

INSTITUTO TECNOLÓGICO DE BUENOS AIRES – ITBA

ESCUELA DE INGENIERÍA Y GESTIÓN

# PONTE

AUTOR/ES: Vera, Juan Sebastian (Leg. N° 53852)

López, Noelia Belén (Leg. N° 53774)

DOCENTE/S TITULAR/ES O TUTOR/ES: Guerrero, Marcela Alejandra

TRABAJO FINAL PRESENTADO PARA LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN INFORMÁTICA

Lugar: Ciudad Autónoma de Buenos Aires, Buenos Aires, Argentina

Fecha: 11 de Junio de 2019

# PONTE

Proyecto Final

2018 - 2019

**VERA** Juan Sebastián

**LOPEZ** Noelia Belén



# Índice

<b>Índice</b>	<b>1</b>
<b>Introducción</b>	<b>3</b>
<b>Etapas de proyecto</b>	<b>5</b>
Nacimiento	5
Relevamiento	5
Definición de objetivos	5
Requisitos	6
Requisitos Funcionales	6
Web	6
Autenticación	6
Donación	7
Backoffice	8
General	8
Estadísticas	9
Validación de instituciones	9
Requisitos No Funcionales	9
Atributos de Calidad	10
Confiabilidad	10
Disponibilidad	10
Seguridad	10
Mantenibilidad	10
Escalabilidad	10
Cronograma y versionado	10
Desarrollo y QA	11
<b>Arquitectura</b>	<b>12</b>
Front End	12
Vistas	13
Landing	14
Autenticación	16
Perfil	16
Instituciones	18
Pedidos	19
Modal de donación	20
Análisis, problemáticas y mejoras	21
Bases de datos & Backend	22
Estructura	25

Autenticación	26
Creación de cuenta	26
Inicio de sesión	26
Mailer	27
Análisis, problemáticas y mejoras	30
Backoffice	32
<b>Notificaciones y Manejo de Errores</b>	<b>34</b>
Errores de Datos	34
Email en uso	34
Usuario Inexistente	35
Contraseña Incorrecta	36
Errores Externos	37
Error de conectividad	38
Errores de servidor	39
Notificaciones de éxito	41
<b>Casos de Uso</b>	<b>43</b>
Institución realiza un pedido particular	44
Donante notifica primero que el bien fue retirado	47
Voluntario notifica primero que el bien fue retirado	49
Entrega en destino	50
Donante realiza un ofrecimiento de un bien no pedido	52
Diagrama de Flujo	53
<b>Casos de Prueba</b>	<b>56</b>
<b>Conclusión</b>	<b>67</b>
<b>Trabajo Futuro</b>	<b>69</b>
Match automático	69
Aplicación mobile	70
Tracking donaciones	70
Geolocalización	70
Programa de fidelización	71
Optimización de la Base de Datos y Backoffice	71

# Introducción

Ponte, nombre que más bien describe un ideal antes que un proyecto, busca ser una asociación civil que responda de la manera más comprensiva posible a una sociedad con muchas necesidades, con agentes invisibles, y con capacidad de ayuda y trabajo en equipo que tiene que ser descubierta.

Ponte, juego de palabras desde el latín puente, quiere ser nexo entre una porción de la sociedad necesitada e invisible, y una porción de la sociedad dispuesta a ayudar pero ignorante ante las problemáticas o el alcance de su ayuda. Ambas porciones de la sociedad se encuentran embebidas en un mundo tecnológico que es el medio ideal para crear puentes. Para ello, Ponte, busca descubrir (desde el sentido más íntimo de la palabra, de hallar lo que estaba escondido o ignorado) aquellos sectores de la sociedad que se encuentran desconocidos y poder tender un puente con aquellos que pueden ayudar.

Nos encontramos en una situación socio-cultural en la que mucha gente invisible tiene necesidades de lo más básicas y muchos están dispuestos a donar o colaborar, pero no saben dónde, son desconfiados, o no están dispuestos a moverse mucho. Ponte apunta a resolver esa problemática.

El objetivo es que el producto final sea una plataforma en la que, por un lado, una persona pueda registrarse como donante de tiempo siendo voluntario en calidad de transportista (en adelante **voluntario**), o como donante de bienes (en adelante **donante**); mientras que, por el otro, una asociación civil u ONG pueda registrarse como tal e informar rubro, necesidades concretas e inmediatas (en adelante **institución**); y mediante un mediador autorizado (en adelante **administradores**) se realice el nexo de manera semi-automática. Es así que, la plataforma, de manera autosustentable ayudará a que un bien o producto llegue a las manos de quien lo necesita por medio de otros voluntarios informando en todo momento el status del mismo.

Por lo tanto el proyecto se encontraría terminado cuando las siguientes tres plataformas estén desarrolladas: sitio web de acceso público, api y backend, backoffice de admin para control, métricas y estadísticas.

Lo interesante de Ponte es que no solamente tiene como finalidad ser un proyecto final de la carrera de Ingeniería en Informática en ITBA, sino que quiere ser funcional y usable para la comunidad en general. Es por ello que, al mismo tiempo en que se desarrolla la aplicación, se crean las bases de un equipo de trabajo interdisciplinario de graduados y alumnos avanzados de distintas carreras y áreas.

El siguiente informe tiene la intención de registrar el proceso de trabajo por el que se atravesó, en particular desde la aplicación informática, siendo reflejo fiel de los aciertos y desaciertos que fueron encontrados en el camino; a fin de poder representar el alcance del proyecto siguiendo los objetivos y requisitos funcionales, y luego registrar los posibles pasos a seguir para que el proyecto pueda seguir creciendo.

# Etapas de proyecto

## Nacimiento

El nacimiento de Ponte, en Febrero de 2018, como ideal y asociación civil, de manera muy ambiciosa, fue originado por el descubrimiento de una realidad muy cruda: existen asociaciones, instituciones, grupos sociales y pequeñas ONG que, sin la capacidad y recursos de muchas ONGs grandes y conocidas, quedan en las sombras de una sociedad que no es capaz de registrar que sus necesidades serían saciadas si tan solo fueran conocidas.

## Relevamiento

Con la certeza de que existen razones concretas por las cuales los bienes necesitados por estas **instituciones** no llegan a las mismas, se comenzó una pequeña etapa de relevamiento. Se descubrió que muchos individuos dicen tener la intención de donar, e incluso encuentran satisfacción y planificación al hacerlo, pero son frenados por tres principales motivos:

1. Desconfianza
2. Desconocimiento
3. No practicidad del sistema de donaciones actual

Al mismo tiempo, quedó en evidencia que cuanto más pequeña la institución, más básicas eran sus necesidades y más sencillo sería saciarlas si fueran conocidas, generando impacto real, concreto y diferencial.

## Definición de objetivos

Con lo antes mencionado en mente, se optó por definir objetivos que apunten a resolver explícitamente las problemáticas más radicales, no solo a nivel institucional sino también informático.

Es de esta manera que se decidió crear una plataforma que permita:

- Traer a la luz instituciones y sus necesidades.
- Presentar la posibilidad de que las donaciones fueran retiradas y entregadas con el menor esfuerzo posible de parte del donante.
- Informar en todo momento el estado de las donaciones.
- Ser intuitiva y simple

A su vez se definieron los requisitos funcionales y no funcionales de la aplicación. Los mismos se encuentran detallados a continuación.

## Requisitos

A partir de la definición de los objetivos presentados se optó por establecer los requisitos funcionales y no funcionales de la plataforma, esperando que para la última versión planteada se puedan cumplir los siguientes

### Requisitos Funcionales

Se definieron los requisitos funcionales en base a los dos grandes componentes en los que la plataforma permite la interacción de usuarios: la *Web* y el *Backoffice*.

#### Web

##### Autenticación

- El sistema debe permitir la generación de cuentas
  - Debe poderse registrar como usuario **voluntario** o **donante** (*default*).
  - Debe poderse registrar como interesado para ser para **institución**.
  - **Todos los usuarios** deben registrar
    - nombre
    - email
    - número de celular
    - dirección (de domicilio)
  - Un usuario **voluntario** debe registrar
    - franja horaria de disponibilidad

- Un usuario que desee mostrar interés para ser **institución** debe registrar:
  - una descripción propia
  - el nombre del referente
  - email del referente
- Un usuario **donante** debe poder solicitar validación para ser **voluntario**.
  - Hasta que un **admin** acepte o rechace esa solicitud, el usuario debe estar en estado **quiero ser voluntario**.
- Ante la generación de un usuario nuevo, el sistema debe permitir la verificación de datos.
  - El sistema debe enviar un email con un *call to action* que verifique la cuenta.
- El sistema debe permitir el ingreso al mismo mediante una **cuenta registrada**
  - Debe existir la posibilidad de ingresar siendo **admin**

#### Donación

- Una donación puede estar
  - **creada**
  - **en búsqueda de institución**
  - **siendo buscada por el voluntario**
  - **marcada como entregada por el donante**
  - **marcada como retirada por el voluntario**
  - **entregada**
  - **cancelada por el usuario**
- El sistema debe poder listar todas las donaciones pedidas
  - El sistema debe poder listar todas las instituciones que solicitan donaciones.
- El sistema debe permitirle a un usuario **donante** seleccionar un **pedido**. Genera un *match* (se crea una **donación**)
  - El **donante** debe poder seleccionar franja horaria de *pick-up*.
  - El sistema debe notificar a los usuarios **admin** para que seleccionen el **voluntario** adecuado.
- El sistema debe permitirle a un usuario registrado ver sus donaciones.

- El sistema debe permitirle a un donante realizar una **oferta** de algo que no esté siendo pedido por alguna institución
  - Una **oferta** debe ser tratada como una **donación en búsqueda de institución**
  - Si no se encuentra institución a la brevedad, debe tener como destino una institución ficticia **Ponte** que representa un depósito.
- El sistema debe permitirle a un **voluntario** notificar la entrega de una donación a una institución o depósito.
- El sistema debe permitirle a un usuario **donante, voluntario, y/o admin** cancelar una donación **creada**
  - Si la donación es cancelada por el **donante**, la donación pasa a ser **cancelada por el donante**.
  - Si la donación es cancelada por el **voluntario**, la donación vuelve a comenzar su curso en estado **creada**.
- El sistema debe informar al **donante** de cada cambio de estado donación.
- Un usuario **voluntario** debe poder marcar una donación como retirada.
  - Una donación **creada** pasa a estar en estado **marcada como retirada por el voluntario**
  - Una donación **marcada como retirada por el donante** pasa a estar en estado **en camino a destino** (final, la institución)
- Un usuario **donante** debe poder marcar una donación como retirada.
  - Una donación **creada** pasa a estar en estado **marcada como retirada por el donante**
  - Una donación **marcada como retirada por el voluntario** pasa a estar en estado **en camino a destino** (final, la institución)
- Un usuario **voluntario** debe poder marcar una donación como entregada, sólo cuando ambas partes (donante y voluntario) la hayan marcada como retirada.

## Backoffice

### General

- El sistema debe permitirle a un usuario administrador ingresar a su cuenta
- El sistema debe poder listar
  - Donaciones pedidas/ofrecidas/creadas
  - Usuarios registrados

- Instituciones
- El sistema debe permitir activar/desactivar
  - Una institución
  - Un usuario
- El sistema debe permitir aceptar/rechazar un usuario que quiere ser voluntario.
- El sistema debe permitirle a un usuario admin validar una institución
- El sistema debe permitirle a un usuario admin agregar instituciones nuevas

#### Estadísticas

- El sistema debe poder registrar métricas
- El sistema debe permitirle a un usuario admin visualizar métricas

#### Validación de instituciones

- Un usuario admin valida una institución cuando la registra
  - La validación se da a partir de los criterios que el equipo de **territorial**<sup>1</sup> propone
  - El registro y/o validación de instituciones debe poder hacerse sólo por usuarios **admin**.
- Una institución podría registrarse por si sola → “demostrar interés por ser parte de la plataforma”
  - La institución se registra como **no validada**
    - Una institución no validada es una institución no es visible a la plataforma.
    - Una institución no validada no tiene visible sus pedidos en la plataforma
  - El sistema debe informarle a los admin que una institución desea registrarse.

## Requisitos No Funcionales

A partir de los requisitos funcionales definidos y de los objetivos establecidos se decidieron los siguientes atributos de calidad.

---

<sup>1</sup> El equipo territorial es el equipo formado, en su gran mayoría, por miembros de Ponte que no son alumnos de Ingeniería en Informática. Son los encargados de entrar a los barrios, registrar las instituciones y sus necesidades.

## Atributos de Calidad

### Confiabilidad

- Tolerante a fallos
- Operaciones transaccionales
  - Poca demora en la respuesta
  - El usuario es quien mide la transparencia
- Poco tiempo de resolución de problemas
- Explicación de errores y manera de resolución

### Disponibilidad

- Tiempo de actividad/inactividad del sistema

### Seguridad

- Log in
- Dónde está mi donación?
- Llegó a destino → Se informa → Ayuda con la confiabilidad

### Mantenibilidad

- Generación de documentación que permita continuar con el proyecto

### Escalabilidad

- Utilización de sprints
  - Crecer de un sprint a otro
- Versionado
  - Agregar features de una versión a otra sin pérdida de información
- Posibilidad de agregar funcionalidades a lo largo del tiempo
  - Incluso después del release de la tercera versión
- Posibilidad de migración a mobile

## Cronograma y versionado

Inicialmente se pensó el desarrollo de Ponte en 3 grandes versiones iterativas en la que cada una tuviera funcionalidades más completas y automatizadas que la versión anterior. Sin embargo, en la práctica y debido a una migración en la arquitectura de la

aplicación (que se encuentra mejor detallada en la sección de este informe correspondiente), la presentación de la plataforma se desarrolló en dos versiones.

Para poder tener una mejor visión de las tareas que se llevaban a cabo, y la relación entre tareas, se decidió utilizar el método *SCRUM* de planificación, por su separación iterativa creciente de *sprints*. Se probaron, como herramientas, *Trello* y *Project*, resultando la primera más intuitiva y fácil de manejar.

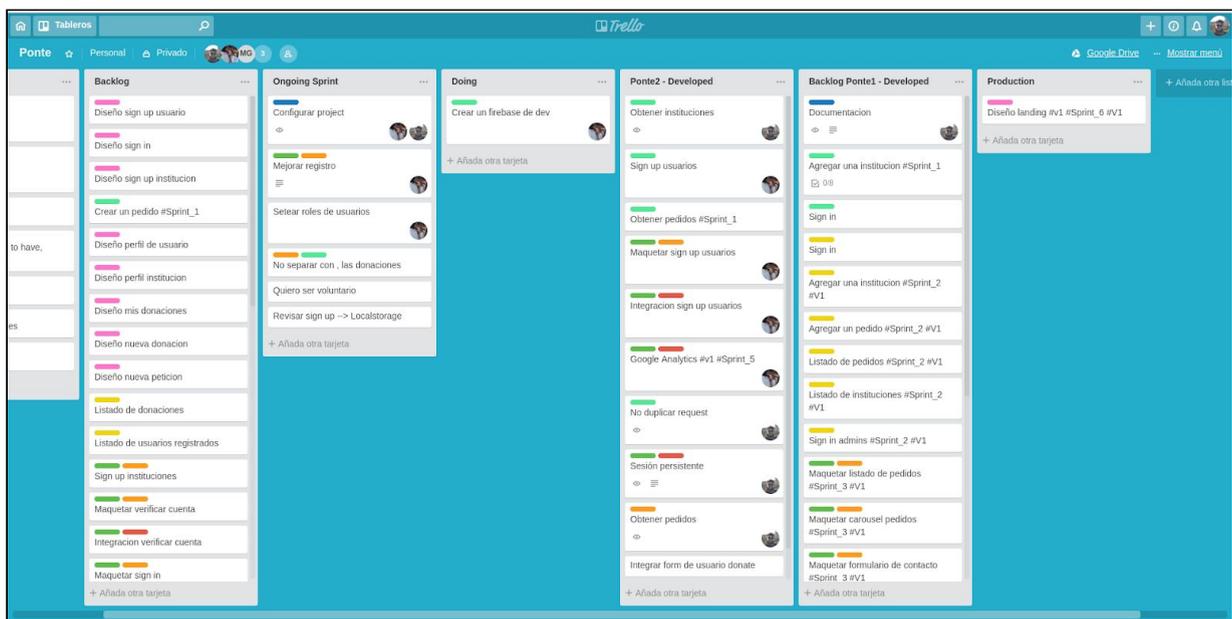


Figura 1. Captura de pantalla del Dashboard de Trello utilizado durante la planificación y llevado a cabo de los sprints.

## Desarrollo y QA

Una vez definidos los objetivos fundamentales que fueran el norte del proceso de desarrollo, se comenzó con el mismo. En la presentación de las dos versiones se obtuvo *feedback* como respuesta a un proceso de *quality assurance* por parte de la tutora y del resto del equipo. Asimismo se obtuvo *feedback* entremedio de las presentaciones de las distintas versiones.

# Arquitectura

A partir de lo planificado inicialmente se optó por encarar el desarrollo de ambas versiones de la plataforma en tres etapas paralelas, el desarrollo del **front end**, el armado de la **base de datos junto con el backend** y la puesta del **backoffice**.

Es en base a estas tres sub-plataformas que se define la arquitectura de la aplicación.

## Front End

El equipo técnico tiene mucha experiencia desarrollando en Front End, por lo que conoce mucho de las tecnologías destacadas actuales: *Angular y React*. Se reconoce que el crecimiento de *React* por sobre *Angular* es mucho mayor en los últimos tiempos y que las soluciones que hoy brinda son fáciles de implementar y eficientes. Es por ello que la pregunta inicial sobre el desarrollo del *Front End* no fue sobre la tecnología a utilizar, ya que desde el principio se optó por *React*, sino que giró en torno al orden en el cual desarrollar las vistas y que se quería priorizar en las mismas.

Para el desarrollo de la primera versión de Ponte se consideró importante mantener los siguientes atributos:

- Que la plataforma explicara, en su vista inicial, de manera sencilla qué es Ponte.
- Que la plataforma permitiera a un usuario completar un formulario.
- Permitirle al equipo tener una manera de comunicarse con el usuario.
- Que la plataforma listara las necesidades de las instituciones registradas.

Se optó por darle suma importancia a la simplicidad de las vistas y la completitud de la *landing*. Es importante remarcar que se contó con la ayuda de un diseñador para el desarrollo de la vista principal.

## Vistas

Siguiendo con lo estipulado inicialmente, el desarrollo continuó siendo en *React* obteniendo las siguientes vistas.

