proceso principal es el de crear y gestionar ventanas de la aplicación (...)"* (Electron, s. f., sección "Process Model", traducción propia). Mientras que, por cada ventana, se encuentra un proceso de renderizado que se encarga de la renderización de la misma.

Integra Node.js, permitiendo a los desarrolladores acceder a funcionalidades de sistema operativo y APIs nativas. Esto facilita la interacción con el sistema de

archivos, operaciones de red, y otras funciones que no están disponibles directamente en un entorno de navegador

Electron proporciona herramientas para empaquetar la aplicación y distribuirla como un archivo ejecutable. Esto simplifica el proceso de distribución de aplicaciones de escritorio sin depender de un navegador externo (Electron, s. f., sección “Application Packaging”).

Una cualidad también importante es la comunidad de Electron, se trata de una comunidad activa, y hay una amplia gama de módulos y complementos disponibles para extender la funcionalidad de las aplicaciones. Además, existen herramientas de desarrollo que facilitan la creación y prueba, pudiendo utilizar las herramientas de debugging para desarrolladores proporcionadas por Google (Electron, s. f., sección “Application Debugging”).

Electron también facilita la implementación de actualizaciones automáticas en las aplicaciones. Los usuarios pueden recibir nuevas versiones sin tener que descargar e instalar manualmente las actualizaciones (Electron, s. f., sección “Updating Applications”).

## **Flutter**

Flutter es un framework de código abierto desarrollado por Google que se utiliza para la creación de interfaces de usuario (UI) multiplataforma (Wikipedia, s. f., sección “Flutter”). Esta tecnología está especialmente diseñada para el desarrollo de aplicaciones móviles, aunque también se puede utilizar para construir aplicaciones web, de escritorio e incluso en embebidos, lo que le otorga una gran versatilidad (Flutter, s. .f, sección “Plataformas”).

Flutter utiliza Dart como lenguaje de programación principal. Dart es un lenguaje moderno, orientado a objetos y de tipado estático, también. Su sintaxis es familiar con respecto a la de otros lenguajes de programación como Java o JavaScript.

Una de las principales fortalezas de Flutter radica en su capacidad para crear aplicaciones multiplataforma de alta calidad, con un solo código base, los desarrolladores pueden crear aplicaciones para iOS, Android, web y escritorio. Facilita la gestión y el mantenimiento del código, reduciendo el tiempo y los

esfuerzos necesarios para desarrollar y mantener aplicaciones en diferentes plataformas.

Flutter utiliza un enfoque de desarrollo basado en widgets, que son elementos de interfaz de usuario como botones, contenedores y listas. Estos widgets son altamente personalizables y se pueden combinar y anidar para construir interfaces complejas. La flexibilidad de los widgets permite a los desarrolladores crear diseños atractivos y personalizados. Además, pueden encontrarse un amplio catálogo de widgets ya generados y puestos a disposición para su uso en la comunidad (Flutter, s. f., sección “Widgets”).

Utiliza el motor de renderizado Skia, conocido por su alto rendimiento y capacidad para proporcionar interfaces de usuario fluidas y rápidas. Flutter utiliza compilación AOT (Ahead of Time), lo que significa que el código se compila directamente a código nativo para la plataforma de destino, mejorando el rendimiento de la aplicación, pero, en etapas de desarrollo de la aplicación, puede disponerse de una compilación JIT (Just in Time), en donde se traduce y ejecuta el código en tiempo de ejecución, a medida que es necesario. Estas características le otorgan un gran dinamismo a Flutter (Dart, s. f., sección “Compilación”).

Una de las características más destacadas de Flutter es su función Hot Reload. Esto permite a los desarrolladores ver los cambios en tiempo real mientras están programando, sin tener que reiniciar la aplicación por completo. Facilita la iteración rápida y la corrección de errores de manera eficiente, esto producto a que se inyecta código fuente actualizado en la máquina virtual de Dart que se encuentra en ejecución (Flutter, s. f., sección “Hot reload”).

Flutter cuenta con una amplia comunidad activa y en crecimiento de desarrolladores que comparten conocimientos, bibliotecas y recursos. Esto proporciona un entorno colaborativo y ayuda a mantener actualizadas las herramientas y bibliotecas relacionadas con Flutter.

## **Django**

Django, un framework de desarrollo web de alto nivel en Python, se destaca principalmente en lo referido al desarrollo del backend en aplicaciones web, no obstante, la versatilidad de Django se extiende más allá de sus funciones backend, permitiendo su uso también en el desarrollo frontend de aplicaciones web. Esta

capacidad se fundamenta en su compatibilidad con tecnologías ampliamente utilizadas en la generación de interfaces web, tales como HTML y CSS para la estructura y diseño, así como JavaScript para la implementación de la lógica en el cliente.

El framework ofrece un enfoque integral al proporcionar sus propias plantillas, conocidas como "Django templates", que permiten la creación dinámica de estructuras HTML, CSS y JavaScript basadas en los datos obtenidos del backend. Este enfoque facilita la construcción de aplicaciones web completas utilizando un solo framework, simplificando el proceso de desarrollo y mantenimiento.

Otra característica de Django, en su enfoque predeterminado, es que es un framework de desarrollo web que sigue la arquitectura del lado del servidor (SSR, Server-Side Rendering). Esto significa que las páginas web se generan en el servidor y se envían al cliente ya renderizadas. El cliente recibe el HTML completo, lo muestra y, a medida que el usuario interactúa, se realizan solicitudes adicionales al servidor para obtener y mostrar datos adicionales.

Si bien Django posee las capacidades de integrarse con otros frameworks y librerías para el apartado del frontend, quedando así relegado al apartado del backend más estrictamente. Esta opción no se desarrollará en esta sección, pues aquí se aborda únicamente lo referido a las capacidades de Django en cuanto a generar y administrar interfaces de usuario.

Es crucial tener en cuenta que, si bien Django ofrece las herramientas para el desarrollo frontend en aplicaciones web, estas pueden ser consideradas más limitadas y menos especializadas en comparación con opciones dedicadas exclusivamente a este apartado, como podrían ser Angular o React, entre otras.

## **Angular**

Angular es un framework basado en componentes para el desarrollo frontend, donde estos componentes son la pieza básica para la creación de aplicaciones en Angular. Diseñado para facilitar la creación de aplicaciones web dinámicas y robustas, Angular ha ganado popularidad por su enfoque modular, su arquitectura basada en componentes y su capacidad para manejar aplicaciones de una sola página (SPA) (Angular, s. f., sección "Bases Angular"). Actualmente es una

de las tecnologías más utilizadas para el desarrollo frontend y de interfaces de usuario.

Angular se basa en TypeScript, un superset de JavaScript que agrega características de tipado estático y otras mejoras al lenguaje. Esto proporciona a los desarrolladores herramientas avanzadas para detectar errores y mejorar la calidad del código (TypeScript, s. f., sección “TypeScript”).

Como ya se mencionó, Angular se basa en una arquitectura de componentes, donde la interfaz de usuario se construye mediante la composición de componentes reutilizables. Los componentes son bloques de construcción modulares que contienen tanto la lógica como la presentación de una parte específica de la interfaz de usuario, esto favorece la reutilización y mantenibilidad del código generado, ahorrando trabajo a los desarrolladores.

También ofrece un enlace de datos bidireccional, lo que significa que los cambios en el modelo afectan automáticamente la vista y viceversa, sincronizando la información de la interfaz con el componente. Esto simplifica la gestión del estado de la aplicación y mejora la coherencia entre los datos y la interfaz de usuario (infragistics, s. f., sección “Two way binding”).

Angular utiliza un sistema de inyección de dependencias que facilita la organización y la gestión de componentes y servicios. Esto promueve la modularidad y la reutilización del código, así como facilita las pruebas unitarias. La inyección de dependencias es un patrón de diseño que permite a un objeto obtener los recursos que necesita de otros objetos sin tener que crearlos o gestionarlos directamente. Se utiliza para proporcionar a los componentes, directivas y servicios los recursos que necesitan para funcionar (Angular, s. f., sección “Inyección de dependencias”).

Angular proporciona directivas (Angular, s. f., sección “Directivas”) para extender el HTML con comportamientos personalizados y pipes (Angular, s. f., sección “Pipes”) para transformar datos en la vista. Esto permite una manipulación más avanzada y dinámica de la interfaz de usuario.

Un punto a favor es que cuenta con una comunidad activa de desarrolladores y es respaldado por Google, lo que garantiza un constante desarrollo, actualizaciones de seguridad y soporte a largo plazo.

En Angular, de manera predeterminada, la renderización es del lado del cliente (CSR), pudiendo modificarse, en caso de ser necesario, a un esquema de renderización en el servidor (SSR)

Angular es una elección sólida para el desarrollo frontend, ofreciendo una estructura organizada, herramientas potentes y una amplia comunidad de desarrolladores que respaldan su evolución y mejora continua. Su enfoque en la creación de aplicaciones escalables y mantenibles lo hace especialmente adecuado para proyectos complejos y de gran envergadura.

Como desventaja, puede ser innecesariamente robusto para proyectos pequeños o aplicaciones simples, donde los beneficios de Angular puedan quedar injustificados frente a opciones más ligeras y ágiles.

Aborda lo referido al frontend, por lo que es necesario plantear una tecnología aparte para afrontar el backend de la aplicación.

## **React**

React es una biblioteca de JavaScript desarrollada y mantenida por Facebook. Se ha convertido en una opción popular para el desarrollo frontend de aplicaciones web debido a su enfoque declarativo y su capacidad para construir interfaces de usuario interactivas y eficientes.

Al basarse en un enfoque declarativo, los desarrolladores describen cómo debería lucir la interfaz de usuario en función del estado de la aplicación. Está estructurado en torno a componentes, que son bloques modulares que encapsulan la lógica y la presentación. Esta abstracción facilita la construcción y el mantenimiento de interfaces complejas, además fomenta la reutilización de código al utilizar dichos componentes repetidas veces a lo largo de la aplicación (React, s. f., sección "React").

React utiliza un Virtual DOM (DOM virtual) para optimizar las actualizaciones en la interfaz de usuario. En lugar de actualizar directamente el DOM cada vez que cambia el estado, React compara el Virtual DOM con el DOM real y realiza solo las actualizaciones necesarias (reactjs, s. f., sección "React DOM"). Esto mejora

significativamente el rendimiento de la aplicación. De forma predeterminada, el modo de renderización es en el lado del cliente (CSR)

Además, utiliza JSX, una extensión de JavaScript que permite escribir código HTML en archivos JavaScript. Esto facilita la creación de componentes de manera más intuitiva y mejora la legibilidad del código, facilitando el desarrollo.

React sigue el principio de unidireccionalidad de los datos. Los datos fluyen en una sola dirección, desde el componente principal a los componentes secundarios. Esto facilita el seguimiento de los cambios en la aplicación y mejora la previsibilidad del flujo de datos (reactjs, s. f., sección “Unidireccionalidad React”). La unidireccionalidad facilita el seguimiento y la comprensión de cómo los datos fluyen a través de la aplicación. El estado se mantiene en un solo lugar y se propaga hacia abajo en la jerarquía de componentes.

La arquitectura basada en componentes de React promueve la reutilización de código. Los componentes pueden ser fácilmente reutilizados en diferentes partes de la aplicación, lo que ahorra tiempo y esfuerzo en el desarrollo.

Una característica de React son los React Hooks, que son funciones especiales que permiten agregar características de estado y comportamientos de ciclo de vida a los componentes funcionales. Los hooks posibilitan el uso de funcionalidades en componentes basados en funciones sin necesidad de convertirlos en clases. Permiten encapsular lógica compleja y reutilizable en funciones separadas, mejorando la modularidad de los componentes, facilitando la composición y reutilización de la lógica en toda la aplicación (reactjs, s. f., sección “React Hooks”).

React Router es una biblioteca adicional que se utiliza con React para facilitar la navegación y el enrutamiento en aplicaciones de una sola página (SPA). Permite la creación de rutas dinámicas y la manipulación de la navegación de manera eficiente.

React cuenta con una comunidad muy activa de desarrolladores y es respaldado por Facebook, lo cual favorece un desarrollo constante, actualizaciones de seguridad y una amplia base de conocimientos compartida.

React es una opción versátil y poderosa para el desarrollo frontend, permitiendo a los desarrolladores construir interfaces de usuario dinámicas y eficientes con un enfoque modular y declarativo. Su popularidad y su amplia adopción en la industria hacen que sea una tecnología confiable y bien respaldada para diversos tipos de proyectos.

En casos donde no se necesite una interactividad compleja, puede ser excesivo introducir React, otras opciones más simples pueden ser más adecuadas y ayudaría a mantener el proyecto más simple. En caso de no estar familiarizado con esta tecnología, puede ser desafiante hacer un uso adecuado correcto de la misma, además, se trata de una solución exclusivamente para el frontend, por lo que es necesario conocer alguna otra tecnología para hacer frente al backend de la aplicación.

## Backend

En base a las posibilidades analizadas previamente, se dan en consideración las siguientes opciones para el desarrollo backend de la aplicación.

### **ExpressJs - Nodejs**

Express.js, también conocido simplemente como Express, es un framework de servidor web para Node.js que se centra en el desarrollo de aplicaciones web y la creación de API, proporcionando funcionalidades para manejar rutas, middleware, vistas, y otros aspectos relacionados con el servidor. Es uno de los marcos más populares y ampliamente utilizados en la comunidad de desarrollo de Node.js debido a su simplicidad y flexibilidad.

Express se destaca por su enfoque minimalista y no invasivo. Proporciona un conjunto básico de funciones y permite a los desarrolladores agregar módulos según sea necesario para satisfacer los requisitos específicos de su aplicación.

Permite definir rutas para manejar diferentes solicitudes HTTP (GET, POST, etc.), el enrutamiento es sencillo y flexible, facilitando la creación de API RESTful y la gestión de diferentes endpoints (Expressjs, s. f., sección "Enrutamiento").

Utiliza un sistema de middleware que permite a los desarrolladores ejecutar funciones en el ciclo de vida de una solicitud HTTP. Esto facilita la implementación de funciones como la autenticación, el registro de solicitudes, el manejo de errores, entre otros (Expressjs, s. f., sección "Middleware").

Un punto fuerte de Express.js es su amplia cantidad de motores de plantillas disponibles que se pueden utilizar con esta tecnología. La flexibilidad en la elección de motores de plantillas y la facilidad de configuración son aspectos destacados y contribuyen a su popularidad y versatilidad, permitiendo adaptar fácilmente el sistema a la solución más adecuada según sean las necesidades (Expressjs, s. f., sección "Plantillas").

Ofrece mecanismos robustos para manejar errores, facilitando la identificación y gestión de problemas durante el desarrollo y la ejecución de la aplicación.

Express no impone un sistema de base de datos específico, pero es compatible con varios sistemas de base de datos, lo que permite la integración sencilla con bases de datos SQL y NoSQL. Si bien puede considerarse como un punto a favor, puesto a que no impone ningún motor de base de datos, ni posee de forma predeterminada un ORM, tiene la contraparte que, cae en responsabilidad del desarrollador agregar la base de datos y, en caso de desearlo, configurar el ORM a utilizarse, antes de poder utilizar y probar el sistema de manera completa.

Cuenta con una comunidad activa y un ecosistema de módulos y middleware extenso, lo que facilita la incorporación de nuevas funcionalidades a través de paquetes de terceros.

Por contraparte, entre los puntos menos favorables de Expressjs se tiene que si se necesita un marco con muchas características integradas (por ejemplo, ORM integrado, sistema de autenticación avanzado), podría considerarse otros marcos que ofrezcan estas características de forma nativa. Pero no es un punto que no pueda sobrepasarse realizando las configuraciones pertinentes.

### **Flask - Python**

Flask es un framework web minimalista para el lenguaje de programación Python, diseñado para ser simple, ligero y fácil de usar. Proporciona las herramientas esenciales para construir aplicaciones web y APIs de manera eficiente, dejando a los desarrolladores la libertad de elegir y agregar componentes según sus necesidades específicas.

Sigue un enfoque minimalista, proporcionando solo las funcionalidades esenciales para el desarrollo web sin imponer una estructura rígida.

Define rutas para manejar diferentes métodos HTTP mediante el uso de decoradores en funciones. El sistema de vistas basado en funciones facilita la asociación directa entre rutas y código, generando la vinculación entre una ruta en particular, y la función a ejecutarse (Flask, s. f., sección "Rutas").

Flask, de manera predeterminada utiliza Jinja2, un motor de plantillas que permite la generación dinámica de contenido HTML. Facilita la separación de la lógica de presentación del código Python. En caso de desearlo es posible modificar el motor utilizado (Flask, s. f., "Plantillas").

Utiliza un sistema de middleware para ejecutar funciones en el ciclo de vida de una solicitud. Esto permite la implementación de lógica común, como autenticación y manejo de errores.

Flask no impone un sistema de base de datos específico, pero es compatible con varios, incluidos sistemas SQL y NoSQL. Los desarrolladores tienen la libertad de elegir y configurar el sistema de base de datos que mejor se adapte a sus necesidades. Al igual que si se desea utilizar un ORM para el manejo de la base de datos.

Favorece la limitación de las funcionalidades base en favor de un marco de trabajo ligero, permitiendo que el desarrollador incorpore lo que necesite según su criterio. Fomenta la incorporación de funcionalidades adicionales mediante la adición de extensiones y paquetes, facilitando así la personalización de la aplicación según los requisitos específicos del proyecto.

Cuenta con una comunidad activa y un ecosistema de extensiones bien establecido. Los desarrolladores pueden aprovechar una variedad de herramientas y complementos desarrollados por la comunidad para ampliar la funcionalidad de sus aplicaciones.

Flask es una excelente opción para aquellos que buscan construir aplicaciones web de manera rápida y sin la sobrecarga de un marco más completo, presenta un enfoque basado en la sencillez y flexibilidad.

Por contraparte, su utilización no es muy recomendable en aplicaciones muy grandes y complejas, que requieran de un mayor conjunto de herramientas integradas. Producto de esta sencillez y ligereza del framework, es altamente probable que sea necesario incorporar funcionalidades que podrían estar incluidas nativamente en otras soluciones.

### **Spring Boot - Java**

Java con Spring Boot es una combinación poderosa y ampliamente utilizada para el desarrollo de aplicaciones backend. Utilizando como base Java, conocido por su robustez y versatilidad, además, posee un ecosistema muy maduro y establecido,

con gran cantidad de librerías y paquetes, facilitando la integración con otras tecnologías y el desarrollo en general.

Spring Boot, parte del ecosistema Spring, es un framework que simplifica y agiliza el desarrollo de aplicaciones Java. Proporciona un conjunto completo de características, incluyendo manejo de dependencias, configuración automática, y desarrollo basado en convenciones. Hace uso extensivo de la inyección de dependencias, lo que facilita la gestión de componentes y la creación de aplicaciones modularizadas (Spring, s. f., sección “Inyección de dependencias”).

La configuración en Spring Boot se realiza principalmente mediante anotaciones y archivos de configuración, simplificando la tarea de configurar y personalizar la aplicación.

Spring Boot ofrece soporte para una variedad de bases de datos, tanto SQL como NoSQL, facilitando la integración con sistemas de almacenamiento de datos. No posee de forma predeterminada un ORM, en caso de desearlo debe configurarse.

Proporciona funcionalidades integradas para la implementación de medidas de seguridad, incluyendo autenticación y autorización (Spring, s. f., sección “Seguridad”).

Aplica el concepto de convenciones sobre configuración (CoC), esta filosofía de diseño se refiere a la práctica de establecer automáticamente configuraciones predeterminadas basadas en convenciones y patrones, en lugar de requerir configuraciones explícitas para cada detalle de una aplicación. En el contexto de Spring Boot, esta idea implica que muchas configuraciones por defecto son aplicadas automáticamente, lo que simplifica el desarrollo y reduce la necesidad de configuraciones explícitas, siempre y cuando sigas las convenciones establecidas por el marco.

Spring Boot es ideal para proyectos de diferentes tamaños, pero se destaca particularmente en proyectos más complejos donde se beneficia de su conjunto completo de características.

Producto de su amplia utilización desde hace tiempo, tiene una amplia gama de herramientas y bibliotecas para facilitar integraciones con otras herramientas y tecnologías, además de para poder obtener ciertas funcionalidades.

Java, de forma nativa, al realizar algún cambio debería volver a compilarse la aplicación para probar el mismo, hay herramientas en Spring boot que solventan esta situación, como Spring Boot DevTools que permiten el "hot reloading" o recarga en caliente. Con esta característica habilitada, algunos cambios en el código fuente pueden aplicarse automáticamente sin la necesidad de reiniciar la aplicación o compilar manualmente (Spring, s. f., sección "Hot reloading").

Java con Spring Boot es una elección sólida para el desarrollo backend, especialmente en proyectos de tamaño medio a grande, donde la modularidad, la seguridad y la facilidad de configuración son aspectos cruciales.

La contraparte de esta combinación Java - Spring boot es que Java, en comparación con algunos lenguajes modernos, puede ser considerado más verboso en términos de líneas de código necesarias para realizar ciertas tareas, aunque Spring Boot ayuda a simplificar muchas cosas, Java todavía puede requerir más código que algunos lenguajes más concisos. En algunos casos, la modularidad extrema de Spring Boot podría llevar a una aplicación con muchas capas y módulos, lo que puede aumentar la complejidad y dificultar la comprensión para proyectos más pequeños.

### **Django - Python**

Django es un framework de desarrollo web de alto nivel y de código abierto escrito en Python. Diseñado para maximizar la productividad del desarrollador y fomentar la escritura de código limpio y reutilizable, Django se ha convertido en uno de los frameworks web más populares y robustos en la comunidad de desarrollo.

Sigue un patrón arquitectónico Model-View-Controller (MVC), aunque en Django se conoce comúnmente como Modelo-Vista-Plantilla (MVT), en el cual el modelo define la estructura y lógica de los datos y entidades de la aplicación. La vista maneja la lógica de presentación y procesamiento de datos, interactuando con el modelo según sea necesario, la vista no solo utiliza la información almacenada en los modelos y la información de la petición, sino que también puede interactuar con el modelo para realizar operaciones adicionales, como la actualización de datos,

además, el procesamiento de la vista no siempre resulta en la renderización de una plantilla, puede devolver diferentes tipos de respuestas, como redirecciones o respuestas JSON, dependiendo de la lógica de la aplicación. Por último, la plantilla se encarga de la presentación y renderización de la interfaz de usuario, la renderización de la plantilla se realiza en la fase de respuesta, permitiendo un control preciso sobre la generación de la salida final enviada al cliente. Facilita la organización modular del código, separando la lógica de presentación de la lógica de negocio.

Django incluye un ORM potente y fácil de usar que permite interactuar con la base de datos utilizando modelos de objetos Python. Esto simplifica la manipulación de datos y elimina la necesidad de escribir consultas SQL directamente.

Incluye también en la generación del proyecto una base de datos SQLite, que si bien en etapas de producción no es aconsejable, es útil tener una base de datos vinculada al proyecto desde el primer momento.

Django proporciona una interfaz de administración automática generada dinámicamente. Los desarrolladores obtienen un panel de administración listo para usar para gestionar modelos de base de datos sin necesidad de escribir código adicional.

Django incorpora medidas de seguridad como protección contra ataques de inyección SQL, Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF). Esto ayuda a construir aplicaciones más seguras por diseño (Django, s. f., sección “Seguridad”).

Ofrece un potente sistema de manejo de URL que facilita la definición de rutas y la asignación de vistas a esas rutas. Esto simplifica la creación de API RESTful y la gestión de endpoints (Django, s. f., sección “Enrutamiento”).

Es altamente extensible a través de aplicaciones y paquetes de terceros. La comunidad activa proporciona numerosos paquetes que pueden ser integrados para añadir funcionalidades adicionales.

Django es compatible con una variedad de bases de datos, incluyendo PostgreSQL, MySQL y SQLite. Esto permite elegir el motor de base de datos que

mejor se adapte a sus necesidades. Es compatible con motores de bases de datos NoSQL, pero la compatibilidad directa está limitada en comparación con los motores de bases de datos relacionales, Django es conocido por su fuerte integración con bases de datos relacionales, especialmente a través de su ORM. La utilización de bases de datos no relacionales puede requerir esfuerzo y configuraciones adicionales.

Es versátil y puede adaptarse a proyectos pequeños y grandes, aunque algunas características pueden parecer excesivas para proyectos simples.

Django favorece las convenciones sobre la configuración (CoC), lo que significa que sigue un conjunto de reglas predefinidas. Esto puede facilitar la configuración, pero algunos desarrolladores pueden preferir un enfoque más configurable.

Puede desaconsejarse en proyectos pequeños o simples, donde la riqueza de características de Django podrían ser consideradas excesivas y sería más eficiente utilizar un framework más ligero y específico. También en aquellos casos donde se utilicen bases de datos no relacionales puede ser útil buscar soluciones más centradas en esa tecnología.

Django, al ser un framework completo, puede requerir más recursos que frameworks más ligeros, si se desarrolla una aplicación con restricciones de recursos significativas, deberían considerarse alternativas más livianas.

## Base de datos

Para el análisis de la base de datos de la aplicación se tuvieron en cuenta las siguientes alternativas.

### **MySQL**

MySQL es un sistema de gestión de bases de datos relacional de código abierto que se utiliza ampliamente en el desarrollo de aplicaciones web y empresariales. MySQL es conocido por su rendimiento confiable, facilidad de uso y su amplia adopción en la comunidad de desarrollo.

MySQL sigue el modelo de bases de datos relacionales, lo que significa que organiza los datos en tablas con relaciones entre ellas. Utiliza el lenguaje SQL (Structured Query Language) para realizar consultas y manipular los datos almacenados.

Está distribuido bajo una licencia de código abierto (GNU General Public License), lo que significa que es accesible de forma gratuita y los usuarios tienen la libertad de modificar y distribuir el software.

Es compatible con diversas plataformas, incluyendo Windows, Linux, macOS y otras, lo que facilita su implementación en una variedad de entornos (MySQL, s. f., sección "Plataformas").

MySQL es conocido por su capacidad de escalabilidad, lo que significa que puede manejar grandes cantidades de datos y un alto volumen de transacciones, siendo adecuado para entornos de producción.

Es adecuado para el almacenamiento de datos transaccionales, lo que lo convierte en una opción común para aplicaciones que requieren operaciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).

Ofrece soporte para una amplia variedad de tipos de datos, incluyendo enteros, decimales, cadenas de texto, fechas, entre otros (MySQL, s. f., sección "Tipo de datos"). Esto permite a los desarrolladores elegir el tipo de datos más apropiado para sus necesidades.

MySQL proporciona herramientas y utilidades de administración que facilitan la configuración, el monitoreo y el mantenimiento de bases de datos. Además, se integra con diversas herramientas de terceros para la administración visual y el análisis de rendimiento.

Un punto fuerte de MySQL es que cuenta con una gran comunidad de usuarios activa y un amplio ecosistema de herramientas, extensiones y recursos de soporte que facilitan su adopción y desarrollo.

