

## **INFORMATARIO PROYECTO: DONANTES DE PLASMA**

### **Autores:**

- ARAUJO ITATI
- FERNANDEZ IAN
- FERNANDEZ JUAN MANUEL
- FLAUTT FACUNDO
- GIMENEZ IVANA
- GONZALEZ FACUNDO
- JUAREZ JUAN JOSE

**Año: 2020**

# 1 CAPÍTULO I

## 1.1 Introducción

El presente proyecto se redacta con carácter de Trabajo Campo el curso del Informatario el tema DONACIÓN DE PLASMA. Nuestro proyecto al que llamaremos ***“Juntos contra el Covid”***, *este eslogan, por la presente situación que estamos atravesando; pretende* colaborar en forma de herramienta informática, con el propósito de poder agilizar la donación de plasma para poder generar anticuerpos para el COVID19.

Para esta primera parte, como alumnos del Informatario lo que implementamos como primera instancia fue contactarnos con un médico mediante la aplicación WhatsApp, donde a través de un audio nos explicó cómo es el procedimiento de la extracción de sangre y cuáles los procedimientos para poder llegar a lo que conocemos como “plasma” que son los generadores de anticuerpos, como así también nos habló acerca de quiénes son los posibles candidatos para poder donarlos.

## 1.2 Objetivo

Diseñar e implementar una herramienta de software que ayude tanto al personal de la salud a una selección más efectiva y rápida para la donación de plasma y al usuario para poder registrarse de manera virtual.

### 1.2.1 Objetivos específicos

Proporcionar una interfaz amigable para la registración de los usuarios.

Proporcionar información referente a la donación de plasma.

Proporciona una geolocalización para poder asistir a los lugares más cercanos de los centros de salud donde se pueda realizar la donación del plasma.

Proporcionar un sector de noticias.

## 1.3 Fundamentación

Durante el cursado notamos la necesidad de realizar más ejercicios para fijar nuestros conocimientos sobre los temas. Por eso surge la iniciativa del diseño de una herramienta de software que ayude tanto al personal de salud como a las personas que deseen ser donantes para ayudar a combatir el COVID-19.

#### **1.4 Breve estado del arte**

Debido a la pandemia y a la poca información o herramientas que solventen estas problemáticas, nos vimos en la necesidad de investigar y ahondar sobre el tema, encontramos un Sitio Web <https://portal-coronavirus.gba.gob.ar/donaplasma/inicio> que nos ayudó a plasmar mejor nuestras ideas.

El mismo se está implementando en la provincia de Buenos Aires y notamos que en nuestra provincia no contamos con esta herramienta.

## CAPÍTULO II

### 2 Especificación de Requerimientos de Software

#### 2.1 Documento de Especificación de Requisitos Sistema.

##### 2.1.1. Propósito (ERS 1)

El software proporcionará a los chaqueños una herramienta que le permita una rápida recolección de personas que puedan ser posibles candidatos a la donación de plasma.

Este documento está destinado al personal de salud, a los docentes del Informatorio y a todas las personas que deseen colaborar con este proyecto.

##### 2.1.3 Alcance (ERS 1.2)

De aquí en adelante nos referiremos al software como **Juntos contra el covid**.

Juntos contra el Covid, proporciona noticias para el personal médico y demás personas que accedan a él.

Juntos contra el Covid, contará con un Login para la registración de los usuarios.

Juntos contra el Covid, recolectar información acerca de cómo los usuarios son o no aptos para la donación y la cantidad de veces que pueden hacerlo con tiempo prudencial.

Juntos contra el Covid, deberá ser una herramienta Open Source (código abierto), disponible para todo aquel que desee colaborar con el desarrollo de la herramienta.

Juntos contra el Covid, deberá estar distribuido bajo la licencia MIT.<sup>1</sup>

##### 2.1.4 Definiciones, Siglas y Abreviaturas (ERS 1.3)

Tabla 1: Definiciones, Siglas y Abreviaturas

SIGLA	Descripción
Juntos contra el covid	<b>Con este eslogan, queremos incentivar a las personas que colaboren a tomar conciencia sobre el alcance que tomo esta pandemia a nivel mundial.</b>
ERS	Especialización de Requerimientos de Software
IEEE	Institute of Electrical and Electronics Engineers/ El Instituto de Ingeniería Eléctrica y Electrónica
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
IT#	Riesgo de Tecnología

<sup>1</sup> Vea el documento: Licencia MIT en el anexo 1 (Documentos).

!P#	Riesgo de Personal
!O#	Riesgo de Organizacional
!H#	Riesgo de Herramientas
!R#	Riesgo de Requerimientos
XP	Extreme development o Programación extrema, es un ciclo de vida de proyecto.
BD	Base de Datos
SGBD	Sistema de Gestión de Base de Datos
DLL	Data Definition Language
DML	Data Manipulation/Management Language
MVC	Modelo Vista-Controlador
EEE	Eficiencia de la eliminación de Errores

## **2.2 Descripción global**

### **2.2.1 Perspectiva del producto**

Los EVEAS, han tomado un gran protagonismo en la vida de los ciudadanos y la nueva normalidad.

Podemos destacar algunos EVEA como Anses, bancos, Afip, moodle, udemy, treeHouse y SourceCode. Juntos contra el covid, no pretende replicar lo ya se ha hecho por estas plataformas; sino más bien pretende ser un software para uso específico de los chaqueños y que la gente tome conciencia de la necesidad de poder donar plasma para poder generar los anticuerpos en aquellas personas que se encuentren inmersos bajo esta enfermedad.

### **2.2.2 Funciones del producto**

Juntos contra el Covid, proveerá soporte al personal de salud para una rápida y eficaz selección de los candidatos a donar.

### **2.2.3 Características del usuario**

La herramienta cuenta con dos tipos de usuario y un colaborador.

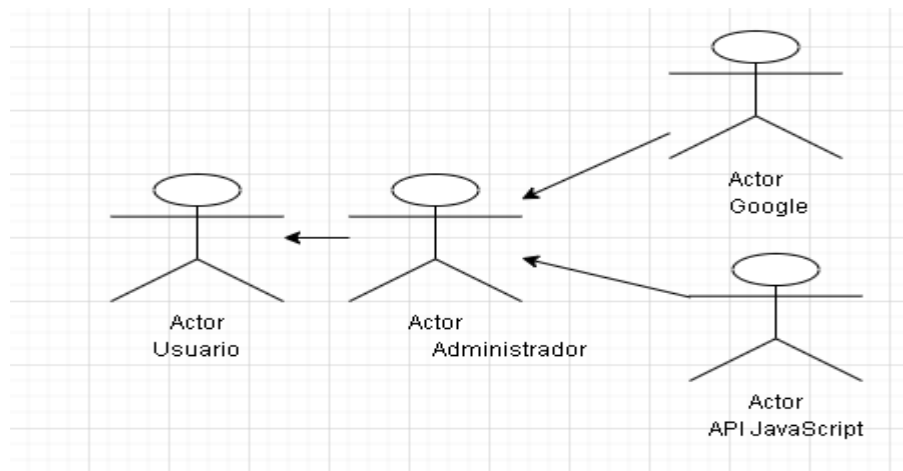


Fig. 1: Usuarios del software

#### 2.2.3.1 Usuario

Es un tipo de usuario del sistema debe identificarse y para ello registrarse para recolectar la información.