# Landing

**PONTE** PEDIDOS INSTITUCIONES DONAR ALGO NUEVO MI PERFIL SALIR

## Colaborar hoy es así de simple

Alicia  
2 bolsos con ropa en estado para donar =>

Hogar "Margarita"  
Donaciones recibidas

Marcos  
Retira donaciones en lo de Alicia

Gracias =)

### ¿Que es ponte?

Una plataforma para que puedas donar todos aquellos objetos que no usas y no te encantan en destino, siendo aprovechado de una mejor manera.

El objetivo es ser el puente entre los usuarios que quieren donar de la sociedad y aquellos que los quieren recibir.

Colaborar hoy es así de simple

- 1. Crea una cuenta para comenzar.
- 2. Contanos que tenés para donar.
- 3. Un recolector pasará a buscarlo.

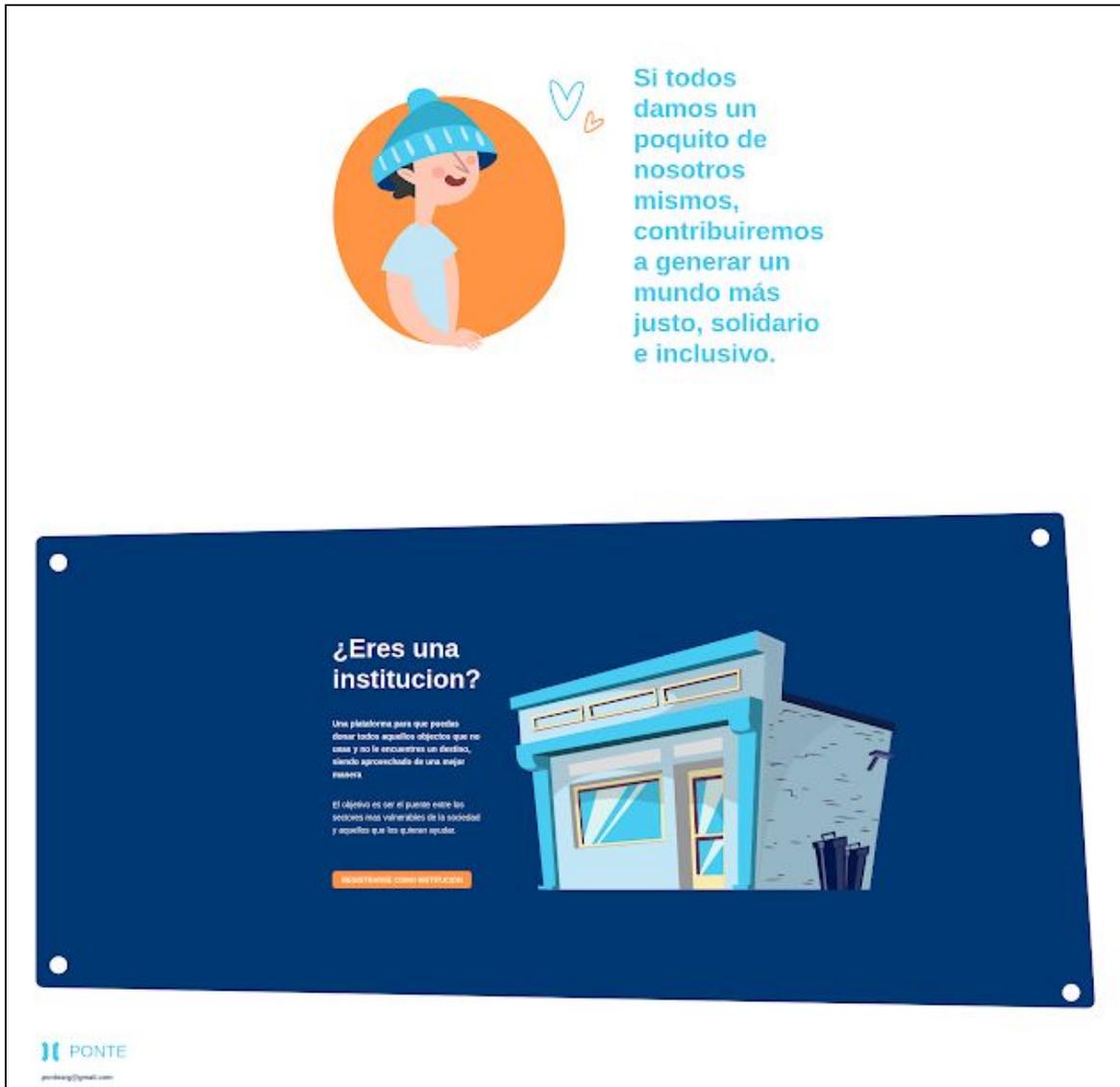
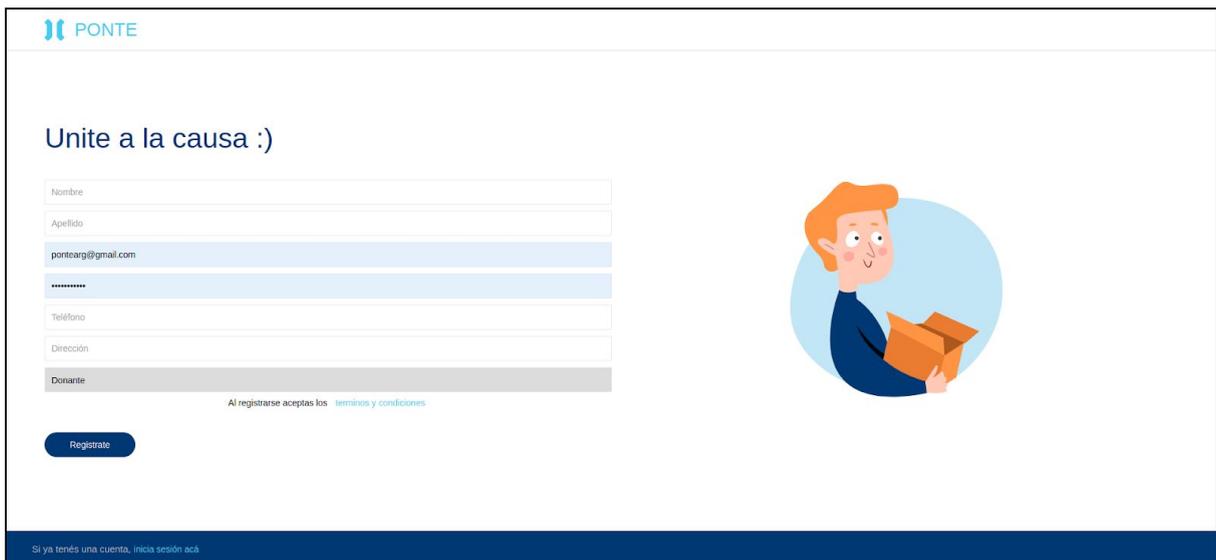


Figura 2. Captura de pantalla de la landing de Ponte. Es la vista inicial de lo que cualquier usuario ve al ingresar a la plataforma. Busca ser amena, explicativa, simple y fácil de usar.

## Autenticación



*Figura 3. Captura de pantalla al Inicio de sesión de Ponte. Se accede a esta vista mediante el botón de inicio de sesión que se encuentra en la barra de navegación. Es obligatorio haber creado una cuenta antes de poder iniciar sesión con ella.*



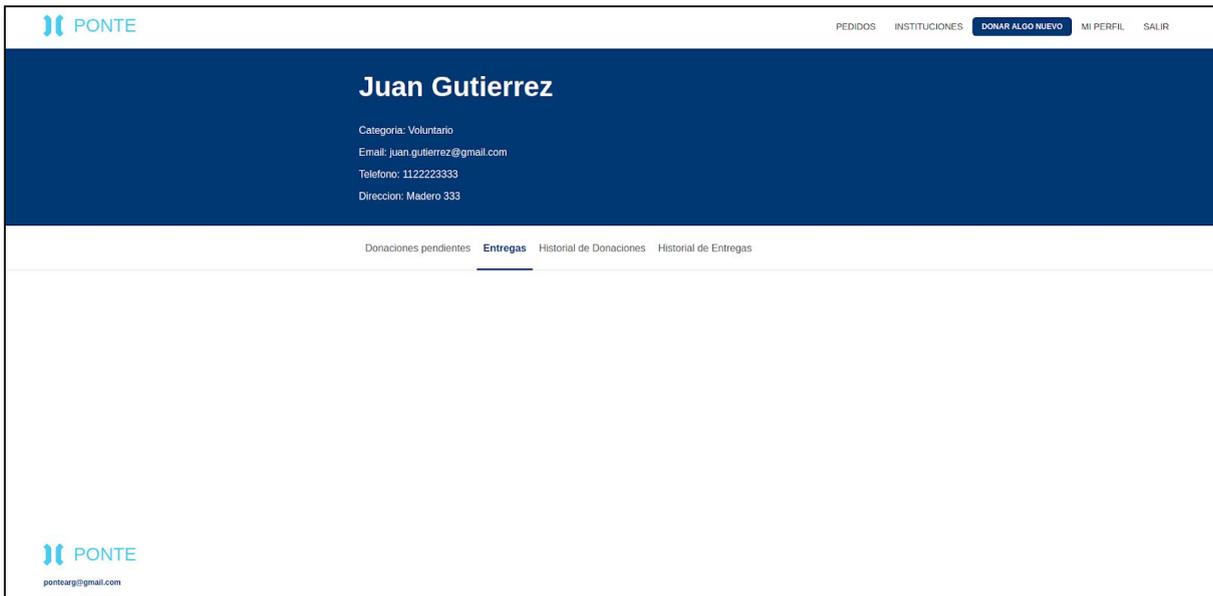
*Figura 4. Captura de pantalla al Registro de Cuenta de Ponte. Mediante esta vista, un usuario cualquiera puede crear una cuenta para ser donante. Al mismo tiempo existe la posibilidad de ya, en este estadio, crear una cuenta de voluntario. Todos los campos son obligatorios y no puede haber más de una cuenta con la misma dirección de email.*

## Perfil

The screenshot shows the profile page for Maria Aguirre on the PONTE website. The page is divided into several sections:

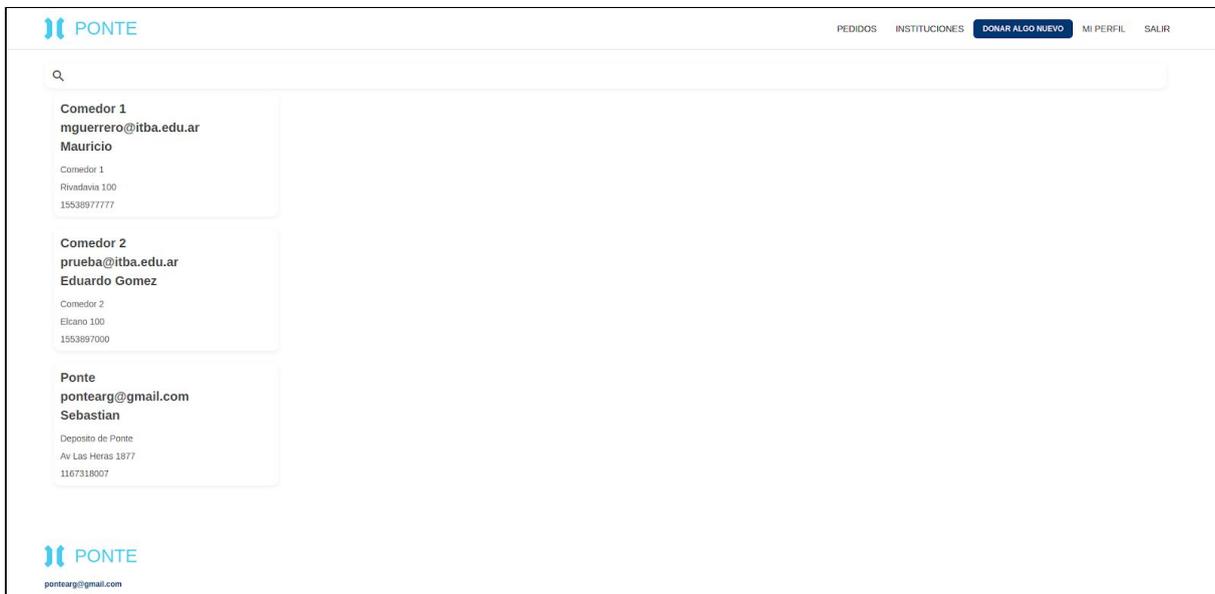
- Header:** The PONTE logo is on the left. On the right, there are navigation links: PEDIDOS, INSTITUCIONES, DONAR ALGO NUEVO (highlighted), MI PERFIL, and SALIR.
- Profile Section:** A dark blue background contains the name "Maria Aguirre" in white. Below the name, the following information is listed:
  - Categoría: Donante
  - Email: maria.aguirre@gmail.com
  - Telefono: 1123456789
  - Dirección: Av. Madero 399An orange button labeled "QUIERO SER VOLUNTARIO" is positioned below the contact information.
- Donations Section:** Below the profile section, there are two tabs: "Donaciones pendientes" (which is selected and underlined) and "Historial de Donaciones".
- Content:** Under the "Donaciones pendientes" tab, the text reads: "Todavía no registras donaciones. Realíz una!".
- Footer:** The PONTE logo and the email address "pontearg@gmail.com" are located at the bottom left of the page.

*Figura 5. Captura de pantalla al Perfil de un donante de Ponte. Se puede observar que un usuario donante tiene la posibilidad de postularse para ser voluntario desde su perfil. Al mismo tiempo, cada perfil de donante cuenta con dos secciones, la de donaciones pendientes y la de historial de donaciones. Cuando una donación termina su ciclo, ya sea por ser entregada o cancelada por el usuario, es visualizada en la sección de historial.*



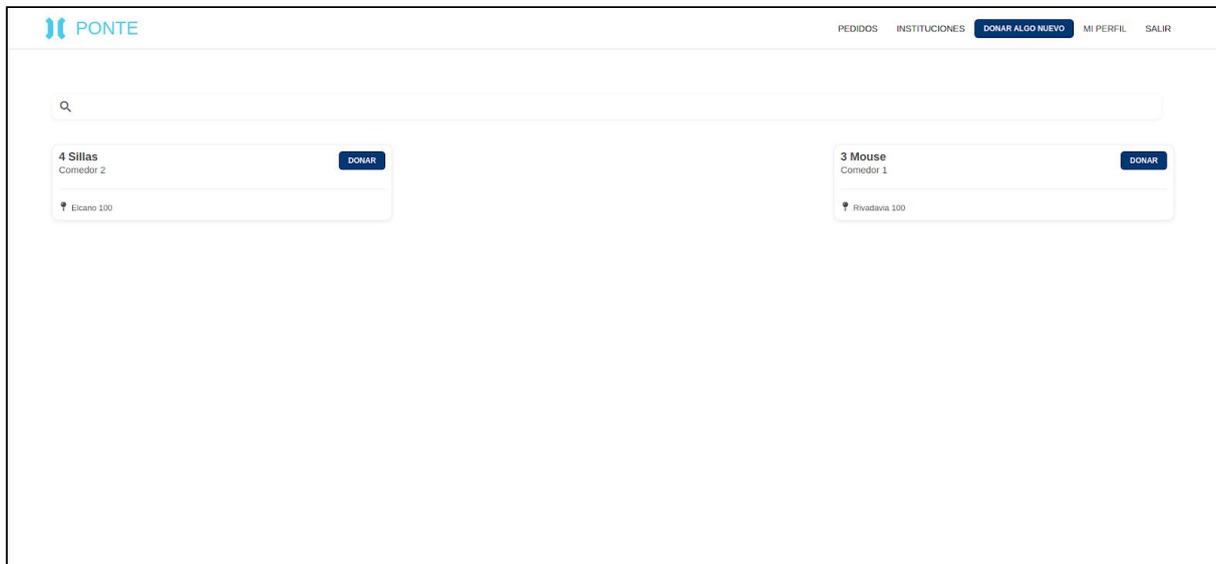
*Figura 6. Captura de pantalla al Perfil de un voluntario de Ponte. Se puede observar como el mismo tiene más secciones incluidas en su perfil, entre las que se encuentran las entregas pendientes y el historial de entregas realizadas. Un usuario puede ser donante previa autorización de los administradores.*

## Instituciones



*Figura 7. Captura de pantalla a la vista de Instituciones de Ponte. Se puede observar la información de las instituciones registradas en Ponte, y como contactarse con ellas si fuera el deseo de los donantes. Una institución figura en esta vista previa autorización de los administradores.*

## Pedidos



*Figura 8. Captura de pantalla a la vista de Pedidos de Ponte. En esta vista se visualizan todos los pedidos realizados por cualquiera de las instituciones que pertenecen a Ponte. A medida que las cantidades pedidas van siendo satisfechas por los donantes, los pedidos se van actualizando hasta que el contador llega a cero, y el pedido deja de ser visible.*

## Modal de donación

The screenshot shows a web application interface for Ponte. At the top, there is a navigation bar with the logo 'PONTE' on the left and links for 'PEDIDOS', 'INSTITUCIONES', 'DONAR ALGO NUEVO', 'MI PERFIL', and 'SALIR' on the right. Below the navigation bar is a search bar. The main content area features two donation cards. The left card is for '3 mouse' from 'noelia institucion' at 'Pedro Goyena 1666, Castelar, Buenos Aires, Argentina'. The right card is for '34 mesa' from the same institution. A modal window is open in the center, titled 'Estas queriendo donar mouse para noelia institucion'. The form inside the modal contains the following fields: '¿Cuál es tu nombre?' (Juan), '¿Cuál es tu email?' (juan.sanchez@gmail.com), '¿Cuál es tu número de celular?' (1144444444), '¿Cuántas unidades podés donar?' (3), and '¿En qué horario podrías buscar las donaciones?' (Day: Lunes, Hour: De 8 a 12). There is also a section for '¿Algún otro comentario?' with a note to 'Agregar cualquier comentario que desees.' and an 'ENVIAR' button.

*Figura 9. Captura de pantalla al modal de donación de Ponte. Este modal ya cuenta con la información del usuario cargada, para poder facilitarle el trabajo y hacer el proceso más rápido. Para incentivar a los donantes a donar, el número de bien ofrecido es por defecto el total de lo pedido por la institución o el restante.*

## Análisis, problemáticas y mejoras

De acuerdo a lo mencionado en los requisitos y en base a cumplir los objetivos, es necesario que la presentación de la plataforma sea lo más clara posible, que sea intuitiva y que, en todo momento se mantenga la transparencia. Sin embargo, existen situaciones en las que la priorización de alguna de dichas características es contraria a otra de ellas.