Una consideración a tener en cuenta es que aunque MySQL es de código abierto, es propiedad de Oracle Corporation, lo que puede generar inquietudes sobre la dirección futura de MySQL y cómo se alinea con la estrategia general de Oracle.

### **PostgreSQL**

PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto y objeto-relacional. Conocido por su robustez, extensibilidad y conformidad con los estándares ANSI SQL, PostgreSQL es una opción popular para una amplia gama de aplicaciones, desde pequeños proyectos hasta grandes sistemas empresariales.

PostgreSQL sigue el modelo de bases de datos relacionales, permitiendo organizar los datos en tablas con relaciones entre ellas. Además, ofrece extensibilidad a través de la creación de tipos de datos y funciones personalizadas.

PostgreSQL está distribuido bajo la Licencia PostgreSQL, una licencia de código abierto que permite su uso, modificación y distribución de manera gratuita (PostgreSQL, s. f., sección "Licencia").

Es altamente extensible y permite a los usuarios crear sus propias funciones, operadores, tipos de datos y agregados. Esto facilita la personalización y adaptación a los requisitos específicos de cada proyecto.

PostgreSQL posee soporte para transacciones ACID desde el 2001, lo que garantiza la integridad y la consistencia de los datos incluso en situaciones de fallos.

Además de los tipos de datos estándar, PostgreSQL admite tipos de datos avanzados como hstore (almacenamiento clave-valor), JSON y arrays multidimensionales, proporcionando flexibilidad para representar diversos tipos de información (PostgreSQL, s. f., sección “Tipo de datos”).

PostgreSQL admite herencia de tablas, lo que permite crear tablas que heredan atributos y métodos de otras tablas, lo que puede facilitar la gestión y organización de datos.

### **Maria DB**

MariaDB es un sistema de gestión de bases de datos relacional de código abierto y bifurcación de MySQL, desarrollado por la comunidad y liderado por el fundador original de MySQL, mantiene una alta compatibilidad con MySQL y busca ofrecer mejoras, nuevas características y una mayor apertura en su desarrollo.

MariaDB es compatible con MySQL, lo que significa que las aplicaciones y datos existentes diseñados para MySQL generalmente pueden ejecutarse en MariaDB sin modificaciones, hay que tener en cuenta que no siempre es el caso (MariaDB, s. f., sección “Compatibilidad”).

MariaDB se adhiere a la filosofía de código abierto y está distribuido bajo la Licencia Pública General de GNU (GPL). Esto permite a los usuarios acceder, modificar y distribuir libremente el software.

Al igual que MySQL y PostgreSQL, MariaDB tiene una comunidad activa de desarrolladores y usuarios que contribuyen al desarrollo y mejora continua del sistema. Existen varias organizaciones que respaldan y contribuyen a la comunidad de MariaDB.

MariaDB incluye mejoras y nuevas características en comparación con MySQL, como el motor de almacenamiento ColumnStore para análisis de datos a gran escala, más cantidad de motores de almacenamiento, la optimización del rendimiento y funciones específicas de MariaDB (MariaDB, s. f., sección “Mejoras vs MySQL”).

Soporta transacciones ACID.

MariaDB, que ha nacido como una bifurcación de MySQL, producto de su adquisición por parte de Oracle, ha hecho hincapié en la transparencia en su desarrollo, permitiendo a la comunidad participar activamente en la toma de decisiones y en la dirección futura del proyecto.

### Comparativa Rendimiento MySQL - PostgreSQL - MariaDB

A continuación se presentarán comparativas técnicas, de rendimiento, entre MySQL, MariaDB y PostgreSQL. Estas comparativas son obtenidas de recursos en línea, en caso de que alguna de las tecnologías mencionadas no se encuentren en la comparativa será aclarado en la misma.

Se realiza la comparación entre estas opciones debido a su similitud en una serie de aspectos que las hace competidoras directas, estas similitudes siendo.

- Uso gratuito.
- Bases de datos relacionales.
- Alta popularidad y amplia comunidad.
- Robustez suficiente para entornos de producción.

La siguiente comparativa (Figura 1.1), se realizó con la herramienta HammerDB utilizada para medir el rendimiento de las bases de datos.

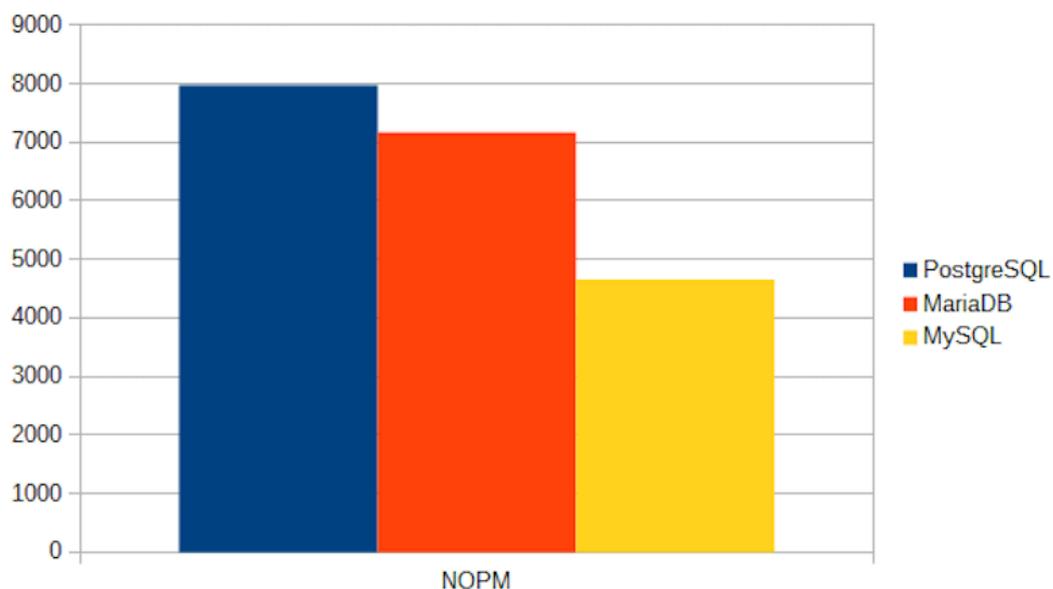


Figura 1.1. Comparativa MariaDB - MySQL - PostgreSQL. Métrica NOPM. Recuperado de <https://stackshare.io/stackups/mariadb-vs-mysql-vs-postgresql>. Información adicional sobre la prueba puede encontrarse en el sitio.

La métrica de la comparativa anterior es el NOPM (New Orders Per Minute), es una métrica de rendimiento independiente de la implementación particular de la base de datos, y por esta razón es recomendada para realizar las comparativas.

De los resultados obtenidos puede observarse que PostgreSQL obtuvo el mejor rendimiento, seguido por MariaDB, y en último lugar MySQL.

En la siguiente comparativa (Figura 1.2), se realiza una prueba similar a la anterior, comparando PostgreSQL y MariaDB únicamente. La métrica es el NOPM, también fue realizado utilizando la herramienta HammerDB

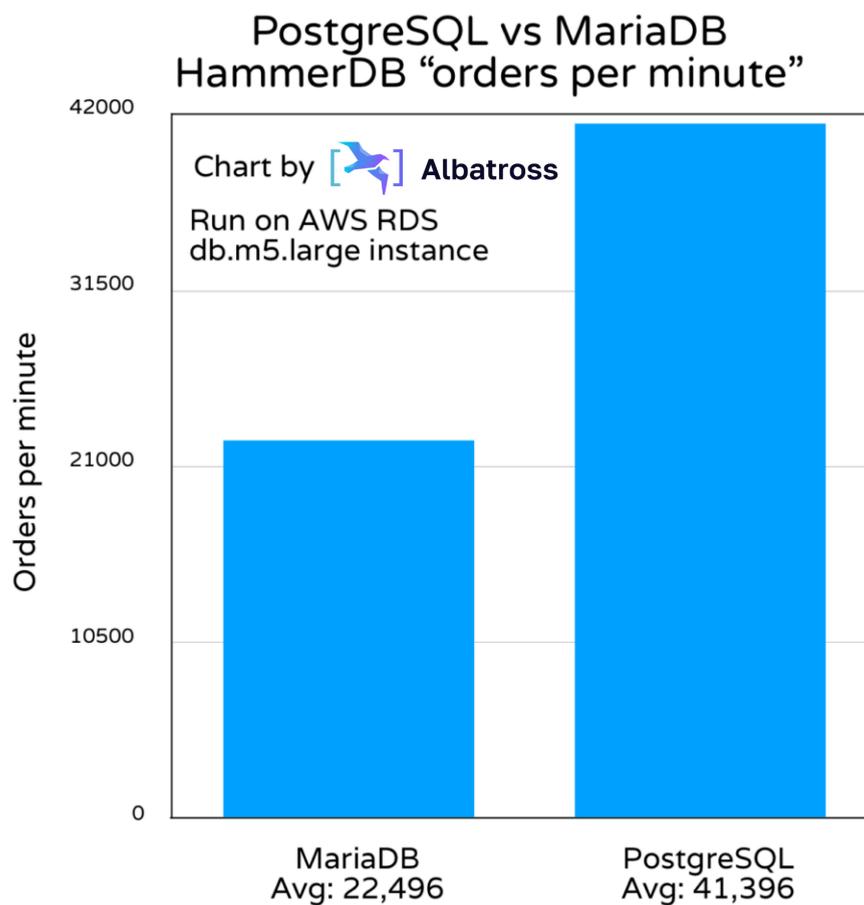


Figura 1.2. Comparativa PostgreSQL - MariaDB. HammerDB, métrica NOPM. Recuperado de <https://www.albatrossmigrations.com/blog/mariadb-postgresql-performance>. Información adicional sobre la prueba puede encontrarse en el sitio

De esta comparativa también puede observarse un mejor rendimiento de PostgreSQL sobre su competidor, si bien la diferencia en la mejora de rendimiento varía de comparación en comparación a causa de la configuración y entorno en particular sobre la cual se realizó dicha comparativa, se mantiene la superioridad de PostgreSQL.

En la siguiente colección de registros sobre comparativas de bases de datos (<https://benchant.com/ranking/database-ranking>). Puede observarse que, generalmente, PostgreSQL obtiene un mejor rendimiento que MySQL.

En conclusión, se puede observar en las pruebas anteriores una mejor prestación por parte de PostgreSQL sobre sus contrapartes, ahora bien, estos resultados no son absolutos, y pueden plantearse casos en los que, dependiendo de la configuración, entorno, tipo de operaciones, tipo de datos manipulados, entre tantas variantes, se puedan observar resultados distintos. Cualquiera de las tres opciones son igualmente válidas y las particularidades de cada proyecto y aplicación deben ser consideradas para hacer la selección final de la tecnología de base de datos a implementarse.

### **MS SQL Server**

Microsoft SQL Server (MS SQL) es un sistema de gestión de bases de datos relacional desarrollado por Microsoft. A diferencia de las opciones anteriores, MS SQL no es de uso gratuito para entornos productivos.

Cuenta con integración con el ecosistema de Microsoft, lo que facilita su uso en entornos que ya emplean tecnologías de Microsoft, como Windows Server, Active Directory, y herramientas como Visual Studio.

MS SQL utiliza T-SQL como lenguaje de programación y consulta. T-SQL es una extensión del estándar SQL con características específicas de Microsoft, lo que permite un mayor control y flexibilidad en la manipulación de datos y el desarrollo de procedimientos almacenados (Microsoft, s. f., sección "T-SQL").

Ofrece servicios de integración (SSIS), diseñado para facilitar la integración de datos en diferentes escenarios, utilizado principalmente para realizar procesos ETL, que implican la extracción de datos desde diversas fuentes, su transformación según las necesidades del negocio y la carga resultante en un almacén de datos o

en otro destino (Microsoft, s. f., sección “SSIS”). También ofrece servicios de análisis (SSAS) diseñados para el análisis de datos y la creación de modelos multidimensionales y tabulares, siendo los modelos multidimensionales utilizados para el análisis OLAP (Online Analytical Processing), permiten a los usuarios explorar los datos de manera jerárquica y realizar análisis complejos, como la perforación (drill-down) y el resumen (roll-up), además de los modelos multidimensionales, SSAS admite modelos tabulares, que son más simples y están diseñados para facilitar la creación de estructuras de datos en columnas. Los modelos tabulares son eficientes y se integran con herramientas de visualización como Power BI, también de Microsoft. Estos servicios en conjunto permiten la integración de datos de diversas fuentes y la creación de cubos OLAP para el análisis multidimensional (Microsoft, s. f., sección “SSAS”).

Ofrece medidas de seguridad, como la autenticación integrada de Windows, roles y permisos a nivel de base de datos, y la capacidad de encriptar datos tanto en reposo como en tránsito.

MS SQL Server destaca por su integración con el ecosistema Microsoft, lo que amplía el alcance de la herramienta, y proporciona una suite que aborda un amplio abanico de necesidades en soluciones integrables fácilmente. Sin embargo, una de las mayores consideraciones que tiene en su contra es que no es de uso gratuito, como sí lo son otros motores de base de datos altamente performantes, además que, estas herramientas del ecosistema Microsoft también son pagas, siendo que, de igual manera que con los motores de base de datos, hay alternativas de uso libre y gratuito.

### **MongoDB**

MongoDB es un sistema de gestión de bases de datos NoSQL orientado a documentos y de código abierto, desarrollado por MongoDB Inc, almacena los datos en formato BSON (Binary JSON), un formato binario de JSON (MongoDB, s. f., sección “BSON”). Cada registro en la base de datos es un documento que puede contener estructuras de datos complejas y anidadas, lo que facilita la representación de información de manera más flexible que en los sistemas de bases de datos relacionales.

A diferencia de las bases de datos relacionales, MongoDB no requiere un esquema fijo para los documentos almacenados en una colección, esto significa que

cada documento en una colección puede tener un conjunto diferente de campos, lo que permite una adaptación fácil a cambios en la estructura de los datos.

Admite la creación de índices para mejorar el rendimiento de las consultas. Además, proporciona un lenguaje de consulta rico que incluye operaciones para filtrar, proyectar, ordenar y realizar operaciones de agregación en los datos almacenados.

MongoDB ofrece un modelo de consistencia eventual por defecto, lo que significa que las lecturas pueden retornar datos desactualizados durante un breve periodo de tiempo antes de que se actualicen (MongoDB, s. f., sección “Consistencia”). Sin embargo, los usuarios tienen la opción de configurar niveles de consistencia más estrictos según sus necesidades.

Dada su capacidad para manejar grandes cantidades de datos no estructurados y semiestructurados, los motores NoSQL son comúnmente utilizados en escenarios de Big Data, donde se procesa y analiza información diversa y a gran escala.

MongoDB cuenta con una amplia comunidad de usuarios y desarrolladores, lo que facilita el intercambio de conocimientos y recursos. Además, MongoDB Inc. proporciona servicios de soporte empresarial para aquellas organizaciones que requieren asistencia técnica y soluciones personalizadas.

Es conocido por su flexibilidad, escalabilidad horizontal, soporte para datos no estructurados y su capacidad para gestionar grandes volúmenes de información.

Entre los factores que pueden hacer desaconsejables la utilización de MongoDB, está que no ofrece transacciones ACID completas en todas las situaciones, especialmente en escenarios que involucran múltiples documentos o colecciones (MongoDB, s. f., sección “ACID”). Si la aplicación requiere un esquema de datos fijo y riguroso, donde cada registro debe seguir una estructura predefinida y consistente, un sistema de base de datos relacional podría ser más apropiado.

## CMS

Un CMS (de sus siglas Content Management System, traducido como Sistema de Gestión de Contenidos), se trata de una aplicación utilizada para crear y administrar los diversos contenidos expuestos en una página o aplicación a través de una interfaz amigable y sencilla, permitiendo que, usuarios no técnicos puedan gestionar dichos contenidos de manera simple y eficiente.

A diferencia de cualquier otra librería o paquete que pudiera implementarse en el sistema a generarse, el CMS cuenta con una sección en particular por su importancia clave en el proyecto. A continuación se exponen las posibles alternativas.

### **WordPress**

WordPress es una de las plataformas más populares para la creación de sitios web y blogs. Programado principalmente en PHP, utilizando una base de datos MySQL para gestionar la información.

WordPress es conocido por su interfaz intuitiva y amigable, lo que facilita a usuarios sin experiencia en programación crear y gestionar sitios web. Ofrece una amplia variedad de temas y complementos (plugins) que permiten personalizar y ampliar las funcionalidades del sitio sin necesidad de aprender a programar, simplificando el proceso de creación y adecuación del sitio web. Esto hace que sea adecuado para una amplia gama de sitios web, desde blogs personales hasta sitios empresariales complejos (WordPress, s. f., sección “Plugins”).

Cuenta con una gran comunidad de usuarios, desarrolladores y contribuyentes que proporcionan soporte, temas y plugins. La comunidad también está involucrada en la mejora continua del software a través de actualizaciones regulares. Siendo WordPress probablemente el CMS más conocido y utilizado mundialmente.

WordPress está diseñado con prácticas de SEO (Optimización para Motores de Búsqueda) en mente, lo que facilita la optimización de contenido para los motores de búsqueda y ayuda a mejorar el ranking del sitio en los resultados de búsqueda (WordPress, s. f., sección “SEO”).

Una consideración es que en sitios web muy grandes y complejos, WordPress puede experimentar problemas de rendimiento. Optimizaciones y técnicas como el uso de servicios de alojamiento dedicados pueden ser necesarios para abordar este problema.

La calidad de la experiencia del usuario y las funciones del sitio a menudo dependen de la elección de temas y plugins. Algunos pueden no ser bien mantenidos o pueden no ser compatibles entre sí, lo que puede causar problemas (WordPress, s. f., sección “Incompatibilidad plugins”).

WordPress suele ser usado para la creación de blogs, sitios web empresariales, sitios de noticias en línea, tiendas online, foros, entre tantos (WordPress, s. f., sección “Casos de uso”).

## **Wagtail**

Wagtail es un sistema de gestión de contenido de código abierto construido sobre el framework web Django. A diferencia de WordPress, que es un CMS más general y ampliamente utilizado, Wagtail se ha destacado como una solución específica para el desarrollo de sitios web y aplicaciones más complejas y personalizadas.

Ofrece una interfaz de usuario amigable y fácil de usar, lo que facilita la gestión del contenido para usuarios no técnicos. Mediante el tablero de administración puede gestionarse los contenidos de las diversas páginas y secciones del sitio, las imágenes cargadas, usuarios, roles y permisos, entre otras cosas.

Se asegura el funcionamiento con bases de datos MySQL, SQLite y PostgreSQL, pudiendo funcionar con otros motores, pero no de manera garantizada. Se recomienda su utilización con el motor PostgreSQL para producción. (Wagtail, s. f., sección “Base de datos”)

Proporciona herramientas para la creación de estructuras de páginas flexibles y personalizables. Pueden definirse modelos de página con campos específicos para adaptarse a las necesidades de contenido de su sitio, permitiendo dar soluciones exactas a necesidades muy específicas. Producto de esto, ofrece una

capacidad de personalización extremadamente alta, tanto en el tipo de contenidos que se pueden gestionar, mediante la modelación de los mismos, como así también la forma en la que se renderizan.

Wagtail ofrece una robusta gestión de activos que facilita la carga y organización de imágenes y documentos. También proporciona herramientas para manipular imágenes directamente dentro de la interfaz.

Ofrece flujos de trabajo de publicación que permiten revisar y aprobar contenido antes de que se publique en el sitio, esto es especialmente útil en entornos donde múltiples usuarios colaboran en la creación y edición de contenido (Wagtail, s. f., sección “Workflows”).

Incluye características diseñadas para facilitar la optimización para motores de búsqueda, como la capacidad de definir fácilmente metadatos y descripciones, promocionando resultados de búsqueda (Wagtail, s. f., sección “SEO”).

Aunque Wagtail ofrece una interfaz amigable, la curva de aprendizaje puede ser mayor en comparación con CMS más simples debido a su flexibilidad y capacidades más avanzadas.

La comunidad de Wagtail es más pequeña en comparación con sistemas más populares como WordPress. A pesar de esto cuenta con una documentación muy amplia y detallada, además de que se encuentra en permanente desarrollo y actualización.

La cantidad de temas y plugins disponibles para Wagtail puede ser limitada. Sin embargo, esto se compensa en parte por la capacidad de personalización y desarrollo de funciones específicas. La contraparte de eso es que esta personalización y desarrollo de funciones específicas debe ser llevada a cabo por un desarrollador/usuario experto, teniendo una alta carga de programación tanto en términos específicos de Wagtail, como también del propio Django y Python.

### **OpenCMS**

OpenCms es un sistema de gestión de contenido de código abierto basado en Java y XML que se utiliza para crear y administrar sitios web (OpenCMS, s. f., sección “Introducción”).

OpenCms está escrito en Java y utiliza tecnologías web como JavaServer Pages (JSP), Servlets y XML. Esto hace que sea compatible con una amplia variedad de entornos de servidor Java.

Ofrece un editor WYSIWYG (What You See Is What You Get) que facilita la creación y edición de contenido sin necesidad de conocimientos de programación o diseño web (OpenCMS, s. f., sección “WYSIWYG”).

Proporciona funciones de gestión de versiones que permiten realizar un seguimiento de los cambios en el contenido y revertir a versiones anteriores si es necesario. También incluye capacidades de flujo de trabajo para gestionar la aprobación y publicación de contenido (OpenCMS, s. f., sección “Workflow”).

Aunque OpenCms es altamente personalizable, puede encontrarse que la personalización avanzada y la creación de funciones personalizadas pueden requerir conocimientos más profundos de desarrollo Java.

Se encuentra en constante desarrollo a modo de código abierto desde el año 2000 hasta la actualidad.

## **Strapi**

Strapi es un sistema de gestión de contenido de código abierto y headless basado en Node.js.

Strapi es un CMS headless centrado en APIs. Esto significa que su enfoque se centra en proporcionar una interfaz fácil de usar para administrar el contenido, mientras que permite a los desarrolladores consumir ese contenido mediante una API RESTful o GraphQL, y luego mostrarlo en diferentes plataformas y dispositivos, separando totalmente el backend del frontend (Strapi, s. f, sección “CMS headless”).

Strapi ofrece una interfaz de usuario amigable que facilita la creación y gestión de contenido sin necesidad de conocimientos de programación. Los usuarios pueden definir tipos de contenido, agregar campos y crear relaciones de manera intuitiva.

Es altamente personalizable y flexible. Los desarrolladores pueden definir sus propios tipos de contenido, campos y relaciones, lo que permite adaptar el sistema a las necesidades específicas del proyecto.

Ofrece un sistema de autenticación y autorización robusto, permitiendo a los desarrolladores definir roles y permisos para controlar el acceso al contenido y las funciones del sistema (Strapi, s. f., sección “Roles y permisos”).

Strapi es compatible con una gran variedad de bases de datos, incluyendo MariaDB, PostgreSQL, MySQL y SQLite. Esto proporciona una gran versatilidad según las preferencias del equipo de desarrollo y los requerimientos del proyecto (Strapi, s. f., sección “Bases de datos”).

Incluye una función de gestión de medios que facilita la carga y manipulación de archivos multimedia, como imágenes, videos, audios y documentos (Strapi, s. f., sección “Multimedia”).

Ofrece soporte tanto para GraphQL como para RESTful, lo que brinda flexibilidad para elegir la tecnología de API que mejor se adapte a su aplicación.

El backend de Strapi se basa en el framework Koa de Node.js, que es conocido por su modularidad y eficiencia, Koa no se especializa en la gestión de contenido o la administración de bases de datos, en cambio, proporciona herramientas para manejar solicitudes, respuestas y middleware (Strapi, s. f., sección “Koa”).

Aunque Strapi es bastante accesible para desarrolladores, para no técnicos puede ser desafiante, especialmente al utilizar características más avanzadas o personalizar el sistema.

## Despliegue

### Render

Render es una plataforma de servicios en la nube que se centra en proporcionar una experiencia sencilla y escalable para el desarrollo y despliegue de aplicaciones web.

Facilita el despliegue automático de aplicaciones directamente desde repositorios Git (GitHub, GitLab, Bitbucket), simplificando el proceso de implementación y actualización de aplicaciones. Pudiendo configurarse que, de forma muy simple, cada cambio subido al repositorio dispare el proceso de despliegue de la aplicación, de forma automática (Render, s. f., sección “despliegue”).

Render en parte nace como una respuesta al cambio de modelo de negocios de una plataforma previamente muy utilizada llamada Heroku (la cual también será analizada). Presentándose como una herramienta más flexible, económica y transparente (Render, s. f., sección “Render vs Heroku”).

Posee a modo de prueba gratuita múltiples tipos de servicios, desde para almacenar servicios web, hasta bases de datos. Esto permite a los desarrolladores aprender la herramienta, sus cualidades y características sin necesidad de incurrir en gastos (Render, s. f., sección “Características gratuitas”).

Además soporta de forma nativa una gran variedad de lenguajes para las aplicaciones, pudiendo ser:

- Node.js
- Python
- Ruby
- Go
- Rust
- Elixir

Y en caso de querer desplegar una aplicación que no se encuentre desarrollada en estos lenguajes, render ofrece la posibilidad de desplegar fácilmente contenedores de docker (Render, s. f., sección “Lenguajes”).

Cuenta con soporte para el despliegue de una base de datos PostgreSQL.

### **Heroku**

Ofrece servicios de plataforma de aplicaciones, siendo una de las primeras y más importantes en este rubro, aunque su utilización y soporte se haya visto afectado producto de las limitaciones aplicadas a sus versiones gratuitas.

Heroku ejecuta las aplicaciones de los clientes en contenedores virtuales, que corren en entornos de ejecución confiables, estos contenedores tienen el nombre de Dynos (Heroku, s. f., sección “Dynos”).

Cuenta con soporte para los siguientes lenguajes de programación:

- Node.js
- Java
- Python
- Scala
- Ruby
- PHP
- Go
- Clojure

Para aquellos lenguajes que no se encuentran en la anterior lista, se utilizan Buildpacks, son scripts que contienen los procesos de construcción para permitir a heroku desplegar una aplicación en un lenguaje no soportado de forma nativa, estos Buildpacks son generados por la comunidad, y hasta pueden ser desarrollados por uno mismo (Heroku, s. f., sección “Lenguajes”).

Además, presenta soporte para una base de datos, con origen en PostgreSQL, llamada Heroku Postgres.

Cuenta con integración con herramientas de control de versiones como Git, facilitando las tareas de despliegue automático (Heroku, s. f., sección “Git”).

## **Railway app**

Railway es una plataforma de despliegue diseñada para soportar el ciclo de vida en el desarrollo de software, buscando facilitar las tareas de integración y despliegue continuo.