#### 2.2.3.2 Administrador

Es un tipo de usuario del sistema que colabora con el desarrollo de funcionalidades o aporta nuevas ideas.

#### 2.2.3.3 Google – API JavaScript

Este actor colabora con la extensión de Google Maps, además de eso dispone de Maps JavaScript API que permite visualizar, y publicar nuestros mapas en la web.

## 2.3 Los requisitos específicos

### Requerimientos No Funcionales

Tabla 2: Requerimientos No Funcionales

Id	Tipo	Descripción	Prioridad
RNF1	Organización	El Formulario IEEE 830 deberá ser adjuntado a la documentación del proyecto.	ALTA
RNF2	Proyecto	La herramienta deberá ser Open Source.	ALTA
RNF3	Proyecto	El código de la herramienta deberá estar disponible para los interesados que deseen acceder al mismo en la plataforma de desarrollo colaborativo GitHub. <sup>2</sup>	ALTA

<sup>2</sup> Código fuente del proyecto Juntos contra el covid disponible en: GitHub:  
<https://github.com/juanmfernandez/plasma-info>

RNF4	Organización	Durante un periodo no inferior a 2 años, los miembros de la organización deberán dar soporte a los clientes y usuario.	MEDIA
RNF5	Producto	La interfaz gráfica del software deberá desarrollarse con los lenguajes CSS, HTML, JavaScript.	MEDIA
RNF6	Producto	El software deberá poder ser ejecutado en la última y en la penúltima versión de Google Chrome para pc para en Windows y Linux.	MEDIA
RNF7	Producto	El software deberá poder ser ejecutado en la última y en la penúltima versión de Google Chrome para móvil en android.	MEDIA
RNF8	Producto	El software se deberá desarrollar de manera que se favorezca y facilite: <ul style="list-style-type: none"> <li>• La comprensión de su código,</li> <li>• Su arquitectura</li> <li>• Su mejora por parte de los usuarios.</li> </ul>	ALTA

## Requerimientos Funcionales

*Tabla 3. Requerimientos Funcionales*

ID	Descripción	Prioridad
RF1	El software deberá poder ser accedido desde Google Chrome.	ALTA
RF2	Deberá contar con una sección de deberá contar con un Login de registración.	ALTA
RF3	El usuario deberá poder ingresar su nombre para que la herramienta lo identifique.	MEDIA

## 2.4 Interfaces externas

### 2.4.1 Interfaces de usuario

El usuario tendrá acceso a la herramienta a través del navegador Web google Chrome en su última versión disponible en el mercado y en las dos previas.

### 2.4.2 Interfaces Hardware

La infraestructura de hardware necesaria para acceder a la herramienta estarán permanentemente ligadas a los requisitos hardware del navegador Google Chrome<sup>3</sup>.

<sup>3</sup> Vea el documento: Requisitos para la instalación de google chrome del Anexo 1 Documentos.

## 2.5. Entorno

A continuación se describe el entorno de la herramienta con un diagrama de contexto.

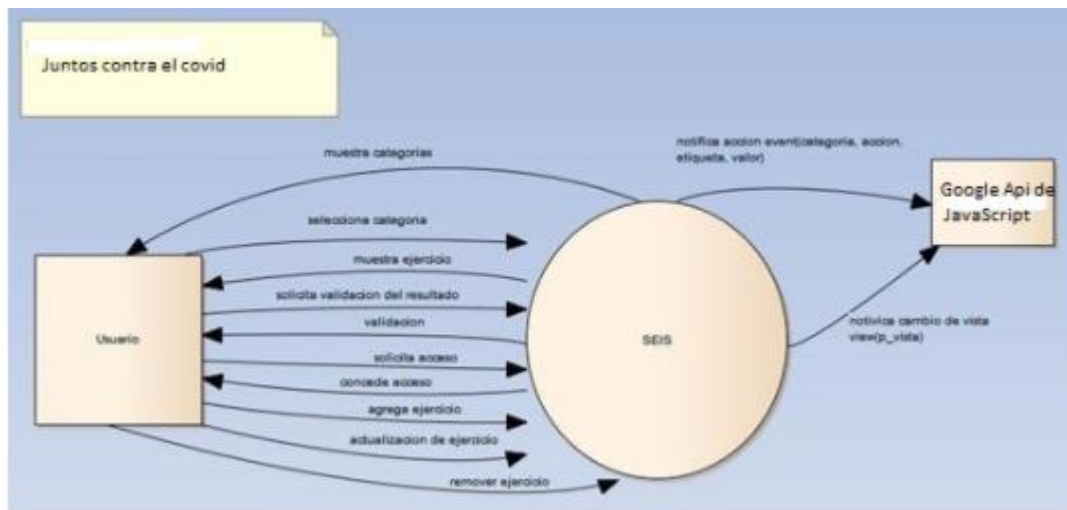


Fig 2: Entorno de la herramienta con un diagrama de contexto

## 2.6 Funciones

Este software proveerá una rápida y eficaz selección de los posibles candidatos a donar, con la facilidad no hacerlo de manera virtual.

### 2.7 Requisitos del desarrollo

- Se desarrollarán un Sitio Web que nos permita una registración tanto de los colaboradores como de los usuarios que lo ocuparán. Donde en la registración se solicitan algunos datos personales que necesitara el personal de salud para poder realizar la selección más apropiada de los candidatos a donar su plasma.
- Además de proporcionar un servicio de Geolocalización, que permite saber qué centro de salud es el más cercano de acuerdo a su ubicación.

#### 2.7.1 Requerimientos Funcionales:

El sistema debe verificar el usuario que ingresa al aplicativo.

El sistema debe verificar la Geolocalización del Usuario.

#### 2.7.2 Requerimientos No Funcionales

##### 2.7.2.1 Requerimientos del Producto

La interfaz deberá ser amigable e intuitiva.



La funcionalidad de la aplicación no deberá ir en contra de los protocolos establecidos por el ministerio de salud.

#### **2.7.2.2 Requerimientos Organizacionales**

Se realizará una primera entrega del proyecto para presentar avances el día 02 de Octubre de 2020.

La capacitación del sistema se deberá realizar en un tiempo no mayor a 7 días.

#### **2.7.2.3 Requisitos del banco de datos lógicos**

##### **BD:**

Un lenguaje de definición de esquema conceptual.

Un sistema de diccionario de datos.

Una estructura simétrica de almacenamiento.

Un lenguaje de consulta de propósito general.

##### **SGBD:**

La manipulación de los datos de manera concisa.

Buena definición de estructura para el almacenamiento de datos.

Además, los SGBD deben incorporar como herramienta fundamental dos tipos de lenguajes: uno para la definición de los datos, y otro para la manipulación de los mismos. El primero se denomina DLL y es el que provee de los medios necesarios para definir los datos con precisión, especificando las distintas estructuras. El segundo se conoce como DML y es el facilitar a los usuarios el acceso y manipulación de los datos.

## **2.8 Restricciones del diseño**

Interfaz acorde a diseños preestablecidos en el prototipo en el apartado de educación de requerimientos desarrollada con la tecnología Web de avanzada al momento de iniciar el desarrollo con el fin de maximizar la vida útil de la herramienta.

La herramienta deberá tener una clara división de capas, que facilite su mantenimiento y mejora y se deberá fomentar esta característica por encima de todas las demás.

### **2.8.1 Atributos del software del sistema**

#### **Rendimiento**

Garantizar que las consultas recurrentes a la base de datos no afecten el desempeño ni la integridad del sistema.