Se desea que todas las instituciones tengan la posibilidad de ser descritas y de presentar la mayor información posible para darse a conocer y ser puestas en la mira de la comunidad, pero al mismo tiempo no se desea abrumar a los usuarios con información e invitarlos a enfocarse solamente en lo funcionalmente importante.

Se optó por mantener la simplicidad a la hora de presentarle al usuario la información pertinente, siendo sugerencia para el desarrollo posterior, la utilización de modales que le permitan al usuario obtener mayor información sin el sentimiento de que se está cambiando reiteradas veces de vista.

Por otro lado, la utilización del recurso de dibujos es sumamente eficiente a la hora de explicarle al usuario el objetivo y formas de Ponte, sin embargo se dificulta a la hora de hacer un sitio *responsive* al 100%. Dada la posibilidad, hoy certera, de que Ponte fuera utilizado como base para futuros proyectos finales dentro de los cuales se desarrollen aplicaciones nativas para *mobile*, se decidió priorizar el desarrollo de una plataforma web entendible y *amistosa*.

## Bases de datos & Backend

Es el área de trabajo en la que se cuenta con menor experiencia y conocimiento. Eso llevó al equipo a encontrarse con muchas problemáticas, desde la correcta elección de tecnologías a utilizar hasta la implementación de funcionalidades eficientes y escalables.

En la instancia de maquetado, se optó por elegir una tecnología a partir de la cual se pudiera conseguir mentoreo y acompañamiento. Se eligió empezar el desarrollo de la db a partir de un backend desarrollado en Rails.

La idea original era utilizar la herramienta **Forest** como *bootstrap* de la db, ya que es fácilmente integrable con varias tecnologías, entre ellas Rails. Se maquetaron los objetos y relaciones que se podían llegar a necesitar y nació la primera versión de **Ponte**.

Se pudo considerar que Ponte, como plataforma, no necesita mucho trabajo de base de datos en cuanto a complejidad de la información o de las relaciones. Desarrollar las *queries* en *Rails*, para una base de datos que estaba siendo complejizada sin un objetivo concreto, resultó ser un freno cuando la totalidad del equipo está orientada al front end. Es por ello que se decidió, antes de seguir avanzando, mudar de tecnología.

Con las necesidades del *backend* y la *db* en mente (que nacen a partir de la planificación inicial y se encuentran especificadas en la sección de requisitos funcionales de este informe), se optó por trabajar con las herramientas *Firebase de Google*. Las mismas permiten armar una base de datos a partir de la utilización de documentos que no necesitan ser creados previamente con determinadas especificaciones.

Al mismo tiempo, *Firebase*, tiene un manejo de los eventos de Creación, Lectura, Modificación y Borrado (en adelante **CRUD**) de sus *db* que permite notificar, en particular por email, cada vez que se agrega información a un repositorio; permitiéndole a un usuario llenar un formulario y enviar la información por mail a un admin.

De esta manera nace la segunda versión de **Ponte**, con *Firestore* como herramienta de *backend* y de manejo del *db*, y se presenta como primera versión funcional del proyecto.

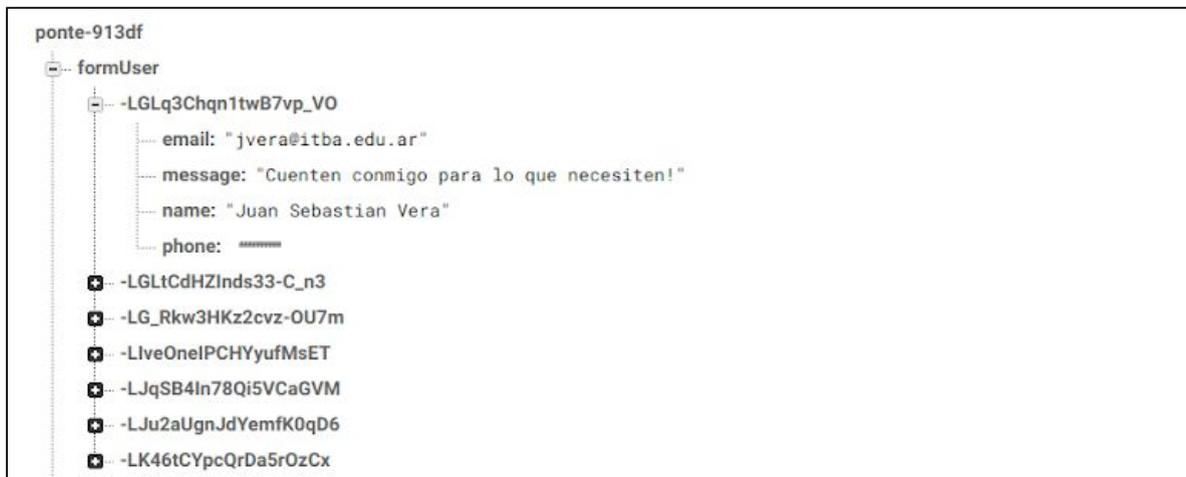


Figura 10. Captura de la tabla de documentos 'formUser' de la primera versión de Ponte.

Sin embargo, utilizar bases de datos orientadas a documentos tiene como mayor contrapartida la relación entre tablas que se traduce en la complejidad para realizar *joins* y en la repetición de información. A medida que el proyecto avanzaba, las *queries* se complejizaban y la lecto-escritura de información por parte del *Front End* aumentaba, quedó en evidencia que la respuesta correcta en cuanto a la tecnología de bases de datos era una relacional.

Se decidió volver a mudar optando esta vez por maquetar las tablas en *Mongoose* y armar el *backend* en *Node.js*. Si bien significó para el equipo tiempo de capacitación en el armado de *API Rest* y maquetado, se contaba con la ventaja de estar desarrollando en *javascript*. De igual manera se optó por seguir utilizando las herramientas de *Firestore* para la creación y autenticación de usuario, y para el envío de emails.

Al mismo tiempo, se descubrió que la primera herramienta en querer ser utilizada, *Forest*, también es compatible con *Node*, por lo que el manejo de información y métricas de diversos entornos de desarrollo es posible de manera muy sencilla y personalizable.

	ADDRESS	CATEGORY	CREATED AT	EMAIL	NAME	PHONE	SURNAME	UPDATED AT
<input type="checkbox"/>	Av. Madero 399	QUIERO SER VOLUNTARIO	20/05/2019 20:54:28	juan.sanchez@gmail.com	Juan	1144444444	Sanchez	20/05/2019 21:18:45
<input type="checkbox"/>	Av. Madero 399	DONANTE	20/05/2019 20:50:35	maria.aguirre@gmail.com	Maria	1123456789	Aguirre	20/05/2019 21:19:33

Figura 11. Captura de la tabla Usuarios de la última versión de Ponte.

Si bien se especifica de mejor manera en la sección correspondiente en este informe, es posible observar en la *Figura 11* que *Forest* es funcional como herramienta de *backoffice* para los administradores.

## Estructura

Como fue mencionado con anterioridad, se utilizan dos sistemas distintos y paralelos para el manejo de los datos en el *Back End*, y su relación con el *Front End*. La intención de esta sección del informe es poder dejar más claro cómo funcionan las interacciones entre las distintas partes.

Se decidió continuar con el uso de *Firebase* para la autenticación y creación de usuarios, ya que cuenta con un sistema ya preparado para lo mismo, y permite la posibilidad de escalar de manera muy sencilla a distintas formas de autenticación (cuenta de Google, celular, entre otros). Al mismo tiempo, *Firebase* cuenta con un conjunto de funciones que permite “escuchar” los cambios en las tablas y reaccionar ante esos cambios. Se utilizan estas funciones para el envío de notificaciones. Además la manera en la que los datos sensibles son encriptados hacen de la herramienta el fuerte de la seguridad de la aplicación, incluso ante posibles fallos en la seguridad o *leaks* de información.

En cualquier caso, como fue mencionado con anterioridad, se optó por tener bases de datos relacionales para almacenar la información que la plataforma consume. Por lo tanto, *Firebase* y la base de datos (almacenada en *Heroku*) no tienen comunicación directa, sino que el *Front End* quien las consume y las ejecuta.

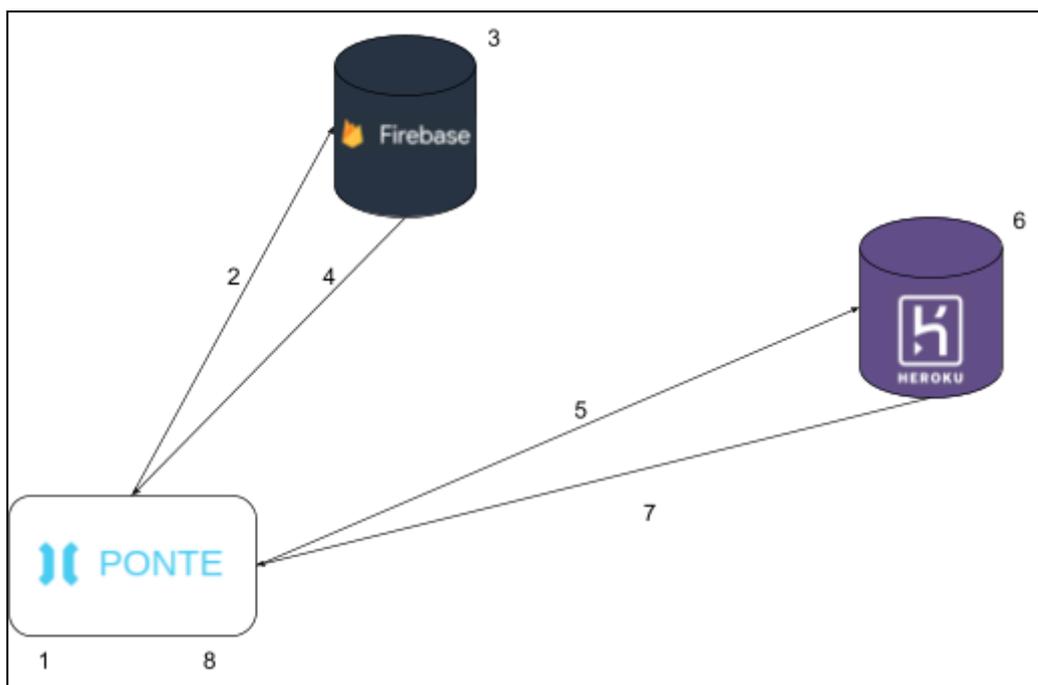


Figura 12. Gráfico de relación entre el *Front End* y las estructuras de *Back End* utilizadas.

## Autenticación

Se utiliza *Firebase* como herramienta para el registro de usuarios con dirección de correo electrónico único; para mantener la coherencia, las tablas de usuarios en la base de datos tiene como *unique key* el mismo campo.

### Creación de cuenta

De esta manera, como se puede observar en la *Figura 12* como referencia, a la hora de crear una cuenta, se realizan las siguientes llamadas.

1. Se registra, completando el formulario, en el sitio de Ponte.
2. Se envía la información a Firebase.
3. Firebase valida la información recibida con su sistema de autenticación. Si el usuario se puede crear y guardar, así se hace
4. Si el usuario ya existe se envía a Ponte el error correspondiente y el proceso no continúa. Si el usuario fue creado, se envía a Ponte la información del usuario creado.
5. Con la información recibida, se envía a la base de datos la información correspondiente, para crear allí el usuario que efectivamente será utilizado.
6. Se crea el usuario con la información recibida.
7. Se retorna código satisfactorio.
8. Si el usuario fue creado correctamente, se redirige a la vista de inicio de sesión.

### Inicio de sesión

De manera equivalente a la creación de cuenta, para iniciar sesión se realizan las siguientes llamadas.

1. Se inicia sesión, completando el formulario, en el sitio de Ponte.
2. Se envía la información a Firebase.
3. Firebase valida la información recibida con su sistema de autenticación. Si el email y la contraseña son correctas y correspondientes, devuelve la información del usuario. En caso contrario, devuelve un error.

4. Se envía la respuesta al *request*. En caso de haber un error, se muestra el mismo y el proceso no continúa.
5. Si se recibe un usuario, se envía la información recibida a la base de datos
6. Con el email recibido, se busca el usuario y se prepara para ser retornado.
7. Se retorna el usuario correspondiente.
8. Se guarda el usuario recibido en el *Local Storage* para ser accedido durante el proceso en la plataforma. Se redirige a la pantalla principal, con las funcionalidades de usuario activadas.

## Mailer

Las funciones de *Firebase* permite enviar *emails* de manera automática al encontrar un cambio en alguna de las tablas almacenadas en su base. Esto significa que, todos los cambios que se registren en *Heroku* que ameriten una notificación al usuario o a los administradores, deben registrarse también en *Firebase*.

Es por ello que se optó por aprovechar la condición de bases de datos orientada a documentos, que define a *Firebase*, para crear una tabla *emailInformation* en donde se pudiera guardar información para enviar emails a partir de la actualización de la misma. Se creó un *listener* que ante un cambio en la tabla *emailInformation*, lee el campo *emailType* para saber qué *template* de email utilizar para notificar al usuario involucrado.

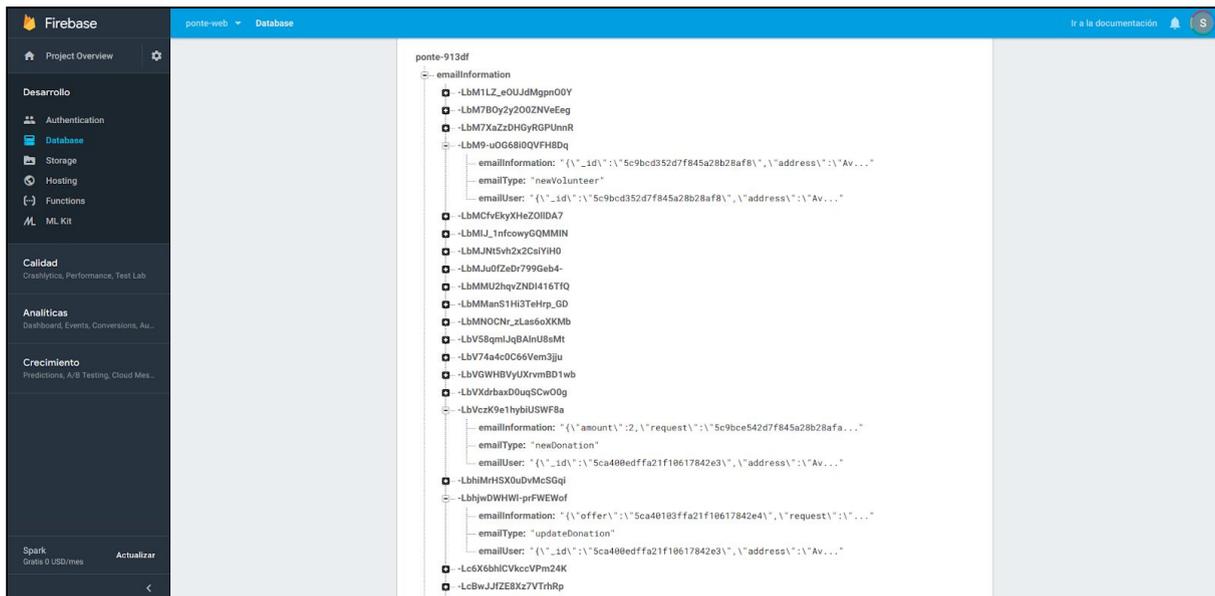


Figura 13. Gráfico la tabla *emailInformation* utilizada para el registro de los emails a ser enviados

De manera equivalente a la autenticación, el circuito para el envío de emails es de la siguiente manera, suponiendo por ejemplo que usuario desea donar algo que no está en la lista de pedidos:

1. El usuario completa el formulario con lo que tiene para donar.
2. (ítem 5) Se envía a *Heroku* la información para crear una nueva oferta con los datos de la misma y del usuario involucrado.
3. (ítem 6) Se crea una entrada nueva en la tabla de ofertas.
4. (ítem 7) En caso de éxito, se retorna la oferta creada.
5. (ítem 8) Si se recibe una oferta, se utiliza la información de la misma para enviar la información a *Firebase*. En caso contrario el proceso no continúa.
6. (ítem 2) En caso de éxito, se envía la información del usuario y del formulario a *Firebase*, junto con el valor *newOffer* en el campo *emailType*.
7. (ítem 3) *Firebase* guarda la información recibida en la tabla *emailInformation*, y esa escritura *dispara* el evento del envío del email.
8. (ítem 4) Retorna si la información pudo ser guardada correctamente en la tabla de *Firebase*.

Es importante remarcar que algunos *listeners* no solo reaccionan enviando emails al usuario sino también a los administradores. Por ejemplo, cuando un usuario desea ser voluntario o se realiza una oferta de algún bien no pedido, los administradores son notificados para poder actuar en consecuencia.

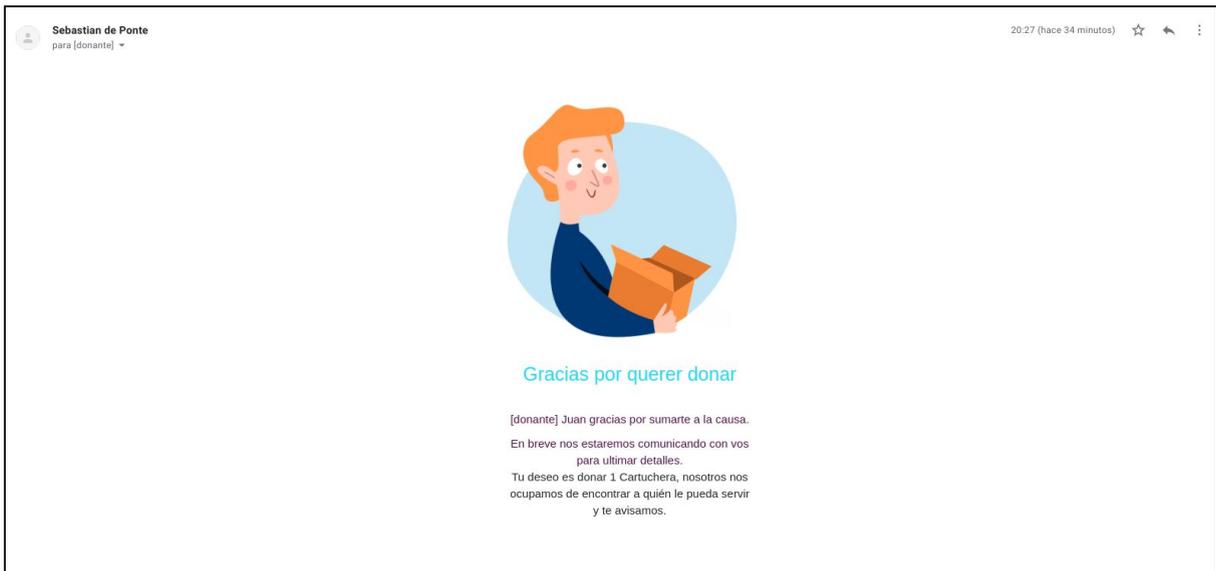


Figura 14. Gráfico de un email enviado a un usuario luego de que ofreciera un bien no pedido

## Análisis, problemáticas y mejoras

Sin lugar a dudas, la doble migración significaron los problemas más grandes para el equipo de trabajo, en particular la última. Habiendo decidido utilizar por completo una base de datos orientada a documentos y sin intención de migrar nuevamente, se optó por adaptar el desarrollo del *Front End* y de los controladores de información a *Firebase*. Eso resultó en una migración engorrosa y compleja del *Front End* y de los controladores.