Las aplicaciones de Railway se agrupan en proyectos, estos proyectos pueden contener un stack de una aplicación, un grupo de servicios, o incluso una colección de grupos de servicios. Estos servicios son donde se implementan los elementos generados por el usuario, pudiendo ser repositorios o imágenes de docker. En caso de que el origen sea un repositorio, Railway tomará dicho código y generará una imagen de docker a partir del mismo (Railway, s. f., sección “Despliegue”).

Railway ofrece soporte para bases de datos:

- PostgreSQL
- MySQL
- MongoDB

Railway además se presenta un esquema de suscripción en donde el usuario se suscribe a un servicio de prestaciones, que le da acceso a las mismas, y además de esto, se cobra en base al consumo específico de recursos realizado. Esto puede otorgar un costo menor que otras opciones, pero también genera que los costos de utilizar el servicio varían mes a mes (Railway, s. f., sección “Costos”).

## Propuesta de solución

Luego de las opciones relevadas, teniendo en mente las características de la problemática y las limitaciones y habilidades personales a la hora de emprender este proyecto, se decidió elegir como conjunto de solución el siguiente grupo de tecnologías.

- Aplicación web
- Django - Python
- Wagtail
- PostgreSQL
- Render

Esta decisión se basa en una serie de consideraciones, que a continuación se describen. En primer lugar, se decide la aplicación web como plataforma de implementación, esto debido a su alta accesibilidad desde cualquier dispositivo. Hoy en día la inmensa mayoría de usuarios potenciales, por no decir todos, poseen dispositivos móviles en donde cuentan con un navegador web, esto permite acceder y hacer uso de la aplicación sin necesidad de realizar la instalación de ninguna otra aplicación extra, siendo de gran comodidad. Además, los usuarios cuentan con una amplia experiencia en la utilización de este tipo de soluciones, por lo que se presenta una curva de aprendizaje muy amena, donde los usuarios pueden hacer uso efectivo del sistema desde un primer momento en base a su experiencia previa.

Contando con la aplicación web, puede fácilmente gestionarse la forma en la que se disponen los elementos de la misma, otorgando una vista simple para los usuarios casuales y voluntarios que realicen acciones en el voluntariado. Y vistas más ricas de opciones e información para los directivos del VUCC y coordinadores.

En cuanto a la elección Django - Python, se tuvo en cuenta la posibilidad de poder realizar la totalidad de la aplicación desde esta tecnología, lo que otorga la gran comodidad de poder administrar la totalidad del código del proyecto desde un único framework con un solo código fuente. Por las características de la aplicación a realizar, el rendimiento obtenido por esta tecnología es más que acorde, ya que no se realizan acciones de alto costo computacional, siendo en gran medida únicamente acciones del tipo ABM (Alta-Baja-Modificación) con la necesidad de realizar ciertas verificaciones y controles únicamente.

También cuenta con una gran cantidad de paquetes, módulos y librerías desarrolladas por la comunidad, lo que otorga la posibilidad de realizar funcionalidades complejas de forma muy fácil, con muy poco esfuerzo. Posee una gran comunidad y muy activa, lo que permite encontrar mucha información y actualizada, además de que el ORM de Django es muy potente y simplifica la interacción con la base de datos.

Como único punto donde podría encontrarse alguna flaqueza de esta elección, es en cuanto a que Django, en comparación con otras herramientas diseñadas exclusivamente para el desarrollo de interfaces y elementos de interacción con el usuario, es más pobre en su diseño de interfaces gráficas, mas aún siendo totalmente capaz de la generación de las mismas, contando con un motor de plantillas muy completo, facilitando la generación dinámica de estructuras HTML-CSS-JavaScript.

Mediante la utilización de algunos frameworks de CSS como Bootstrap, se logra realizar diseños altamente adecuados, con muy poco esfuerzo por parte del desarrollador, y sin la necesidad de aprender un conjunto de tecnologías diferente para el desarrollo del backend y frontend, siendo altamente conveniente.

Wagtail, el módulo de CMS para Django, es muy potente y flexible, permitiendo un elevado grado de personalización y ajuste de manera muy simple. Cuenta con una gran cantidad de documentación actualizada y se encuentra en constante desarrollo, pudiendo encontrarse actualizaciones de forma frecuente. Cuenta con una gran cantidad de utilidades y funciones, lo que la convierte en una herramienta muy completa desde el primer momento.

PostgreSQL es elegido a partir de cuatro consideraciones claves. En primer lugar, por la naturaleza del sistema a desarrollarse, lo más adecuado es la utilización de una base de datos relacional, en donde las entidades y relaciones puedan ser plasmadas, generando una estructura sólida para la gestión de los datos. Por otro lado, es un motor de bases de datos extremadamente completo y holgadamente capaz de hacer frente a las exigencias de la aplicación a desarrollarse, quedando plasmado en las comparaciones técnicas su alto rendimiento en relación con otros motores que son considerados su competencia más directa. En tercer lugar, es de uso gratuito, por lo que es de alta conveniencia. Y como último punto, Wagtail cuenta con garantía de funcionamiento con este motor de base de datos, pudiendo Django también integrarse perfectamente con esta herramienta, generando un conjunto de tecnologías perfectamente cohesivo y efectivo para hacer frente al problema.

En último lugar se decidió utilizar Render como plataforma de despliegue de la aplicación, esto producto de su soporte nativo de Python y capacidad de desplegar bases de datos PostgreSQL, lo cual también asienta el conjunto de tecnologías aplicadas, en donde hay una gran cohesión y soporte nativo entre todas ellas. Render cuenta además con una documentación muy detallada sobre cómo realizar el despliegue de una aplicación de Django, vinculando la misma con la base de datos PostgreSQL, facilitando en gran medida esta etapa del proyecto. Sus prestaciones considerando su costo es acorde a lo que se observa en otras plataformas de implementación, y mediante su vinculación con un repositorio remoto de GitHub, es muy sencillo y rápido llevar a producción los cambios realizados.

## Alcance funcional

A continuación se presentarán, agrupadas por afinidad, las funcionalidades principales del sistema.

- Funcionalidades de contenidos
  - ABM contenidos
  - Preguntas frecuentes
  - Datos de contacto
  - Información de donaciones
  - Visualización de contenidos
- Funcionalidades de reportes
  - Resumen asistencias por actividad por voluntario
  - Resumen asistencias por voluntariado por voluntario
  - Resumen asistencias por voluntario
- Funcionalidades de actividades
  - Operaciones ABM actividades
- Funcionalidades de voluntarios
  - Alta de voluntarios
  - Gestión de sesión
  - Inscripción a actividades
- Funcionalidades de vehículos
  - ABM vehículos
  - Vinculación con actividades
- Funcionalidades de coordinadores
  - Inscripción voluntarios voluntariados
  - Lista de inscriptos
  - Toma de asistencias
- Funcionalidades de directivos
  - ABM voluntarios
  - ABM voluntariados
  - Inscripción voluntarios voluntariados
  - Reinicio de año

Se procede a detallar en mayor medida los orígenes de dichas funcionalidades, a partir de breves historias de usuarios.

## Funcionalidades de contenidos

- Como directivo, quiero realizar operaciones de ABM sobre los contenidos de las páginas para mantener actualizada y adecuada la información.
- Como directivo, quiero administrar una página de preguntas frecuentes para responder las preguntas más habituales.
- Como directivo, quiero administrar las propuestas de donación existentes a los voluntariados para informar las necesidades de los mismos.
- Como directivo, quiero exponer métodos de contacto, para facilitar el acercamiento con el voluntariado.
- Como directivo, quiero previsualizar los cambios realizados en una página para validar que esté satisfecho con los mismos.

## Funcionalidades de reportes

- Como coordinador, quiero ver un historial de asistencias de un voluntario para mantenerme informado sobre su nivel de participación en las actividades de voluntariado.
- Como coordinador, quiero observar las asistencias en una actividad, agrupada por día, para estar al tanto de las participaciones en una fecha determinada.
- Como coordinador, quiero observar las asistencias en una actividad, agrupada por voluntario, para estar al tanto de las participaciones de cada voluntario en una actividad.
- Como coordinador, quiero observar las asistencias de cada voluntario en los voluntariados donde estoy inscripto, para estar al tanto de la participación de cada voluntario en el voluntariado.
- Como directivo, quiero observar las asistencias de cada voluntario en cualquier voluntariado, para estar al tanto de la participación de cada voluntario en el voluntariado.
- Como voluntario, quiero observar mis asistencias registradas, para corroborar el estado de mis registros de asistencias.

## Funcionalidades de actividades

- Como directivo, quiero realizar operaciones de ABM sobre las actividades para mantener actualizadas las propuestas de los voluntariados.

## Funcionalidades de voluntarios

- Como usuario no logueado, quiero iniciar sesión para ingresar a las funcionalidades del sistema.
- Como voluntario, quiero inscribirme en una actividad para poder participar en la misma.
- Como voluntario, quiero des-inscribirme de una actividad para cancelar mi participación en la misma.
- Como usuario no logueado, quiero registrarme para crear mi usuario en el sistema.

## Funcionalidades de vehículos

- Como voluntario, quiero crear y eliminar mis vehículos para mantener actualizados mis datos de movilidad.
- Como voluntario, quiero vincular un vehículo a una actividad para informar mis datos de movilidad.
- Como voluntario, quiero des-vincular un vehículo de una actividad para informar mis datos de movilidad.

## Funcionalidades de coordinadores

- Como coordinador, quiero tomar asistencia en una actividad para generar un registro de los presentes.
- Como coordinador, quiero observar quienes están inscriptos a una actividad para poder gestionar la logística de la misma.
- Como coordinador, quiero inscribir voluntarios en los voluntariados donde estoy inscripto, para habilitarlos a participar en los mismos.
- Como coordinador, quiero des-inscribir voluntarios en los voluntariados donde estoy inscripto, para revocarles la capacidad de participar en los mismos.

## Funcionalidades de directivos

- Como directivo, quiero iniciar un nuevo ciclo del VUCC para reiniciar las inscripciones de los voluntarios en los voluntariados.
- Como directivo, quiero iniciar un nuevo ciclo del VUCC para que los reportes muestren información del periodo actual.
- Como directivo, quiero inscribir voluntarios en cualquier voluntariado para habilitarlos a participar en los mismos.
- Como directivo, quiero des-inscribir voluntarios en cualquier voluntariado para des-habilitarlos de participar en los mismos.
- Como directivo, quiero habilitar voluntarios para que puedan acceder a las funcionalidades del sistema.
- Como directivo, quiero des-habilitar voluntarios para revocarles su acceso a las funcionalidades del sistema.
- Como directivo, quiero establecer voluntarios como coordinadores para que obtengan permisos extra.
- Como directivo, quiero retirar la designación de voluntarios como coordinadores para revocar sus permisos extra.
- Como directivo, quiero realizar operaciones de ABM sobre los voluntarios para gestionar sus cuentas e información.
- Como directivo, quiero realizar operaciones de ABM sobre los voluntariados para actualizar las propuestas del VUCC.

Es importante señalar que la jerarquía de voluntarios, coordinadores y directivos sigue un modelo hereditario. En este contexto, los coordinadores poseen las capacidades inherentes a un voluntario, complementadas con una gama adicional de permisos, acciones y recursos, por su parte, los directivos poseen las habilidades comunes a los voluntarios y coordinadores, ampliadas con una serie adicional de permisos, acciones y recursos.

Las características de la herramienta generada deben ser tales que no sea burocrática ni tediosa su utilización. Debe centrarse en la sencillez y facilidad de uso, para así lograr su cometido de aligerar la tarea de los coordinadores de los voluntariados y directivos del VUCC. Este es un aspecto clave de la aplicación.

## Solución generada

A partir de las necesidades planteadas a lo largo del documento, mediante la utilización de las tecnologías definidas previamente, se llega al siguiente modelado.

La aplicación web, producto de la estructura organizativa de Django, es separada en aplicaciones (Django apps), pudiendo ser consideradas como módulos del sistema, cada una con sus propios modelos y vistas, que luego se integran e interactúan entre sí a modo de dar la solución final.

El repositorio donde se encuentra el sistema es el siguiente: <https://github.com/Manuel-Bobadilla/Trabajo-Final/>. De este puede observarse la estructura con todo nivel de profundidad.

A continuación se mencionan y describen a grandes rasgos las apps generadas, a modo de especificar las características principales de las mismas.

### Actividades

Este módulo da respuesta a las necesidades del VUCC para representar sus actividades, las actividades del VUCC son específicamente las tareas que realizan los equipos de voluntariado. Un equipo de voluntariado puede realizar múltiples actividades.

Cuando un voluntario se inscribe a una actividad, lo que representa es la manifestación de participar en la próxima iteración de la actividad. Esto parte de que un voluntario inscrito a un voluntariado, puede no participar de la actividad planificada en alguna ocasión, por la razón que sea. A los coordinadores de los voluntariados, les es muy útil conocer quiénes y cuántos son los voluntarios que se van a presentar, a modo de poder organizar correctamente la actividad. Por esto, antes de la próxima realización de la actividad, los voluntarios deben hacer explícita su intención de participar nuevamente en la misma.

En esta aplicación se modelizan las actividades de los voluntariados. Para su modelado se utilizan funcionalidades aportadas por Wagtail, permitiendo generar las actividades y su información a partir de una estructura predefinida, en la cual el usuario pueda definir que contenidos, dentro de un conjunto de contenidos, quiere utilizar.

Los modelos dentro de esta aplicación son los siguientes:

- **ActivityListingPage:** Dentro de este modelo se genera la estructura capaz de almacenar dentro de sí todas las actividades disponibles en el VUCC. Posee la lógica de presentación de las actividades.
- **ActivityDetailPage:** Aquí se define efectivamente el modelo de las actividades.

Atributos de ActivityListingPage.

- **custom\_title.** Campo de texto, define el título de la sección

Atributos de ActivityDetailPage:

- **custom\_title.** Campo de texto, define el título de la sección
- **volunteers.** Referencia muchos a muchos con voluntarios, establece la relación con el modelo Volunteers, donde una actividad puede tener muchos voluntarios, y un voluntario puede participar en muchas actividades.
- **volunteering.** Referencia de clave foránea con los voluntariados, establece la relación con el modelo Volunteerings, donde una actividad pertenece a un voluntariado, pero un voluntariado puede poseer múltiples actividades.
- **activity\_image.** Relación de clave foránea con imágenes, un modelo de Wagtail. Establece la relación con las imágenes, una actividad posee una imagen, una misma imagen a su vez puede estar presente en múltiples actividades.
- **activity\_start\_date.** Campo de tipo fecha. Define la fecha de inicio de la actividad.
- **activity\_end\_date.** Campo de tipo fecha. Establece la fecha de cierre de la actividad.
- **content.** Tipo de dato StreamField, proveniente de Wagtail, en donde puede agregarse contenidos, dentro de una colección de contenidos disponibles, a modo de adaptarse al formato deseado. Forma el cuerpo en cuanto a información de la actividad.

En base a estos modelos, con sus atributos, se define la estructura básica de presentación de las actividades de los voluntariados.

En cuanto a la lógica implementada en esta sección, se rescata lo siguiente.

Dentro de `ActivityListingPage`, se obtienen todas las `ActivityDetailPage` que se encuentren publicadas (donde efectivamente se encuentra la información de las actividades), con estas actividades, la información del voluntario y la información del contexto, se renderizan las actividades del voluntariado.

Por último, en la sección de vistas de este módulo de la solución se encuentra la lógica para.

- Agregar la inscripción de un voluntario a una actividad.
- Remover la inscripción de un voluntario a una actividad.
- Visualizar la lista de inscriptos en una actividad.
- Obtener la lista para tomar asistencia en una actividad.
- Generar los registros de asistencia de una actividad.
- Exposición de las asistencias de una actividad, agrupadas por día.
- Exposición de las asistencias de una actividad, agrupadas por voluntario.
- Reinicio de las inscripciones a una actividad, para poder iniciar una nueva iteración de la actividad.

## Asistencias

Aquí se modelan los registros de asistencia del VUCC, estos registros son clave puesto su gran importancia para el desarrollo de las actividades en el voluntariado. Se utilizan atributos y funcionalidad provenientes de Django.

Las asistencias son el registro de que un voluntario participó en una actividad en una fecha determinada. Este registro es la base de la mayoría de los reportes generados en el sistema, pues conocer la cantidad de participaciones registradas por cada voluntario es de gran necesidad para el VUCC.

Dentro de esta Django app se encuentra el modelo.

- Attendance. Registro de asistencia de un voluntario en una actividad.

Atributos de Attendance:

- volunteer. Referencia por clave foránea con los voluntarios, establece una relación con el modelo Volunteer. Un voluntario puede tener muchos registros de asistencia, un registro de asistencia es de un único voluntario.
- activity. Referencia por clave foránea con las actividades, establece una relación con el modelo ActivityDetailPage. Una actividad puede tener muchos registros de asistencia, pero un registro de asistencia sólo puede referir a una actividad.
- date. Campo de tipo fecha. Almacena la fecha del registro de asistencia.
- activity\_title. Campo de tipo texto. Utilizado para persistir el nombre de la actividad a la que referencia la asistencia, en caso de que dicha actividad se elimine del sistema.

Este módulo del sistema no cuenta con lógica del negocio implementada.

## Flex

Es una estructura de contenidos definida utilizando las prestaciones de Wagtail, a diferencia de ActivityListingPage, que está diseñada únicamente para gestionar los contenidos de las actividades disponibles, Flex está pensada para ser altamente flexible a las necesidades del usuario, pudiendo gestionar mayor cantidad de contenidos, siendo utilizada como un modelo de página general en la aplicación.

Este módulo surge para brindar al VUCC una plantilla sobre la cual puedan exponer los contenidos que les sean necesarios, de la forma más adecuada.

El modelo que se encuentra dentro de este módulo es el siguiente.

- FlexPage. Encargado de mantener la información de páginas de contenido genéricas dentro de la aplicación, y su lógica de renderizado.

Dentro del modelo FlexPage se encuentran los siguientes atributos.

- content. Atributo del tipo StreamField, utilizado para gestionar de manera dinámica y personalizada la disposición de los contenidos, dentro de un conjunto predefinido de contenidos disponibles.
- subtitle. Atributo de tipo texto, utilizado para almacenar el texto que se muestra en la sección superior de la página.

Este módulo del sistema no cuenta con lógica del negocio implementada.

## Home

Es una estructura de contenidos definida utilizando las prestaciones de Wagtail, a diferencia de FlexPage, que está diseñada para gestionar los contenidos genéricos de forma flexible, Home está pensada para exponer los contenidos de la landing page de la aplicación, siendo más restringida en sus posibilidades.

El modelo que se encuentra dentro de este módulo es el siguiente.

- HomePage. Encargado de mantener la información de la landing page, y su lógica de renderizado.

Dentro del modelo HomePage se encuentran los siguientes atributos.

- content. Atributo del tipo StreamField, utilizado para gestionar de forma dinámica los contenidos que se desean exponer en esta página, dentro de un conjunto predefinidos de contenidos disponibles.
- subtitle. Atributo de tipo texto, utilizado para almacenar el texto que se muestra en la sección superior de la página.

## Streams

En este módulo se encuentran todos los tipos de contenidos que fueron generados de forma personalizada, para que luego sean utilizadas en las diversas páginas que hacen uso de Wagtail para su manipulación.

Representan la unidad más atómica sobre la cual se construyen los contenidos en el sistema.

Los modelos que se encuentran en este módulo son los siguientes, debido a que son elementos relativamente simples en comparación con el resto de modelos, se expone sus características principales, y los atributos que poseen, a excepción del modelo `BulletinListingPage` que cuenta con mayor complejidad.

- `BulletinListingPage`. Modelo encargado de almacenar y renderizar los diversos boletines informativos que son generados, utilizado para generar la página de boletines del VUCC.
- `TitleAndTextBlock`. Estructura de contenido que presenta un título y a continuación un texto.
- `RichTextBlock`. Estructura de contenido que presenta un texto, con funcionalidades enriquecidas, como la capacidad de definir texto en negrita, crear listas, insertar links, imágenes, entre tantas otras opciones.
- `SimpleTextBlock`. Estructura de contenido que presenta un texto.
- `Bulletin`. Estructura de contenido que presenta un texto, acompañado de su fecha de creación.
- `CardElement`. Estructura de contenido para generar tarjetas, cuenta con una imagen, un texto a modo de cuerpo, y la posibilidad de agregar un link para la generación automática de un botón que referencie a dicho link.
- `HorizontalAlignElements`. Estructura de contenido que distribuye los elementos que se le asignen de forma horizontal. Puede almacenar de manera dinámica una cantidad no definida de elementos tipo `CardElement` (tarjetas)
- `Question`. Estructura de contenido utilizada para almacenar un texto que represente una consulta, y sus subyacentes respuestas. Empleada en la sección de preguntas frecuentes. La pregunta almacenada es un elemento de texto, y las respuestas son tantas como el usuario defina para una pregunta dada, estas respuestas son elementos de tipo texto.
- `FAQ`. Estructura de contenido utilizada para almacenar las preguntas (`Questions`) y sus subyacentes respuestas. Empleada en la sección de preguntas frecuentes, puede almacenar tantas preguntas como el usuario defina.
- `Instagram`. Estructura de contenido utilizada para almacenar la información de contactos de Instagram. Cuenta con un atributo de tipo texto donde se almacena el nombre del usuario de Instagram.

- Whatsapp. Estructura de contenido utilizada para almacenar la información de contactos de Whatsapp, números telefónicos. Cuenta con un atributo de tipo texto donde se almacena el respectivo número.
- DonacionesVoluntariado. Estructura de contenido utilizada para almacenar las propuestas de donación y apoyo a los voluntariados. Cuenta con un atributo de tipo texto donde se almacena el voluntariado o la causa social de la donación, una imagen para exponer junto a este texto, un elemento de tipo texto con el fin de almacenar un alias bancario, una estructura dinámica de texto para exponer los elementos sugeridos a ser donados, y por último una lista dinámica de los contactos para esta campaña de donación.
- Donación. Estructura de contenido utilizada para almacenar de forma dinámica una cantidad no definida de DonacionesVoluntariado (elementos de donación)

Cabe destacar que, además de los atributos mencionados anteriormente, cada uno de estos modelos define la plantilla con la cual se renderizarán los contenidos. También, como puede observarse, se presentan estructuras anidadas, donde una estructura puede encontrarse dentro de otra, y así sucesivamente, para lograr generar diseños más complejos y dinámicos.

#### Atributos del modelo BulletinListingPage.

- custom\_title. Tipo de dato texto, almacena el texto que se muestre en la parte superior de la página.
- content. Tipo de dato StreamField. almacena de forma dinámica, y sin una cantidad definida, elementos de tipo Bulletin (los boletines o novedades propiamente, elemento definido en el apartado anterior)

Además, el modelo BulletinListingPage cuenta con la lógica para, además de presentar los boletines, generar tarjetas en la parte superior de la página, donde cada una de ellas hace referencia a uno de los voluntariados a los que está inscripto el voluntario, permitiendo ingresar a las actividades de los mismos al hacerle click al botón presente dentro de cada tarjeta.

## Site settings

Módulo acotado, en este se almacena la información de contacto, de la que luego se nutren los elementos del footer de la página para la exposición de los mismos.

Cuenta con un único modelo, `SocialMediaSettings`, que presenta tres atributos.

- `whatsapp`. Atributo de tipo texto en donde se almacena el número telefónico referido al contacto de whatsapp (en caso de existir) del área del VUCC.
- `instagram`. Atributo de tipo texto en donde se almacena el nombre de la cuenta de Instagram (en caso de existir) del área del VUCC.
- `email`. Atributo de tipo texto en donde se almacena el correo electrónico (en caso de existir) del área del VUCC.

Mediante la utilización de funcionalidades de Wagtail, estos atributos son manipulados en una pestaña dentro de la página de administración de la aplicación.

## Usuarios

Módulo destinado a la manipulación de los voluntarios, su información y funcionalidades referidas.

Los voluntarios son los miembros participantes del VUCC, tanto los que realicen las actividades, los coordinadores, y hasta el directivo del VUCC es un voluntario dentro del sistema, cada uno con sus permisos y atributos asignados de forma adecuada.

Cuenta con un único modelo.

- `Volunteer`. Dentro de este modelo se encuentran los atributos de los voluntarios.

Atributos del modelo `Volunteer`.

- `user`. Atributo de clave foránea, referencia al modelo `User` (importado, es obtenido de Django como un modelo utilizado de forma estándar en la gestión de usuarios dentro de un sistema). `User` posee los atributos básicos de un usuario dentro de un sistema web (contraseña, mail, nombre, apellido, etc), mientras que `Volunteer` extiende los mismos para adaptarse al modelo

de negocio de los voluntariados. Relación uno a uno, un voluntario refiere a un usuario, y un usuario puede ser referido a un único voluntario.

- address. Atributo de tipo texto, almacena la dirección de la residencia del voluntario.
- phone. Atributo de tipo texto, almacena el número telefónico del voluntario.
- neighborhood. Atributo de tipo texto, almacena el barrio donde reside el voluntario.
- university. Atributo de tipo texto, almacena la universidad de la que proviene el voluntario.
- university\_file. Atributo de tipo texto, almacena el legajo universitario del voluntario.
- career. Atributo de tipo texto, almacena la carrera universitaria que realiza el voluntario.
- birthdate. Atributo de tipo fecha, almacena la fecha de nacimiento del voluntario.
- dni. Atributo de tipo texto, almacena el número de documento del voluntario.
- ingreso. Atributo de tipo fecha, almacena la fecha en la que el voluntario inició sus actividades en el VUCC.
- coordinador. Atributo del tipo boolean, almacena si el voluntario es coordinador.
- validated. Atributo del tipo boolean, almacena si el voluntario ha sido validado para realizar actividades de los voluntariados.

Además de esto, en la sección de vistas de este módulo de la solución se encuentra la lógica para.

- Generar el reporte para voluntarios, en el que se puede observar las asistencias que posee dicho voluntario.
- Generar el reporte para coordinadores y directivos en el cual pueden observar las asistencias de un voluntario, dentro del ciclo actual del VUCC.

## Voluntariados

Módulo referido a la manipulación de los voluntariados del VUCC, su información y funcionalidades.

Los voluntariados representan la suborganización interna del VUCC, si bien todos los voluntarios forman parte del VUCC, cada uno de ellos forma parte de uno, o más, voluntariados en específico. Los voluntarios participan en actividades, producto de su vinculación con algún voluntariado.

Cuenta con un único modelo.

- Volunteering. Dentro de este modelo se encuentra la información de los voluntariados.

Los atributos del modelo Volunteering son los siguientes.

- name. Atributo de tipo texto, refiere al nombre del voluntariado.
- description. Atributo de tipo texto, aquí se almacena una descripción del voluntariado.
- short\_description. Atributo de tipo texto, aquí se almacena una descripción del voluntariado más resumida, para ser expuesta en dispositivos móviles en lugar del contenido almacenado en *description*.
- volunteers. Atributo de relación muchos a muchos. Referencia al modelo Volunteer, de la forma que un voluntario puede estar en múltiples voluntariados, y un voluntariado puede tener muchos voluntarios.
- coordinadores. Atributo de relación muchos a muchos. Referencia al modelo Volunteer, aquí se define cuáles voluntarios son coordinadores de los voluntariados. Un voluntario puede ser coordinador de muchos voluntariados, un voluntariado puede tener múltiples coordinadores.
- image. Atributo del tipo imagen, se utiliza para almacenar el logo del voluntariado.