Establecer redundancia en los servidores favoreciendo el tráfico de datos por las redes informáticas y reduciendo el esfuerzo de los servidores.

#### **Seguridad**

Garantizar el acceso a la información y a opciones aplicables a dicha información por medio de niveles de seguridad de cuentas, requiriendo usuarios y contraseñas de acceso.

#### **Fiabilidad**

Garantizar que el sistema funcione de manera adecuada durante un periodo de tiempo extenso o según el tiempo de vida que se le asigne al sistema.

Mantener las interfaces y funciones del sistema según lo establecido y ya conocido.

#### **Disponibilidad**

La disponibilidad del sistema debe ser continua con un porcentaje de tiempo de servicio mayor o igual al 99%, para esto se deberá contar con un plan de contingencia ante errores que permita mantener activos los servicios sin afectar a los usuarios.

#### **Mantenibilidad**

El sistema debe disponer de una documentación fácilmente actualizable que permita realizar operaciones de mantenimiento.

La interfaz debe estar complementada con un soporte de ayuda.

#### **Portabilidad**

El sistema tiene una portabilidad de las más altas, puede funcionar en cualquier plataforma operativa que cuente con MS Windows, Linux, Android o IOS, en su última, o penúltima versión. Además se adapta a todas las resoluciones de pantalla, bajo el estándar impuesto por empresa Twitter en la tecnología Bootstrap.

## **2.9 Análisis de Riesgos de Proyecto.**

### **2.9.1 Introducción**

En el orden de minimizar el impacto de la ocurrencia de los eventos adversos a continuación se hará un listado que representan acontecimientos desfavorables para el desarrollo del software, se desarrollará en este apartado las respectivas estrategias para minimizar su incidencia en este proyecto.

### **2.9.2 Propósito**

El objetivo de este apartado es emprender acciones en caso de que un evento futuro pueda ocasionar un perjuicio.

### **2.9.3 Alcance**

Se enumeran la lista de los posibles riesgos que pueden aparecer:

- i) Riesgos de Tecnología.
- ii) Riesgos de Personal.
- iii) Riesgos Organizacionales.
- iv) Riesgos de Herramientas.
- v) Riesgos de Requerimientos.
- vi) Riesgos de Estimación.

### **2.9.4 Metodologías**

#### **2.9.4.1 Identificación de los Riesgos.**

##### **a) Riesgos de Tecnología**

- 1) Las fallas en el hardware ocasionan pérdida del código desarrollado.
- 2) Las bases de datos no sean compatibles con el software.
- 3) Falta de energía eléctrica.
- 4) Por fallo de hardware no sea pueda guardar la información que está siendo procesada.
- 5) Fallos de red.
- 6) No disponer de medios de almacenamientos para la realización de backup.
- 7) La arquitectura de las máquinas no se adapta a los requerimientos del software propuestos.
- 8) Las medidas protectivas no satisfacen la seguridad del sistema.

**b) Riesgos de Personal**

- 1) Personal con experiencia abandona el proyecto antes de su finalización.
- 2) El personal del proyecto no cuenta con el tiempo necesario para trabajar en el proyecto.
- 3) Personal clave contrae enfermedad y se ve forzado a tomar licencia.
- 4) No hay disponibilidad de contratación de personal externo.
- 5) No hay disponibilidad de capacitación para el personal.
- 6) Falta de incentivo económico.
- 7) Condiciones de trabajo no favorables.
- 8) Baja moral del equipo.

**c) Riesgos Organizacionales**

- 1) Falta de presupuesto para llevar a cabo el software.
- 2) Reestructuración organizacional de la empresa.
- 3) Implementación de nuevas políticas de gestión organizacional.

**d) Riesgos de Herramientas**

- 1) Mala utilización de las Herramientas CASE.
- 2) Falta de integración de las Herramientas CASE.

**e) Riesgos de Requerimientos**

- 1) Interpretación errónea del analista acerca de lo que el cliente plantea.
- 2) Petición de cambios en el sistema por parte del cliente.
- 3) Falta de visión por parte del cliente sobre las modificaciones realizadas.
- 4) El cliente solicita instrucciones innecesarias para el sistema.
- 5) Falta de interés o tiempo por parte del cliente.
- 6) El mercado no acepta el producto.

**f) Riesgos de Estimación**

- 1) El tiempo requerido para desarrollar el software está subestimado.
- 2) La tasa de reparación de defectos está subestimada.
- 3) El tamaño del software está subestimado.
- 4) Mal diseño en la planificación causan demoras de implementación.
- 5) El tiempo del análisis no es considerado.

## 2.10 Ciclo de vida del producto

### 2.10.1 Introducción

Para el ciclo de vida de <Juntos contra el covid/> seleccionamos el ciclo de vida ideal de XP (Extreme Development o Programación Extrema).

### 2.10.2 Fundamentación

Nuestra selección de esta metodología se sostiene en tres pilares:

1. Se ajusta a las necesidades del entorno en el que se llevará adelante el proyecto al proveer pequeñas y sucesivas entregas apoyadas en un fluido proceso de retroalimentación y validación.
2. Consideramos que es la más adecuada para trabajar con el personal y los recursos de los que disponemos.

### 2.10.3 Etapas del ciclo de vida de XP.

A continuación, describiremos las etapas del ciclo de vida de la metodología de desarrollo extremo.

#### 2.10.3.1 Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.

#### 2.1.3.2 Historia de Usuarios

Para registrar las historias de usuario y tenerlas a disposición del cliente y los desarrolladores usamos la herramienta Stories On Board<sup>4</sup>, que nos permite mantener un registro colaborativo en la nube de las historias de usuario y además interactuar de manera remota con el cliente y sus representante. vea también: archivo adjunto *plasma-info \_ecxe\_storiesOnBoard.xls* y *plasma-info \_png\_storiesOnBoard.png*.

---

<sup>4</sup> Stories On Board es una herramienta web que provee una interface para la carga y getion de historias de usuario <https://storiesonboard.com/>



Fig. 3: Historias de usuario generadas con la herramienta: Stories On Board.

### 2.1.3.3 Herramientas

Tabla 4: Herramientas utilizadas.

Nombre	Uso	Trazabilidad
Pycharm	IDE, plataforma de desarrollo.	<a href="#">ERS-RNF10</a>
Git	Herramienta de control de versionado.	
GitHub	Comunidad de desarrollo.	<a href="#">ERS(RNF4 y RNF5)</a>
html	Lenguaje de programación.	<a href="#">ERS(RNF7)</a>
javascript	Lenguaje de programación	<a href="#">ERS(RNF7)</a>
css	para darle una apariencia amigable.	<a href="#">ERS(RNF7)</a>

### 2.1.3.4 Planificación de la Entrega (Release)

Se educa por medio de encuesta a los clientes el orden de prioridades de las historias de usuario., vea [1] y los programadores estiman el esfuerzo de cada tarea.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Estimamos que el primer release estará disponible 3 semanas luego del inicio de la fase de desarrollo.

### 2.1.3.5 Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.

#### **2.1.3.6 Producción**

La fase de producción requerirá de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase. Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación.

#### **2.1.3.7 Mantenimiento**

Mientras la primera versión se encuentra en producción, el equipo mantendrá el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

#### **2.1.3.8 Muerte del Proyecto**

Es cuando el cliente no tiene más historias para ser incluidas en el sistema.

En cuanto el sistema deje de cumplir con los objetivos una vez finalizada la pandemia.