Al mismo tiempo, al optar por dejar de utilizar *Firebase* como base de datos, sin ser conscientes del todo, se optó por dejar de utilizar *Firebase* como herramienta de autenticación y de comunicación. Es por ello que se decidió trabajar con algunas pequeñas modificaciones para poder seguir utilizando el sistema de autenticación de *Firebase* y el envío de emails.

De esta manera, al generarse un nuevo usuario, se generan las credenciales en *Firebase* y luego se guardan en la base de datos. A la hora de un nuevo login, sólo se accede a la base de datos si el *request* a *Firebase* es satisfactorio.

Por otro lado, para poder contar con el envío de emails, se optó por crear una tabla en *Firebase* que guardase la información modificada, el usuario involucrado y la característica de la modificación, para, con la ayuda de algunas funciones propias y templates de emails, enviar los correos correspondientes a las partes pertinentes.

Todo esto mencionado facilita las transacciones en la aplicación haciendo que el *backend* no se encargue de procesar toda la información junta en tiempo real. Es así que si un usuario realiza una modificación, la misma llama a los *triggers* mencionados y muestra una respuesta en tiempo real mientras *Firebase* y el Servidor ejecutan las funciones pertinentes, haciendo que las transacciones sean más rápidas y mejor informadas al usuario.

En un plano paralelo, se encontraron complejidades a la hora de maquetar el concepto del Voluntario. En primer lugar, desde el inicio del proyecto, es deseo que cualquier usuario de Ponte tenga la capacidad de donar, por lo que, un voluntario es, antes que nada, donante. En segundo lugar, *Firebase* permite, ante una donación, asignar de manera muy sencilla un voluntario si y sólo si existe una tabla de voluntarios independiente. En tercer lugar, un voluntario tiene los mismos campos que un usuario pero tiene también algunos específicos, por lo que la tabla de voluntarios hereda de la tabla de usuarios.

Estos tres aspectos y el restringido conocimiento del área en el equipo de trabajo, llevó al mismo a decidir crear una tabla especial de voluntarios (que permitiría la mejor utilización del *backoffice*), cometiendo la imprudencia de duplicar información y de tener que ser modificada dos veces.

Se considera seguir capacitando al equipo para la mejora en estos aspectos.

Al mismo tiempo, se concluyó que para mantener las sesiones persistentes, sería ideal realizar la autenticación por medio de *tokens* y no a través de usuarios crudos. Si bien es algo en lo que el equipo ya está trabajando, en este momento se encuentra en fase de desarrollo y es presentado dentro de la categoría de “futuras mejoras”.

## Backoffice

La herramienta del *backoffice* no solo es fundamental para el manejo de métricas y analíticas sino también como espacio funcional de los administradores del proyecto. Es por ello que, siendo por lo general individuos de conocimientos generales informáticos promedio, es necesario que el *backoffice* sea igual de intuitivo y fácil de utilizar que la plataforma en sí.

El *backend* se refleja directamente en el *backoffice*, y todos los cambios que sucedan en el primero, por cuestiones de lógica computacional, se suceden en el segundo. Es por ello que las migraciones que afectaron el desarrollo del *backend*, afectaron también de manera directa el desarrollo, la elección de herramientas, y la utilización del *backoffice*.

Inicialmente, al haber pensado el backend en Rails, se pensó en la herramienta Forest para el manejo de las db ya que presenta una herramienta autogestionable de backoffice, como fue mencionado anteriormente.

Con la misma se buscaba poder centralizar las herramientas de los administradores en un sólo lugar, que fuera de fácil acceso para la parte del equipo que no es técnica / desarrolladora, y que permitiera el registro de instituciones de manera sencilla.

Las ventajas que se creía que esto traería a nivel comodidad o facilidad de trabajo, no superaban las desventajas que se creían surgirían de estar complejizando el desarrollo del backend y se optó por dejar Forest de lado.

En su lugar, para la segunda versión de **Ponte**, se decide hacer una vista dentro del mismo sitio, que funcionara para el trabajo administrativo, como ser el registro de instituciones. Se pensó que lo mejor sería poder tener un */admin* que requiera autenticación en donde se encuentren las herramientas del backoffice.

Se había decidido no utilizar ninguna herramienta externa como *backoffice* aunque eso significara sumar trabajo al desarrollo de la plataforma en cuanto a diseño e integración con el backend, hasta que se llevó a cabo la última migración del *backend*.

# Notificaciones y Manejo de Errores

Uno de los principales objetivos a nivel plataforma es que el usuario sienta en todo momento que recibe toda la información necesaria y que entiende el proceso por el que está pasando. Poder informar tanto a los usuarios donantes como a los usuarios voluntarios de manera correcta, clara y eficiente es fundamental no solo para ganar su confianza y asegurar la continua utilización de la plataforma, sino también para lograr que más donaciones se realicen y se completen en el menor tiempo posible.

Es por ello que el equipo decidió invertir tiempo en notificar a los usuarios de todo lo que ocurre en la plataforma. Parte de ese tiempo invertido resultó en el *mailer*, mencionado con anterioridad, en la notificación de estados de una donación, mencionado en la próxima sección, y en la notificación de errores, mencionada a continuación.

Es sabido que, en algunas ocasiones, los errores en las plataformas web se generan por razones que son externas al usuario, y es por ello que se hace énfasis en el tipo de error y se acompaña con una posible solución, cuando el usuario puede hacer algo.

## Errores de Datos

Un usuario puede equivocarse a la hora de crear una cuenta o de iniciar sesión, introduciendo información que no es correcta o ya está siendo usada por otro usuario, es por ello que se optó por utilizar los siguientes mensajes de error.

### Email en uso

Cuando un usuario quiere registrarse, como donante o como voluntario, en la plataforma, debe ingresar su dirección de email. Esta será utilizada como *key* en las tablas de usuario, por lo que no pueden existir dos usuarios con el mismo email.

Por lo tanto, si un usuario quisiera registrarse con una dirección de email que ya está siendo usada, recibiría un error como el de la *Figura 15*.

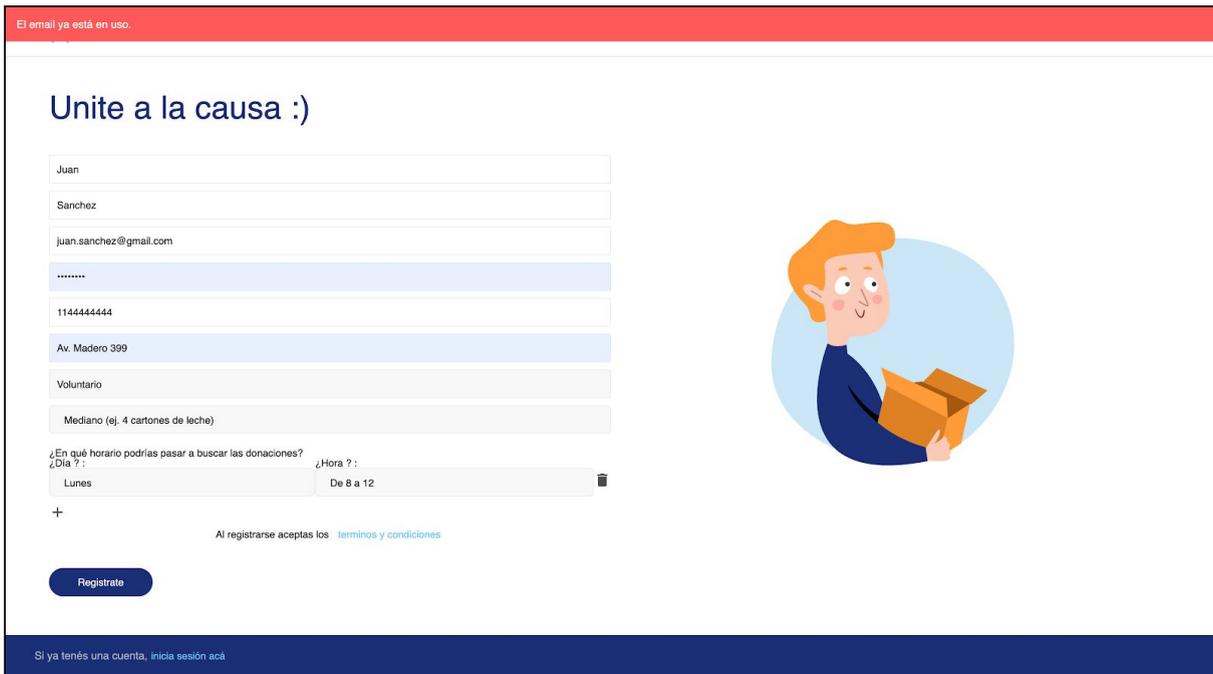


Figura 15. Gráfico de la vista de registro de cuenta con un error causado por utilizar una dirección de email que ya está en uso



Figura 16. Gráfico del error ocasionado por utilización de dirección de email repetida en el registro de cuenta

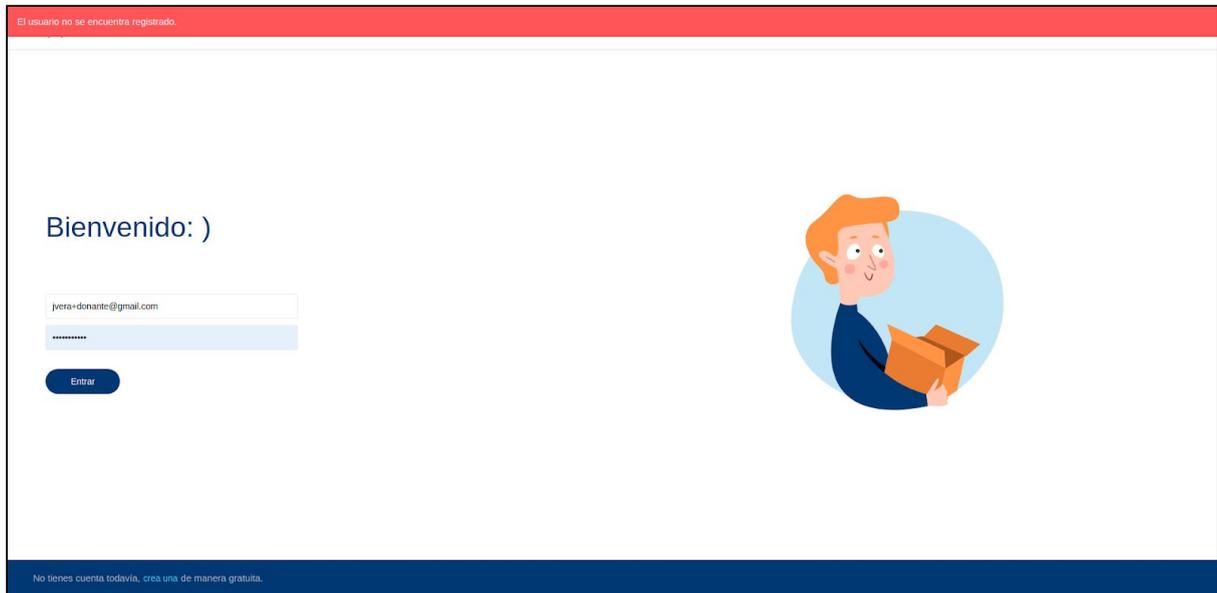
## Usuario Inexistente

Como fue mencionado con anterioridad, la dirección de email es *key* en la tabla de usuarios y es por ello que funciona como nombre de usuario. Eso quiere decir que un usuario que quiere ingresar a su cuenta, para poder interactuar con las herramientas de donación y para ver su perfil, debe hacerlo ingresando su email.

Puede pasar que un usuario quiera ingresar a su cuenta utilizando un email inexistente. A nivel usuario, en realidad, no es posible saber si es que en la utilización del email hubo un error de sintaxis o si se ingresó realmente un email que no fue registrado con anterioridad.

En cualquiera de los casos, para el sistema del *back end*, se intentó ingresar con una dirección de email que no se encuentra guardada en la tabla de usuarios; si ese fuera el

caso, el usuario, ya sea voluntario o donante, recibiría el error que se muestra en la *Figura 17*.



*Figura 17. Gráfico de la vista de inicio de sesión con un error causado por utilizar una dirección de email inexistente*



*Figura 18. Gráfico del error ocasionado por utilización de dirección de email inexistente en el inicio de sesión*

## Contraseña Incorrecta

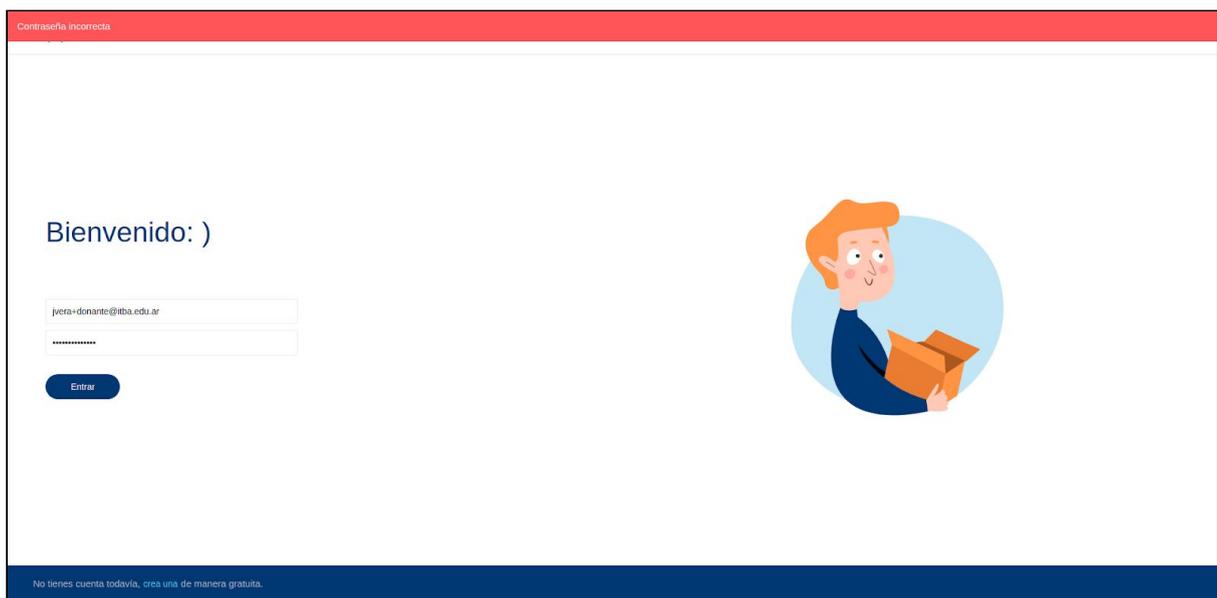
De manera análoga a lo mencionado en el error por dirección de email inexistente, el usuario, para iniciar sesión debe introducir, junto con su dirección de email que funcionara como nombre de usuario, la contraseña que eligió para registrarse.

Puede pasar que el usuario introduzca correctamente su dirección de email pero que se equivoque en la contraseña introducida. Si este es el caso, se le mostrará el error de la *Figura 19*.

Existen varias teorías sobre la información que recibe el usuario sobre los errores y que pueden inducir al respecto. Por ejemplo, *Amazon Web Services*, en alguna de sus

funcionalidades de servidores, no informa absolutamente nada si un inicio no fue satisfactorio, ni siquiera que el inicio resultó en un error. Esto incrementa, de alguna manera, a la seguridad ya que el usuario no puede saber si es que la plataforma “se tildó”, si perdió conectividad a internet, si el usuario es incorrecto, si el email es incorrecto o si la dirección del servidor es incorrecta.

Sabiendo las ventajas y desventajas de informar a los usuarios el error específico ocasionado, se optó por ser explícito con la esperanza de que “no saber qué sucedió” no sea un motivo para que los usuarios dejen de utilizar la plataforma.