Además de esto, en la sección de vistas de este módulo de la solución se encuentra la lógica para.

- Exponer la totalidad de los voluntariados disponibles, ofreciendo visualmente las funcionalidades asignadas según el nivel de acceso del voluntario (poder realizar inscripciones, ver asistencias).
- Acceder a las actividades referidas a un voluntariado en particular.
- Exponer los voluntarios pertenecientes al voluntariado y aquellos que no.
- Poder realizar la inscripción de voluntarios al voluntariado.

- Poder remover la inscripción de voluntarios al voluntariado.
- Poder otorgar permisos de coordinador a un voluntario.
- Remover los permisos de coordinador a un voluntario.
- Obtener la cantidad de asistencias, de cada voluntario inscripto en el voluntariado, en actividades referidas al voluntariado en cuestión.
- Al remover la validación de un voluntario, automáticamente se anula la inscripción del mismo a las actividades a las que esté vinculado.

## Vehículos

Módulo referido a la manipulación de los vehículos de los voluntarios, su información, y funcionalidades.

Si bien el sistema no tiene como fin manejar con gran granularidad, ni corroborar exhaustivamente la información brindada por los voluntarios en este aspecto, es de gran utilidad para los coordinadores conocer las capacidades de movilidad de los voluntarios que se van a presentar a una actividad, esto les sirve, por ejemplo, para gestionar el movimiento de materiales, y organizar cómo es que los voluntarios van a ir y volver de las actividades.

Cuenta con un único modelo.

- Vehicle. Dentro de este modelo se almacena la información de los vehículos.

Los atributos del modelo Vehicle son los siguientes.

- domain. Atributo de tipo texto, refiere a la patente del vehículo.
- brand. Atributo de tipo texto, refiere a la marca del vehículo.
- model. Atributo de tipo texto, refiere al modelo del vehículo.
- proprietary. Atributo de tipo clave foránea, referencia al modelo Volunteer. Un vehículo pertenece a un voluntario, un voluntario puede poseer múltiples vehículos.
- activity. Atributo de clave foránea, referencia al modelo ActivityDetailPage mediante una relación muchos a muchos. Un vehículo puede estar vinculado a muchas actividades, y una actividad puede tener asignada múltiples vehículos.

Además de esto, en la sección de vistas de este módulo de la solución se encuentra la lógica para.

- Exponer los vehículos vinculados a un voluntario.
- Agregar un nuevo vehículo a un voluntario.
- Eliminar un vehículo de un voluntario.
- Aumentar las plazas disponibles de un vehículo.
- Decrementar las plazas disponibles de un vehículo.
- Vincular un vehículo con una actividad.
- Desvincular un vehículo con una actividad.

## Reinicio

Módulo referido a la manipulación de los reinicios de ciclos del VUCC, su información y funcionalidades.

Este módulo sirve para representar el inicio de un nuevo año del VUCC, en el caso de una institución educativa correspondiente con un nuevo ciclo lectivo. A principio de años, es necesario que el director del VUCC pueda desinscribir a todos los voluntarios de sus voluntariados, y que, algunos de los reportes referidos a las asistencias también se reinicien, volviendo nuevamente a cero.

Cabe aclarar que, si bien se desvincula a los voluntarios de los voluntarios, su información, sus asistencias vinculadas, sus vehículos cargados siguen referenciados al mismo. No elimina información de ningún tipo, únicamente desliga a los voluntarios de los voluntariados, a fin de que los mismos tengan que volver a manifestar su voluntad de participar nuevamente en el VUCC durante el año corriente.

Cuenta con un único modelo.

- Restart. En este modelo se almacena la información referida a los reinicios de las etapas del VUCC.

Los atributos del modelo Restart son los siguientes.

- date. Atributo de tipo fecha, almacena la fecha en la que se llevó a cabo el reinicio del VUCC.

Además de esto, en la sección de vistas de este módulo de la solución se encuentra la lógica para.

- Desinscribir a todos los voluntarios de sus respectivos voluntariados.
- Desinscribir a todos los voluntarios de las actividades.
- Desvincular a todos los vehículos de las actividades.

## Módulos importados

A continuación se comentarán aquellos módulos no triviales que son importados para completar la solución generada, se obviará aquellos que sean parte de forma predeterminada de Django y Wagtail.

### Django All-Auth

Este módulo es importado para hacer frente a los procesos de inicio de sesión, registro de nuevos usuarios, recuperación de contraseñas, gestión de las sesiones durante la actividad de un usuario en el sistema, entre otras funciones. Ofrece un paquete de funcionalidades muy completo y robusto para la gestión de los usuarios en el sistema de forma integral.

### psycopg2-binary

Es un paquete de Python que proporciona una implementación de la interfaz de programación de bases de datos (DB-API) para PostgreSQL. Es un controlador o adaptador de base de datos que permite a los programas de Python interactuar con bases de datos PostgreSQL.

### Whitenoise

Es una biblioteca de middleware para aplicaciones web Django que sirve archivos estáticos directamente desde la aplicación en lugar de delegar esta tarea al servidor web. Los archivos estáticos incluyen cosas como hojas de estilo CSS, archivos JavaScript, imágenes, etc.

Cuando una aplicación Django se despliega en producción, se enfrenta al problema de servir archivos estáticos de manera eficiente y segura. Whitenoise ayuda a abordar este problema al servir archivos estáticos directamente desde la aplicación, eliminando la necesidad de configuraciones y servidores adicionales.

### dj-database-url

Proporciona una utilidad para cargar la configuración de la base de datos desde la URL de la base de datos. En lugar de tener que especificar los detalles de la base de datos directamente en el archivo de configuración de Django (settings.py), puedes utilizar una URL de base de datos única y este paquete se encargará de parsear esa URL para obtener la configuración de la base de datos.

## Gunicorn

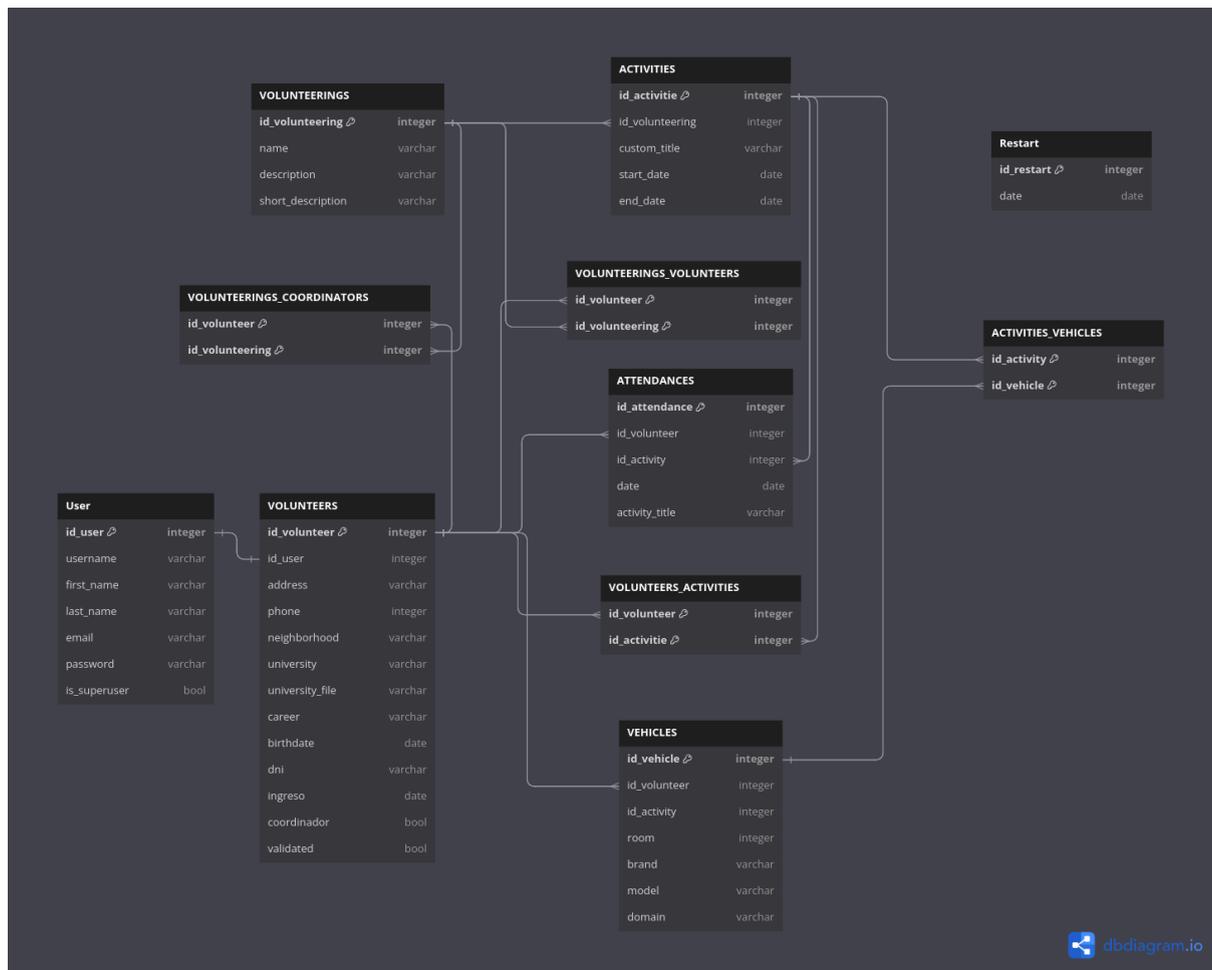
Es un servidor web HTTP para aplicaciones web Python, en el contexto de Django, gunicorn se utiliza comúnmente como un servidor WSGI (Web Server Gateway Interface) para desplegar aplicaciones web Django en entornos de producción, en este caso desde Render. WSGI define una interfaz estándar entre las aplicaciones web escritas en Python y los servidores web.

## python-decouple

Utilizado para separar la configuración del código. Es útil debido a que permite almacenar en variables de entorno información sensible o privada de la aplicación, como pudieran ser claves para acceder a recursos externos. En el servicio web de Render se definen variables de entorno, las cuales pueden accederse fácilmente mediante este paquete.

## Diagrama de base de datos

Se presenta un diagrama simplificado de la base de datos que representa las tablas y sus relaciones utilizadas para abordar los procesos de negocio desarrollados en el VUCC. Este diagrama excluye deliberadamente los modelos destinados a la gestión de contenidos. La omisión de estos modelos se justifica por la complejidad inherente a la forma en que Wagtail administra y organiza los contenidos para abordar su amplia variedad de posibilidades, la inclusión de estas múltiples tablas en el diagrama dificultaría la claridad visual y añadiría escasa información relevante en relación con la solución implementada. Por tanto, el diagrama presentado se centra específicamente en las tablas y relaciones pertinentes a los procesos del VUCC, excluyendo aquellas relacionadas con la gestión de contenidos.



## Usuarios del sistema

A continuación se mencionan los tipos de usuario que interactúan con el sistema, sus necesidades, responsabilidades y limitaciones.

### Usuario anónimo

Se trata de aquel usuario que no ha iniciado sesión en el sistema, presumiblemente no tenga un usuario registrado. Este usuario tiene la necesidad de visualizar los diversos voluntariados existentes, sus actividades, propuestas de donación, una sección de preguntas frecuentes e información de contacto del VUCC.

Su interacción con el sistema tiene fines informativos y de exploración, no pudiendo, por ejemplo, acceder a las funcionalidades para inscribirse en una actividad, cargar sus vehículos, ver sus asistencias, etc.

Es el usuario con acceso más restringido, dentro de sus acciones y funcionalidades principales están las de:

- iniciar sesión
- registrarse en el sistema (en el caso de no haberlo hecho previamente)

### Voluntario no validado

Es aquel usuario registrado en el sistema, y que su atributo *validated* no se encuentra habilitado. En el proceso de negocios del VUCC se trataría presumiblemente de aquel voluntario que recientemente ha creado su cuenta y todavía no ha sido habilitada por parte del director del VUCC para poder acceder a las funcionalidades del sistema, es decir, tiene valor *False*.

Debido a esto es el usuario registrado con menor acceso a las funcionalidades del sistema, teniendo las mismas posibilidades de visualización de contenidos que un usuario anónimo, más la posibilidad de acceder a las asistencias que tenga cargadas a su nombre, si es que existen.

## Voluntario validado

Se trata del usuario registrado con el nivel de acceso inmediatamente posterior al voluntario no validado, se diferencia de este producto a que su atributo *validated* está habilitado, es decir, es *True*. En el modelo de negocio del VUCC se trata de los voluntarios que participan en las actividades, sin involucrarse en la organización de las mismas.

Hereda todas las capacidades y posibilidades del voluntario no validado.

Dentro de los procesos del VUCC, puede:

- Agregar vehículos.
- Eliminar vehículos.
- Vincular vehículos de una actividad a la que esté inscripto.
- Desvincular vehículos de una actividad a la que esté inscripto.
- Inscribirse en una actividad de un voluntariado al que esté inscripto.
- Desinscribirse de una actividad de un voluntariado al que esté inscripto.
- Acceder a la sección de los boletines del VUCC.

## Coordinador voluntariado

Se trata del usuario registrado con el nivel de acceso inmediatamente posterior al voluntario validado, se diferencia de este producto a que su atributo *coordinador* está habilitado, es decir, es *True*. En el caso del voluntario validado no coordinador, este atributo se encuentra deshabilitado (tiene valor *False*).

En el modelo de negocio del VUCC se trata de los voluntarios que participan en las actividades, involucrándose en la organización de las mismas. Toman las asistencias, son la presencia organizativa del VUCC durante la ejecución de las actividades, participando también en la planificación de las mismas.

Heredan todas las capacidades y posibilidades del voluntario validado.

Dentro de los procesos del VUCC, puede:

- Inscribir voluntarios en los voluntariados donde el coordinador se encuentre inscripto (un coordinador solo puede inscribir voluntarios dentro del voluntariado en el que se encuentre inscripto).
- Desinscribir voluntarios en los voluntariados donde el coordinador se encuentre inscripto (idem en cuanto a restricciones del ítem anterior).
- Observar la totalidad de las asistencias de todos los voluntarios.
- Acceder al reporte de las asistencias de cada voluntario, dentro del voluntariado en el que se encuentre inscripto el coordinador, durante el periodo de actividad actual del VUCC.
- Observar la lista de inscriptos, dentro de las actividades del voluntariado en el que esté inscripto.
- Tomar asistencias dentro de las actividades del voluntariado en el que esté inscripto.
- Observar los registros de asistencias de una actividad, agrupados por día, del periodo actual del VUCC, de los voluntariados en el que esté inscripto.
- Observar los registros de asistencias de una actividad, agrupados por voluntario, del periodo actual del VUCC, de los voluntariados en el que esté inscripto.
- Reiniciar las inscripciones en las actividades del voluntariado en el que esté inscripto.

## Directivo VUCC

Se trata del usuario registrado con mayor nivel de acceso del sistema, se diferencia del resto de usuario producto a su atributo *is\_superuser* proveniente del modelo User vinculado al voluntario, el cual se encuentra habilitado (tiene valor *True*). En el caso del resto de usuarios registrados, este atributo se encuentra deshabilitado (tiene valor *False*).

En el modelo de negocio del VUCC se trata del director del mismo, centrándose en aspectos organizativos de mayor nivel. Es, entre tantas cosas, quien valida a los candidatos a voluntarios para participar en el VUCC y quien designa a los coordinadores. En el marco del sistema es además quien gestiona los contenidos del mismo.

Heredan todas las capacidades y posibilidades del voluntario coordinador.

Dentro de los procesos del VUCC, puede:

- Inscribir voluntarios en cualquier voluntariado.
- Desinscribir voluntarios en cualquier voluntariado.
- Inscribir coordinadores en cualquier voluntariado.
- Desinscribir coordinadores en cualquier voluntariado.
- Acceder al reporte de las asistencias de cada voluntario, dentro de cualquier voluntariado, durante el periodo de actividad actual del VUCC.
- Observar la lista de inscriptos, dentro de las actividades de cualquier voluntariado.
- Tomar asistencias dentro de las actividades de cualquier voluntariado.
- Observar los registros de asistencias de una actividad, agrupados por día, del periodo actual del VUCC, de cualquier voluntariado.
- Observar los registros de asistencias de una actividad, agrupados por voluntario, del periodo actual del VUCC, de cualquier voluntariado.
- Reiniciar las inscripciones en las actividades de cualquier voluntariado.
- Luego del 1 de febrero de cada año, ejecutar el reinicio de ciclo del VUCC, para iniciar un nuevo periodo.
- Validar voluntarios.
- Desvalidar voluntarios.
- Otorgar permisos de coordinador.
- Remover permisos de coordinador.
- Ejecutar acciones del tipo ABM sobre las cuentas de los voluntarios.
- Ejecutar acciones del tipo ABM sobre los voluntariados existentes.
- Ejecutar acciones del tipo ABM sobre las actividades de los voluntariados..

Además, tiene acceso total para modificar la información y contenidos disponibles en la plataforma.

## Pantallas del sistema

Las siguientes imágenes son capturas tomadas del sitio publicado en internet, si bien los contenidos pueden modificarse y variar, la estructura se mantiene.

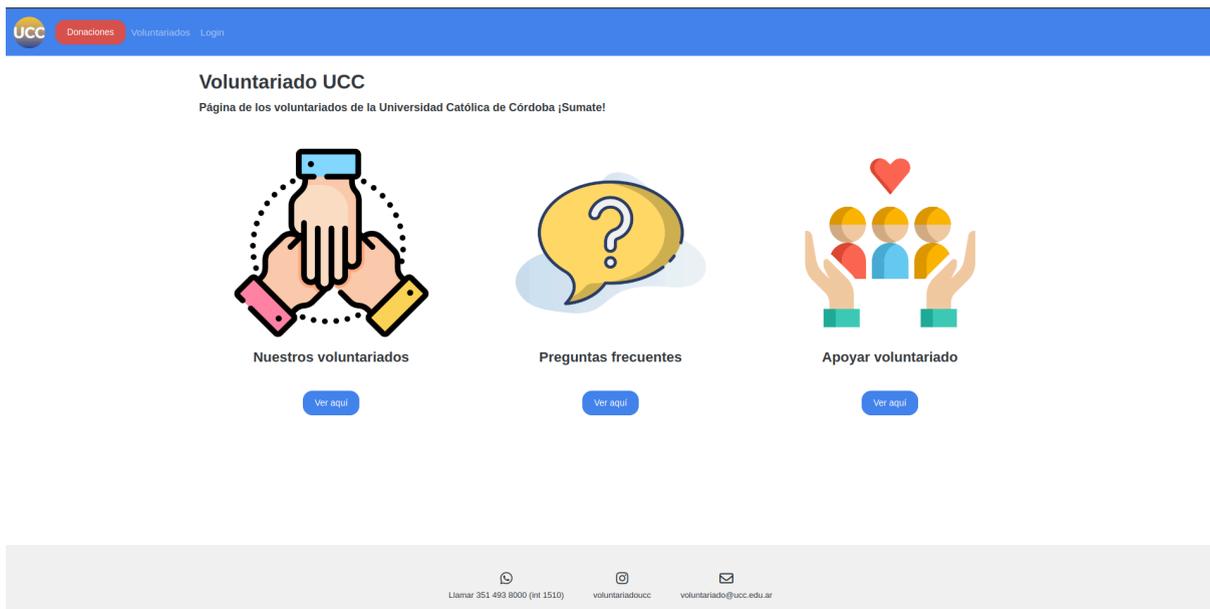
No se mostrarán todas las posibles pantallas dentro de la aplicación, puesto a su gran cantidad, y, en algunos casos, poca relevancia para con la solución final implementada. Se exponen las principales pantallas del sistema, cuyo valor es clave para hacer frente a las necesidades del VUCC, acompañadas de una breve descripción de las mismas.

Se expondrán también los cambios que se observan según el tipo de usuario que se encuentre interactuando con el sistema, los usuarios de mayor jerarquía heredan los elementos y contenidos de aquellos usuario con un nivel jerárquico inferior. Esta característica prevalece en todas las pantallas, por lo que no se mencionará en cada caso, sino que se da por sentado.

### Home page

A continuación se detalla la página de inicio del sistema.

#### Usuario anónimo



The screenshot shows the home page of the VUCC website. At the top, there is a blue navigation bar with the UCC logo and links for 'Donaciones', 'Voluntariados', and 'Login'. Below the navigation bar, the main heading is 'Voluntariado UCC' with the subtitle 'Página de los voluntariados de la Universidad Católica de Córdoba ¡Sumate!'. The page features three main content blocks, each with an icon, a title, and a 'Ver aquí' button:

- Nuestros voluntariados**: Represented by an icon of three hands shaking.
- Preguntas frecuentes**: Represented by a yellow speech bubble with a question mark.
- Apoyar voluntariado**: Represented by an icon of three people with a heart above them.

At the bottom of the page, there is a grey footer bar containing contact information: 'Llamar 351 493 8000 (int 1510)', 'voluntariadoucc', and 'voluntariado@ucc.edu.ar'.

Esta pestaña es la que se obtiene al ingresar al sitio, presenta a modo de título y subtítulo una breve referencia de los contenidos y temática del sitio.

También se observan accesos a sectores clave de la página, como pueden ser los voluntariado (primer ítem), preguntas frecuentes (segundo ítem) y las diversas formas de apoyar al VUCC y sus voluntariado (tercer ítem)

En la parte inferior de la pantalla se observa la información del contacto del VUCC, cargada por los directivos del mismo.

Al estar en el sistema como modo de usuario anónimo (no inició sesión), en la barra de navegación de la parte superior se observan 3 elementos, el primero, con texto "Donaciones" redirige al apartado donde se encuentran los métodos de apoyo al VUCC, en el segundo elemento "Voluntariados" redirige a la sección donde se encuentran la totalidad de los voluntariados del VUCC, por último, con el texto "Login" se encuentra el elemento que redirecciona al formulario para iniciar sesión en el sistema.

### Usuario voluntario, validado y no validado

The screenshot shows the 'Voluntariado UCC' page. At the top is a blue navigation bar with the UCC logo and links for 'Donaciones', 'Voluntariados', 'Vehículos', 'Asistencias', 'Boletín', and 'Logout'. Below the navigation bar, the page title is 'Voluntariado UCC' with the subtitle 'Página de los voluntariados de la Universidad Católica de Córdoba ¡Sumate!'. The main content area features three cards:

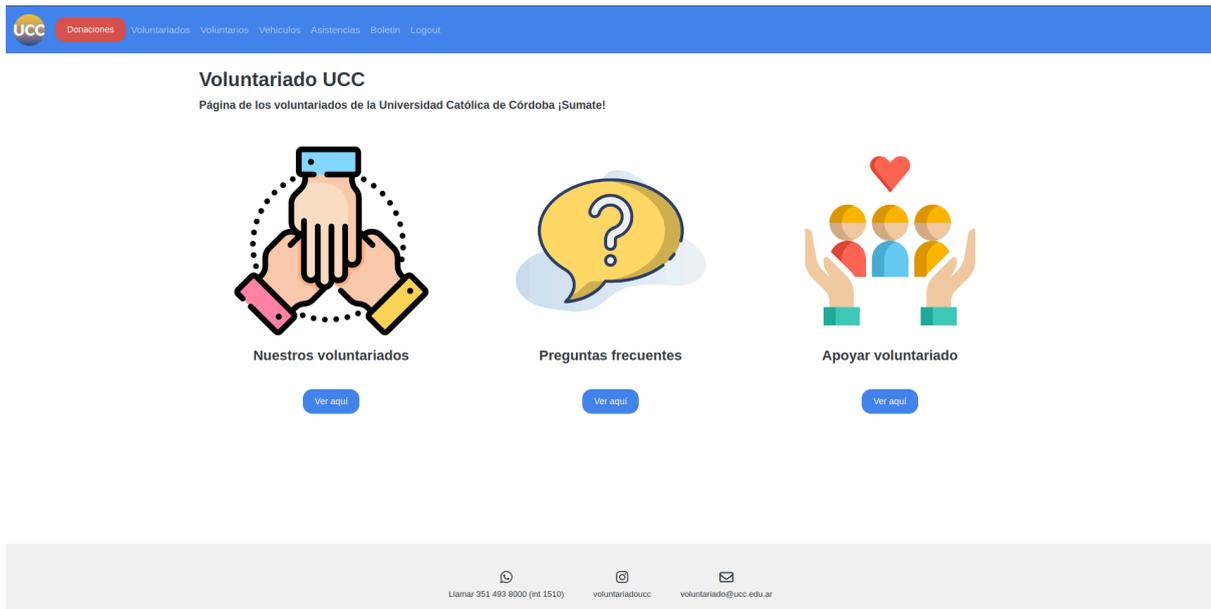
- Nuestros voluntariados:** An icon of two hands shaking, with the text 'Nuestros voluntariados' and a 'Ver aquí' button below it.
- Preguntas frecuentes:** An icon of a speech bubble with a question mark, with the text 'Preguntas frecuentes' and a 'Ver aquí' button below it.
- Apoyar voluntariado:** An icon of three people with a heart above them, with the text 'Apoyar voluntariado' and a 'Ver aquí' button below it.

At the bottom of the page, there is a light gray footer containing contact information: 'Llamar 351 493 8000 (nt 1510)', 'voluntariadoucc', and 'voluntariado@ucc.edu.ar'.

En el caso de aquellos usuarios que inicien sesión en el sistema, el contenido se mantiene igual a excepción de la barra de navegación, en donde se agregan 3 elementos, el primero "Vehículos" redirige a la pantalla donde se pueden observar y gestionar los vehículos vinculados con el voluntario en cuestión, el segundo "Asistencias" redirige a la sección en donde un voluntario puede consultar las asistencias que tiene registradas a su

nombre, por último, el ítem “Boletín” redirige a la pantalla donde se exponen los voluntariados al que se encuentra inscripto el voluntario, y las novedades publicadas por el VUCC; en caso de que el voluntario no se encuentre validado (usuario del sistema [voluntario no validado](#)) se expone un cartel en el que se le indica al voluntario su estado de no validado.

## Coordinador



UCC Donaciones Voluntariados Voluntarios Vehiculos Asistencias Boletín Logout

### Voluntariado UCC

Página de los voluntariados de la Universidad Católica de Córdoba ¡Sumate!