Cuando los usuarios dejen de colaborar con el proyecto.

Cuando se encuentre la vacuna.

### **2.11 Educción de Requerimientos**

#### **2.11.1 Introducción**

Enumeramos a continuación los métodos de educación de requerimientos que utilizamos.

#### **2.11.2 Métodos Utilizados**

##### **2.11.2.1 Observación**

El proceso de educación de requerimientos inicia con la observación durante un periodo de entre 2 y 3 semanas, tiempo durante el cual los miembros del equipo de análisis lograron el entendimiento del dominio del problema.

##### **2.11.2.2 Tormenta de ideas**

Pasamos a realizar sesiones de tormenta de ideas organizada con los miembros del equipo de desarrollo, para detectar los posibles objetivos que cumpla con los requerimientos del cliente, seleccionamos el objetivo general del sistema.

### 2.11.2.3 Estudio de documentos

Del estudio de un documento que establece los contenidos que el cliente se compromete a certificar que los usuarios han adquirido con su supervisión, extrajimos los objetivos específicos.<sup>5</sup>

### 2.11.2.4 Cuestionarios

Apoyados en una serie de cuestionarios realizados a:

- El cliente.
- Los empleados de nivel gerencial.
- Los futuros usuarios.
- Los posibles beneficiarios de la implementación del sistema.

Educimos los principales requerimientos del software a desarrollar.

## 2.11.3 Validación de objetivos

Con el fin de detectar posibles inconsistencias en los requerimientos educidos<sup>6</sup> realizamos:

### 2.11.3.1 Caso de Uso

En el siguiente diagrama de caso de uso podemos ver en contexto a usuario del sistema, colaboradores y las acciones que puede realizar.

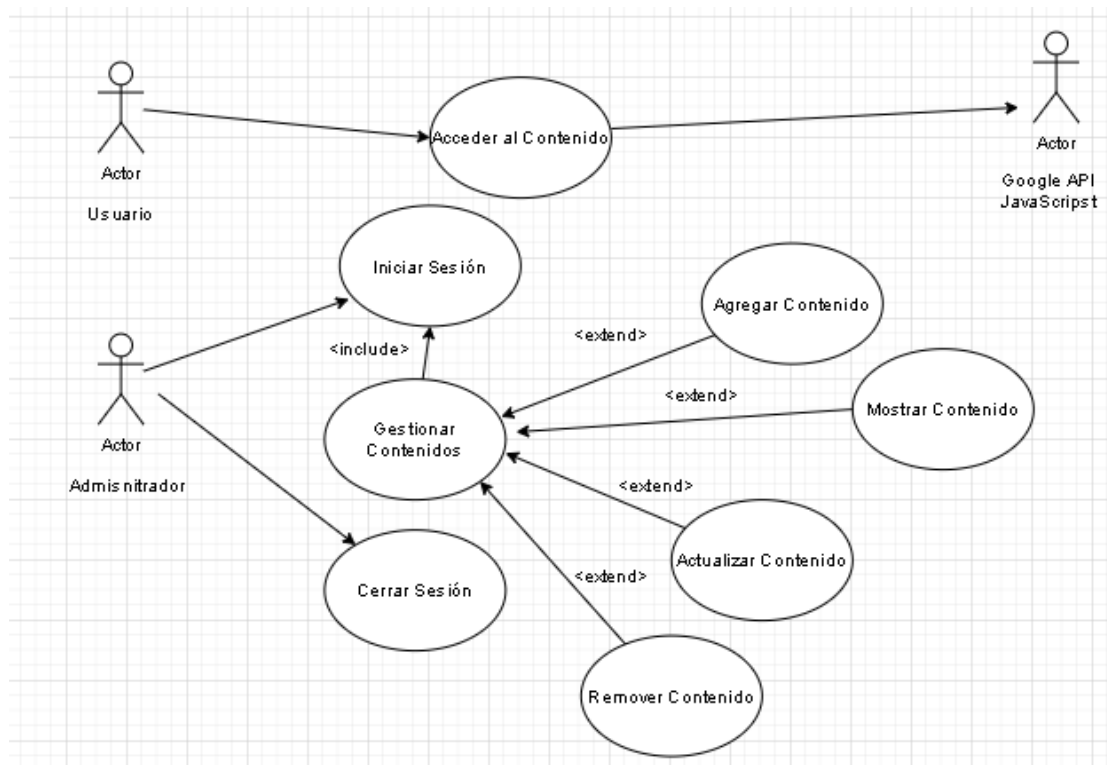


Fig. 4: Caso de Uso – Juntos contra el Covid

<sup>5</sup> Programa del Informatario.

<sup>6</sup> Vea también apartado de riesgos !R2, y !R4 (riesgos de requerimiento) y sus respectivas supervisiones en el apartado de riesgos.



### 2.11.3.2 Prototipo 1

Partiendo de un primer bosquejo realizado en una de las sesiones de tormentas de ideas, realizamos el primer prototipo de la herramienta utilizando la herramienta CASE Pencil Project-2.0.5

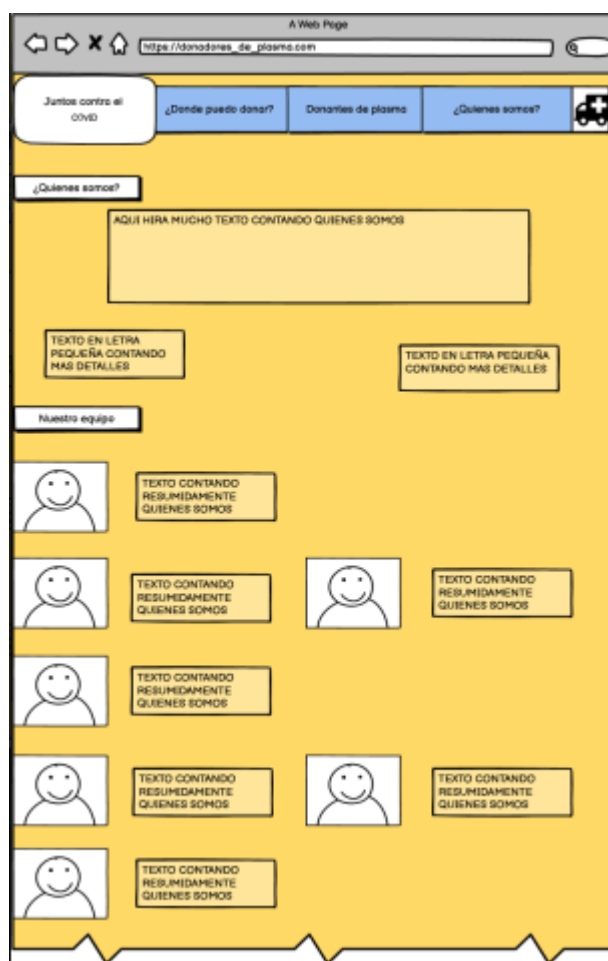


Fig. 5: Bosquejo de la interfaz gráfica – Juntos contra el Covid

### 2.11.3.3 Prototipo 2

El prototipo 2 es equivalente a un demo con alrededor del 10% de la funcionalidad total de la herramienta. Mientras el 'prototipo 1' se desarrollaron exclusivamente para apoyar el proceso de educación de requerimientos, este prototipo se desarrolló con un doble propósito:<sup>7</sup>

1. Validar los objetivos educidos en el proceso de educucción de requerimientos.<sup>8</sup>
2. El obtener la máxima precisión posible de la estimación de esfuerzo.<sup>9</sup>

<sup>7</sup> El prototipo 2 se encuentra disponible en <https://plasma-info.alwaysdata.net/>

<sup>8</sup> Vea apartado 2.2 Educucción de Requerimientos.