*Figura 19. Gráfico de la vista de inicio de sesión con un error causado por utilizar una contraseña incorrecta junto con la dirección de email*



*Figura 20. Gráfico del error ocasionado por utilización de una contraseña incorrecta junto con la dirección de email en el inicio de sesión*

## Errores Externos

Así como hay errores que se ocasionan porque el usuario, pudiendo haber hecho las cosas de manera distinta, la hizo de una manera que resultó en un error; existen errores en los que el usuario no podría haber hecho nada para evitarlos. De igual manera, ante esta

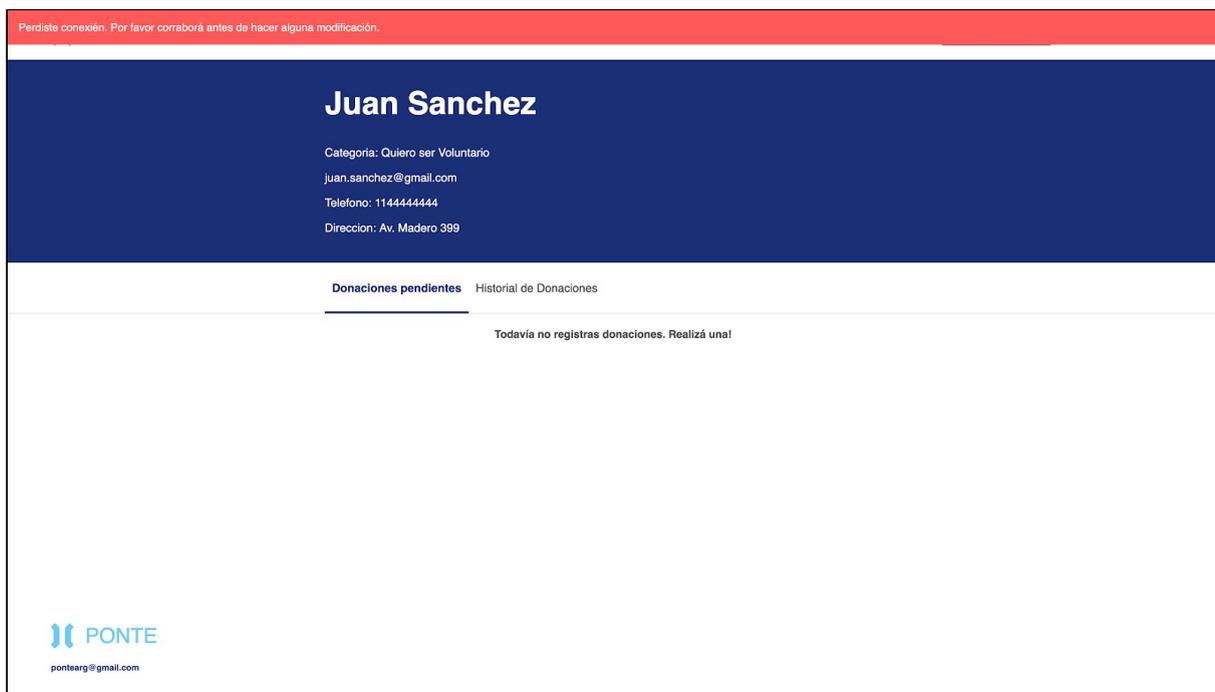
situación, se optó por notificarle al usuario que ocurrió un error y brindarle alguna acción que puede realizar para contribuir a su solución.

## Error de conectividad

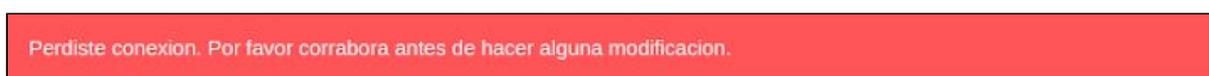
Como fue mencionado en la sección de arquitectura, Ponte utiliza bases de datos y herramientas almacenadas en servidores externos, con los cuales la conexión es mediante internet.

No es posible asumir que un usuario está corroborando todo el tiempo su conectividad, así como tampoco es posible asumir que todos los usuarios saben que, al no tener conectividad, no pueden hacer ninguna modificación ya que esta no tendrá efecto.

Es por ello que se decidió notificar al usuario, ante la falta de conexión a internet, con el error de la *Figura 21*.



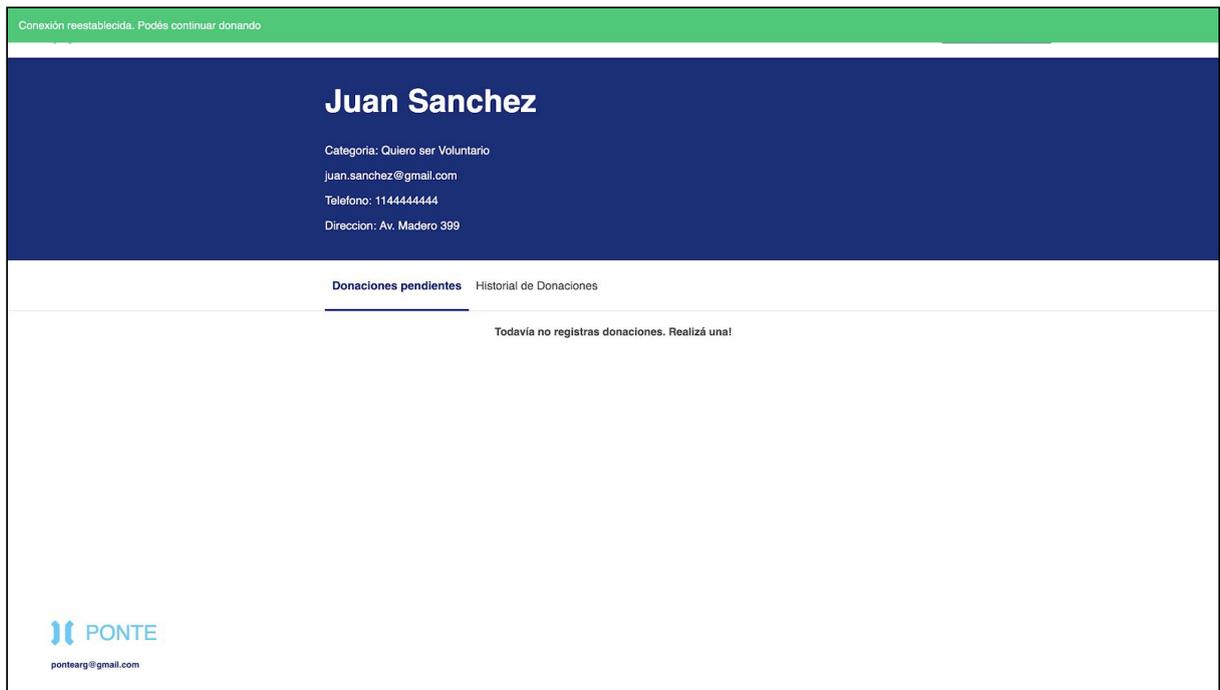
*Figura 21. Gráfico de la vista de perfil con un error causado por la pérdida de conexión a internet*



*Figura 22. Gráfico del error ocasionado por la pérdida de conectividad*

A diferencia de la notificación del resto de los errores en la plataforma, esta es la única notificación que no desaparece luego de algunos segundos, sino que lo hace cuando se ha restablecido la conexión.

Cuando eso sucede, el usuario también es notificado de la manera que se muestra en la *Figura 23*.



*Figura 23. Gráfico de la vista de perfil con la notificación de conexión restablecida*



*Figura 24. Gráfico de la notificación de conectividad restablecida*

Como se puede observar, en la *Figura 22* y en la *Figura 24*, se hace uso de la diferencia de colores para notificar al usuario y diferenciar entre los diferentes tipos de notificación.

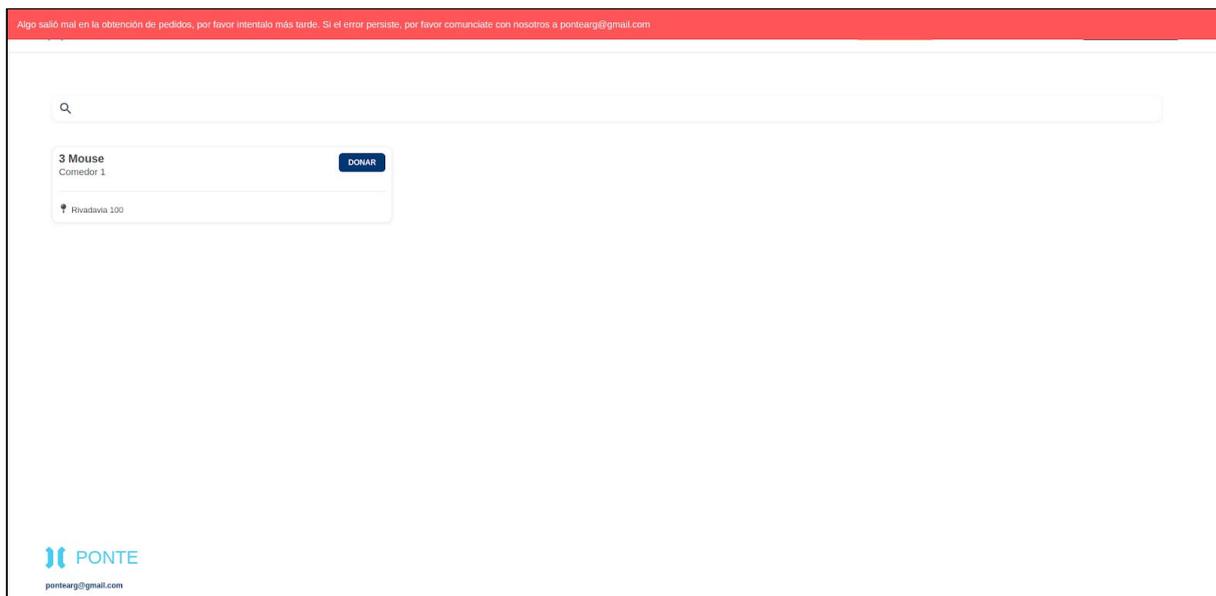
## Errores de servidor

Los errores en el servidor son imprevistos, en particular para el usuario, y no hay nada desde el lado del mismo para solucionarlos. Si bien el equipo de desarrollo es notificado ante los errores de servidor, es necesario informarle al usuario de los mismos para mantener la transparencia en la comunicación.

Al mismo tiempo, se concluyó que sería contraproducente informarle al usuario que fue lo que ocasionó el error, cuando este es en el lado de los servidores, ya que esto puede asustar a los usuarios menos informados en el área informática.

En la mayoría de los casos, los errores de servidor dejan de existir luego de un pequeño lapso de tiempo, ya sea por que es ese tiempo el que le tarda al equipo de desarrollo solucionarlo, o es el tiempo en el que las plataformas externas con las que se trabajan, tarda en restaurar sus herramientas. Esa es la razón por la que se les notifica a los usuarios, como se puede ver en la *Figura 25*, que la acción que estaban realizando no se completó satisfactoriamente y que vuelvan a intentarlo más adelante.

Al mismo tiempo, en la notificación, se le provee a los usuarios con un medio de comunicación para que puedan expresar sus inquietudes y notificar, ellos a su vez, cómo fue que se encontraron con ese error, para que el equipo de desarrollo tenga más herramientas para poder solucionarlo.



*Figura 25. Gráfico de la vista de pedidos con un error en el servidor*



*Figura 26. Gráfico de la notificación de error en el servidor*

## Notificaciones de éxito

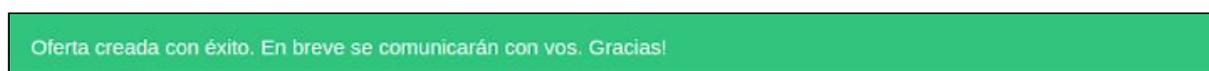
Como fue explicitado con anterioridad, uno de los objetivos fundamentales en la plataforma es mantener la transparencia en la comunicación y que el usuario pueda sentir en todo momento que sabe que es lo que está sucediendo.

Es así que se decidió notificar los errores que ocurren en la plataforma, como fue mencionado hasta recién, y se optó por notificar a los usuarios cuando sus acciones son ejecutadas con éxito. Para ello, no solo se recurre al envío de emails sino también a las notificaciones dentro de la plataforma.

En la *Figura 27* y en la *Figura 29*, se muestran algunos ejemplos de notificaciones de éxito.



*Figura 27. Gráfico de la vista inicial con la notificación de donación realizada con éxito*



*Figura 28. Gráfico de la notificación de donación creada con éxito*

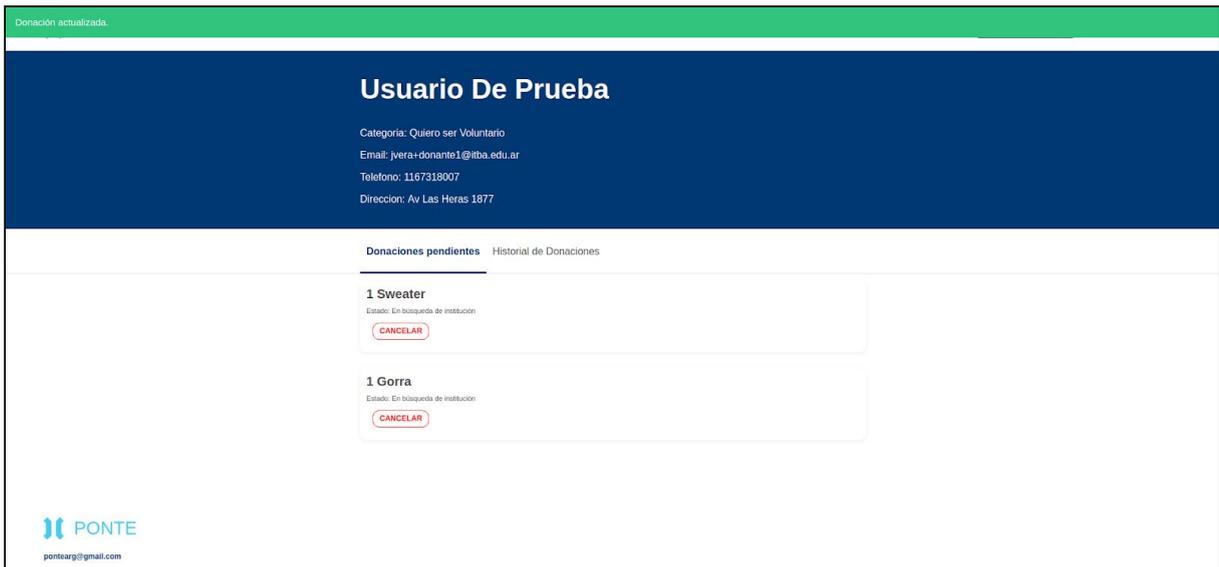


Figura 29. Gráfico de la vista del perfil con la notificación de donación actualizada con éxito



Figura 30. Gráfico de la notificación de donación actualizada con éxito

# Casos de Uso

El proceso entero, desde que un bien es entregado por un donante hasta que llega a una institución que lo necesita, involucra a muchos agentes y tiene diferentes maneras de ser iniciado.

A continuación se listan las diferentes situaciones en las cuales se puede iniciar una donación, los diferentes agentes que pueden ser involucrados según el momento del proceso en el que se encuentra la donación y las acciones que los mismos pueden realizar.

Para mayor comprensión, se optó por dividirlos, inicialmente según como pueden ser originadas.

Siguiendo con ese pensamiento, los estados de las donaciones serían los mostrados en el flujo de estados de la *Figura 31*.

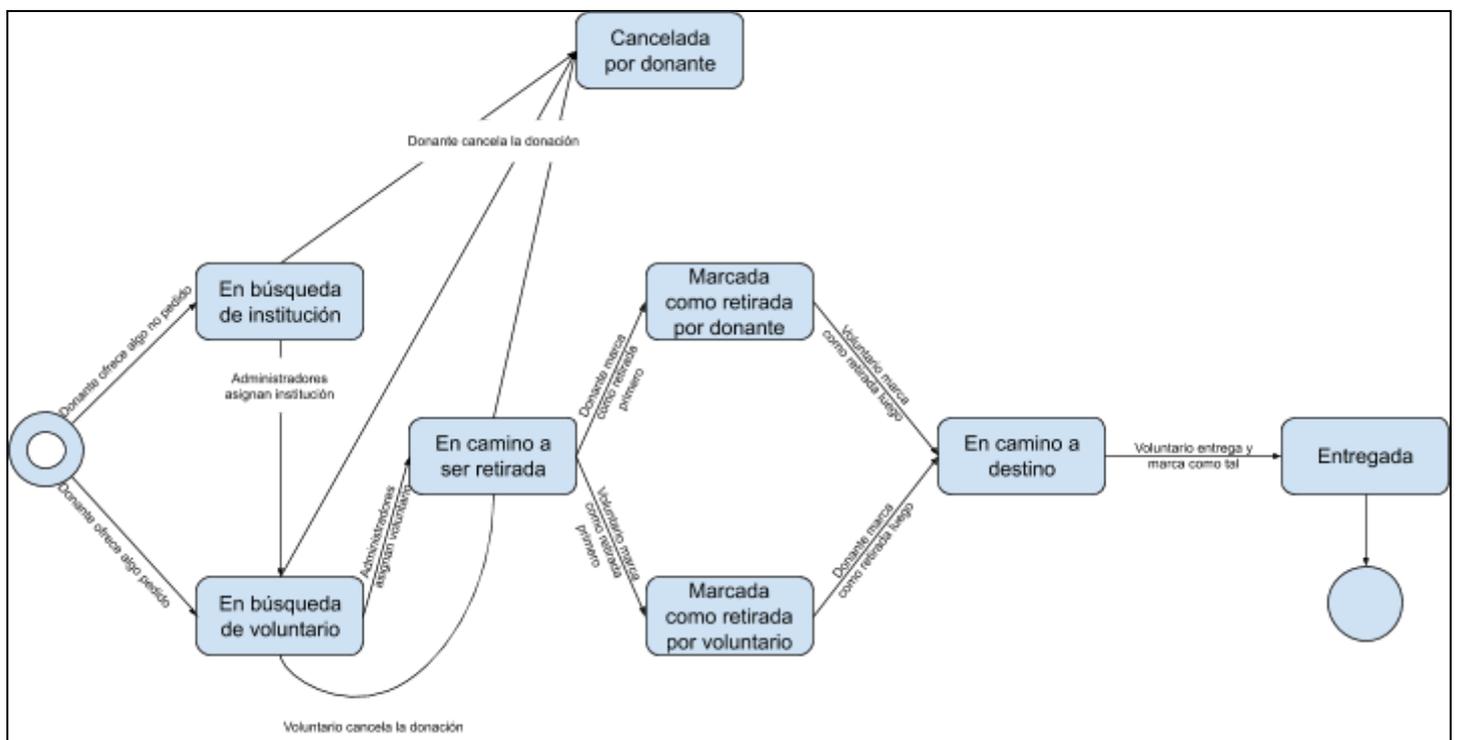


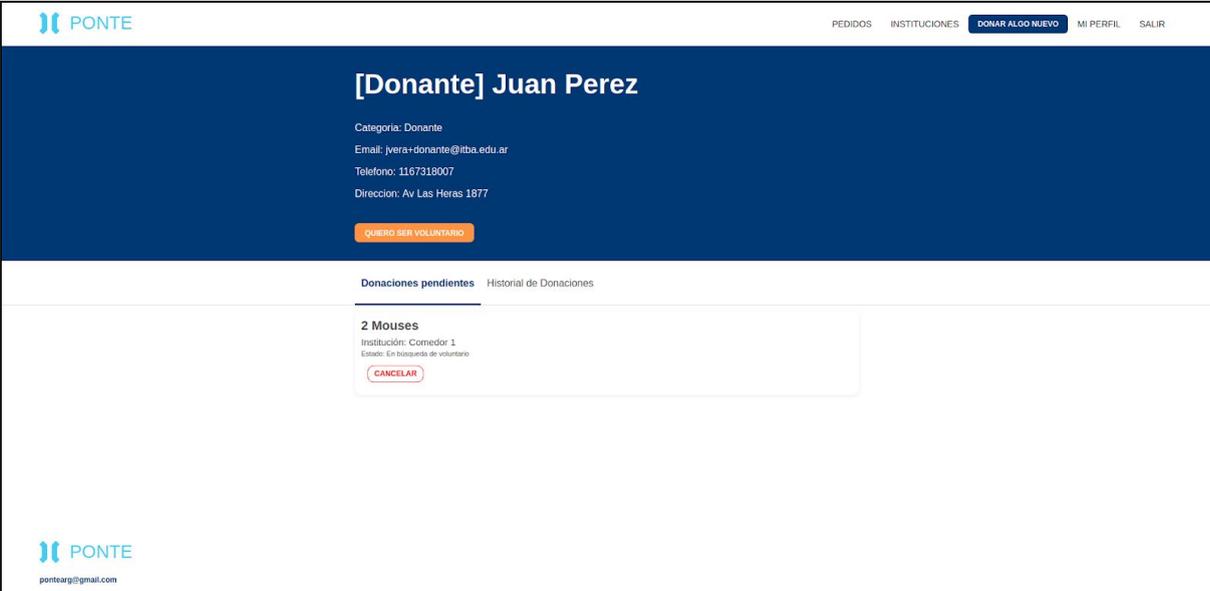
Figura 31. Gráfico del flujo sobre los estados de las donaciones

## Institución realiza un pedido particular

Cuando una institución habilitada realiza un pedido de un bien en particular, el mismo empieza a figurar en la vista de pedidos, como se ve en la *Figura 8*. A partir de ese momento, un usuario puede donar ese bien para esa institución. Si la cantidad pedida fuera mayor a 1, el donante puede elegir entre donar la totalidad pedida o una cantidad menor.