 Nuestros voluntariados [Ver aquí](#)

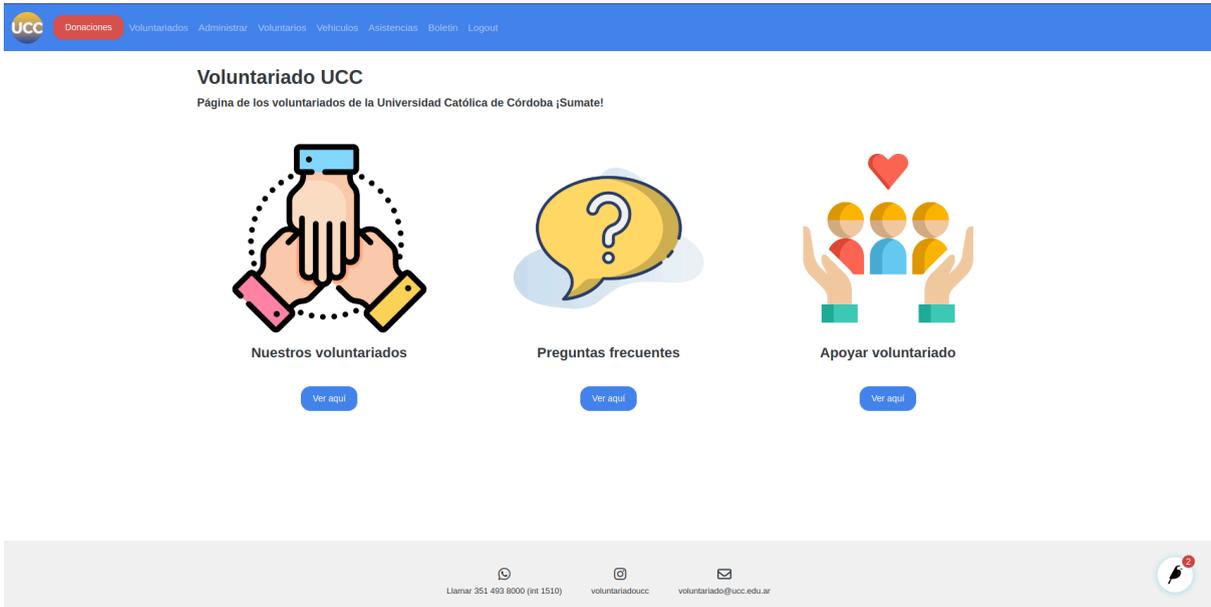
 Preguntas frecuentes [Ver aquí](#)

 Apoyar voluntariado [Ver aquí](#)

Llamar 351 493 8000 (int 1510)  voluntariadoucc  voluntariado@ucc.edu.ar

Para aquellos usuario que inicien sesión en el sistema con un [usuario coordinador](#), en la barra de navegación se les agrega un elemento con texto “Voluntarios”, este redirige a la sección en donde se pueden observar las asistencias de un voluntario en un periodo determinado.

## Directivo VUCC



The screenshot shows the 'Voluntariado UCC' page. At the top is a blue navigation bar with the UCC logo and menu items: Donaciones, Voluntariados, Administrar, Voluntarios, Vehículos, Asistencias, Boletín, and Logout. Below the navigation bar, the page title is 'Voluntariado UCC' with the subtitle 'Página de los voluntariados de la Universidad Católica de Córdoba ¡Sumate!'. The main content area features three cards: 1. 'Nuestros voluntariados' with an icon of three hands shaking and a 'Ver aquí' button. 2. 'Preguntas frecuentes' with a question mark icon and a 'Ver aquí' button. 3. 'Apoyar voluntariado' with an icon of three people and a heart, and a 'Ver aquí' button. At the bottom is a footer with contact information: phone number 351 493 8000 (ext 1510), Instagram handle @voluntariadoucc, email voluntariado@ucc.edu.ar, and a social media icon with a notification badge.

En el caso de aquellos usuarios que inicien sesión en el sistema con un usuario [directivo del VUCC](#), el contenido se mantiene igual a excepción de la barra de navegación, en donde se agregan un elemento con texto “Administrar”, este redirige a la sección provista por Wagtail para los administradores del sitio, en donde se gestiona la información del mismo.

Estos cambios que se observan en la barra de navegación están presentes en el resto de pantallas donde se encuentre la misma, por lo que estos no se volverán a destacar en el resto de vistas del sistema.

El footer, en donde se muestra la información de contacto del VUCC, también se encuentra presente en el resto de pantallas (a excepción de la vista de administración provista por Wagtail), por lo que su presencia no se mencionará en las siguientes pantallas, se da por sentado.

## Voluntariados

### Usuario anónimo - Voluntario (tanto no validado como validado)

UCC
Donaciones
Voluntariados
Vehículos
Asistencias
Boletín
Logout



Voluntario

**Villa Rivadavia**

Somos un grupo de estudiantes que desde inicios del año 2004 trabajamos con un grupo de niños/as en el barrio Villa Rivadavia Anexo al sur de nuestra ciudad, realizando básicamente apoyo integral, fomentando cuestiones tales como la educación, creatividad, integración, salud, etc.

[Ver](#)



**La Candelaria**

Somos estudiantes y jóvenes profesionales que brindan de manera desinteresada, comprometida, responsable y continua parte de su tiempo para compartir momentos con las comunidades de zonas rurales de la zona de La Candelaria en el departamento Cruz del Eje de la provincia de Córdoba.

[Ver](#)



**Rural Monte del Rosario**

Disponemos de parte de nuestro tiempo para compartir momentos con las comunidades de zonas rurales del interior de la provincia de Córdoba, en Monte del Rosario. Esta actividad se caracteriza por tratar a la persona de forma integral.

[Ver](#)



**Puentes**

Somos un grupo de estudiantes de distintas carreras de la Universidad que, a partir del año 2021, nos propusimos "crear puentes" entre comunidades. Como grupo, acompañamos un sábado al mes a niños, niñas y adolescentes que viven en una residencia de la Secretaría Nacional de Niñez, Adolescencia y Familia.

[Ver](#)



**Nuestro Hogar III**

Nuestra tarea principal es el apoyo escolar, basándonos en una formación integral de cada niño/a, fomentando la creatividad, el respeto por el otro, el compañerismo y la amistad, y la salud, entre otros en el barrio Nuestro Hogar III.

[Ver](#)

Aquí se exponen las diversas ofertas de voluntariados que se encuentran en el VUCC, se presentan en modo de tarjetas los diversos voluntariados, un logo de los mismos, su nombre y una breve descripción, además, en el extremo derecho se encuentra un botón con texto "Ver" en donde al pulsarlo redirige al usuario a las actividades cargadas para este voluntariado.

En la esquina superior izquierda de dichas tarjetas se expone si el usuario se encuentra inscripto como voluntario en dicho voluntariado.

## Coordinador

UCC
Donaciones
Voluntariados
Voluntarios
Vehiculos
Asistencias
Boletem
Logout

Coordinador



**Villa Rivadavia**

Somos un grupo de estudiantes que desde inicios del año 2004 trabajamos con un grupo de niños/as en el barrio Villa Rivadavia Anexo al sur de nuestra ciudad, realizando básicamente apoyo integral, fomentando cuestiones tales como la educación, creatividad, integración, salud, etc.

Inscriptos

Asistencias

Ver

Voluntario



**La Candelaria**

Somos estudiantes y jóvenes profesionales que brindan de manera desinteresada, comprometida, responsable y continua parte de su tiempo para compartir momentos con las comunidades de zonas rurales de la zona de La Candelaria en el departamento Cruz del Eje de la provincia de Córdoba.

Ver

RURAL Voluntariado



**Rural Monte del Rosario**

Disponemos de parte de nuestro tiempo para compartir momentos con las comunidades de zonas rurales del interior de la provincia de Córdoba, en Monte del Rosario. Esta actividad se caracteriza por tratar a la persona de forma integral.

Ver

Puentes



**Puentes**

Somos un grupo de estudiantes de distintas carreras de la Universidad que, a partir del año 2021, nos propusimos "crear puentes" entre comunidades. Como grupo, acompañamos un sábado al mes a niños, niñas y adolescentes que viven en una residencia de la Secretaría Nacional de Niñez, Adolescencia y Familia.

Ver

Nuestro Hogar III



**Nuestro Hogar III**

Nuestra tarea principal es el apoyo escolar, basándonos en una formación integral de cada niño/a, fomentando la creatividad, el respeto por el otro, el compañerismo y la amistad.

Ver

Además de la información detallada previamente, en aquellos voluntariados en donde el voluntario se encuentre con permisos de coordinador, se agregan dos botones extra, el primero “Inscriptos”, permite acceder a la lista de voluntarios que se encuentran inscriptos, y los que no, al voluntariado, brindando la posibilidad de definir cuáles voluntarios pertenecen al voluntariado. El segundo botón que se agrega, “Asistencias” muestra un reporte en donde se observan todos los voluntarios inscriptos al voluntariado, y sus respectivas asistencias en actividades del mismo.

En estos voluntariados donde el voluntario es coordinador se expone en la esquina superior derecha un texto “Coordinador”.

## Directivo VUCC

UCC

[Donaciones](#)
[Voluntariados](#)
[Administrar](#)
[Voluntarios](#)
[Vehículos](#)
[Asistencias](#)
[Boletín](#)
[Logout](#)

Voluntario
Coordinadores

**Villa Rivadavia**

Somos un grupo de estudiantes que desde inicios del año 2004 trabajamos con un grupo de niños/as en el barrio Villa Rivadavia Anexo al sur de nuestra ciudad, realizando básicamente apoyo integral, fomentando cuestiones tales como la educación, creatividad, integración, salud, etc.

Inscriptos
Asistencias
Ver

Voluntario
Coordinadores

**La Candelaria**

Somos estudiantes y jóvenes profesionales que brindan de manera desinteresada, comprometida, responsable y continua parte de su tiempo para compartir momentos con las comunidades de zonas rurales de la zona de La Candelaria en el departamento Cruz del Eje de la provincia de Córdoba.

Inscriptos
Asistencias
Ver

Voluntario
Coordinadores

**Rural Monte del Rosario**

Disponemos de parte de nuestro tiempo para compartir momentos con las comunidades de zonas rurales del interior de la provincia de Córdoba, en Monte del Rosario. Esta actividad se caracteriza por tratar a la persona de forma integral.

Inscriptos
Asistencias
Ver

Voluntario
Coordinadores

**Rural Monte del Rosario**

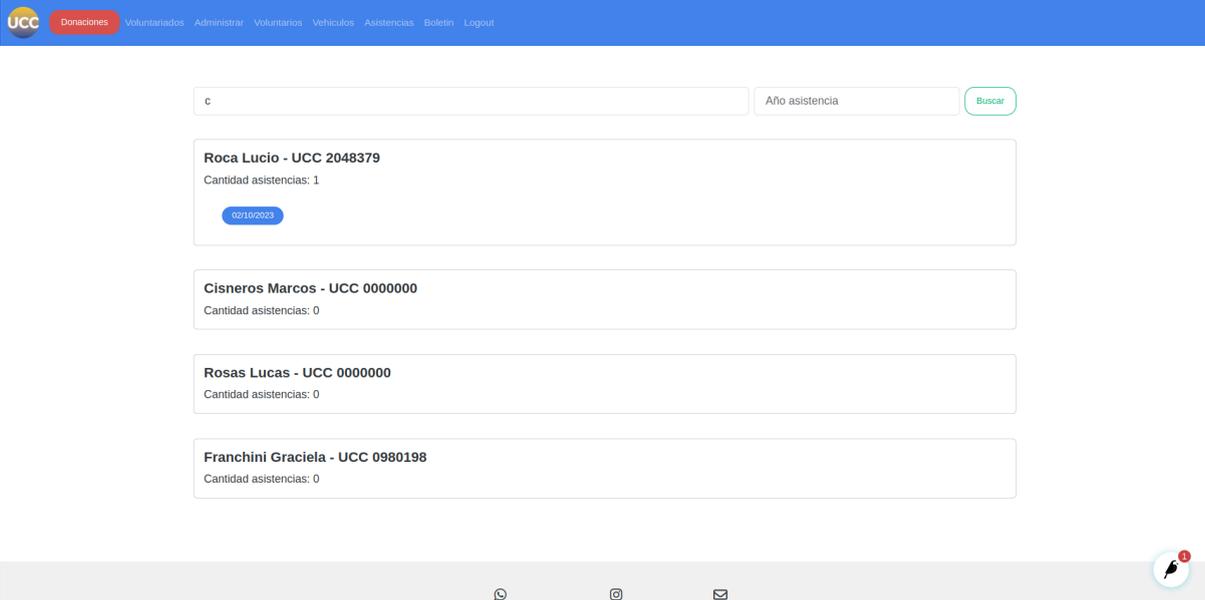
Disponemos de parte de nuestro tiempo para compartir momentos con las comunidades de zonas rurales del interior de la provincia de Córdoba, en Monte del Rosario. Esta actividad se caracteriza por tratar a la persona de forma integral.

Inscriptos
Asistencias
Ver

A aquellos usuarios directivos del VUCC se presentan las cualidades expuestas para los coordinadores en todos los voluntariado (no únicamente en aquellos voluntariados donde se encuentren inscriptos con permisos de coordinador), además, se agrega un botón extra, “Coordinadores”, el cual posibilita el acceso a la lista de voluntarios coordinadores, tanto aquellos vinculados en calidad de coordinador como aquellos que aún no forman parte del voluntariado en dicho rol, brindando la posibilidad de definir cuáles voluntarios son coordinadores del voluntariado.

## Asistencias voluntarios

### Coordinadores - Directivos del VUCC



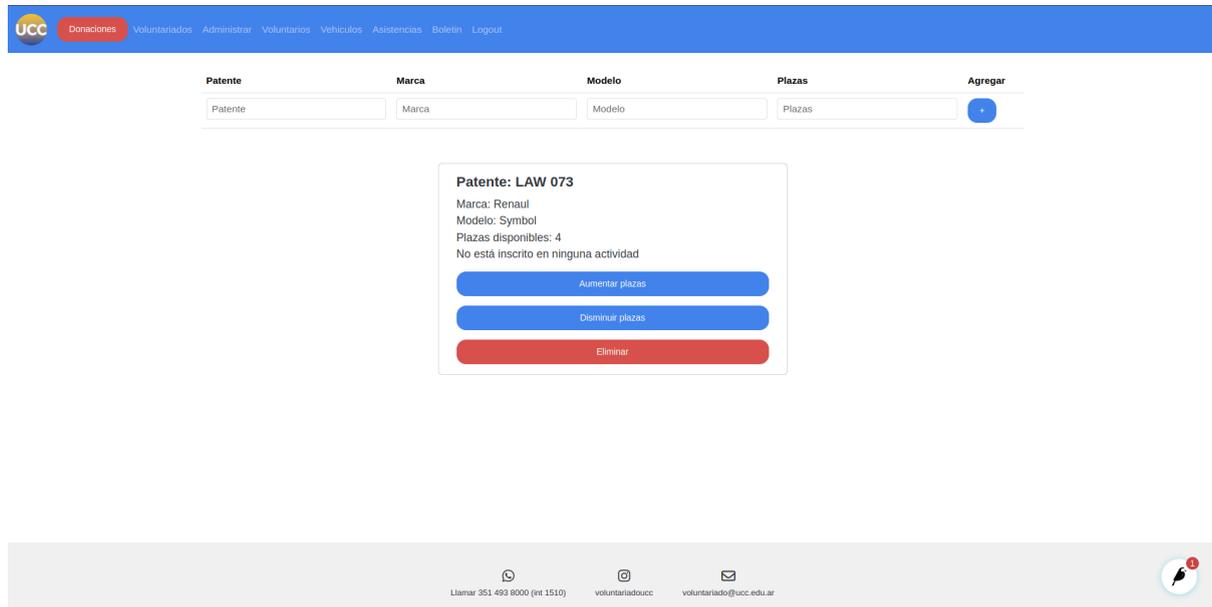
The screenshot shows a web application interface for managing volunteer attendances. At the top, there is a navigation bar with the UCC logo and several menu items: Donaciones, Voluntariados, Administrar, Voluntarios, Vehículos, Asistencias, Boletín, and Logout. Below the navigation bar, there is a search form with two input fields and a search button. The first input field contains the letter 'c', and the second input field is labeled 'Año asistencia'. The search button is labeled 'Buscar'. Below the search form, there are four search results displayed in a list format. Each result shows the name of the coordinator, their UCC ID, and the number of attendances. The first result is 'Roca Lucio - UCC 2048379' with 1 attendance, and a date '02/10/2023' is shown below it. The other three results are 'Cisneros Marcos - UCC 0000000', 'Rosas Lucas - UCC 0000000', and 'Franchini Graciela - UCC 0980198', all with 0 attendances. At the bottom of the page, there is a footer with social media icons for WhatsApp, Instagram, and Email, and a notification icon with a red circle containing the number 1.

En esta pantalla se muestran las asistencias de los voluntarios, tiene presente dos entradas de texto, en base al texto que se ingrese en la primera (de izquierda a derecha) se filtran los voluntarios expuestos según coincidencias del texto con su nombre, apellido y/o dni. La segunda entrada de texto permite ingresar un año sobre el cual se filtrarán las asistencias expuestas. Por ejemplo, si en el primer campo se ingresa “Manuel”, y en el segundo “2022”, se expondrán todas las asistencias registradas para aquellos voluntarios que presenten coincidencias en su nombre/apellido/dni con “Manuel” y que tengan como año de la fecha registrada el año 2022.

Al presionar el botón con texto “Buscar” se realiza la búsqueda de las asistencias.

## Vehículos

### Voluntarios validados - Coordinadores - Directivos VUCC



The screenshot shows the VUCC system interface. At the top, there is a navigation bar with the UCC logo and menu items: Donaciones, Voluntariados, Administrar, Voluntarios, Vehículos, Asistencias, Boletín, and Logout. Below the navigation bar is a form with five input fields: Patente, Marca, Modelo, Plazas, and a blue button with a "+" sign labeled "Agregar".

Below the form, a card displays the details for a specific vehicle:

- Patente: LAW 073
- Marca: Renault
- Modelo: Symbol
- Plazas disponibles: 4
- No está inscrito en ninguna actividad

At the bottom of the card are three buttons: "Aumentar plazas" (blue), "Disminuir plazas" (blue), and "Eliminar" (red).

At the bottom of the page, there is a footer with contact information: "Llamar 351 493 8000 (ext 1510)", "voluntariadoucc", and "voluntariado@ucc.edu.ar". There is also a small circular icon with a red notification bubble in the bottom right corner.

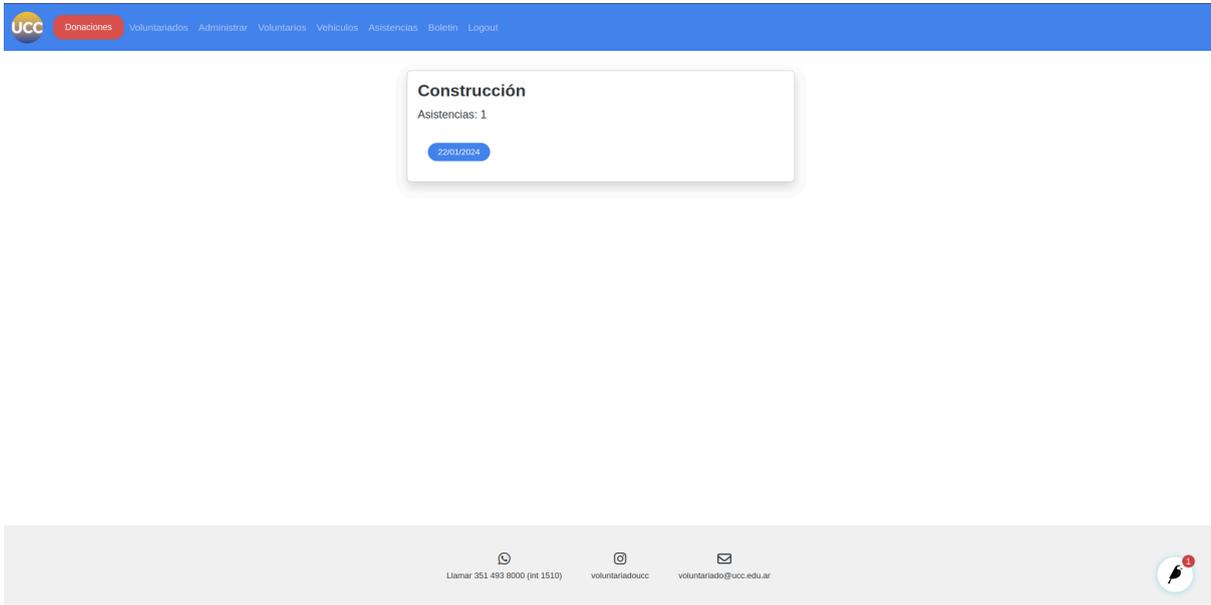
En esta pantalla se exponen los vehículos vinculados a un voluntario, también es donde se realizan las acciones de alta, baja y modificación de los mismos.

En la parte superior se observa un formulario de cuatro campos, en el primero "Patente" se ingresa la patente del vehículo, en el segundo "Marca" se ingresa la marca del vehículo, en el tercero "Modelo" se ingresa el modelo del vehículo, por último, en el cuarto "Plazas" se indican la cantidad de plazas disponibles del vehículo. Como último elemento de este formulario se encuentra un botón con el signo "+", al pulsarlo se agrega el vehículo, siempre y cuando se hayan cargado todos los campos mencionados previamente.

Debajo de este formulario, se encuentra, en forma de tarjetas, todos los vehículos del voluntario, se expone la información del mismo (patente, marca, modelo, cantidad de plazas disponibles) en adición a las actividades a la que se encuentra vinculado dicho vehículo. También se encuentran tres botones, el primero (de arriba para abajo) con texto "Aumentar plazas" al pulsarlo incrementa en uno la cantidad de plazas disponibles del vehículo, el segundo "Disminuir plazas" al pulsarlo decrementa en uno la cantidad de plazas disponibles del vehículo, por último, el tercer botón "Eliminar" al pulsarlo ejecuta la rutina para eliminar el vehículo.

## Asistencias

### Voluntarios validados y no validados - Coordinadores - Directivos VUCC



The screenshot displays the VUCC system interface. At the top, there is a blue navigation bar with the UCC logo and menu items: Donaciones, Voluntariados, Administrar, Voluntarios, Vehículos, Asistencias, Boletín, and Logout. Below the navigation bar, a card titled "Construcción" is shown. Inside the card, it states "Asistencias: 1" and lists a date "22/01/2024" in a blue button. At the bottom of the page, there is a footer with contact information: "Llamar 351 493 8000 (nt 1510)", social media icons for Instagram (@voluntariadoucc) and Email (voluntariado@ucc.edu.ar), and a notification icon with a red "1" badge.

En esta pantalla se exponen las asistencias del voluntario, se muestran a modo de tarjetas, en donde el título refiere al título de la actividad sobre la que hacen referencia los registros de asistencia contenidos en dicha tarjeta, a continuación se muestra un texto "Asistencias: " y procede la cantidad de asistencias que se encuentran en dicha tarjeta. Por último, se encuentran como ítems las fechas de los registros de asistencia.

## Boletines

### Voluntarios no validados

Aquí se le expone al voluntario su estado de no validado por parte de un directivo del VUCC.

### Voluntarios validados - Coordinadores - Directivos VUCC

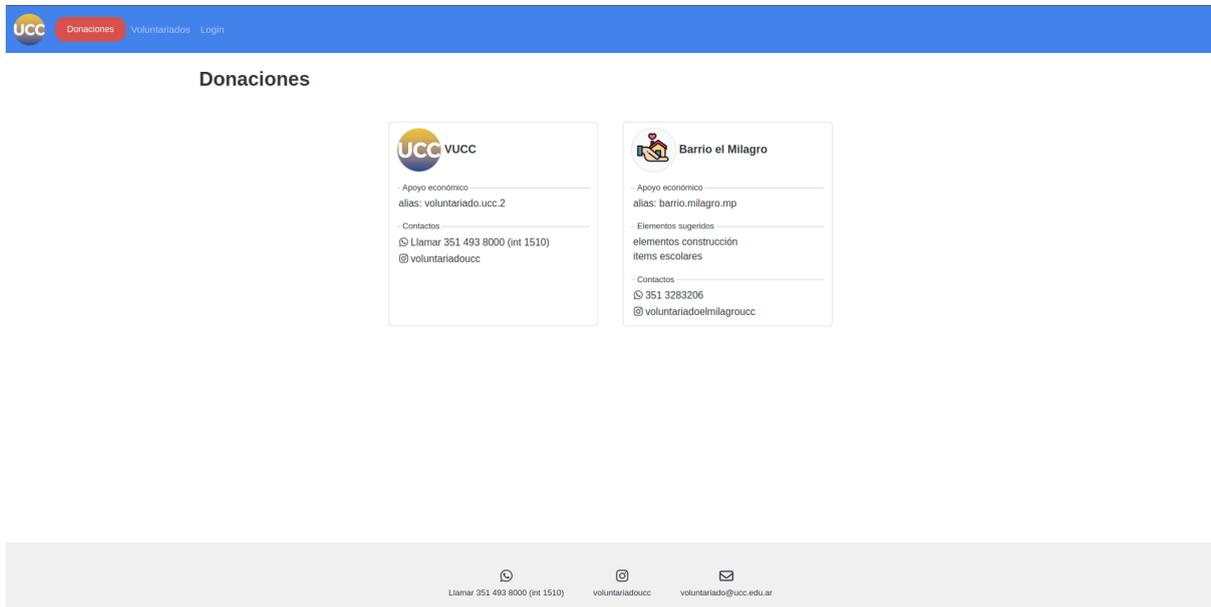
Se saluda al voluntario mediante un mensaje en la parte superior izquierda de la pantalla, luego, se encuentran la lista de voluntariados, a modo de tarjetas, con los que se encuentra vinculado el voluntario, ya sea porque es voluntario o coordinador del mismo. En

estas tarjetas se muestra el logo del voluntariado, seguido por un texto que indica la condición del voluntario en el voluntariado, pudiendo ser el mismo “Voluntario” o “Coordinador” según corresponda, por último se encuentra un botón con texto “Ver”, al pulsarlo se redirige al usuario a las actividades de dicho voluntariado.

Por último, se encuentran los boletines (novedades) publicadas por los directivos del VUCC para mantener informado a los voluntarios.