<sup>9</sup> Vea apartado 2.3.2 Análisis de Riesgo, riesgo clave !E3.

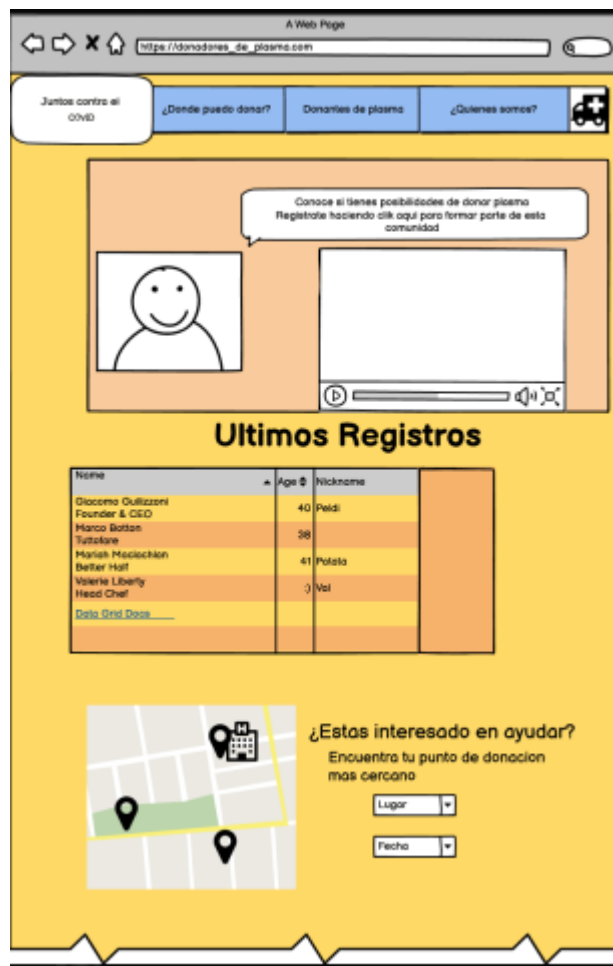


Fig. 6: Prototipo 2 - Juntos contra el Covid

#### 2.11.4 Visión General del documento

Este apartado contiene una lista de riesgo organizada según los tipos de riesgo descritos por Sommerville<sup>10</sup>.

### 2.12 Análisis de Riesgos de Proyecto.

#### 2.12.1 Introducción

En el orden de minimizar el impacto de la ocurrencia de los eventos adversos a continuación se hará un listado que representan acontecimientos desfavorables para el desarrollo del software, se desarrollará en este apartado las respectivas estrategias para minimizar su incidencia en este proyecto.

<sup>10</sup> Sommerville, Ingeniería del Software, ISBN: 84-7829-074-5, 7ma ed., Pearson Educación. S.A., Madrid, 2005, pp. 95-101.

### **2.12.2 Propósito**

El objetivo de este apartado es emprender acciones en caso de que un evento futuro pueda ocasionar un perjuicio.

### **2.12.3 Alcance**

Se enumeran la lista de los posibles riesgos que pueden aparecer:

- i) Riesgos de Tecnología.
- ii) Riesgos de Personal.
- iii) Riesgos Organizacionales.
- iv) Riesgos de Herramientas.
- v) Riesgos de Requerimientos.
- vi) Riesgos de Estimación.

### **2.12.4 Visión General del documento**

Este apartado contiene continuación presentamos una lista de los riesgos que el organizada según los tipos de riesgo descritos por Sommerville<sup>11</sup>.

### **2.12.5 Metodologías**

#### **2.12.5.1 Identificación de los Riesgos.**

##### **a)Riesgos de Tecnología**

- 1)Las fallas en el hardware ocasionan pérdida del código desarrollado.
- 2)Las bases de datos no sean compatibles con el software.
- 3)Falta de energía eléctrica.
- 4)Por fallo de hardware no sea pueda guardar la información que está siendo procesada.
- 5)Fallos de red.
- 6)No disponer de medios de almacenamientos para la realización de backup.
- 7)La arquitectura de las máquinas no se adapta a los requerimientos del software propuestos.
- 8) Las medidas protectivas no satisfacen la seguridad del sistema.

##### **b)Riesgos de Personal**

- 1)Personal con experiencia abandona el proyecto antes de su finalización.
- 2)El personal del proyecto no cuenta con el tiempo necesario para trabajar en el proyecto.

---

<sup>11</sup> Sommerville, Ingeniería del Software, ISBN: 84-7829-074-5, 7ma ed., Pearson Educación. S.A., Madrid, 2005, pp. 95-101.

- 3) Personal clave contrae enfermedad y se ve forzado a tomar licencia.
- 4) No hay disponibilidad de contratación de personal externo.
- 5) No hay disponibilidad de capacitación para el personal.
- 6) Falta de incentivo económico.
- 7) Condiciones de trabajo no favorables.
- 8) Baja moral del equipo.

#### **c) Riesgos Organizacionales**

- 1) Falta de presupuesto para llevar a cabo el software.
- 2) Reestructuración organizacional de la empresa.
- 3) Implementación de nuevas políticas de gestión organizacional.

#### **d) Riesgos de Herramientas**

- 1) Mala utilización de las Herramientas CASE.
- 2) Falta de integración de las Herramientas CASE.

#### **e) Riesgos de Requerimientos**

- 1) Interpretación errónea del analista acerca de lo que el cliente plantea.
- 2) Petición de cambios en el sistema por parte del cliente.
- 3) Falta de visión por parte del cliente sobre las modificaciones realizadas.
- 4) El cliente solicita instrucciones innecesarias para el sistema.
- 5) Falta de interés o tiempo por parte del cliente.
- 6) El mercado no acepta el producto.

#### **f) Riesgos de Estimación**

- 1) El tiempo requerido para desarrollar el software está subestimado.
- 2) La tasa de reparación de defectos está subestimada.
- 3) El tamaño del software está subestimado.
- 4) Mal diseño en la planificación causan demoras de implementación.
- 5) El tiempo del análisis no es considerado.