Una vez que el donante elige la cantidad a donar y realiza la donación, la misma empieza a figurar en su perfil como aparece en la *Figura 31*. Es importante remarcar que a nivel interno, cuando un donante ofrece un bien, se crea una *oferta*. Cuando una *oferta* y un *pedido* se combinan, se crea una lo que a nivel interno se denomina *donación*. Estas diferencias en la terminología pueden confundir al usuario, por lo que, para el usuario, todo es una donación desde que el bien es ofrecido.

Una vez que la donación es creada, los administradores son notificados y es su responsabilidad asignar un voluntario para ir a retirar lo ofrecido y llevarlo hacia la institución que lo requiere. Hasta que los administradores asignen un voluntario a la donación, la misma se encontrará en estado “*en búsqueda de voluntario*”.



The screenshot shows the user profile for Juan Perez, a donor. The profile includes contact details such as email, phone number, and address. A prominent orange button labeled 'QUIERO SER VOLUNTARIO' is visible. Below the profile, there are tabs for 'Donaciones pendientes' and 'Historial de Donaciones'. Under 'Donaciones pendientes', a single donation is listed: '2 Mouses' from 'Comedor 1', with a status of 'En búsqueda de voluntario' and a 'CANCELAR' button.

Figura32. Gráfico de la vista del perfil de un usuario donante en el que figura una donación en búsqueda de voluntario

A partir de este momento, y hasta que la donación sea retirada por el voluntario del domicilio del donante, el donante puede cancelar la donación en cualquier momento. En realidad, lo que estaría cancelando es la oferta. En este caso, el pedido vuelve a la cantidad pedida antes de que la oferta se hubiera hecho, la donación pasa a estar con el estado “cancelada por el usuario” y empezará a figurar en la pestaña de “Historial de Donaciones”.

Una vez que los administradores asignan un voluntario para buscar lo ofrecido y llevarlo a destino, la donación pasa a estar en estado “a ser retirada por voluntario”, y es visible también para el voluntario asignado. Se pueden observar estos cambios en la Figura 33 y en la Figura 34.

Se puede observar, también, que para los usuarios voluntarios, las donaciones que tienen que buscar y entregar, no figuran en la pestaña de “Donaciones”, sino en la pestaña de “Entregas”.

A partir de este momento, un voluntario puede cancelar la donación, indicando que no puede buscar la donación debido a un imprevisto. En ese caso, la donación pasa a estar nuevamente “en búsqueda de voluntario”, ya que el ofrecimiento no fue cancelado por el usuario.

The screenshot shows the PONTÉ user interface. At the top, there is a navigation bar with the PONTÉ logo and links for PEDIDOS, INSTITUCIONES, DONAR ALGO NUEVO, MI PERFIL, and SALIR. The main header area is dark blue and displays the user's name [Donante] Juan Perez, along with their category (Donante), email (jvera+donante@itba.edu.ar), phone number (1167318007), and address (Av Las Heras 1877). Below this is an orange button labeled 'QUIERO SER VOLUNTARIO'. The main content area is white and shows a tabbed interface with 'Donaciones pendientes' selected. Under this tab, there is a list of donations. One donation is highlighted, showing '2 Mouses' from 'Institución: Comedor 1'. The status is 'Estado: En camino a ser retirada'. Below the donation details are two buttons: 'MARCAR COMO RETIRADA' and 'CANCELAR'. At the bottom left of the page, there is a footer with the PONTÉ logo and the email address pontearg@gmail.com.

Figura 33. Gráfico de la vista del perfil de un usuario donante en el que figura una donación en camino a ser retirada por el voluntario asignado

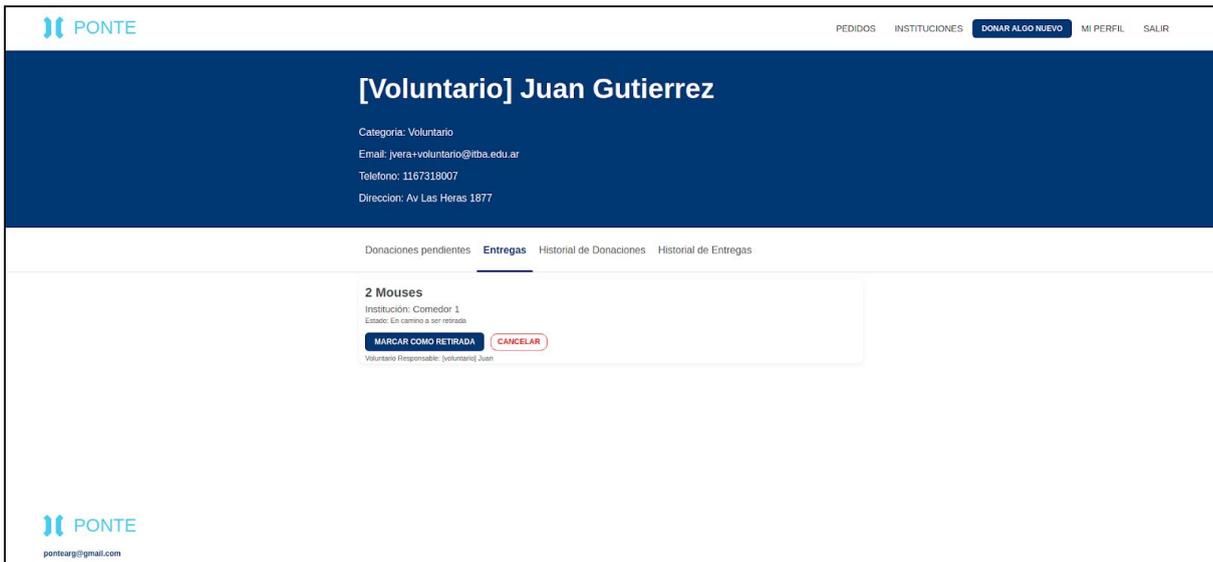


Figura 34. Gráfico de la vista del perfil de un usuario voluntario en el que figura una donación en camino a ser retirada por él que está asignado

A partir de este momento, los posibles caminos de una donación aumentan. Para mantener la transparencia sobre el punto en el que se encuentra la donación, se optó por darle la responsabilidad de indicar el estado de la donación, a las dos partes involucradas.

El siguiente estado de una donación, siguiendo un razonamiento lógico, sería que lo ofrecido fuera retirado por el voluntario y la donación estuviera en camino a la institución destino.

Sin embargo, para tratar de solucionar, algunos problemas externos a lo informático y poder asegurar que efectivamente lo ofrecido fue retirado del domicilio del donante, se optó por pedirle a ambas partes que notifiquen cuando el bien fue retirado.

Es así que puede ocurrir que, incluso si lo hicieran al mismo tiempo, a nivel pedidos al servidor, el donante realice esta actualización primero; y puede pasar que el voluntario realice esta actualización primero.

## Donante notifica primero que el bien fue retirado

En caso que el donante notifique primero que lo ofrecido fue retirado de su domicilio, la donación pasa a estar en estado “*marcada como retirada por el donante*” como se puede observar en la *Figura 35* y en la *Figura 36*.

The screenshot shows the profile of a donor user named Juan Perez. The header includes the PONTE logo and navigation links: PEDIDOS, INSTITUCIONES, DONAR ALGO NUEVO, MI PERFIL, and SALIR. The user's profile information is displayed in a dark blue box, including their category (Donante), email (jvera-donante@itba.edu.ar), phone number (1167318007), and address (Av Las Heras 1877). Below this is an orange button labeled 'QUIERO SER VOLUNTARIO'. The main content area shows a list of donations under the heading 'Donaciones pendientes' and 'Historial de Donaciones'. A single donation is listed: '2 Mouses' from 'Institución: Comedor 1', with the status 'Estado: Marcada como entregada por donante'. A red 'CANCELAR' button is visible next to the donation details. The footer contains the PONTE logo and the email address 'pontearg@gmail.com'.

Figura 35. Gráfico de la vista del perfil de un usuario donante en el que figura una donación marcada como retirada por el donante

The screenshot shows the profile of a volunteer user named Juan Gutierrez. The header is identical to Figure 35. The user's profile information is displayed in a dark blue box, including their category (Voluntario), email (jvera+voluntario@itba.edu.ar), phone number (1167318007), and address (Av Las Heras 1877). Below this is a 'MARCAR COMO RETIRADA' button. The main content area shows a list of donations under the heading 'Donaciones pendientes', 'Entregas', 'Historial de Donaciones', and 'Historial de Entregas'. A single donation is listed: '2 Mouses' from 'Institución: Comedor 1', with the status 'Estado: Marcada como entregada por donante'. A blue 'MARCAR COMO RETIRADA' button and a red 'CANCELAR' button are visible next to the donation details. The footer contains the PONTE logo and the email address 'pontearg@gmail.com'.

Figura 36. Gráfico de la vista del perfil de un usuario voluntario en el que figura una donación marcada como retirada por el donante

En este estado, el único que puede realizar una acción para continuar el proceso esperado de la donación, es el voluntario que puede indicar él también que la donación fue retirada del domicilio del donante.

Hasta que la donación no esté marcada como retirada por ambas partes, la donación no podrá ser marcada como entregada. Una vez que el voluntario marca que la donación fue retirada, la donación pasa a estar en estado “*en camino a destino*”, como se puede observar en la *Figura 37* y en la *Figura 38*.

The screenshot shows a web interface for a donor profile. At the top left is the 'PONTE' logo. At the top right are navigation links: 'PEDIDOS', 'INSTITUCIONES', 'DONAR ALGO NUEVO', 'MI PERFIL', and 'SALIR'. The main header area is dark blue and contains the name '[Donante] Juan Perez'. Below the name, the following information is listed: 'Categoría: Donante', 'Email: jvera+donante@itba.edu.ar', 'Telefono: 1167318007', and 'Direccion: Av Las Heras 1877'. An orange button labeled 'QUIERO SER VOLUNTARIO' is positioned below the contact details. The main content area is white and features two tabs: 'Donaciones pendientes' (selected) and 'Historial de Donaciones'. Under the 'Donaciones pendientes' tab, there is a list of 2 donations. The first donation is '2 Mouses', with details: 'Institución: Comedor: 1', 'Estado: En camino a destino', and 'Voluntario Responsable: [voluntario] Juan'. At the bottom left of the page, the 'PONTE' logo and email 'pontearg@gmail.com' are displayed.

*Figura 37. Gráfico de la vista del perfil de un usuario donante en el que figura una donación en camino a destino*



Figura 38. Gráfico de la vista del perfil de un usuario voluntario en el que figura una donación en camino a destino

A partir de este momento, ninguna de las dos partes puede cancelar la donación.

### Voluntario notifica primero que el bien fue retirado

En caso que el voluntario notifique primero que lo ofrecido fue retirado del domicilio del donante, la donación pasa a estar en estado “*marcada como retirada por el voluntario*” como se puede observar en la *Figura 39* y en la *Figura 40*.

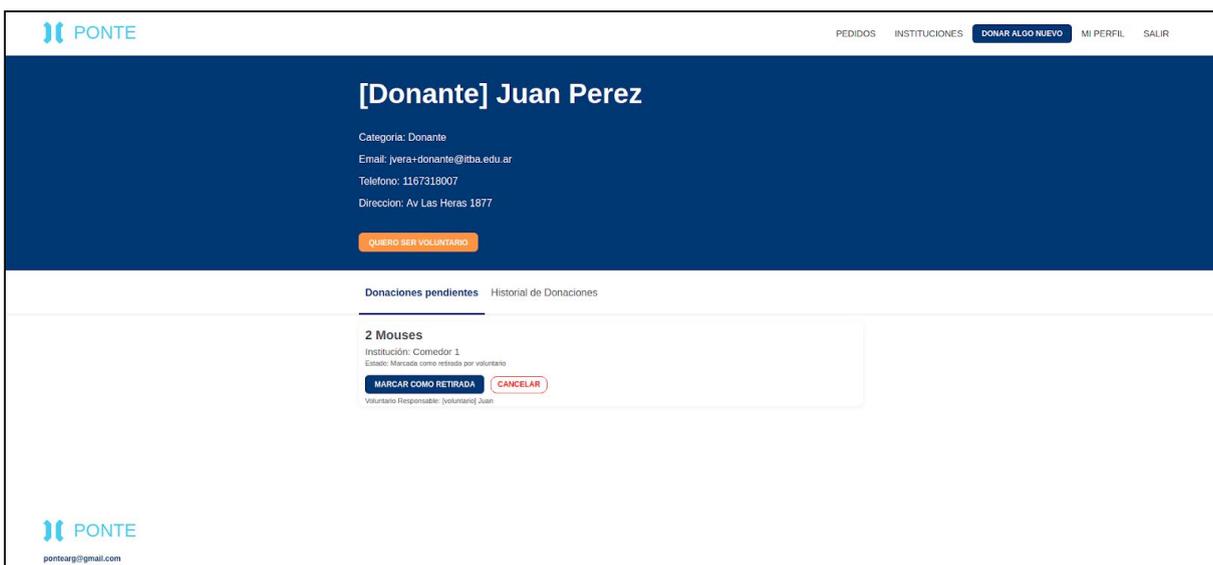


Figura 39. Gráfico de la vista del perfil de un usuario donante en el que figura una donación marcada como retirada por el voluntario

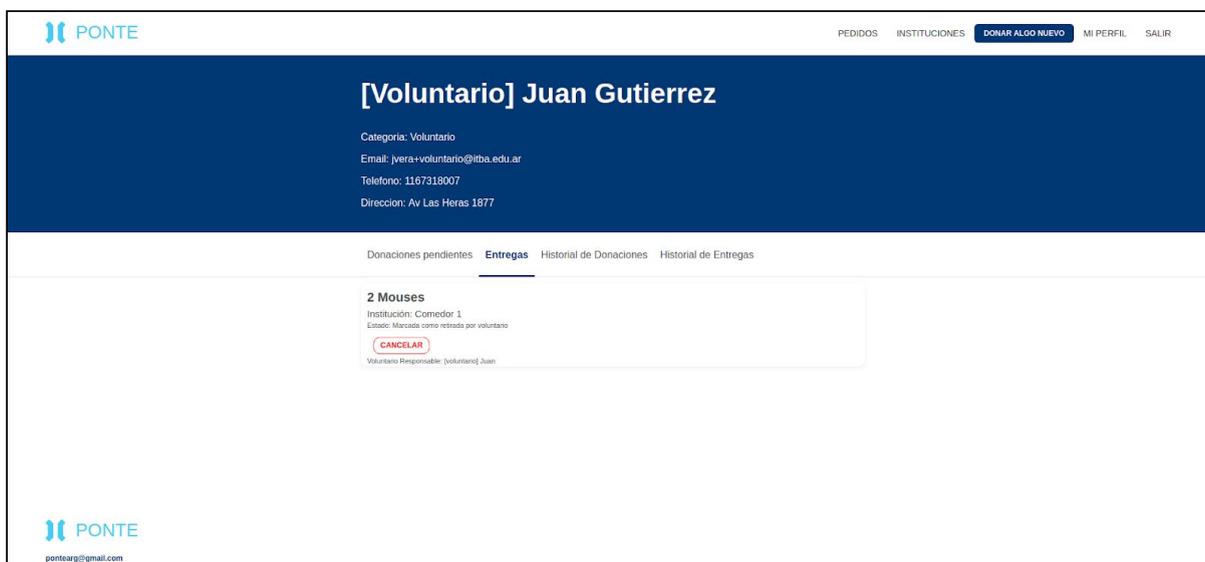


Figura 40. Gráfico de la vista del perfil de un usuario voluntario en el que figura una donación marcada como retirada por el voluntario

En este estado, el único que puede realizar una acción para continuar el proceso esperado de la donación, es el donante que puede indicar él también que la donación fue retirada de su domicilio.

Nuevamente, hasta que la donación no esté marcada como retirada por ambas partes, la donación no podrá ser marcada como entregada. Una vez que el donante marca que la donación fue retirada, la donación pasa a estar en estado “*en camino a destino*”, como se puede observar nuevamente en la *Figura 37* y en la *Figura 38*.

## Entrega en destino

Una vez que ambas partes marcaron la donación como retirada, y la donación se encuentra en estado “*en camino a destino*”, es responsabilidad del voluntario marcar la donación como entregada.

Una vez que el voluntario entregó la donación y marcó la misma con este estado, la donación pasa a estar en estado “*entregada*” y deja de figurar en las pestañas de “*donaciones*” y “*entregas*” para el donante y el voluntario respectivamente, sino que pasa a figurar en los historiales correspondientes, como se puede observar en la *Figura 41* y en la *Figura 42*.