## Donaciones

### Todos los usuarios



UCC Donaciones Voluntariados Login

### Donaciones



- Apoyo económico  
alias: voluntariado.ucc.2

- Contactos  
☎ Llamar 351 493 8000 (int 1510)  
@ voluntariadoucc



- Apoyo económico  
alias: barrio.milagro.mp

- Elementos sugeridos  
elementos construcción  
items escolares

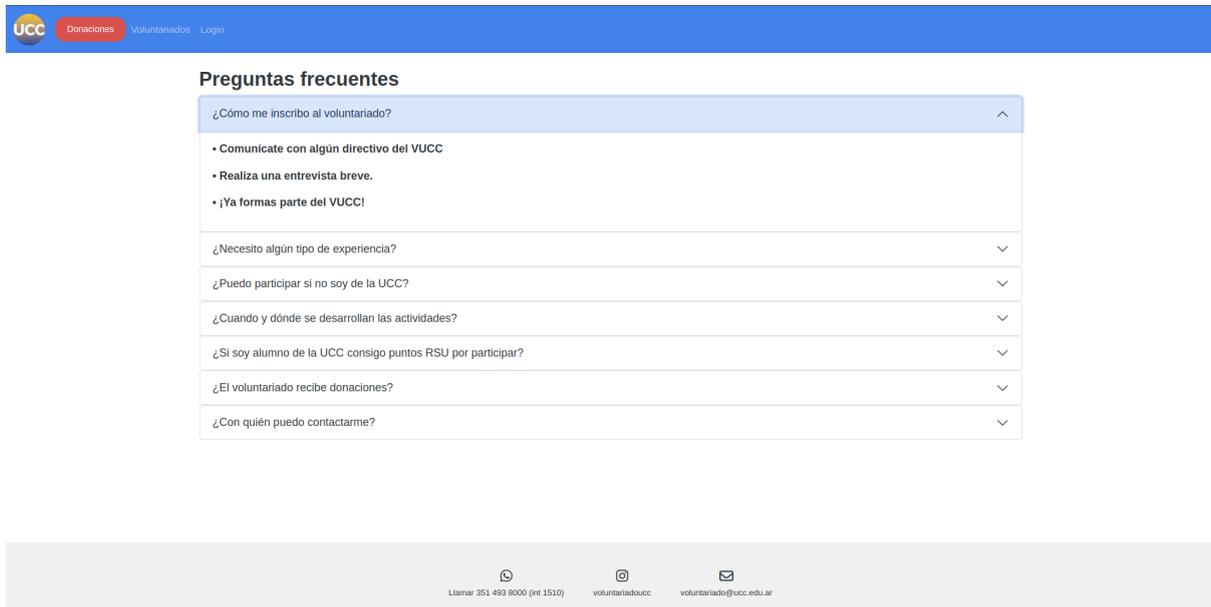
- Contactos  
☎ 351 3283206  
@ voluntariadoelmilagroucc

Llamar 351 493 8000 (int 1510) voluntariadoucc voluntariado@ucc.edu.ar

En esta pantalla se encuentran, a modo de tarjetas, todas las campañas disponibles para apoyar al VUCC y sus voluntariados, estas están formadas por una imagen y título a modo de presentar la misma, y se exponen los elementos sugeridos para ser donados, seguidos por los métodos de contacto para gestionar dichas donaciones.

## Preguntas frecuentes

### Todos los usuarios



The screenshot shows a navigation bar with 'UCC', 'Donaciones', 'Voluntariado', and 'Login'. Below it is a 'Preguntas frecuentes' section with a list of questions and answers. The first question is expanded, showing the answer: 'Comunicate con algún directivo del VUCC', 'Realiza una entrevista breve.', and '¡Ya formás parte del VUCC!'. Other questions include: '¿Necesito algún tipo de experiencia?', '¿Puedo participar si no soy de la UCC?', '¿Cuándo y dónde se desarrollan las actividades?', '¿Si soy alumno de la UCC consigo puntos RSU por participar?', '¿El voluntariado recibe donaciones?', and '¿Con quién puedo contactarme?'. At the bottom, there are contact icons for phone, Instagram, and email, with the phone number 351 493 8000 (ext 1510) and email voluntariado@ucc.edu.ar.

Aquí se encuentran, a modo de secciones retraibles, las preguntas más frecuentes (cargadas por un directivo del VUCC) en conjunto con su respuesta. Al presionar sobre una pregunta se despliega su respuesta, si se vuelve a presionar la misma pregunta se repliega la respuesta, quedando oculta.

# Login

## Usuario anónimo

regístrate primero.' Below this are two input fields: 'Email' and 'Contraseña'. There is a checkbox labeled 'Recordarme: '. At the bottom of the form is a blue button labeled 'Ingresar' and a link labeled 'Olvidaste la contraseña?'. The footer contains contact information: 'Llamar 351 493 8000 (nt 1510)', social media icons for Instagram and email, and the email address 'voluntariado@ucc.edu.ar'." data-bbox="120 162 879 433"/>

**Log In**

Si todavía no has creado una cuenta, [regístrate](#) primero.

Email

Contraseña

Recordarme:

[Ingresar](#) [Olvidaste la contraseña?](#)

Llamar 351 493 8000 (nt 1510) [voluntariadoucc](#) [voluntariado@ucc.edu.ar](mailto:voluntariado@ucc.edu.ar)

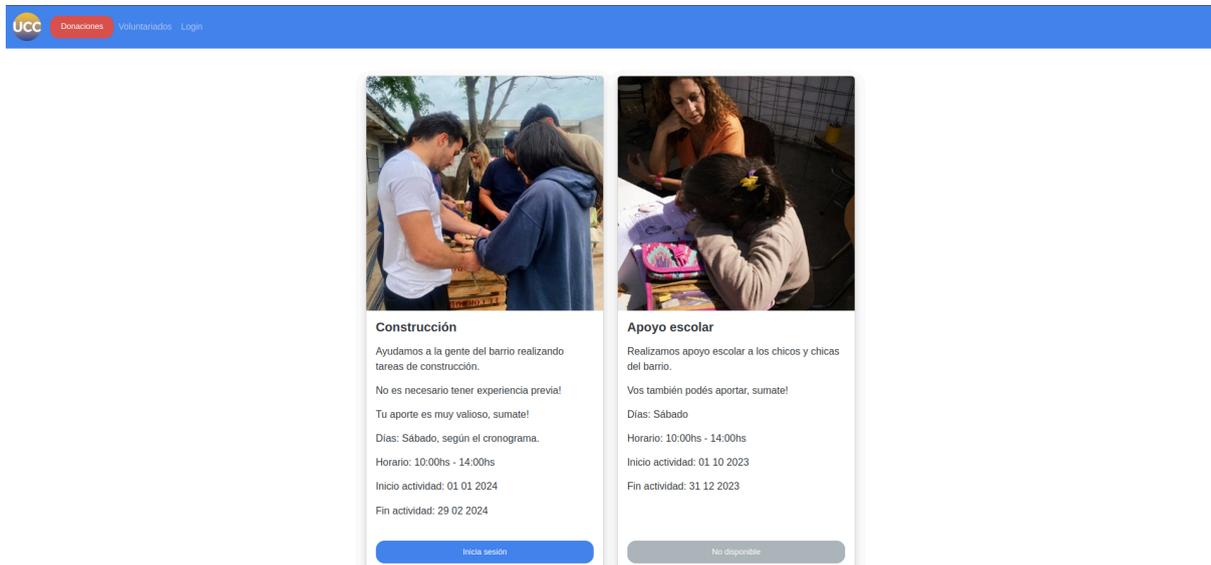
Únicamente disponible para aquellos usuarios que no hayan iniciado sesión, expone un sencillo formulario para realizar el inicio de sesión.

En la parte superior, a través del texto “regístrate” se encuentra una redirección para acceder al formulario en donde se crean los usuarios del sistema, debajo se observa un campo de texto en donde el usuario ingresa su email, seguido por otro campo de texto en donde ingresa su contraseña. Luego hay un ítem tipo checkbox en donde permite al usuario iniciar sesión automáticamente al ingresar al sitio. Por último, se encuentra un botón con texto “Ingresar” para iniciar sesión en el sistema validando la información brindada anteriormente (email y contraseña), acompañado por un texto “Olvidaste la contraseña?” que permite restablecer la misma en caso de ser necesario.



## Actividades

### Usuario anónimo



UCC Donaciones Voluntariados Login

**Construcción**  
Ayudamos a la gente del barrio realizando tareas de construcción.  
No es necesario tener experiencia previa!  
Tu aporte es muy valioso, sumate!  
Días: Sábado, según el cronograma.  
Horario: 10:00hs - 14:00hs  
Inicio actividad: 01 01 2024  
Fin actividad: 29 02 2024  
Inicia sesión

**Apoyo escolar**  
Realizamos apoyo escolar a los chicos y chicas del barrio.  
Vos también podés aportar, sumate!  
Días: Sábado  
Horario: 10:00hs - 14:00hs  
Inicio actividad: 01 10 2023  
Fin actividad: 31 12 2023  
No disponible

En esta pantalla se pueden observar las actividades de un voluntariado en particular, por cada actividad se encuentra una tarjeta, la misma posee una imagen representativa, el título de la actividad, una descripción y la fecha de inicio y fin de la misma. Por último hay un botón en la parte inferior, para aquellas actividades disponibles (aquellas en donde la fecha actual se encuentra dentro de la fecha desde y hasta de la actividad) el botón redirige a la pantalla de login, indicando que debe iniciar sesión, para aquellas actividades en donde la fecha actual no se encuentra dentro de las fechas donde la actividad se encuentra disponible, el botón se encuentra deshabilitado, con un texto que indica este estado.

### Voluntario no validado

The screenshot shows the VUCC website interface. At the top, there is a navigation bar with the UCC logo and menu items: Donaciones, Voluntariados, Vehículos, Asistencia, Boletín, and Logout. Below this, two activity cards are displayed side-by-side. The left card is titled 'Construcción' and includes the following text: 'Ayudamos a la gente del barrio realizando tareas de construcción.', 'No es necesario tener experiencia previa!', 'Tu aporte es muy valioso, sumate!', 'Días: Sábado, según el cronograma.', 'Horario: 10:00hs - 14:00hs', 'Inicio actividad: 01 01 2024', and 'Fin actividad: 29 02 2024'. At the bottom of this card is a grey button that says 'Cuenta no verificada'. The right card is titled 'Apoyo escolar' and includes: 'Realizamos apoyo escolar a los chicos y chicas del barrio.', 'Vos también podés aportar, sumate!', 'Días: Sábado', 'Horario: 10:00hs - 14:00hs', 'Inicio actividad: 01 10 2023', and 'Fin actividad: 31 12 2023'. At the bottom of this card is a grey button that says 'No disponible'.

En el caso de aquellos voluntarios que no han sido validados, el botón que previamente dirigía al login ahora se encuentra deshabilitado indicando que la cuenta no se encuentra habilitada.

### Voluntario validado

The screenshot shows the VUCC website interface for a validated user. The navigation bar is the same as in the previous screenshot. The two activity cards are identical in their text content. However, the 'Construcción' card now features a dropdown menu below the activity details, which is currently set to 'Vehículo LAW 073'. At the bottom of the 'Construcción' card is a blue button that says 'No podrá asistir'. At the bottom of the 'Apoyo escolar' card is a blue button that says '¡Voy a asistir!'.

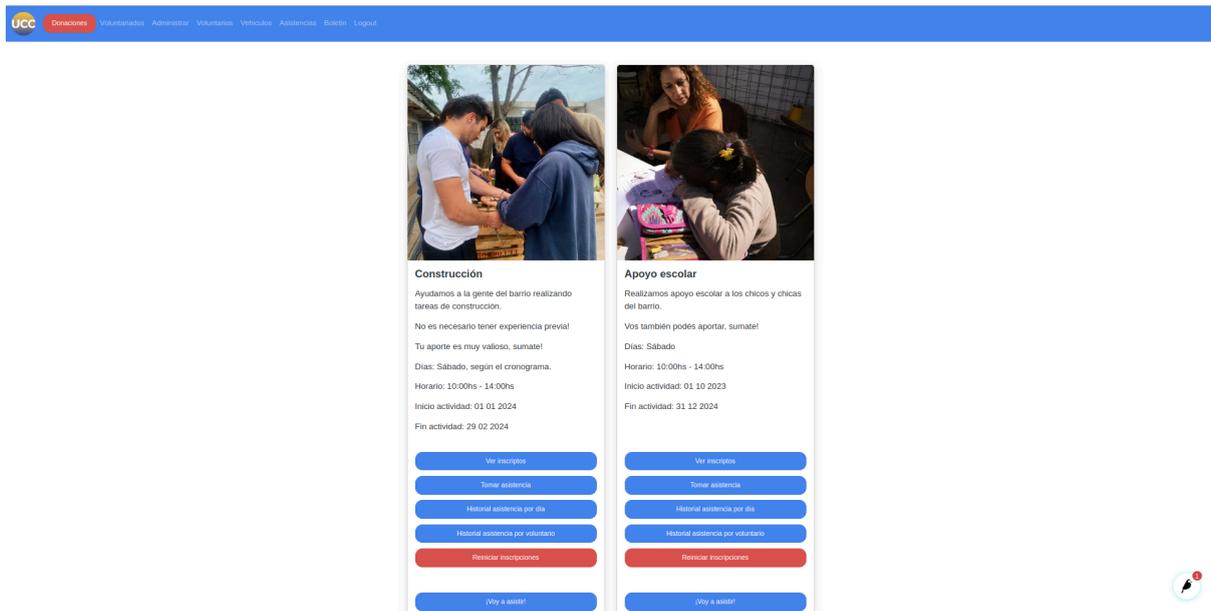
Para aquellos voluntarios validados y que se encuentran inscriptos en el voluntariado, podrán acceder a la funcionalidad para manifestar su deseo de asistir a la próxima instancia de la actividad, esto mediante pulsar el botón que se encuentra al final de

la tarjeta de la actividad que presenta el texto “¡Voy a asistir!”, una vez presionado, aparece un elemento tipo selector en donde se le permite al voluntario vincular el vehículo con el que pretende ir a la actividad, estas opciones de vehículos a seleccionar se obtienen de la lista de vehículos cargadas por el voluntario, además tiene la posibilidad de no vincular ningún vehículo en caso de que no posea esta posibilidad, y por último puede indicar que es pasajero en un vehículo de otro voluntario.

Una vez presionado el botón para manifestar la intención de asistir a la actividad, el botón modifica su texto a “No podré asistir”, en caso de presionar este botón se anula la inscripción del voluntario en la actividad.

Para aquellos voluntarios validados que ingresen a voluntariados donde no se encuentren inscriptos la experiencia es similar a la de los usuarios anónimos y voluntarios no validados. El botón para manifestar la participación se encuentra deshabilitado con un texto que indica que el voluntario no se encuentra inscripto al voluntariado.

## Coordinador - Directivo del VUCC



The screenshot displays the VUCC web interface for a coordinator or director. It features a blue header with the UCC logo and navigation links. Below the header, two activity cards are shown side-by-side:

- Construcción:**
  - Header image: People working on a construction site.
  - Title: **Construcción**
  - Description: Ayudamos a la gente del barrio realizando tareas de construcción.
  - Text: No es necesario tener experiencia previa! Tu aporte es muy valioso, sumate!
  - Días: Sábado, según el cronograma.
  - Horario: 10:00hs - 14:00hs
  - Inicio actividad: 01 01 2024
  - Fin actividad: 29 02 2024
  - Buttons: Ver inscriptos, Tomar asistencia, Historial asistencia por día, Historial asistencia por voluntario, Renovar inscripciones, ¡Voy a asistir!
- Apoyo escolar:**
  - Header image: A woman helping a child with schoolwork.
  - Title: **Apoyo escolar**
  - Description: Realizamos apoyo escolar a los chicos y chicas del barrio.
  - Text: Vos también podés aportar, sumate!
  - Días: Sábado
  - Horario: 10:00hs - 14:00hs
  - Inicio actividad: 01 10 2023
  - Fin actividad: 31 12 2024
  - Buttons: Ver inscriptos, Tomar asistencia, Historial asistencia por día, Historial asistencia por voluntario, Renovar inscripciones, ¡Voy a asistir!

A los coordinadores que ingresen a las actividades de los voluntariados en los que fueron asignados como coordinadores y a los directivos del VUCC se les presenta de esta forma las actividades. Se les expone cinco botones extra, el primero “Ver inscriptos” permite acceder a la lista de inscriptos a la actividad, el segundo “Tomar asistencia” permite ingresar al formulario autogenerado para poder indicar la presencia de los voluntarios inscriptos en la actividad cuando se desarrolló la misma, al presionar el tercer botón “Historial asistencia por

día” se obtiene un reporte de las asistencias registradas sobre dicha actividad agrupadas por día, el cuarto botón “Historial asistencia por voluntariado” permite acceder a un reporte similar al anterior, con la única diferencia de que en vez de encontrarse agrupadas por día, las asistencias se agrupan por voluntario, por último, el quinto botón con texto “Reiniciar inscripciones” permite ejecutar la acción por la cual se desvincula a todos los voluntarios de la actividad, de forma tal que tengan que renovar nuevamente su intención de participar en la próxima iteración de la actividad en cuestión.

## Inscriptos actividad

### Coordinador - Directivo VUCC

UCC
Donaciones
Voluntariados
Administrar
Voluntarios
Vehículos
Asistencias
Boletín
Logout

#### Inscriptos

Nombre	Teléfono	Dirección	Barrio	Vehículo
Bobadilla Manuel	3513283206	Gustavo Brandan 773	Residencial Velez Sarsfield	
Bobadilla Manuel	3513283206	Brandan 773	Res. Velez Sarsfield	Renaul Symbol 4 plazas

  
Llamar 351 493 8000 (int 1510)

  
voluntariadoucc

  
voluntariado@ucc.edu.ar

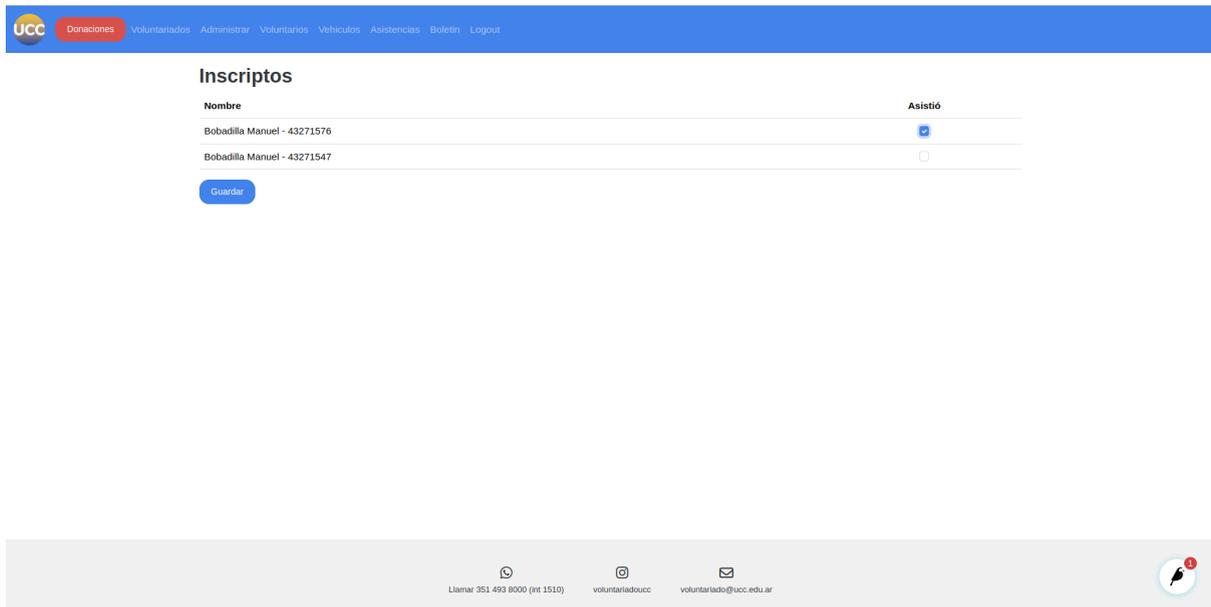


Los coordinadores de los voluntariados y directivos del VUCC tienen acceso a la lista de inscriptos a una actividad, en esta lista se muestran aquellos voluntarios que hayan manifestado su deseo de participar en la próxima realización de una actividad. Esta lista muestra el nombre y apellido, teléfono, dirección, barrio y método de transporte del voluntario.

En caso de no encontrarse inscripto ningún voluntario se muestra un mensaje que expone esta situación.

## Tomar asistencia

### Coordinadores - Directivos VUCC



Nombre	Asistió
Bobadilla Manuel - 43271576	<input checked="" type="checkbox"/>
Bobadilla Manuel - 43271547	<input type="checkbox"/>

Guardar

Llamar 351 493 8000 (int 1510) | voluntariadoucc | voluntariado@ucc.edu.ar

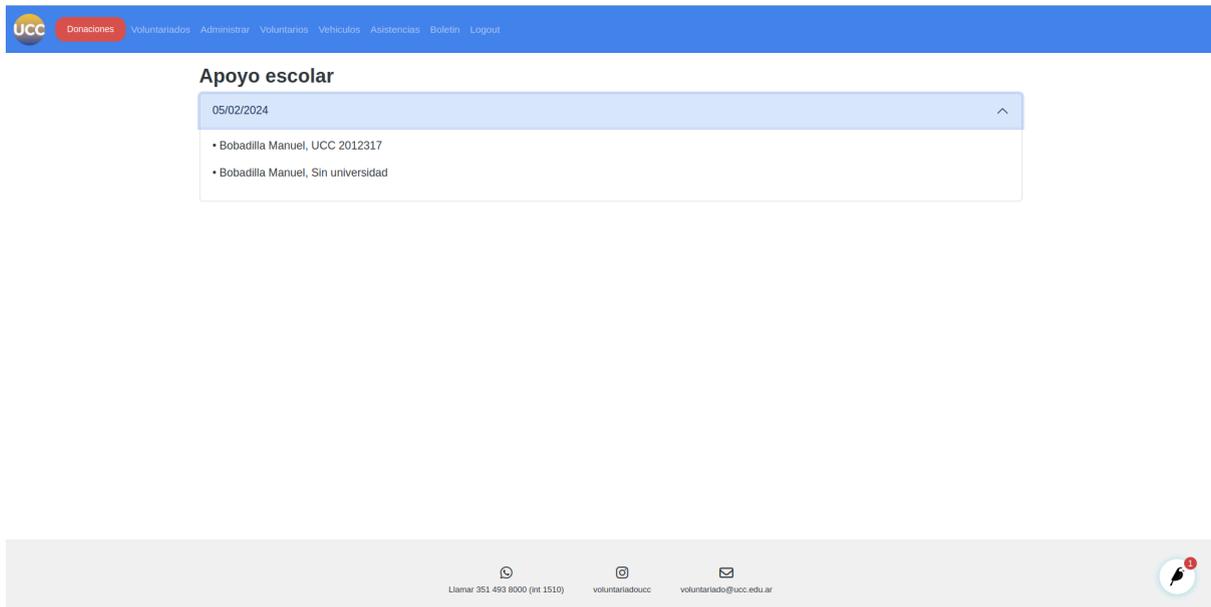
Los coordinadores de los voluntariados y directivos del VUCC tienen acceso al formulario autogenerado para tomar asistencia en una actividad, esta lista está poblada con aquellos voluntarios que hayan manifestado su intención de participar en la instancia de la actividad.

Se muestran dos campos, el primero conformado por el nombre, apellido y dni del voluntario, y en el segundo se encuentran los checkbox donde se indica la asistencia o no del voluntario. Por último, al final del voluntario hay un botón con texto "Guardar", al presionarlos se persisten las asistencias según se indica en el formulario.

En caso de no haber voluntarios inscriptos sobre los cuales tomar la asistencia se muestra un cartel indicando esa situación.

## Historial asistencia por día

### Coordinadores - Directivos del VUCC



UCC Donaciones Voluntariados Administrar Voluntarios Vehiculos Asistencias Boletín Logout

#### Apoyo escolar

05/02/2024

- Bobadilla Manuel, UCC 2012317
- Bobadilla Manuel, Sin universidad

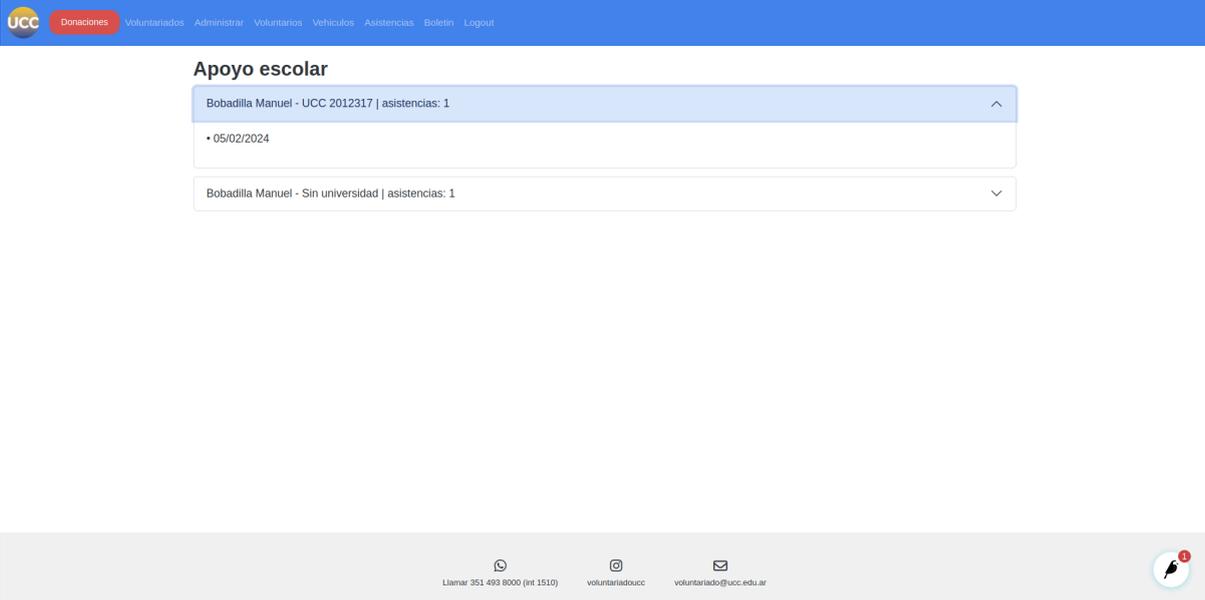
Llamar 351 493 8000 (int 1510) voluntariadoucc voluntariado@ucc.edu.ar

Los coordinadores de los voluntariados y directivos del VUCC tienen acceso al reporte donde se exponen las asistencias vinculadas a una actividad en particular agrupadas por días, al pulsar sobre el elemento que contiene la fecha de la asistencia se expande el mismo mostrando el detalle de quienes son los voluntarios que conforman la asistencia de esa fecha, al volver a presionar el elemento esta información vuelve a ocultarse.

En caso de no haber ninguna asistencia vinculada a la actividad se muestra un cartel que indica dicha situación.