### **2.12.6 Análisis de Riesgos y Planificación**

*Tabla 5: Riesgos de Tecnología*

<b>Id</b>	<b>Descripción</b>	<b>probabilidad</b>	<b>nivel riesgo</b>	<b>estrategia</b>
!T1	Las fallas en el hardware ocasionan pérdida del código desarrollado.	25 %	CATASTRÓFICO	MINIMIZACIÓN

!T2	Las bases de datos no sean compatibles con el software.	20%	SERIO	PREVENCIÓN
!T3	Falta de energía eléctrica.	5%	TOLERABLE	PREVENCIÓN
!T4	Por fallo de hardware no sea pueda guardar la información que está siendo procesada.	30%	SERIO	MINIMIZACIÓN
!T5	Fallos de red.	5%	TOLERABLE	PREVENCIÓN
!T6	No disponer de medios de almacenamientos para la realización de backup.	40%	SERIO	PREVENCIÓN
!T7	La arquitectura de las máquinas no se adapta a los requerimientos del software propuestos.	15%	SERIO	PREVENCIÓN
!T8	Las medidas protectivas no satisfacen la seguridad del sistema.	30%	SERIO	PLAN DE CONTINGENCIA

*Tabla 6. Riesgos de Personal*

<b>Id</b>	<b>Descripción</b>	<b>probabilidad</b>	<b>nivel riesgo</b>	<b>estrategia</b>
!P1	Personal con experiencia abandona el proyecto antes de su finalización.	5%	SERIO	MINIMIZACIÓN
!P2	El personal del proyecto no cuenta con el tiempo necesario para trabajar en el proyecto.	70%	SERIO	MINIMIZACIÓN
!P3	Personal clave contrae enfermedad y se ve forzado a tomar licencia.	8%	SERIO	MINIMIZACIÓN
!P4	No hay disponibilidad de contratación de personal externo.	80%	SERIO	MINIMIZACIÓN
!P5	No hay disponibilidad de capacitación para el personal.	40%	TOLERABLE	PLAN DE CONTINGENCIA
!P6	Falta de incentivo económico.	75%	TOLERABLE	MINIMIZACIÓN
!P7	Condiciones de trabajo no favorables.	50%	TOLERABLE	MINIMIZACIÓN
!P8	Baja moral del equipo.	35%	TOLERABLE	MINIMIZACIÓN

*Tabla 7: Riesgos de Organización*

<b>Id</b>	<b>Descripción</b>	<b>probabilidad</b>	<b>nivel riesgo</b>	<b>estrategia</b>
!O1	Falta de presupuesto para llevar a cabo el software.	50%	CATASTRÓFICO	PLAN DE CONTINGENCIA
!O2	Reestructuración organizacional de la organización.	1%	SERIO	PLAN DE CONTINGENCIA

!O3	Implementación de nuevas políticas de gestión organizacional.	37%	TOLERABLE	PLAN DE CONTINGENCIA
-----	---	-----	-----------	----------------------

*Tabla 8: Riesgos de Herramienta*

Id	Descripción	Probabilidad	Nivel riesgo	Estrategia
!H1	Mala utilización de las Herramientas CASE.	45%	SERIO	PREVENCIÓN
!H2	Falta de integración de las Herramientas CASE.	8%	TOLERABLE	MINIMIZACIÓN

*Tabla 9: Riesgos de Requerimiento*

Id	Descripción	Probabilidad	Nivel riesgo	Estrategia
!R1	Interpretación errónea del analista acerca de lo que el cliente plantea.	50%	SERIO	PLAN DE CONTINGENCIA
!R2	Petición de cambios en el sistema por parte del cliente.	85%	TOLERABLE	PLAN DE CONTINGENCIA
!R3	Falta de visión por parte del cliente sobre las modificaciones realizadas.	90%	SERIO	PLAN DE CONTINGENCIA
!R4	El cliente solicita instrucciones innecesarias para el sistema.	75%	TOLERABLE	PLAN DE CONTINGENCIA
!R5	Falta de interés o tiempo por parte del cliente.	32%	INSIGNIFICANTE	PLAN DE CONTINGENCIA
!R6	El mercado no acepta el producto.	7%	SERIO	MINIMIZACIÓN

*Tabla 10: Riesgos de Estimación*

Id	Descripción	Probabilidad	Nivel Riesgo	Estrategia
!E1	El tiempo requerido para desarrollar el software está subestimado.	95%	SERIO	MINIMIZACIÓN
!E2	La tasa de reparación de defectos esta subestimada.	85%	SERIO	MINIMIZACIÓN
!E3	El tamaño del software está subestimado.	50%	SERIO	PREVENCIÓN
!E4	Mal diseño en la planificación causan demoras de implementación.	98%	TOLERABLE	MINIMIZACIÓN
!E5	El tiempo del análisis no es considerado.	75%	TOLERABLE	MINIMIZACIÓN

#### **2.12.6.1 Supervisión de riesgos clave**

**RIESGOS: !T1** la estrategia para supervisar este riesgo es la Prevención, el [RNF#5](#) establece la necesidad el uso de un sistema de versiones(git versión control), para evitar este riesgo.

**RIESGOS: !T3** La estrategia para supervisar este riesgo es la PREVENCIÓN, donde se propone proveer a las máquinas UPS, para evitar que se vean afectadas por los cortes de luz o niveles de tensión.

**RIESGOS: !T4** la estrategia para supervisar este riesgo es la MINIMIZACIÓN , realizar copias de seguridad cada 30 segundos de la información que está siendo procesada cuando se detecta un alto tráfico en la red.

**RIESGOS: !T5** la estrategia para supervisar este riesgo es la PREVENCIÓN, trabajar en sistemas distribuidos para evitar una colisión.

**RIESGOS: !P1** la estrategia para supervisar este riesgo es la MINIMIZACIÓN, se debe plantear en el contrato una penalización por el abandono del proyecto.

**RIESGOS: !P3** la estrategia para supervisar este riesgo es la MINIMIZACIÓN, se debe prever esta situación y reorganizar al equipo de forma que haya solapamiento en el trabajo.

**RIESGOS: !O1** la estrategia para supervisar este riesgo es el PLAN DE CONTINGENCIA, preparar un documento para el gestor principal que muestre las ventajas del proyecto y se pueda plantear una refinanciación.

**RIESGOS: !E1** la estrategia para supervisar este riesgo es la MINIMIZACIÓN, al diseñar la planificación se debe incluir más tiempo de lo que se estipula que llevará el desarrollo del proyecto por cualquier imprevisto.

**RIESGOS: !E3** Con el fin de estimar con la máxima confiabilidad el tamaño del software a desarrollar, se desarrollara un demo, con no más del 10% de la funcionalidad y se pondera su tamaño para obtener un estimado del total de líneas de código a desarrollar.

RIESGOS: !E4 la estrategia para supervisar este riesgo es la MINIMIZACIÓN, preparar una presentación donde se le explique al cliente de manera clara y concisa porque se demora la implementación del producto.

## 2.8 Arquitectura

Con el fin de maximizar la escalabilidad y disponibilidad de la herramienta y minimizar los costos se decidió utilizar como arquitectura principal la arquitectura: *cliente-servidor*.

*Para esto se dividió el desarrollo de la herramienta en dos partes:*

1. *La parte que se ejecuta del lado de cliente/usuario a la que nos referiremos de aquí en adelante como plasma-info\_frontend.*
2. *La parte que se ejecuta en la nube a la que nos referiremos como plasma-info\_backend.*

## 2.9 Planificación de tareas

### 2.9.1 Planificación de la etapa de análisis

A continuación se puede ver un resumen de la planificación de tareas de la etapa de análisis y la una lista de los recursos.

NOMBRE	DURACION	NOMBRE DEL RECURSO
ANALISIS	5 DIAS	ITATI-IVANA
PROPOSITO DEL SISTEMA	3 DIAS	JUAN
OBJETIVOS	2 DIAS	JUAN
FUNDAMENTACION	6 DIAS	FACUNDO G
PLANIFICACION DE RIESGOS	2 DIAS	FACUNDO
EDUCCION DE REQUERIMIENTOS	3 DIAS	FACUNDO
CUESTIONARIO	4 DIAS	IAN
PROTOTIPADO	5 DIAS	IAN
ESPECIFICACION DE REQUERIMIENTO DE SOFTWARE	6 DIAS	ITATI-IVANA
CICLO DE VIDA	7 DIAS	FACUNDO G
HISTORIA DE USUARIO	1DIAS	JUANJO
ESTIMACIONES DE COSTO	3 DIAS	JUANJO

*Fig. 8.1: Planificación de Actividades*



## CAPÍTULO III

### 3.1 Herramientas y/o lenguajes de programación

#### 3.1.1 Herramientas Utilizadas

**Sparks Enterprise Architect v10**

Herramienta de modelado.