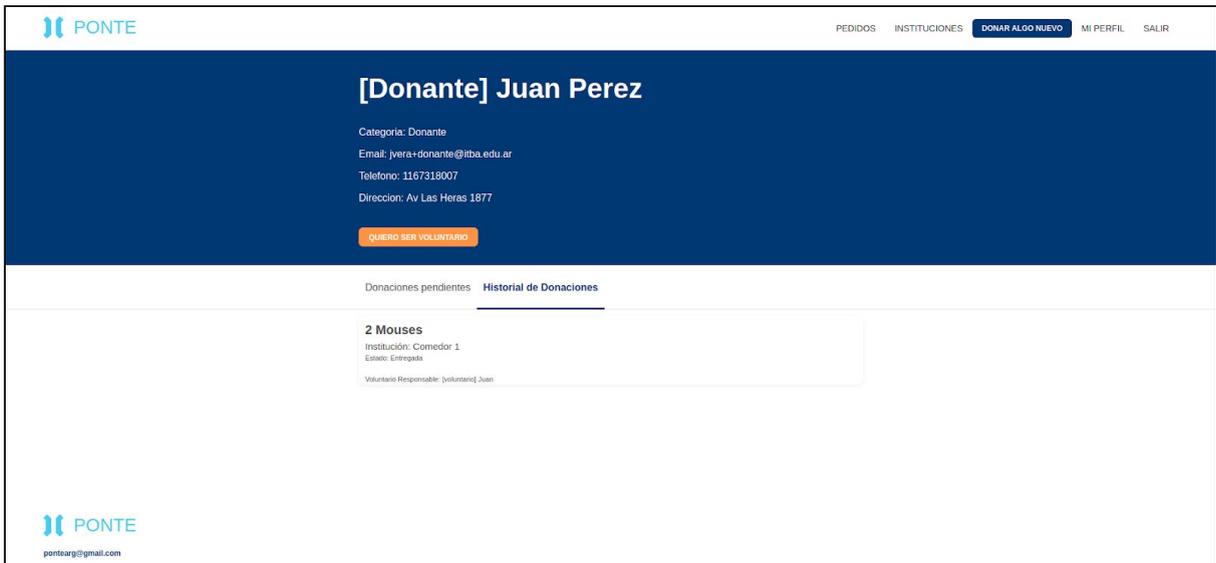


Figura 41. Gráfico de la vista del perfil de un usuario donante en el que figura una donación como entregada en la pestaña de historial de donaciones



Figura 42. Gráfico de la vista del perfil de un usuario voluntario en el que figura una donación como entregada en la pestaña de historial de entregas

## Donante realiza un ofrecimiento de un bien no pedido

Paralelamente, un usuario donante puede querer donar un bien que no está siendo necesitado por ninguna institución en ese momento. La plataforma le da la posibilidad a los usuarios de realizar esta acción.

En este caso, los administradores reciben una notificación de que un usuario desea donar un bien que no está siendo pedido y es responsabilidad suya asignarle a la oferta un lugar de destino.

Como fue mencionado anteriormente, a nivel interno, una donación sólo existe a partir de la combinación de un pedido y de una oferta; por lo tanto, en esta situación no existiría una donación, a falta de un pedido.

Los administradores pueden buscar una institución que necesite el bien ofrecido en la cantidad ofrecido y crear un pedido desde el *backoffice*, y a partir de las mismas crear una donación. Así mismo, los administradores pueden crear un pedido hacia una institución ficticia que representa el depósito de Ponte, y crear una donación a partir de las mismas.

Es importante remarcar que uno de los objetivos principales de la plataforma es que el donante espere el menor tiempo posible hasta que la donación es retirada de su domicilio. Es por ello que se creó la opción del depósito, para que la donación pueda seguir su curso normal, hasta que se encuentre una institución que pueda hacer uso de lo ofrecido.

Por otro lado, es importante remarcar, también, que esta institución ficticia, al igual que los pedidos ficticios que con la misma, no son visibles a ningún usuario.

Hasta que el ofrecimiento tenga un pedido con el que se pueda combinar para generar, a nivel interno, una donación; para el usuario, la donación figurará en estado “*en búsqueda de institución*”, como se puede observar en la *Figura 43*.

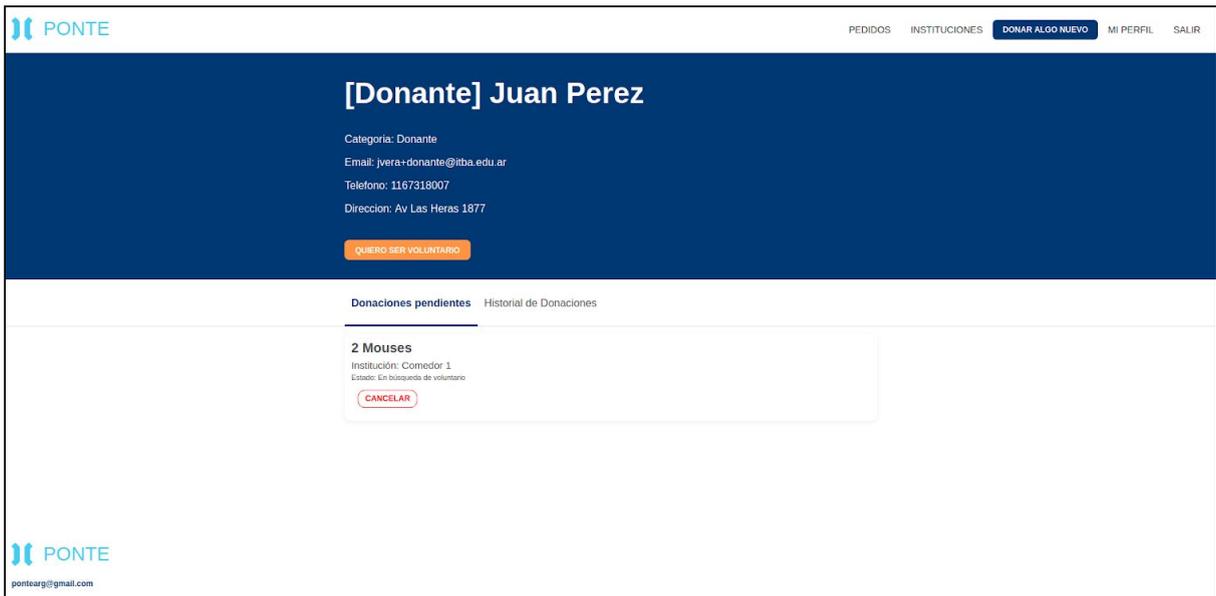


Figura 43. Gráfico de la vista del perfil de un usuario donante en el que figura una donación en búsqueda de institución

## Diagrama de Flujo

Para acompañar el detalle de lo antes mencionado, se decidió armar el siguiente diagrama de flujo con los estados posibles de las donaciones y los actores involucrados.

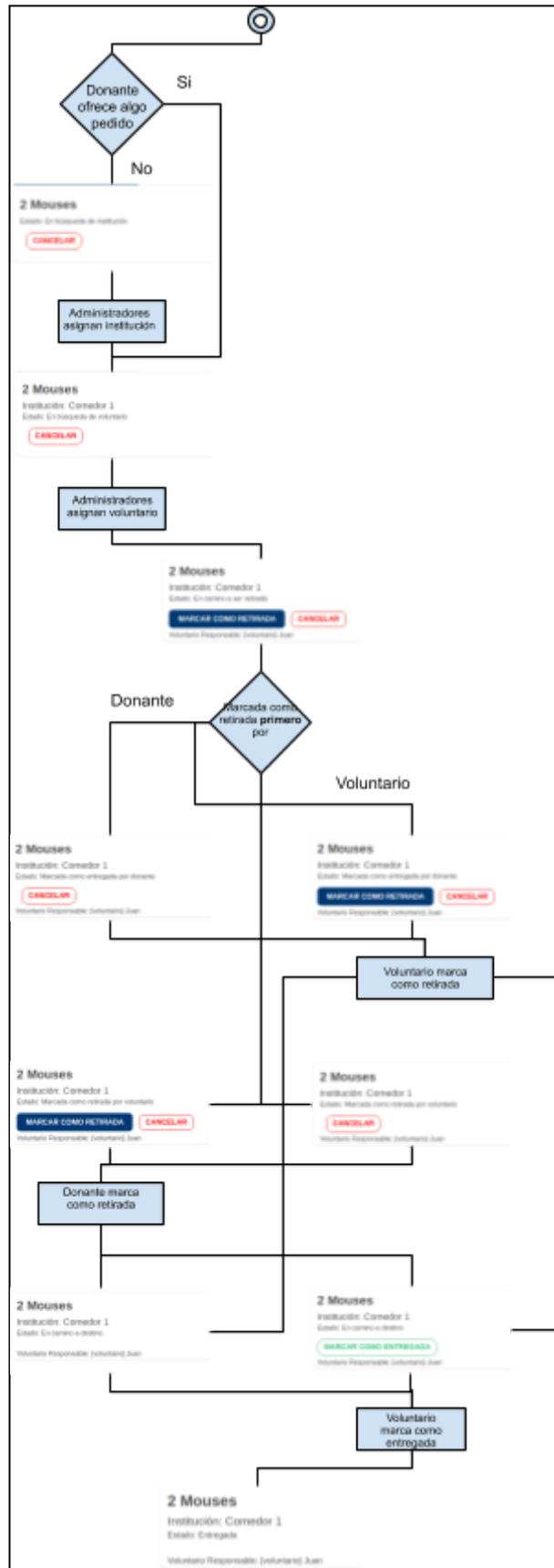


Figura 44. Gráfico del diagrama de flujo sobre los estados de las donaciones

Como se puede observar en la *Figura 44*, del lado izquierdo del diagrama se encuentran los estados y acciones que involucran únicamente al donante; así mismo, del lado derecho del diagrama se encuentran los estados y acciones que involucran únicamente al voluntario. Por otro lado, en el medio del diagrama se encuentran los estados que son vistos por ambas partes de igual manera.

Para mantener la claridad en el diagrama, se consideró oportuno no agregar las posibilidades que tienen ambas partes, donante y voluntario, de cancelar una donación. Como fue mencionado con anterioridad, si el donante cancela la donación, la misma evoluciona en un estado nuevo llamado *“cancelado por el usuario”*; mientras que si la cancela el voluntario, la misma vuelve al estado *“en busca de voluntario”*.

# Casos de Prueba

<b>#Test</b>	1		
<b>Nombre del Test</b>	Crear una cuenta donante		
<b>Objetivo</b>	Un usuario cualquiera podrá ingresar a la plataforma y crear una cuenta para poder realizar donaciones		
<b>Pre-condiciones</b>	-		
<b>Entradas</b>	Datos de ingreso obligatorios: <ul style="list-style-type: none"> <li>● Nombre</li> <li>● Apellido</li> <li>● Email</li> <li>● Contraseña</li> <li>● Teléfono</li> <li>● Dirección</li> </ul>		
<b>Sistemas Involucrados</b>	<div style="background-color: #ADD8E6; border: 1px solid black; padding: 5px; display: inline-block;"> <b>APLICACIÓN WEB</b> </div>		
<b>#Paso</b>	<b>Descripción del paso</b>	<b>Resultado esperado</b>	<b>Pasó / Falló (Just.)</b>
<b>1</b>	Se ingresa en el link de la plataforma web	La página muestra la landing explicativa	<b>Pasó</b>
<b>2</b>	Se selecciona Iniciar Sesión en la barra de navegación	La página muestra el formulario para iniciar sesión	<b>Pasó</b>
<b>3</b>	En la parte inferior se selecciona 'Crear una'	La página muestra el formulario para crear una cuenta	<b>Pasó</b>
<b>4</b>	Se cargan parcialmente los datos	Se muestra un mensaje de error indicando que hay datos faltantes	<b>Pasó</b>
<b>5</b>	Se cargan datos utilizando un email ya utilizado	Se muestra un mensaje de error indicando que el email está en uso	<b>Pasó</b>
<b>6</b>	Se cargan datos correctamente	Se muestra un mensaje de éxito y se redirecciona a la vista de inicio de sesión	<b>Pasó</b>

<b>#Test</b>	2		
<b>Nombre del Test</b>	Iniciar sesión en una cuenta existente		
<b>Objetivo</b>	Un usuario de Ponte podrá ingresar a la plataforma con su cuenta creada previamente		
<b>Pre-condiciones</b>	El usuario tiene que haber creado una cuenta donante previamente		
<b>Entradas</b>	Datos de ingreso obligatorios: <ul style="list-style-type: none"> <li>• Email</li> <li>• Contraseña</li> </ul>		
<b>Sistemas Involucrados</b>	<div style="border: 1px solid black; background-color: #ADD8E6; padding: 5px; display: inline-block;"> <b>APLICACIÓN WEB</b> </div>		
<b>#Paso</b>	<b>Descripción del paso</b>	<b>Resultado esperado</b>	<b>Pasó / Falló (Just.)</b>
<b>1</b>	Se ingresa en el link de la plataforma web	La página muestra la landing explicativa	<b>Pasó</b>
<b>2</b>	Se selecciona Iniciar Sesión en la barra de navegación	La página muestra el formulario para iniciar sesión	<b>Pasó</b>
<b>3</b>	Se cargan parcialmente los datos	Se muestra un mensaje de error indicando que hay datos faltantes	<b>Pasó</b>
<b>4</b>	Se cargan datos utilizando un email no existente	Se muestra un mensaje de error indicando que el email no está en uso	<b>Pasó</b>
<b>5</b>	Se cargan datos utilizando una contraseña que no corresponde al email	Se muestra un mensaje de error indicando que la contraseña es incorrecta	<b>Pasó</b>
<b>6</b>	Se cargan datos correctamente	Se muestra un mensaje de éxito y se redirecciona a la landing con nuevas funcionalidades en la barra de navegación	<b>Pasó</b>

<b>#Test</b>	3		
<b>Nombre del Test</b>	Ver los pedidos realizados por las instituciones		
<b>Objetivo</b>	Un usuario de Ponte podrá ingresar a la plataforma para poder ver los pedidos realizados por las instituciones		
<b>Pre-condiciones</b>	El usuario tiene que haber creado una cuenta donante previamente		
<b>Entradas</b>	-		
<b>Sistemas Involucrados</b>	<div style="border: 1px solid black; border-radius: 10px; padding: 5px; display: inline-block; background-color: #ADD8E6;"> <b>APLICACIÓN WEB</b> </div>		
<b>#Paso</b>	<b>Descripción del paso</b>	<b>Resultado esperado</b>	<b>Pasó / Falló (Just.)</b>
<b>1</b>	Se ingresa en el link de la plataforma web	La página muestra la landing explicativa	<b>Pasó</b>
<b>2</b>	Se selecciona 'Pedidos' en la barra de navegación	La página muestra los pedidos realizados con la información de la institución que lo realiza	<b>Pasó</b>

<b>#Test</b>	4		
<b>Nombre del Test</b>	Ver las instituciones que realizan o realizaron pedidos		
<b>Objetivo</b>	Un usuario de Ponte podrá ingresar a la plataforma para poder ver las instituciones involucradas		
<b>Pre-condiciones</b>	El usuario tiene que haber creado una cuenta donante previamente		
<b>Entradas</b>	-		
<b>Sistemas Involucrados</b>	<div style="border: 1px solid black; border-radius: 10px; padding: 5px; display: inline-block; background-color: #ADD8E6;"> <b>APLICACIÓN WEB</b> </div>		
<b>#Paso</b>	<b>Descripción del paso</b>	<b>Resultado esperado</b>	<b>Pasó / Falló (Just.)</b>
<b>1</b>	Se ingresa en el link de la plataforma web	La página muestra la landing explicativa	<b>Pasó</b>
<b>2</b>	Se selecciona 'Instituciones' en la barra de navegación	La página muestra las instituciones involucradas con su respectiva información	<b>Pasó</b>

<b>#Test</b>	5		
<b>Nombre del Test</b>	Donar un bien pedido		
<b>Objetivo</b>	Un usuario de Ponte podrá ingresar a la plataforma para donar un bien que alguna institución esté pidiendo		
<b>Pre-condiciones</b>	El usuario tiene que haber creado una cuenta donante previamente		
<b>Entradas</b>	Datos de ingreso obligatorios: <ul style="list-style-type: none"> <li>• Unidades a donar</li> <li>• Disponibilidad horaria</li> </ul> Datos de ingreso opcional: <ul style="list-style-type: none"> <li>• Comentarios adicionales</li> </ul>		
<b>Sistemas Involucrados</b>	<div style="border: 1px solid black; border-radius: 10px; background-color: #ADD8E6; padding: 5px; display: inline-block;"> <b>APLICACIÓN WEB</b> </div>		
<b>#Paso</b>	<b>Descripción del paso</b>	<b>Resultado esperado</b>	<b>Pasó / Falló (Just.)</b>
<b>1</b>	Se ingresa en el link de la plataforma web	La página muestra la landing explicativa	<b>Pasó</b>
<b>2</b>	Se selecciona 'Pedidos' en la barra de navegación	La página muestra los pedidos realizados con la información de la institución que lo realiza	<b>Pasó</b>
<b>3</b>	Se selecciona 'Donar' en cualquier pedido	Se muestra el formulario de donación con datos del usuario ya cargados y sin posibilidad de edición	<b>Pasó</b>
<b>4</b>	Se envía el formulario con una cantidad mayor a la pedida / menor igual a 0	Se muestra un mensaje de error indicando que la cantidad no es correcta	<b>Pasó</b>
<b>5</b>	Se eliminan todas las posibilidades de franja horaria	Se muestra un mensaje de error indicando que es obligatorio presentar al menos una posibilidad	<b>Pasó</b>
<b>6</b>	Se cargan datos correctamente	Se muestra un mensaje de éxito y se modifica la cantidad pedida con respecto a lo ofrecido. El usuario recibe un email	<b>Pasó</b>