## Historial asistencia por voluntario

### Coordinadores - Directivos VUCC



UCC Donaciones Voluntariados Administrar Voluntarios Vehiculos Asistencias Boletín Logout

#### Apoyo escolar

- Bobadilla Manuel - UCC 2012317 | asistencias: 1
  - 05/02/2024
- Bobadilla Manuel - Sin universidad | asistencias: 1

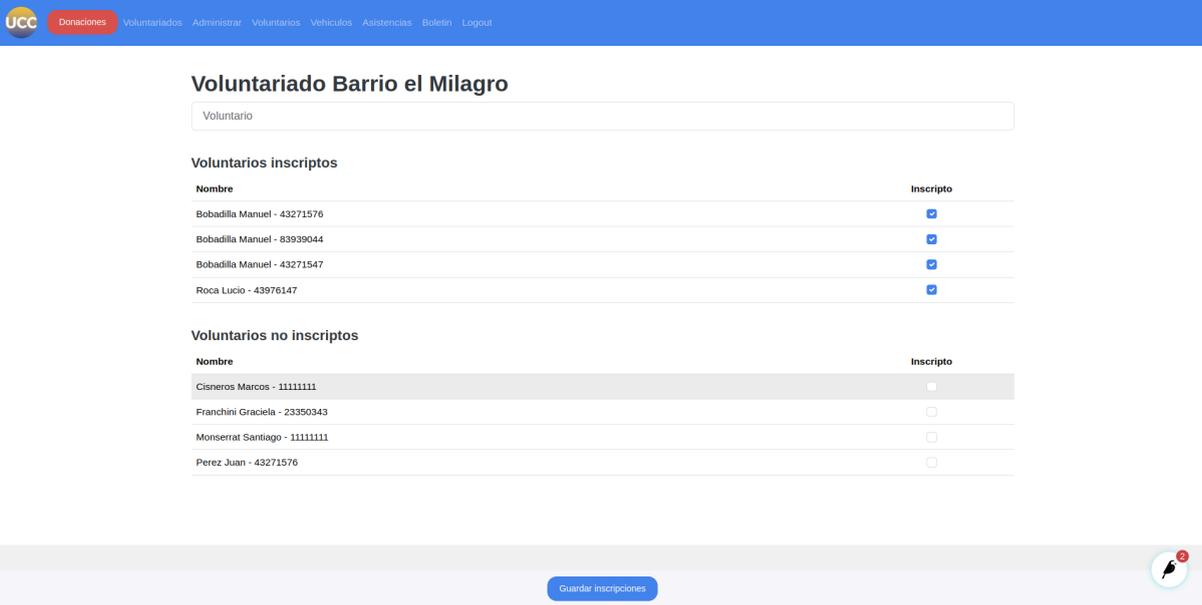
Llamar 351 493 8000 (int 1510) voluntariadoucc voluntariado@ucc.edu.ar

Los coordinadores de los voluntariados y directivos del VUCC tienen acceso al reporte donde se exponen las asistencias vinculadas a una actividad en particular agrupadas por cada voluntario, al pulsar sobre el elemento que contiene la información del voluntario se expande el mismo mostrando el detalle de cuáles son las fechas de las asistencias que tiene registradas, al volver a presionar el elemento esta información vuelve a ocultarse.

En caso de no haber ninguna asistencia vinculada a la actividad se muestra un cartel indicando dicha situación.

## Inscripciones voluntariado

### Coordinadores - Directivos VUCC



**Voluntariado Barrio el Milagro**

Voluntario

**Voluntarios inscriptos**

Nombre	Inscripto
Bobadilla Manuel - 43271576	<input checked="" type="checkbox"/>
Bobadilla Manuel - 83939044	<input checked="" type="checkbox"/>
Bobadilla Manuel - 43271547	<input checked="" type="checkbox"/>
Roca Lucio - 43976147	<input checked="" type="checkbox"/>

**Voluntarios no inscriptos**

Nombre	Inscripto
Cisneros Marcos - 11111111	<input type="checkbox"/>
Franchini Graciela - 23350343	<input type="checkbox"/>
Montserrat Santiago - 11111111	<input type="checkbox"/>
Perez Juan - 43271576	<input type="checkbox"/>

Guardar inscripciones

Los coordinadores de los voluntariados y directivos del VUCC tienen acceso al formulario donde se indica cuáles voluntarios, de la lista de voluntarios validada, pertenecen a un voluntariado determinado, y cuáles no.

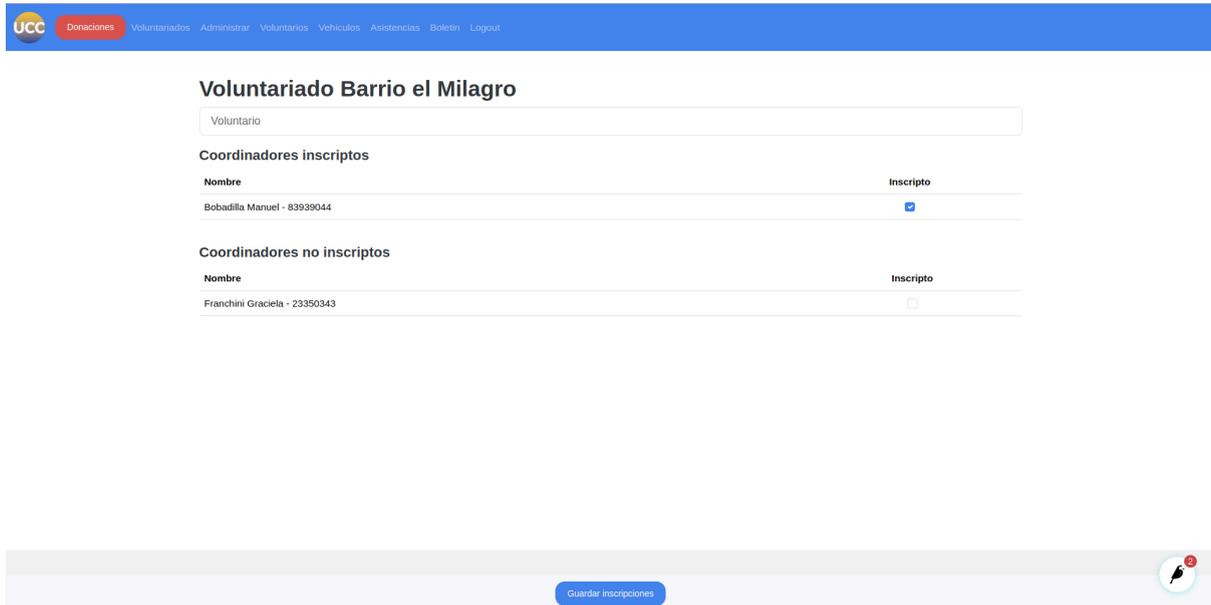
En la parte superior se encuentra un campo de texto en donde el texto ingresado se utiliza para filtrar los voluntarios de la lista, a través de buscar coincidencias entre dicho texto y el nombre en conjunto con el apellido del voluntario, en caso de presentarse la coincidencia se muestra el voluntario, en caso contrario no. Si no se ingresa ningún valor en el campo se muestran todos los voluntarios.

En el formulario se expone el nombre, apellido y dni del voluntario, junto a un elemento tipo checkbox en donde se indica si el voluntario forma parte o no del voluntariado. En la parte inferior de la pantalla se puede observar un botón con texto "Guardar inscripciones" que al pulsarlo se persisten las inscripciones según se indica en el formulario.

En caso de no haber ningún voluntario validado para ser seleccionado se muestra un cartel que expone dicha situación.

## Inscripciones coordinadores voluntariado

### Directivo VUCC



**Voluntariado Barrio el Milagro**

Voluntario

**Coordinadores inscriptos**

Nombre	Inscripto
Bobadilla Manuel - 83939044	<input checked="" type="checkbox"/>

**Coordinadores no inscriptos**

Nombre	Inscripto
Franchini Graciela - 23350343	<input type="checkbox"/>

Guardar inscripciones

Aquellos usuarios que sean directivos del VUCC son los únicos que pueden acceder a este formulario, en este se indica cuáles de los voluntarios validados con permisos de coordinador son coordinadores de un voluntariado determinado.

En la parte superior se encuentra un campo de texto en donde el texto ingresado se utiliza para filtrar los coordinadores de la lista, a través de buscar coincidencias entre dicho texto y el nombre en conjunto con el apellido del coordinador, en caso de presentarse la coincidencia se muestra el coordinador, en caso contrario no. Si no se ingresa ningún valor en el campo se muestran todos los coordinadores.

En el formulario se expone el nombre, apellido y dni del coordinador, junto a un elemento tipo checkbox en donde se indica si el coordinador forma parte o no del voluntariado. En la parte inferior de la pantalla se puede observar un botón con texto "Guardar inscripciones" que al pulsarlo se persisten las inscripciones según se indica en el formulario.

En caso de no haber ningún voluntario validado para ser seleccionado se muestra un cartel que expone dicha situación.

## Panel administración

### Directivos VUCC

**Bienvenido al CMS Wagtail VUCC**  
Manuel Bobadilla

8 Páginas    9 Imágenes    0 Documentos

▲ Wagtail upgrade available. Your version: 5.0.5. New version: 5.2.3. [Lea los detalles de la versión.](#)

**Tus ediciones más recientes**

Apoyo escolar	VER EN VIVO	51 minutos antes
Preguntas	VER EN VIVO	2 horas antes
Donaciones	VER EN VIVO	5 horas antes
Actividad San Carlos	VER EN VIVO	6 horas antes
Home	VER EN VIVO	7 horas antes

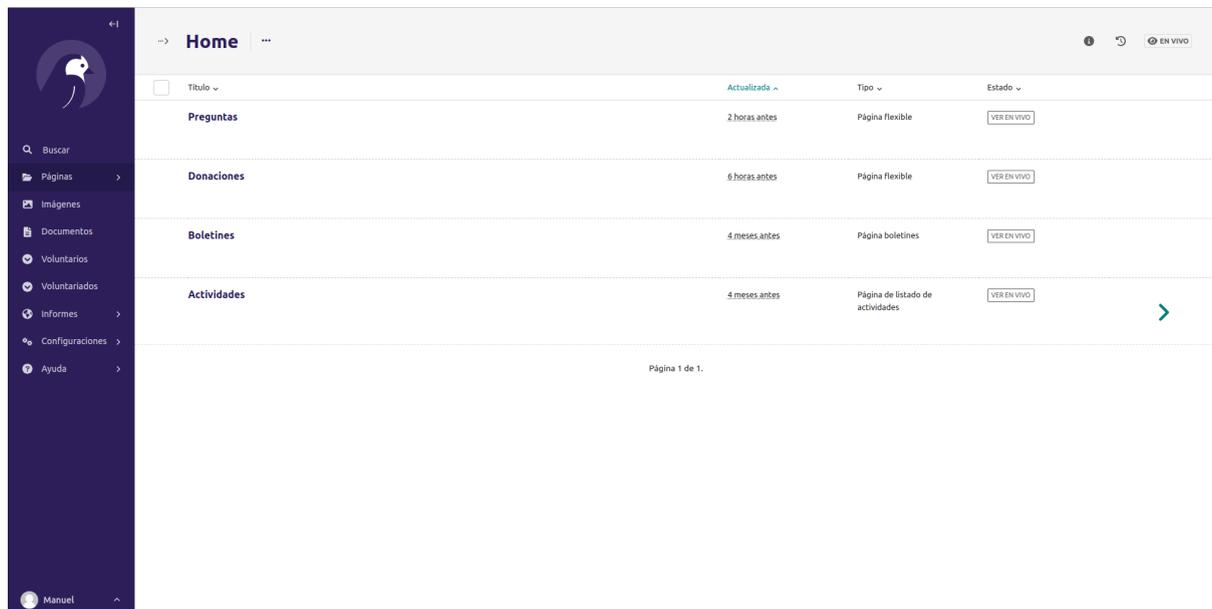
Este panel de administración de los contenidos del sitio es generado por el módulo Wagtail, en él se puede observar en la barra lateral, una serie de elementos a los que se puede acceder para definir ciertos valores clave del sitio, entre estos se destacan, debido a su utilización activa en la solución implementada, los elementos “Páginas”, “Imágenes”, “Voluntarios”, “Voluntariados”, dentro del ítem con texto “Configuración” se encuentran los elementos “Usuarios” y “Contacto redes sociales”. Los contenidos de estos elementos serán abordados más adelante.

En esta pantalla se muestra la información en general del sitio, en la sección principal se exponen cuales son las páginas que recientemente han sido modificadas, y también información sobre la cantidad de páginas cargadas en el sistema, imágenes y documentos.

En esta vista no se realiza ninguna acción propiamente, más allá de ser útil para dirigirse al apartado en particular que se quiera manipular.

## Páginas

### Directivos VUCC

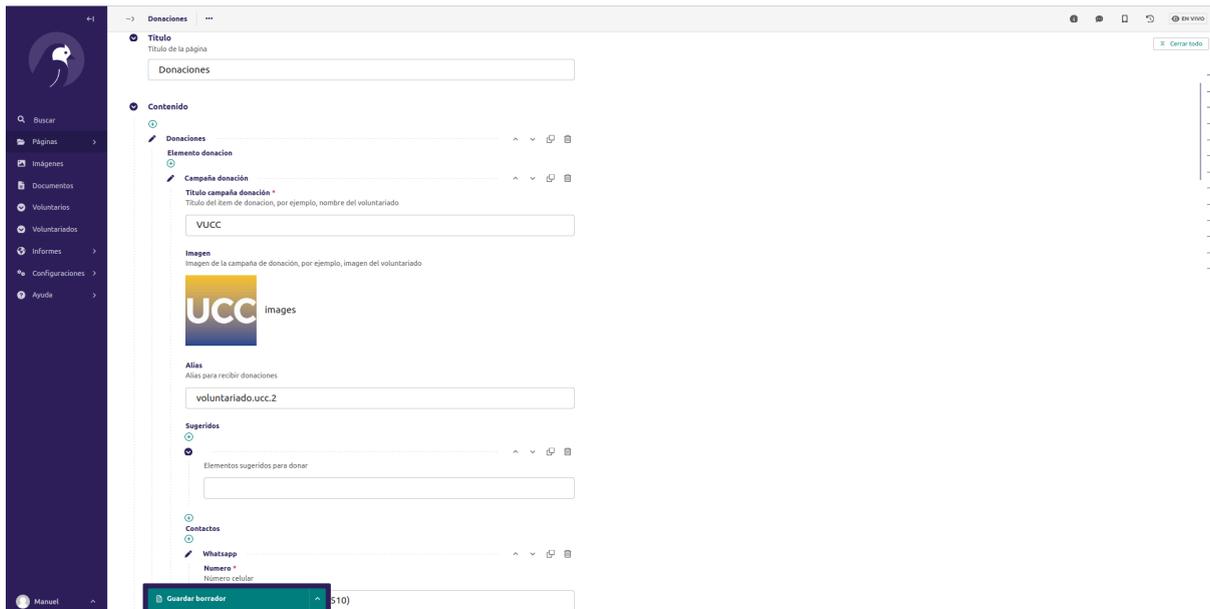


The screenshot shows a web interface for 'Directivos VUCC'. On the left is a dark blue sidebar with a search bar and a menu containing: 'Páginas', 'Imágenes', 'Documentos', 'Voluntarios', 'Voluntariados', 'Informes', 'Configuraciones', and 'Ayuda'. The main content area is titled 'Home' and contains a table of pages. The table has columns for 'Título', 'Actualizado', 'Tipo', and 'Estado'. The rows are: 'Preguntas' (updated 2 hours ago, flexible page), 'Donaciones' (updated 6 hours ago, flexible page), 'Boletines' (updated 4 months ago, bulletin page), and 'Actividades' (updated 4 months ago, list page). Each row has a 'VER EN VIVO' button. At the bottom of the table, it says 'Página 1 de 1'.

Título	Actualizado	Tipo	Estado
Preguntas	2 horas antes	Página flexible	VER EN VIVO
Donaciones	6 horas antes	Página flexible	VER EN VIVO
Boletines	4 meses antes	Página boletines	VER EN VIVO
Actividades	4 meses antes	Página de listado de actividades	VER EN VIVO

Únicamente visible para los directivos del VUCC, la sección de páginas muestra las diversas páginas cargadas en el sistema, las páginas se agrupan de forma tal que hay páginas padre, y bajo estas se encuentran páginas hijas, este esquema es útil para la organización y fácil navegación en la pestaña de administración de las páginas cargadas.

Al presionar sobre alguna de las páginas se muestra en detalle la información de la misma, permitiendo su manipulación, en este caso, por ejemplo, se ingresó a la página “Donaciones”.

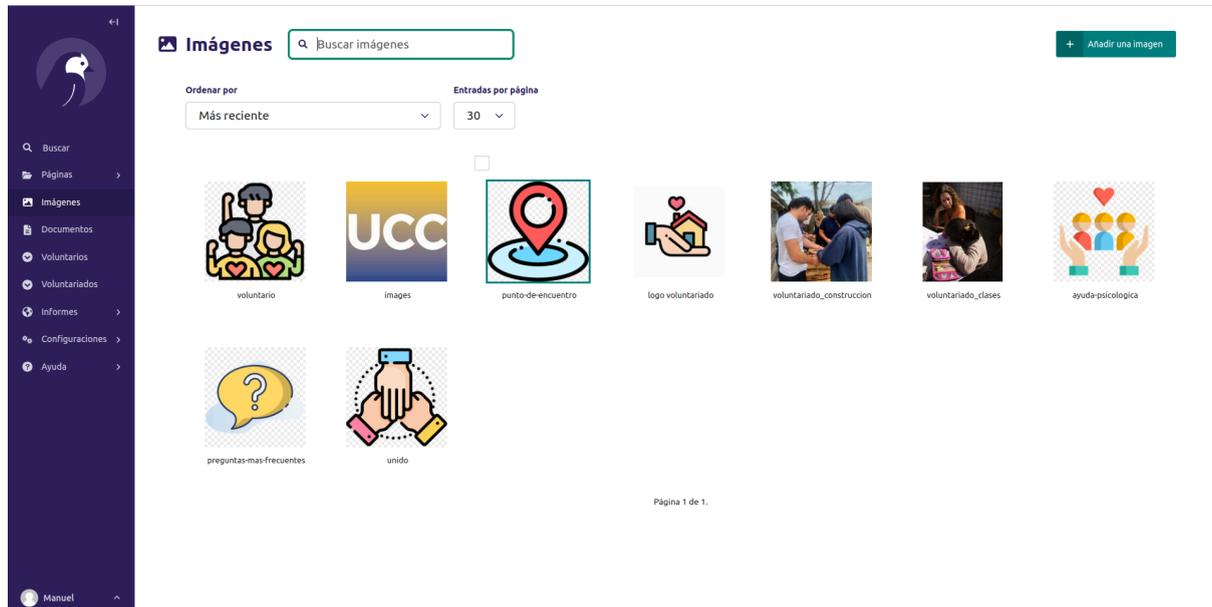


Se observa entonces el detalle de la información cargada en esta página, pudiendo manipularse en caso de desearlo para mantener el sitio y su información de la forma que se desee.

La información propiamente que se expone en cada página depende de la misma, en la parte inferior de esta pantalla se encuentra un botón “Guardar borrador” en donde los cambios realizados se guardan sin exponerse en la página visible para el público, esto permite luego seguir trabajando en los cambios que se realizan. Cuando se está conforme con los cambios realizados se expande las opciones del botón y se selecciona la opción “publicar” para efectivizar los cambios realizados.

## Imágenes

### Directivos VUCC



Únicamente disponible para directivos del VUCC, en este apartado se encuentran las imágenes cargadas en el sitio, estas imágenes luego pueden vincularse con elementos dentro de las páginas, a modo que la imagen queda cargada una única vez en la aplicación y puede mostrarse en múltiples lugares de la misma.

En esta sección pueden cargarse imágenes al pulsar el botón que se encuentra en la esquina superior derecha.

Si se pulsa en una imagen se accede a una información más detallada de la misma, pudiendo también eliminarla en el caso de así desearlo, aquellos contenidos de las páginas que hagan uso de la imagen eliminada ya no mostrarán la misma.

## Voluntarios

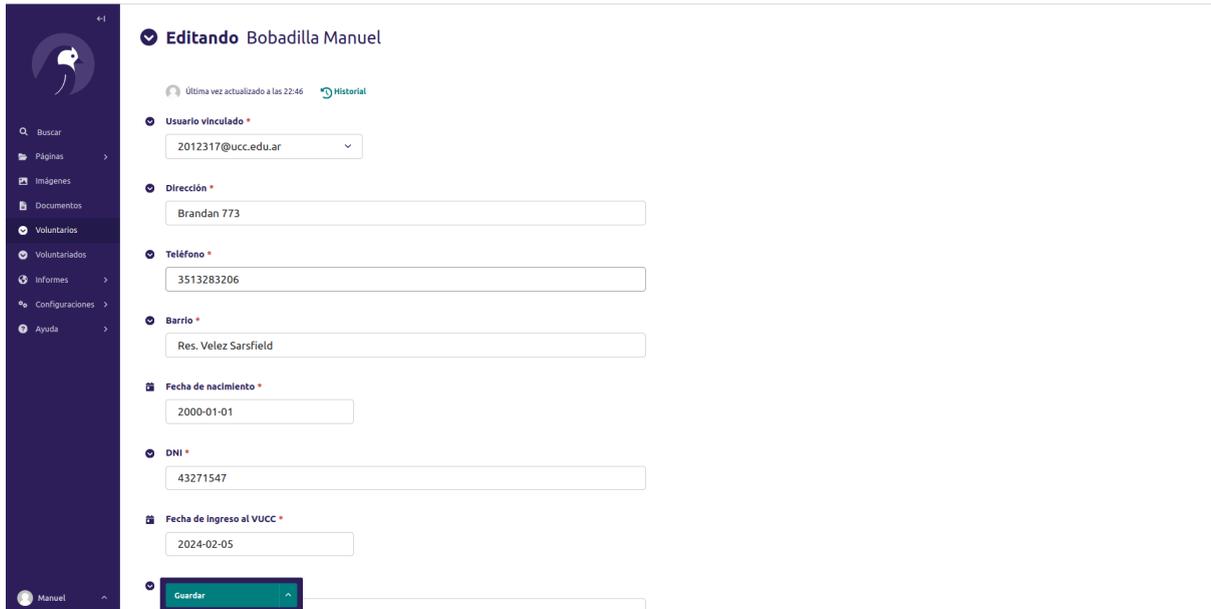
### Directivos VUCC

Voluntario	Direccion	Telefono	Universidad	Legajo	Validado	Coordinador del VUCC
Bobadilla Manuel	Ehijidá	6888677	UCC	82828394	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Bobadilla Manuel	Brandan 773	3513283206	Sin universidad	-	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Franchini Graciela	Luna y Cárdenas 2130	3515451250	UCC	0980198	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Rosas Lucas	calle1	3211231231	UCC	0000000	<input type="checkbox"/>	<input type="checkbox"/>
Cisneros Marcos	calle1	3211231231	UCC	0000000	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Monserrat Santiago	calle1	3211231231	UCC	0000000	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Bobadilla Manuel	Gustavo Brandan 773	3513283206	UCC	2012317	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Roca Lucio	Av. Velez Sarsfield 478	3515369702	UCC	2048379	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Perez Juan	Manuel Baigorria 522	3516322970	UCC	2012317	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Sección accesible únicamente para los directivos del VUCC, aquí se puede observar los voluntarios cargados en el sistema, si se presiona el nombre de alguno se entra en detalle sobre toda la información del mismo, incluyendo la posibilidad de modificar dicha información. De este modo, por ejemplo, es en el que se validan a los voluntarios, mediante el ingreso a la información del mismo y tildar la casilla “Validado”.

En esta pantalla además de poder modificar y visualizar los voluntarios, pueden crearse voluntarios de forma manual en caso de así desearlo, y además, tanto al ingresar a un voluntario en cuestión, o cuando se posiciona el cursor sobre el mismo se expone la opción para eliminar dicho voluntario, en caso de presionar este botón y confirmar la operación se elimina el voluntario.

La pantalla que se muestra a continuación corresponde a la vista que se obtiene cuando se pulsa sobre un voluntario, en esta se puede modificar los atributos del elemento detallado. Se visualizan los campos acompañados por el título del mismo, y un texto de ayuda en caso de haberse cargado. Aquellos campos que poseen un asterisco (\*) color rojo significa que son campos obligatorios y no pueden dejarse vacíos.



**Editar** Bobadilla Manuel

Última vez actualizado a las 22:46 [Historial](#)

**Usuario vinculado \***  
2012317@ucc.edu.ar

**Dirección \***  
Brandan 773

**Teléfono \***  
3513283206

**Barrio \***  
Res. Velez Sarsfield

**Fecha de nacimiento \***  
2000-01-01

**DNI \***  
43271547

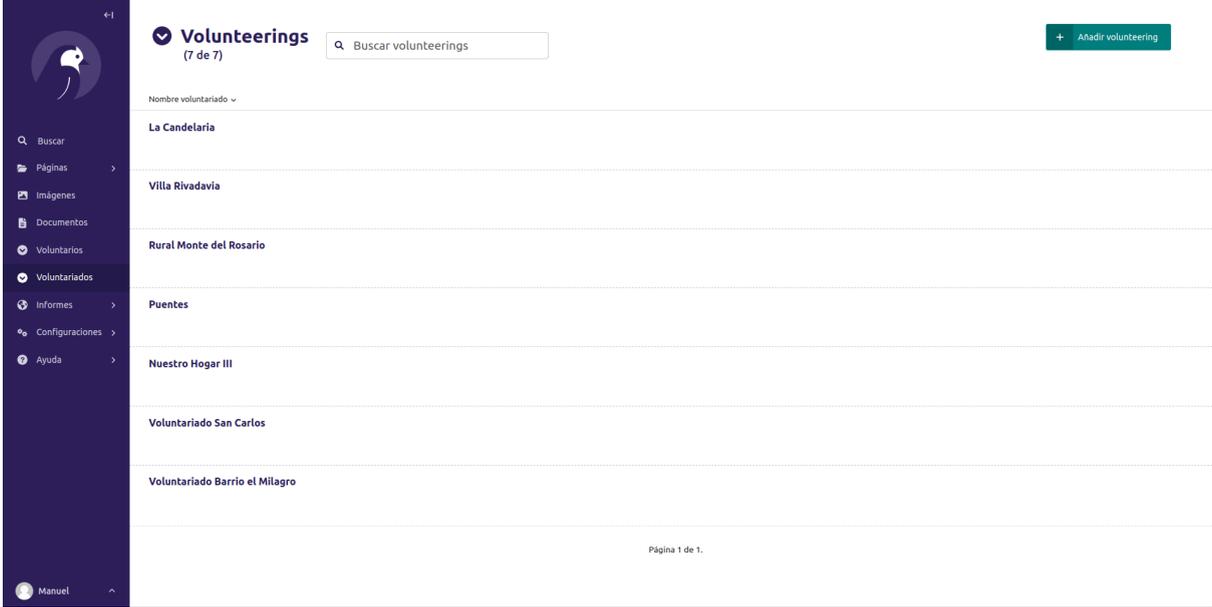
**Fecha de ingreso al VUCC \***  
2024-02-05

**Guardar**

En la parte inferior de la pantalla se observa el botón “Guardar”, al pulsarlo se persisten los cambios realizados

## Voluntariados

### Directivos VUCC



The screenshot displays the 'Volunteerings' section of the VUCC system. On the left is a dark sidebar with navigation options: 'Buscar', 'Páginas', 'Imágenes', 'Documentos', 'Voluntarios', 'Voluntariados', 'Informes', 'Configuraciones', and 'Ayuda'. The main content area is titled 'Volunteerings (7 de 7)' and includes a search bar with the placeholder 'Buscar volunteerings' and a green '+ Añadir volunteerings' button. Below the header is a table listing seven volunteer activities:

Nombre voluntariado
La Candelaria
Villa Rivadavia
Rural Monte del Rosario
Puentes
Nuestro Hogar III
Voluntariado San Carlos
Voluntariado Barrio el Milagro

At the bottom of the page, it indicates 'Página 1 de 1'.