Usamos esta herramienta para generar todos los diagramas en el documento.

URL: <http://www.sparxsystems.com.ar/>

**Draw.io:**

Herramienta de gestión de proyectos.

Usamos este software para planificar la etapa de análisis de nuestro proyecto.

URL <https://app.diagrams.net>

**Stories on board:**

Herramienta para la gestión de historias de usuario.

Usamos esta herramienta para gestionar las historias de usuario que educimos en la etapa de educación. y posteriormente la usaremos para ordenar los futuros releases de la aplicación.

URL: <https://app.storiesonboard.com>

**Google drive:**

Herramienta para el desarrollo colaborativo de documentos.

Utilizamos esta herramienta para mantener los archivos del proyecto a disposición de todos los miembros del equipo.

URL: [https://www.google.com/intl/es\\_ar/drive/](https://www.google.com/intl/es_ar/drive/)

**Git:**

Herramienta de control de versiones.

URL: <https://git.com/>

#### 3.1.2 Lenguaje de programación y marcado.

**CSS:**

Utilizamos el lenguaje de estilos en cascada para darle una apariencia amigable e intuitiva a los documentos Html.

**HTML:**

Utilizamos el HiperText Markup Language nos permitió crear las estructura básicas para la aplicación.

**Javascript:**

Utilizamos este lenguaje para desarrollar los algoritmos necesarios para crear el frontend de la aplicación.

**Python**

Utilizamos este lenguaje de programación para desarrollar el sistema de gestión de contenidos de la herramienta.

### **Django**

Utilizamos esta herramienta para favorecer la separación en capas de la aplicación.

## CAPÍTULO IV

### 4.1 Diagrama de caso de uso:

A continuación el diagrama de caso de uso que describe las distintas opciones de acción que posee los actores del sistema.

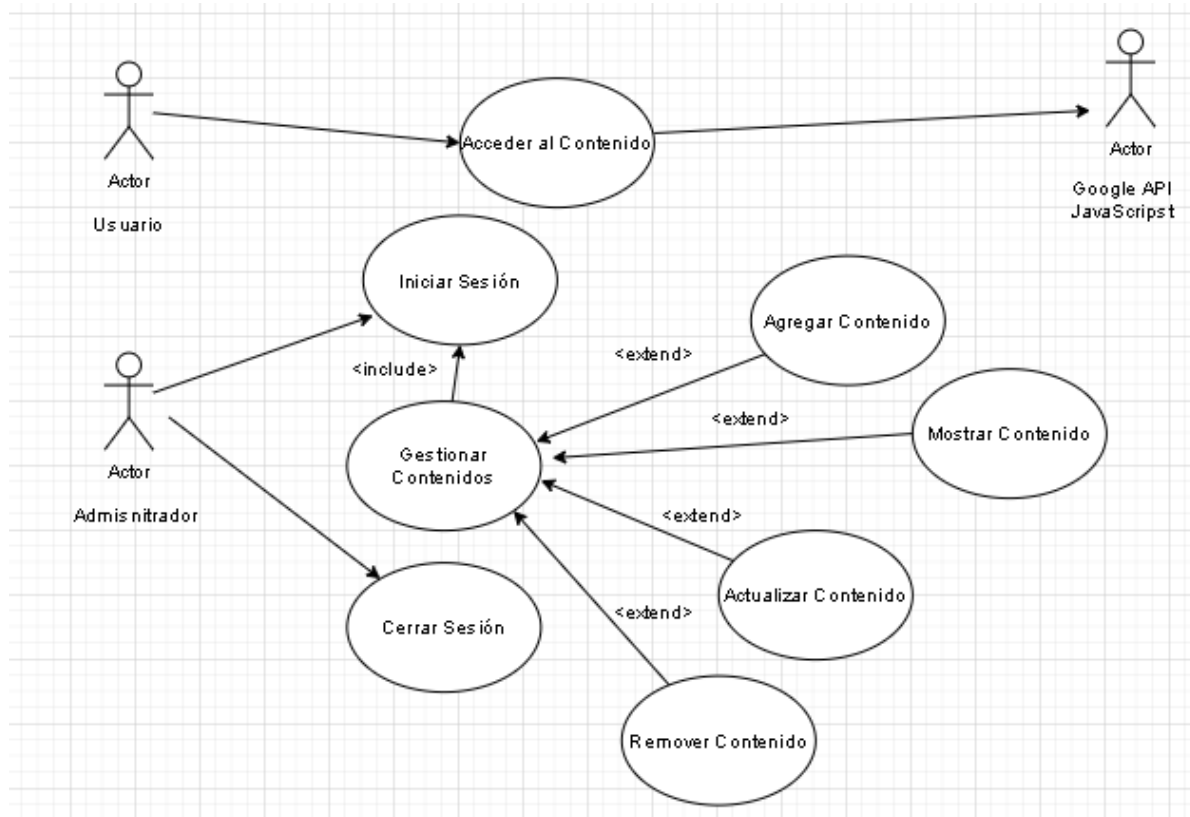
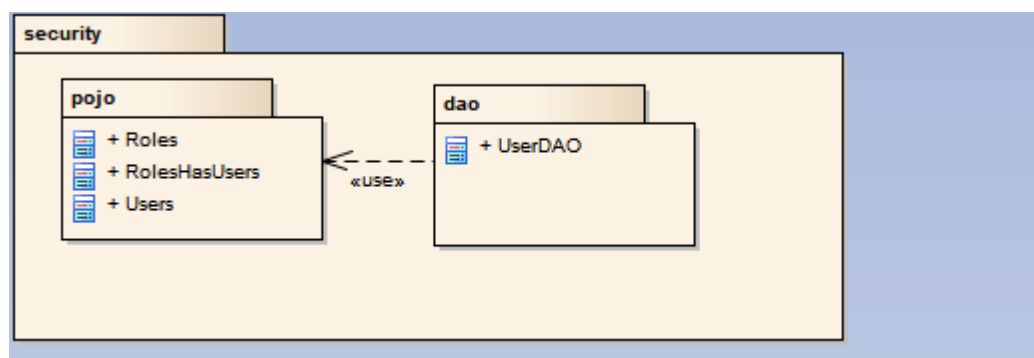


Fig. 9: Diagrama caso de uso.

### 4.2 Diagrama de Clases.

A continuación podemos ver diagramas de clases que describe la estructura básica de la herramienta a distintos niveles de abstracción.