<b>#Test</b>	6		
<b>Nombre del Test</b>	Donar un bien no pedido		
<b>Objetivo</b>	Un usuario de Ponte podrá ingresar a la plataforma para donar un bien cualquiera independientemente de si alguna institución lo pide o no		
<b>Pre-condiciones</b>	El usuario tiene que haber creado una cuenta donante previamente		
<b>Entradas</b>	Datos de ingreso obligatorios: <ul style="list-style-type: none"> <li>• Bien a donar</li> <li>• Unidades a donar</li> <li>• Disponibilidad horaria</li> </ul> Datos de ingreso opcional: <ul style="list-style-type: none"> <li>• Comentarios adicionales</li> </ul>		
<b>Sistemas Involucrados</b>	<div style="border: 1px solid black; border-radius: 10px; background-color: #00aaff; color: white; padding: 5px; display: inline-block;"> <b>APLICACIÓN WEB</b> </div>		
<b>#Paso</b>	<b>Descripción del paso</b>	<b>Resultado esperado</b>	<b>Pasó / Falló (Just.)</b>
<b>1</b>	Se ingresa en el link de la plataforma web	La página muestra la landing explicativa	<b>Pasó</b>
<b>2</b>	Se selecciona 'Donar algo nuevo' en la barra de navegación	Se muestra el formulario de donación con datos del usuario ya cargados y sin posibilidad de edición	<b>Pasó</b>
<b>3</b>	Se envía el formulario con el nombre del bien sin completar	Se muestra un mensaje de error indicando que el nombre del bien es obligatorio	<b>Pasó</b>
<b>4</b>	Se envía el formulario con una cantidad mayor a la pedida / menor igual a 0	Se muestra un mensaje de error indicando que la cantidad no es correcta	<b>Pasó</b>
<b>5</b>	Se eliminan todas las posibilidades de franja horaria	Se muestra un mensaje de error indicando que es obligatorio presentar al menos una posibilidad	<b>Pasó</b>
<b>6</b>	Se cargan datos correctamente	Se muestra un mensaje de éxito. El usuario recibe un email	<b>Pasó</b>

<b>#Test</b>	7		
<b>Nombre del Test</b>	Ver las donaciones realizadas como donante y el estado de las mismas		
<b>Objetivo</b>	Un usuario de Ponte podrá ingresar a la plataforma para ver el estado de las donaciones que haya realizado		
<b>Pre-condiciones</b>	El usuario tiene que haber creado una cuenta donante previamente. El usuario tiene que haber donado algo previamente		
<b>Entradas</b>	-		
<b>Sistemas Involucrados</b>	<div style="border: 1px solid black; border-radius: 10px; background-color: #ADD8E6; padding: 5px; display: inline-block;"> <b>APLICACIÓN WEB</b> </div>		
<b>#Paso</b>	<b>Descripción del paso</b>	<b>Resultado esperado</b>	<b>Pasó / Falló (Just.)</b>
<b>1</b>	Se ingresa en el link de la plataforma web	La página muestra la landing explicativa	<b>Pasó</b>
<b>2</b>	Se selecciona 'Perfil' en la barra de navegación	La página muestra las donaciones realizadas, el estado de las mismas, y la información del usuario.	<b>Pasó</b>

<b>#Test</b>	8		
<b>Nombre del Test</b>	Cancelar una donación realizada		
<b>Objetivo</b>	Un usuario de Ponte podrá ingresar a la plataforma para cancelar cualquiera de las donaciones que haya realizado		
<b>Pre-condiciones</b>	El usuario tiene que haber creado una cuenta donante previamente. El usuario tiene que haber donado algo previamente		
<b>Entradas</b>	-		
<b>Sistemas Involucrados</b>	<div style="border: 1px solid black; border-radius: 10px; background-color: #ADD8E6; padding: 5px; display: inline-block;"> <b>APLICACIÓN WEB</b> </div>		
<b>#Paso</b>	<b>Descripción del paso</b>	<b>Resultado esperado</b>	<b>Pasó / Falló (Just.)</b>
<b>1</b>	Se ingresa en el link de la plataforma web	La página muestra la landing explicativa	<b>Pasó</b>
<b>2</b>	Se selecciona 'Perfil' en la barra de navegación	La página muestra las donaciones realizadas, el estado de las mismas, y la información del usuario.	<b>Pasó</b>
<b>3</b>	Se selecciona en 'Cancelar' de cualquier donación	Se muestra un mensaje de éxito. La donación desaparece de la sección de donaciones pendientes. La donación ahora aparece en el historial con estado 'cancelada por el donante'. El donante recibe un email	<b>Pasó</b>

<b>#Test</b>	9		
<b>Nombre del Test</b>	Crear una cuenta Voluntario desde el momento inicial		
<b>Objetivo</b>	Un usuario de Ponte podrá ingresar a la plataforma y crear una cuenta voluntario que le permitirá, además de realizar donaciones, ser considerado voluntario, post aprobación.		
<b>Pre-condiciones</b>	-		
<b>Entradas</b>	Datos de ingreso obligatorios: <ul style="list-style-type: none"> <li>● Nombre</li> <li>● Apellido</li> <li>● Email</li> <li>● Contraseña</li> <li>● Teléfono</li> <li>● Dirección</li> <li>● Tamaño de vehículo</li> <li>● Franja horaria de disponibilidad</li> </ul>		
<b>Sistemas Involucrados</b>	<b>APLICACIÓN WEB</b>		
<b>#Paso</b>	<b>Descripción del paso</b>	<b>Resultado esperado</b>	<b>Pasó / Falló (Just.)</b>
<b>1</b>	Se ingresa en el link de la plataforma web	La página muestra la landing explicativa	<b>Pasó</b>
<b>2</b>	Se selecciona Iniciar Sesión en la barra de navegación	La página muestra el formulario para iniciar sesión	<b>Pasó</b>
<b>3</b>	En la parte inferior se selecciona 'Crear una'	La página muestra el formulario para crear una cuenta	<b>Pasó</b>
<b>4</b>	Se cambia Donante por Voluntario	El formulario muestra más campos	<b>Pasó</b>
<b>5</b>	Se cargan parcialmente los datos	Se muestra un mensaje de error indicando que hay datos faltantes	<b>Pasó</b>
<b>6</b>	Se cargan datos utilizando un email ya utilizado	Se muestra un mensaje de error indicando que el email está en uso	<b>Pasó</b>
<b>7</b>	Se cargan datos correctamente	Se muestra un mensaje de éxito y se redirecciona a la vista de inicio de sesión	<b>Pasó</b>

<b>#Test</b>	10		
<b>Nombre del Test</b>	Crear una cuenta Voluntario a partir de una donante		
<b>Objetivo</b>	Un usuario de Ponte podrá ingresar a la plataforma y crear una cuenta voluntario, ya siendo donante, que le permitirá, además de realizar donaciones, ser considerado voluntario, post aprobación.		
<b>Pre-condiciones</b>	El usuario tiene que haber creado una cuenta donante previamente.		
<b>Entradas</b>	Datos de ingreso obligatorios: <ul style="list-style-type: none"> <li>• Tamaño de vehículo</li> <li>• Franja horaria de disponibilidad</li> </ul> Datos de ingreso opcional: <ul style="list-style-type: none"> <li>• Comentarios adicionales</li> </ul>		
<b>Sistemas Involucrados</b>	<b>APLICACIÓN WEB</b>		
<b>#Paso</b>	<b>Descripción del paso</b>	<b>Resultado esperado</b>	<b>Pasó / Falló (Just.)</b>
<b>1</b>	Se ingresa en el link de la plataforma web	La página muestra la landing explicativa	<b>Pasó</b>
<b>2</b>	Se selecciona 'Perfil' en la barra de navegación	La página muestra las donaciones realizadas, el estado de las mismas, y la información del usuario.	<b>Pasó</b>
<b>3</b>	En la parte superior se selecciona 'Quiero ser Voluntario'	La página muestra el formulario para ser voluntario	<b>Pasó</b>
<b>4</b>	Se eliminan todas las posibilidades de franja horaria	Se muestra un mensaje de error indicando que es obligatorio presentar al menos una posibilidad	<b>Pasó</b>
<b>5</b>	Se cargan datos correctamente	Se muestra un mensaje de éxito y se redirecciona a la vista de perfil en donde la categoría del usuario se modificó	<b>Pasó</b>

<b>#Test</b>	11		
<b>Nombre del Test</b>	Aceptar / Rechazar un voluntario		
<b>Objetivo</b>	Cuando un usuario de Ponte expresa su deseo de ser voluntario, su categoría es la de 'quiero ser voluntario'. A partir de allí, los administradores podrán aceptarlo o rechazarlo.		
<b>Pre-condiciones</b>	El usuario administrador debe tener acceso a la plataforma de backoffice. Tiene que haber usuarios que hayan expresado su deseo de ser voluntarios		
<b>Entradas</b>	-		
<b>Sistemas Involucrados</b>	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="background-color: #00AEEF; color: white; padding: 5px 10px; border-radius: 5px;">APLICACIÓN WEB</div> <div style="background-color: #003366; color: white; padding: 5px 10px; border-radius: 5px;">BACKOFFICE</div> </div>		
<b>#Paso</b>	<b>Descripción del paso</b>	<b>Resultado esperado</b>	<b>Pasó / Falló (Just.)</b>
<b>1</b>	Se ingresa en el link del backoffice	Se listan todos los elementos creados	<b>Pasó</b>
<b>2</b>	Se seleccionan los usuarios.	Se listan todas las usuarios creados, junto con la información de cada uno.	<b>Pasó</b>
<b>3</b>	Se edita un usuario cuyo estado sea "quiero ser voluntario" para cambiarlo a "voluntario"	El usuario se guarda correctamente. Desde la plataforma Web, el usuario ya puede visualizar el perfil de voluntario	<b>Pasó</b>
<b>4</b>	Se seleccionan los voluntarios.	Se listan todos los voluntarios creados, junto con la información de cada uno. El que recientemente modificó usuario, figura como no modificado en esta lista.	<b>Pasó</b>
<b>5</b>	Se edita el voluntario en cuestión, para cuyo estado sea "voluntario".	El voluntario se guarda correctamente. A partir de este momento, el voluntario es elegible desde el backoffice para ser asignado a una donación	<b>Pasó</b>

<b>#Test</b>	12		
<b>Nombre del Test</b>	Asignar un voluntario a una donación		
<b>Objetivo</b>	Cuando una donación es creada (existe un donante con deseo de donar un bien), los administradores podrán asignar un voluntario para que retire la donación y la entregue en destino		
<b>Pre-condiciones</b>	El usuario administrador debe tener acceso a la plataforma de backoffice. Tiene que haber donaciones creadas con anterioridad. Tiene que haber voluntarios registrados.		
<b>Entradas</b>	-		
<b>Sistemas Involucrados</b>			
<b>#Paso</b>	<b>Descripción del paso</b>	<b>Resultado esperado</b>	<b>Pasó / Falló (Just.)</b>
<b>1</b>	Se ingresa en el link del backoffice	Se listan todos los elementos creados	<b>Pasó</b>
<b>2</b>	Se seleccionan las donaciones	Se listan todas las donaciones creadas, los estados de las mismas y la información correspondiente	<b>Pasó</b>
<b>3</b>	Se edita una donación para agregarle un voluntario y se cambia el estado de la donación al correspondiente	La donación se guarda correctamente. En la plataforma Web, el donante ve reflejado el voluntario que le fue asignado. En la plataforma Web, el voluntario ve la donación como pendiente de entrega.	<b>Pasó</b>

# Conclusión

A medida que el proyecto avanzó, fue quedando en evidencia que el mismo es un proyecto muy ambicioso por su naturaleza de proyecto social. Como fue mencionado a lo largo de este informe, fueron surgiendo imprevistos, no solo desde lo tecnológico; sino también desde lo estructural y “sociológico” que incidían directamente en el desarrollo del proyecto (cómo hacer para estar seguros de que una institución es quien dice ser).

Si bien el equipo cuenta con mucha experiencia en el área de Front End, fue gracias al desarrollo de la plataforma, que se adquirió experiencia en otras tecnologías, acompañados por lo aprendiendo a lo largo de la carrera. A su vez, poder trabajar con alumnos y egresados de otras áreas, permitió el desarrollo de habilidades más *soft*, con las que se pudo poner en *lenguaje intermedio y general* cuestiones informáticas y problemáticas sociales para que ambas partes del equipo de Ponte pudieran entenderlas y trabajar sobre ellas.

Es ver el desarrollo de la plataforma lo que permite ser conscientes del proceso por el cual atravesó el trabajo hasta llegar hasta su estadio final, entendido como proyecto final de Ingeniería en Informática.

Es por ello que se considera importante detenerse un momento en asimilar el origen o aprendizaje de muchas de las herramientas utilizadas para llevar a cabo el desarrollo integral de este proyecto, que se menciona. Se concluye que estas herramientas brotaron de las siguientes materias:

- Ingeniería de Software I
- Ingeniería de Software II
- Proyecto de Aplicaciones Web
- Human Computer Interface (HCI)
- Base de Datos I y II
- Criptografía y Seguridad
- Gestión de Proyectos Informáticos
- Derecho para Ingenieros

- Sociología Política

Es la amplitud en lo desarrollado, aprendido, y superado, lo que llevó al equipo a reconocer que, a la hora de concluir, es fundamental volver a las bases fundantes del proyecto, sobretodo leídas en clave de trabajo final de Ingeniería en Informática.

*“Ponte busca ser una herramienta mediante la cual dos porciones de la sociedad desconectadas puedan unirse para tratar de saciar necesidades que, además de ser desconocidas para la gran mayoría, no son satisfechas.”*

Se considera que este, el norte fundamental del proyecto, se encuentra resuelto. La última versión *deployada* de Ponte, le permite a un usuario donar un bien específico con el deseo previo de donarlo, pero al mismo tiempo le presenta las necesidades particulares de instituciones que no son conocidas, no tienen los recursos para darse a conocer, y trabajan, en general, desde el anonimato.

Finalmente, el equipo es muy consciente de que Ponte, como proyecto y como aplicación tiene mucho potencial. Sería ideal poder lograr que la visita, de los usuarios, a la plataforma sea casi diaria para poder generar una mayor concientización y mejor flujo de donaciones. A partir de este potencial conocido, es que se decidió explicitar, en la siguiente sección, el trabajo futuro; entendiendo *trabajo futuro* como aspectos en los que Ponte puede seguir creciendo a nivel tecnológico y funcional.

# Trabajo Futuro

Ponte nació de la necesidad de sumar un granito de arena propio a la sociedad y desde el momento cero fue acompañado por una gran motivación y compromiso para con este proyecto. Esta motivación nació siempre de la mano de mucha energía, proactividad e ideas que se querían sumar al proyecto que hoy se considera propio.

En un principio, tanta energía motivó a querer incluir todas las ideas y fomentó la creencia de que era posible alcanzarlas en el tiempo estipulado por la cátedra para finalizar la tesis. Dado que era un trabajo entre dos personas y los tiempos eran limitados, se decidió limitar a armar un MVP con las funcionalidades básicas de la idea central del proyecto: las donaciones.

Ponte tiene un gran potencial y existen muchas funcionalidades que se podrían incluir en el producto que hoy se entrega, para sumarle valor al que ya tiene hoy. Algunas de las ideas que se proponen para sumar en un futuro son las siguientes:

## Match automático

Hoy por hoy, los voluntarios son asignados por los administradores según la disponibilidad de los mismos, la disponibilidad de los donantes, el tamaño de vehículo y del bien a ser donado, y la certeza de que los primeros voluntarios serán gente conocida; con el objetivo de que se concrete bien el horario y detalles de la búsqueda del bien, entre el donante y voluntario por otros medios de comunicación.

Se considera de gran utilidad poder implementar un sistema de inteligencia artificial para que la asignación del voluntario a una donación se haga de manera automática según las especificaciones de la donación, donante y voluntario.

## Aplicación mobile

Si bien, la plataforma Web funciona dentro de los parámetros esperados y tiene funcionalidades que permiten diversas acciones, se busca que los usuarios estén en constante contacto y notificados de las actualizaciones de la plataforma. Es sabido que los dispositivos móviles tienden cada vez más a ser más utilizados y se busca poder lanzar una aplicación *mobile*, que cuente con las funcionalidades ya existentes y esté a mayor alcance.

## Tracking donaciones

Con Ponte hay un flujo definido para las donaciones, una persona tiene algo para donar, se asigna esa donación a un voluntario el cual es el encargado de llevar la misma hasta una institución. El tiempo que tarda un voluntario en hacer el recorrido, o incluso la posición actual del mismo, no es algo que hoy por hoy se vea reflejado dentro de la aplicación y se cree que con un *tracking* de las donaciones en tiempo real se podría mejorar mucho más la experiencia del usuario y su *engagement* con la aplicación. Se piensa en incluir un mapa al estilo *Uber* y *Rappi* donde el usuario pueda ver en tiempo real cómo se va 'moviendo' y 'viajando' esta donación desde el usuario hasta la institución. Esto aumentaría la confianza de los usuarios al ver en tiempo real que efectivamente su bien está 'viajando' a donde fue pactado.

## Geolocalización

El público al que apunta la aplicación viaja todos los días hacia sus respectivos trabajos y oficinas. Algunos viajan grandes distancias durante varias horas y es parte de la rutina. Es por esto que se consideró que sería ideal aprovechar estos viajes rutinarios de los usuarios.

¿Por qué? Porque se cree que se puede mejorar exponencialmente más la dinámica de las donaciones. A veces hacerse el tiempo para manejar o viajar hasta un lugar fuera de la rutina se hace difícil y sería beneficioso que en la rutina de una persona está el viajar todos los días de Morón a Palermo por trabajo y alguien de Morón tiene algo para donar por

Palermo, poder aprovechar el viaje para llevar esa donación durante el día laboral. No habría necesidad de asignar las donaciones manualmente como se venía haciendo.

La parte más difícil de este proceso es optimizar al máximo la ubicación de los voluntarios y la ubicación de las donaciones que se les asigna.

## Programa de fidelización

Se considera importante fomentar el uso de la aplicación y darle una mirada más del tipo juego o red social. Donde el usuario tenga que ir completando su perfil y coleccionando distintas medallas y cumplir distintos objetivos. Es un modelo similar al que implementa *Mercadolibre* hoy en día. Por ejemplo el usuario tendría los siguientes desafíos que iría coleccionando una vez realizados y generando así un sentimiento de pertenencia y superación.

- Hacé tu primera donación!
- Convertite en voluntario
- Transporta tu primer donación
- Hiciste 5 donaciones

## Optimización de la Base de Datos y Backoffice

Las diversas migraciones en el área del *BackEnd* lograron solucionar y pulir varios mecanismos para que el proceso por el cual una donación sale de un usuario donante y llega a una institución, sea claro y efectivo. Sin embargo, con la fecha de entrega aproximándose y la poca experiencia inicial por parte del equipo en esta área, hicieron que queden funcionalidades por eficientizar.

Se considera al alcance la mejora de las herramientas utilizadas para 'levantar' el *back end*. Un ejemplo de mejoras es la implementación de *tokens* de *sesión*.