Pantalla únicamente accesible para los directivos del VUCC, en esta se pueden observar los voluntariados cargados en el sistema.

En la esquina superior derecha se observa un botón que al pulsarlo permite cargar un nuevo voluntariado en el sistema.

En caso de querer modificar un voluntariado ya cargado, se pulsa el nombre del mismo, abriendo el mismo formulario que en el caso de desear cargar un nuevo voluntariado, con la diferencia de que los campos del formulario ya se encuentran completados con la información del voluntariado en cuestión.

A continuación se muestra la vista que se obtiene al querer modificar, o inspeccionar, los valores referidos a un voluntariado



**Editando** Voluntariado Barrio el Milagro

Última vez actualizado 7 horas antes [Historial](#)

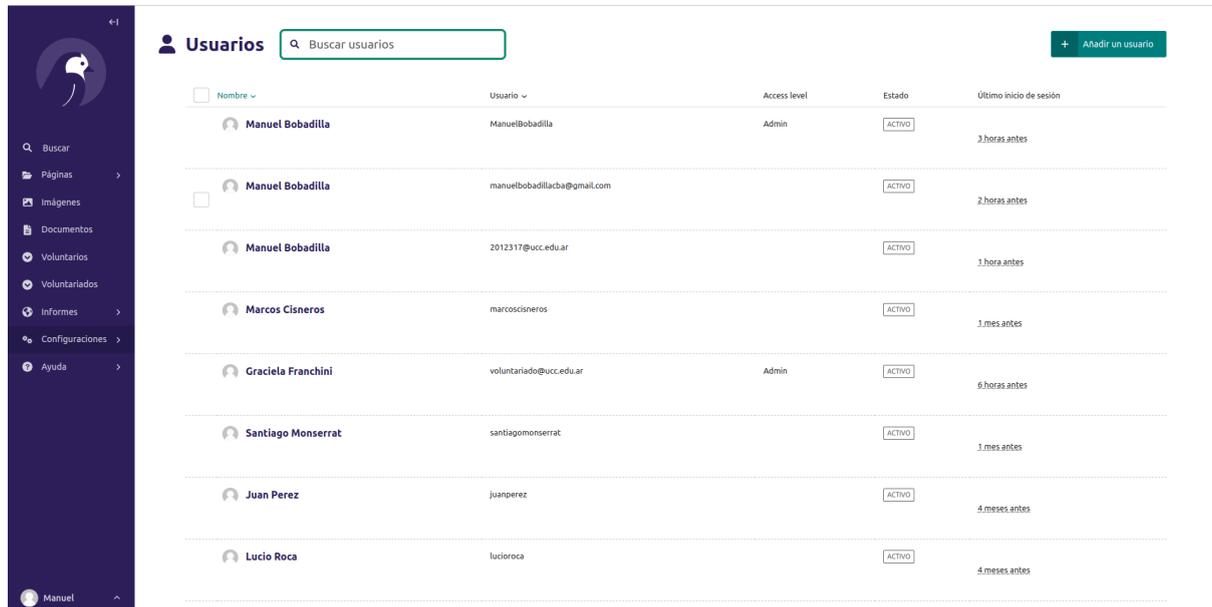
- Nombre voluntariado \***  
Voluntariado Barrio el Milagro
- Descripción del voluntariado \***  
Somos un grupo de estudiantes universitarios/as de la Universidad, que desde el año 2017 tra
- Breve descripción, para vista en celulares**  
Trabajamos en el barrio El Milagro en el mejoramiento y construcción de módulos habitacione
- Imagen voluntariado \***  
Logo del voluntariado  
Actualmente: logo\_voluntariado\_UCFmGSx.jpg  
Cambiar:  
Choose File | No file chosen  
No file chosen

Manuel **Guardar**

Pueden observarse los campos cargados en el modelo del voluntariado.

## Usuarios

### Directivos VUCC

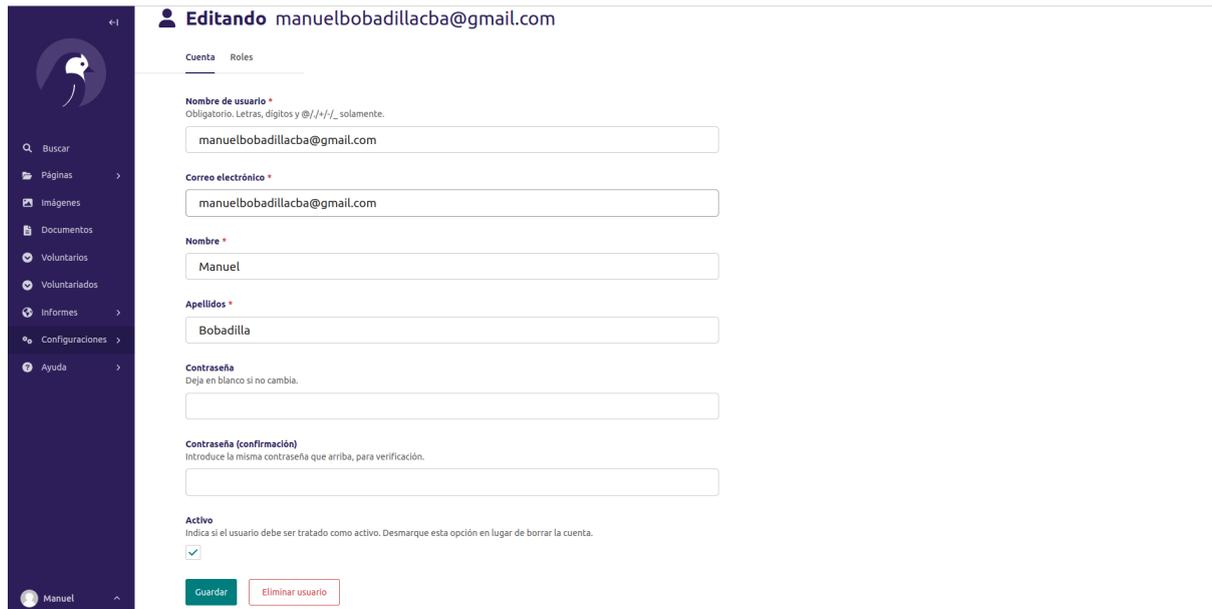


Nombre	Usuario	Access level	Estado	Último inicio de sesión
Manuel Bobadilla	ManuelBobadilla	Admin	ACTIVO	3 horas antes
Manuel Bobadilla	manuelbobadilla@bca@gmail.com		ACTIVO	2 horas antes
Manuel Bobadilla	2012317@ucc.edu.ar		ACTIVO	1 hora antes
Marcos Cisneros	marcoscisneros		ACTIVO	1 mes antes
Graciela Franchini	voluntariado@ucc.edu.ar	Admin	ACTIVO	6 horas antes
Santiago Monserrat	santiagomonserrat		ACTIVO	1 mes antes
Juan Perez	juanperez		ACTIVO	4 meses antes
Lucio Roca	lucioroca		ACTIVO	4 meses antes

Únicamente visible para directivos del VUCC, en esta pantalla se observan los usuarios cargados en el sistema, como ya se mencionó anteriormente (en la [descripción del modelo Volunteer](#)) el modelo *Volunteer* (donde se almacena la información de los voluntarios) se relaciona mediante una relación uno a uno con el modelo *User* (en donde se almacena la información básica referida a un usuario dentro de un sistema informático, alguno de los atributos que aquí se almacenan son el email, contraseña, nombre, apellido, entre otros). Dentro de esta pantalla se pueden observar las instancias generadas en el sistema referidas al modelo *User*, pudiendo modificarse, eliminarse y crear nuevas desde esta vista si así se desea.

Al pulsar sobre una de las instancias de *User* que se exponen se ingresa a una vista donde se pueden modificar los atributos.

A continuación se presenta un ejemplo:



**Editando** manuelbobadillacba@gmail.com

Cuenta Roles

**Nombre de usuario \***  
Obligatorio. Letras, dígitos y @/./+/\_ solamente.

**Correo electrónico \***

**Nombre \***

**Apellidos \***

**Contraseña**  
Deja en blanco si no cambia.

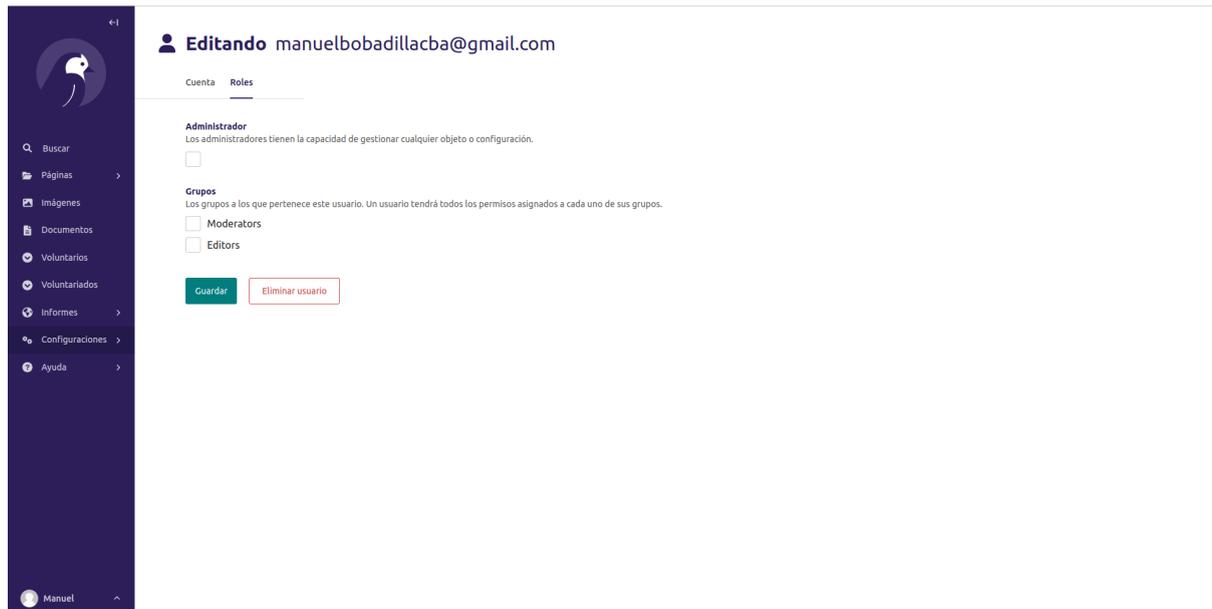
**Contraseña (confirmación)**  
Introduce la misma contraseña que arriba, para verificación.

**Activo**  
Indica si el usuario debe ser tratado como activo. Desmarque esta opción en lugar de borrar la cuenta.

Aquí pueden observarse los campos del modelo *User* con los valores correspondientes a la instancia que se está modificando, los atributos expuesto son “Nombre de usuario”, “Correo electrónico”, “Nombre”, “Apellido”, “Contraseña” y “Contraseña (confirmación)”. El campo de contraseña se encuentra vacío más allá de que el usuario en cuestión si posee una contraseña registrada en el sistema, esto es así a modo de preservar la privacidad del usuario y no hacer accesible la contraseña que haya cargado. En caso de desear cambiar la contraseña de un usuario ya cargado a través de esta pantalla debe completarse en los campos “Contraseña” y “Contraseña (confirmación)” con la nueva clave para el usuario.

Además de esto, en la parte superior del formulario se encuentran dos pestañas, “Cuenta” y “Roles”, si se presiona la segunda (“Roles”), se ingresa a un formulario donde se pueden asignar permisos al usuario que se está modificando, siguiendo el modelo de negocios, si se quisiera hacer que el usuario tenga permisos de directivo del VUCC se debería tildar el checkbox donde se indica si el usuario es administrado.

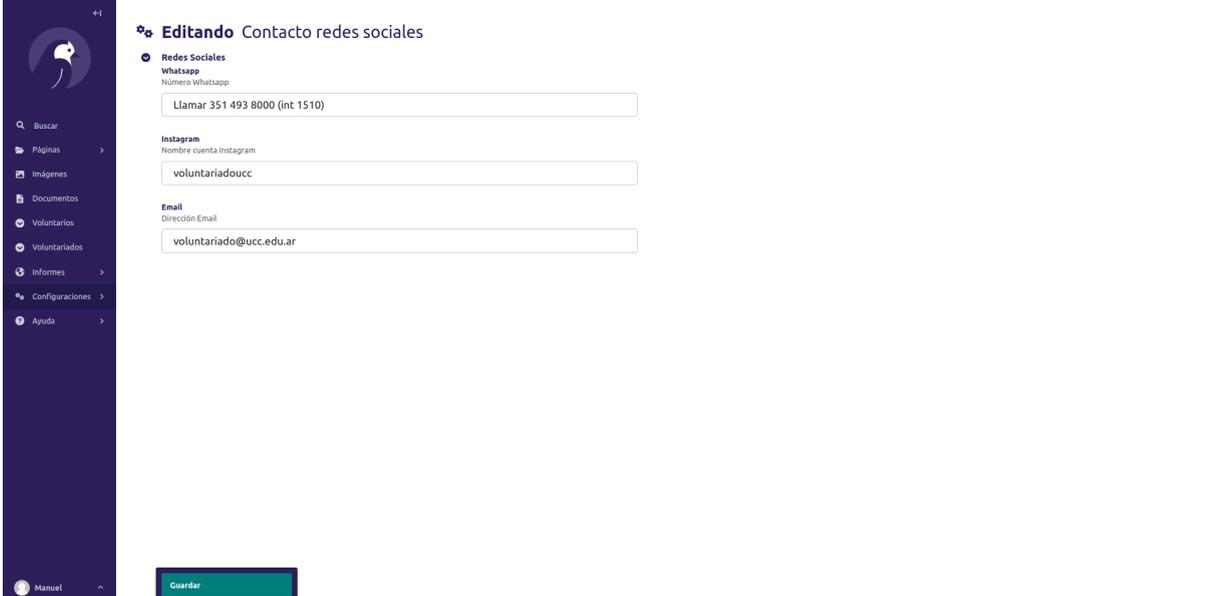
A continuación se muestra dicho formulario.



En caso de que el usuario sea directivo del VUCC se tilda el checkbox “Administrador” y se pulsa el botón “Guardar” para persistir los cambios.

## Contactos redes sociales

### Directivos VUCC



**Editando** Contacto redes sociales

**Redes Sociales**

**Whatsapp**  
Numero Whatsapp  
Llamar 351 493 8000 (int 1510)

**Instagram**  
Nombre cuenta Instagram  
voluntariadoucc

**Email**  
Dirección Email  
voluntariado@ucc.edu.ar

Guardar

Pantalla accesible únicamente por directivos del VUCC, en esta se setean los valores referidos a los medios de contacto expuestos en el footer del sitio web.

Es un formulario simple de tres campos, “Whatsapp”, “Instagram” e “Email”, en donde se debe cargar (en caso de así desearlo) la información referida a estos medios de contacto. En caso de que uno de los campos no se complete con ninguna información será excluido del footer, en modo tal de mantener una presentación adecuada.

En la parte inferior del formulario se presenta un botón “Guardar”, al pulsarlo se persisten los cambios realizados.

## Conclusión

El desarrollo de este proyecto representó un desafío desde el principio, presentando constantemente situaciones nuevas y exigentes. Sin embargo, este desafío no solo fue estimulante, sino también enormemente enriquecedor y gratificante.

Resultó fundamental mantener una investigación constante y estar en un proceso continuo de aprendizaje para llevar a cabo el proyecto. Las tecnologías empleadas eran nuevas desde un punto de vista de los conocimientos personales, y tampoco se tenía un precedente personal para un sistema de esta magnitud, en el cual se abarcaron todas las áreas del mismo, desde la codificación de la aplicación en sí, hasta el diseño de la base de datos y despliegue de la solución final en una plataforma. Esto llevó a explorar en profundidad áreas que anteriormente habían sido examinadas superficialmente.

Compartir con un cliente real como es el VUCC también fue una nueva experiencia, en donde la validación y el diálogo son elementos clave para lograr el correcto desarrollo de un producto que verdaderamente haga frente a los problemas y necesidades planteadas.

Además, el carácter social del cliente, siendo el área de voluntariados de la universidad, en donde se promueven y desarrollan actividades tan importantes y con un impacto tan real y positivo en las comunidades, barrios y personas desfavorecidas, otorga una mayor satisfacción e incentivo durante el desarrollo del proyecto.

El proyecto se completó con éxito, alcanzando los objetivos establecidos y obteniendo la aprobación y satisfacción tanto del VUCC como de sus voluntariados. Se logró desarrollar una solución que cumpla con sus expectativas y que aborda de manera efectiva sus procesos y necesidades de información.

También, el conjunto de tecnologías que se eligió estuvo a la altura para hacer frente a las necesidades técnicas del proyecto, generando una solución adecuada y eficaz desde un punto de vista de capacidad de procesamiento.

## Bibliografía

- Jeremy Wagner. (s. f.). INP. Recuperado de <https://web.dev/articles/inp?hl=es-419>
- Jeremy Wagner y Barry Pollard. (s. f.). TTFB. Recuperado de <https://web.dev/articles/ttfb?hl=es-419>
- Philip Walton. (s. f.). FCP. Recuperado de <https://web.dev/articles/fcp?hl=es-419>
- Philip Walton. (s. f.). TBT. Recuperado de <https://web.dev/articles/tbt?hl=es-419>
- Addy Osmani y Jason Miller. (s. f.). Renderización. Recuperado de <https://web.dev/articles/rendering-on-the-web?hl=es-419>
- Electron. (s. f.). Process Model. Recuperado de <https://www.electronjs.org/docs/latest/tutorial/process-model>
- Electron. (s. f.). Application Packaging. Recuperado de <https://www.electronjs.org/docs/latest/tutorial/application-distribution>
- Electron. (s. f.). Updating Applications. Recuperado de <https://www.electronjs.org/docs/latest/tutorial/updates>
- Electron. (s. f.). Application Debugging. Recuperado de <https://www.electronjs.org/docs/latest/tutorial/application-debugging>
- Wikipedia. (s. f.). Flutter. Recuperado de [https://es.wikipedia.org/wiki/Flutter\\_\(software\)](https://es.wikipedia.org/wiki/Flutter_(software))
- Flutter. (s. f.). Plataformas. Recuperado de <https://flutter.dev/multi-platform>
- Flutter. (s. f.). Widgets. Recuperado de <https://docs.flutter.dev/ui>
- Dart. (s. f.). Compilación. Recuperado de <https://dart.dev/tools/dart-compile>
- Flutter. (s. f.). Hot reload. Recuperado de <https://docs.flutter.dev/tools/hot-reload>
- Angular. (s. f.). Bases Angular. Recuperado de <https://angular.io/guide/what-is-angular>
- TypeScript. (s. f.). TypeScript. Recuperado de <https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>
- Infragistics. (s. f.). Two way binding. Recuperado de <https://www.infragistics.com/products/ignite-ui-angular/angular/components/genernal/wpf-to-angular-guide/two-way-binding>
- Angular. (s. f.). Inyección de dependencias. Recuperado de <https://angular.io/guide/dependency-injection>

- Angular. (s. f.). Directivas. Recuperado de <https://angular.io/guide/built-in-directives>
- Angular. (s. f.). Pipes. Recuperado de <https://angular.io/guide/pipes-overview>
- React. (s. f.). React. Recuperado de <https://es.react.dev/>
- reactjs. (s. f.). React DOM. Recuperado de <https://www.reactjs.wiki/cuales-son-las-caracteristicas-principales-de-react>
- reactjs. (s. f.). Unidireccionalidad React. Recuperado de <https://www.reactjs.wiki/cuales-son-las-caracteristicas-principales-de-react>
- reactjs. (s. f.). React Hooks. Recuperado de <https://legacy.reactjs.org/docs/hooks-intro.html>
- Expressjs. (s. f.). Enrutamiento. Recuperado de <https://expressjs.com/es/guide/routing.html>
- Expressjs. (s. f.). Middleware. Recuperado de <https://expressjs.com/en/guide/using-middleware.html>
- Expressjs. (s. f.). Plantillas. Recuperado de <https://expressjs.com/en/guide/using-template-engines.html> y <https://expressjs.com/en/resources/template-engines.html>
- Flask. (s. f.). Rutas. Recuperado de <https://flask.palletsprojects.com/en/3.0.x/patterns/viewdecorators/>
- Flask. (s. f.). Plantillas. Recuperado de <https://flask.palletsprojects.com/en/2.3.x/templating/>
- Spring. (s. f.). Inyección de dependencias. Recuperado de <https://docs.spring.io/spring-framework/reference/core/beans/dependencies/factory-collaborators.html>
- Spring. (s. f.). Seguridad. Recuperado de <https://spring.io/projects/spring-security/>
- Spring. (s. f.). Hot reloading. Recuperado de <https://docs.spring.io/spring-boot/docs/1.3.8.RELEASE/reference/html/howto-hotswapping.html>
- Django. (s. f.). Seguridad. Recuperado de <https://docs.djangoproject.com/en/5.0/topics/security/>
- Django. (s. f.). Enrutamiento. Recuperado de <https://docs.djangoproject.com/en/5.0/topics/http/urls/>
- MySQL. (s. f.). Plataformas. Recuperado de <https://www.mysql.com/support/supportedplatforms/database.html>
- MySQL. (s. f.). Tipo de datos. Recuperado de <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

- PostgreSQL. (s. f.). Licencia. Recuperado de <https://www.postgresql.org/about/licence/>
- PostgreSQL. (s. f.). Tipo de datos. Recuperado de <https://www.postgresql.org/docs/current/datatype.html>
- MariaDB. (s. f.). Compatibilidad. Recuperado de <https://mariadb.com/kb/en/mariadb-vs-mysql-compatibility/>
- MariaDB. (s. f.). Mejoras vs MySQL. Recuperado de <https://mariadb.com/kb/en/mariadb-vs-mysql-features/>
- Microsoft. (s. f.). T-SQL. Recuperado de <https://learn.microsoft.com/es-es/training/modules/get-started-transact-sql-programming/>
- Microsoft. (s. f.). SSIS. Recuperado de <https://learn.microsoft.com/es-es/sql/integration-services/sql-server-integration-services?view=sql-server-ver16>
- Microsoft. (s. f.). SSAS. Recuperado de <https://learn.microsoft.com/es-es/analysis-services/ssas-overview?view=asall-products-allversions>
- MongoDB. (s. f.). BSON. Recuperado de <https://www.mongodb.com/json-and-bson>
- MongoDB. (s. f.). Consistencia. Recuperado de <https://www.mongodb.com/community/forums/t/what-is-causal-consistency-in-mongodb/172041>
- MongoDB. (s. f.). ACID. Recuperado de <https://www.mongodb.com/basics/acid-transactions>
- WordPress. (s. f.). Plugins. Recuperado de <https://wordpress.com/go/website-building/what-are-wordpress-plugins-and-themes-a-beginners-guide/>
- WordPress. (s. f.). SEO. Recuperado de <https://wordpress.com/support/seo/>
- WordPress. (s. f.). Incompatibilidad plugins. Recuperado de <https://wordpress.org/plugins/incompatibility-status/>
- WordPress. (s. f.). Casos de uso. Recuperado de <https://wordpress.com/go/website-building/what-is-wordpress-com-used-for/>
- Wagtail. (s. f.). Workflows. Recuperado de [https://docs.wagtail.org/en/v4.0.4/editor\\_manual/administrator\\_tasks/managing\\_workflows.html](https://docs.wagtail.org/en/v4.0.4/editor_manual/administrator_tasks/managing_workflows.html)
- Wagtail. (s. f.). SEO. Recuperado de <https://docs.wagtail.org/en/latest/reference/contrib/searchpromotions.html>

- Wagtail. (s. f.). Base de datos. Recuperado de [https://docs.wagtail.org/en/v5.2.2/advanced\\_topics/performance.html](https://docs.wagtail.org/en/v5.2.2/advanced_topics/performance.html)
- OpenCMS. (s. f.). Introducción. Recuperado de <http://www.opencms.org/en/>
- OpenCMS. (s. f.). WYSIWYG. Recuperado de <http://www.opencms.org/en/demo/features/version5/wysiwygeditor.html>
- OpenCMS. (s. f.). Workflow. Recuperado de <http://www.opencms.org/en/demo/tour/index.html?file=37527559-2f8c-11dc-b6d6-b7a33137a728&index=10>
- Strapi. (s. f.). Headless. Recuperado de <https://strapi.io/what-is-headless-cms>
- Strapi. (s. f.). Roles y permisos. Recuperado de <https://docs.strapi.io/user-docs/users-roles-permissions>
- Strapi. (s. f.). Bases de datos. Recuperado de <https://docs.strapi.io/dev-docs/configurations/database#alternative-database-driver-packages>
- Strapi. (s. f.). Multimedia. Recuperado de <https://strapi.io/features/media-library>
- Strapi. (s. f.). Koa. Recuperado de [https://docs.strapi.io/dev-docs/backend-customization/requests-responses#:~:text=The%20Strapi%20back%20end%20server,policies%2C%20controllers%2C%20services\).](https://docs.strapi.io/dev-docs/backend-customization/requests-responses#:~:text=The%20Strapi%20back%20end%20server,policies%2C%20controllers%2C%20services).)
- Render. (s. f.). Despliegue. Recuperado de <https://docs.render.com/deploy-a-commit>
- Render. (s. f.). Render vs Heroku. Recuperado de <https://docs.render.com/render-vs-heroku-comparison>
- Render. (s. f.). Características gratuitas. Recuperado de <https://docs.render.com/free>
- Render. (s. f.). Lenguajes. Recuperado de <https://docs.render.com/language-support#:~:text=Render%20natively%20suports%20Node..specify%20a%20branch%20to%20deploy.>
- Heroku. (s. f.). Dynos. Recuperado de <https://www.heroku.com/dynos>
- Heroku. (s. f.). Lenguajes. Recuperado de <https://www.heroku.com/languages>
- Heroku. (s. f.). Git. Recuperado de <https://devcenter.heroku.com/articles/git>
- Railway. (s. f.). Despliegue. Recuperado de <https://docs.railway.app/overview/the-basics>
- Railway. (s. f.). Costos. Recuperado de <https://docs.railway.app/reference/pricing>