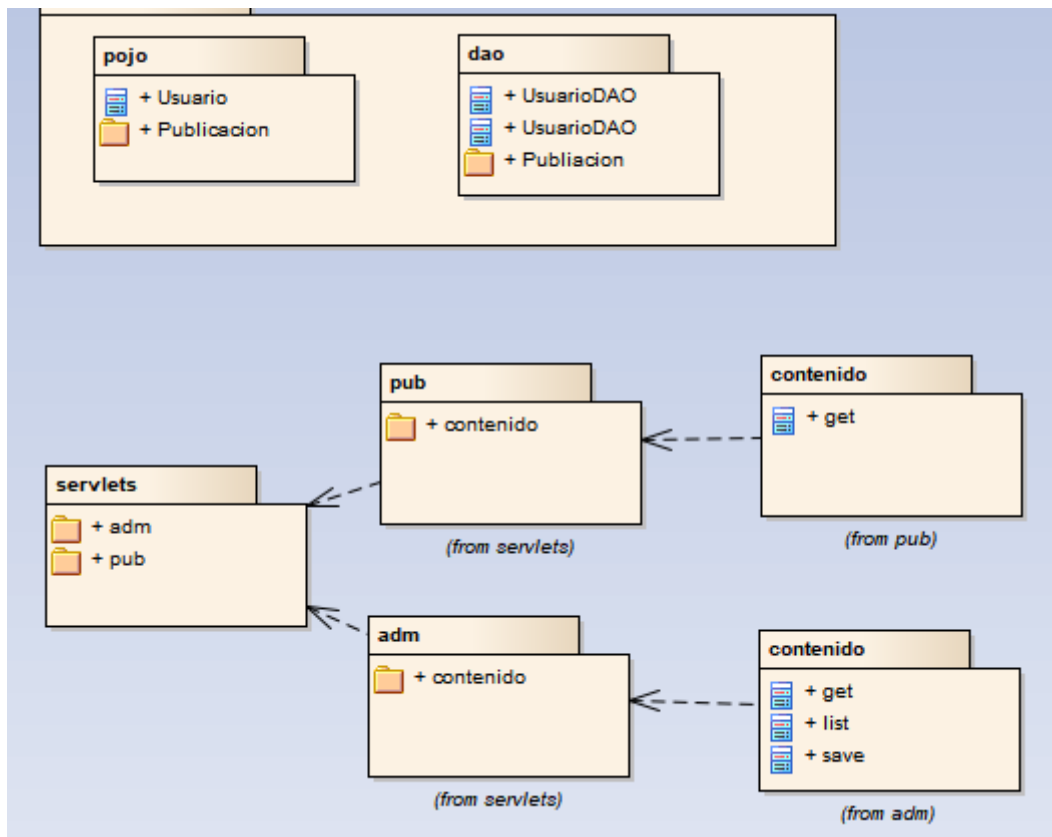


Fig. 15: Diagrama de clases que describe la estructura básica de la herramienta.

#### 4.2.1 Diagrama de Clases capa de modelo de datos.

El paquete pojos contiene clases que modelan los objetos de que forman parte de la lógica de negocios y sus correspondientes mapas de conexión con los datos de la base de datos.

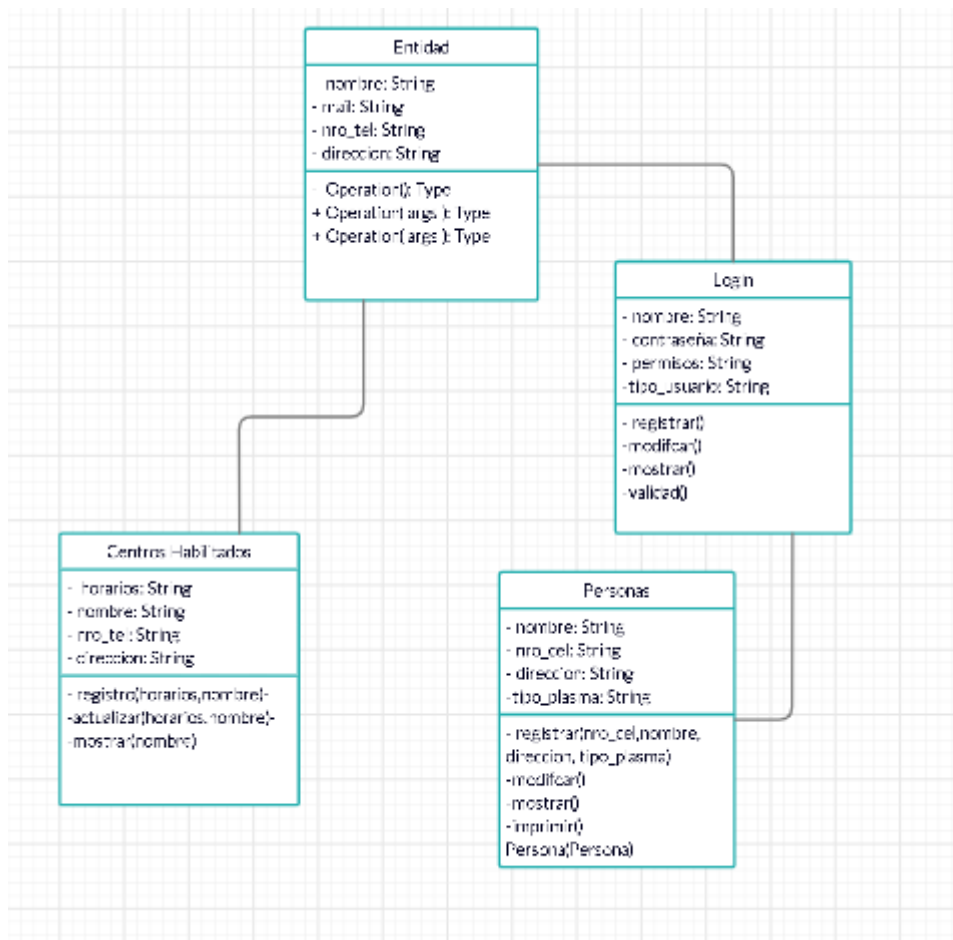


Fig. 16: Diagrama de Clases capa de modelo de datos

#### 4.2.2 Diagrama de Clases capa de la capa de seguridad.

El paquete security contiene clases necesarias para modelar la estructura de control de acceso.

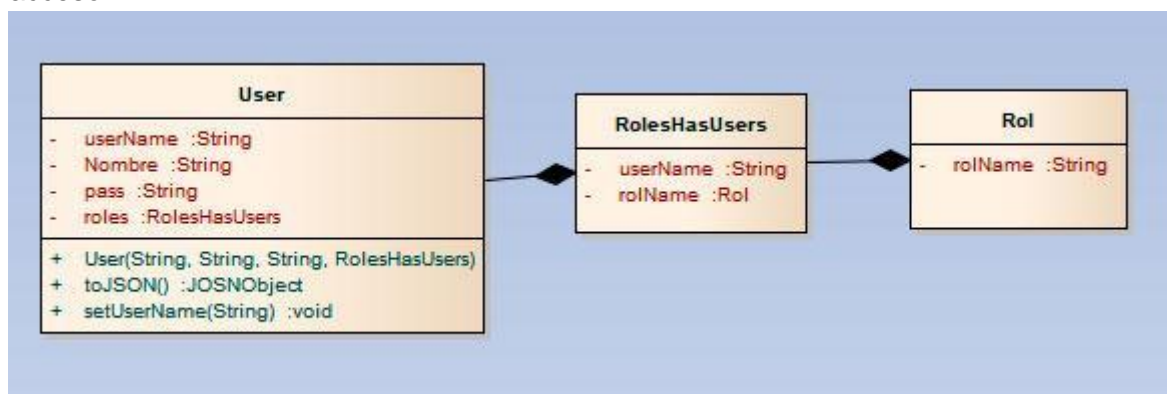


Fig 17: Diagrama de Clases, Capa de la Capa de Seguridad.

#### 4.2 Diagrama de entidad relación.

A continuación podemos ver un diagrama de de entidad relación que describe la estructura de datos desde el punto de vista lógico.

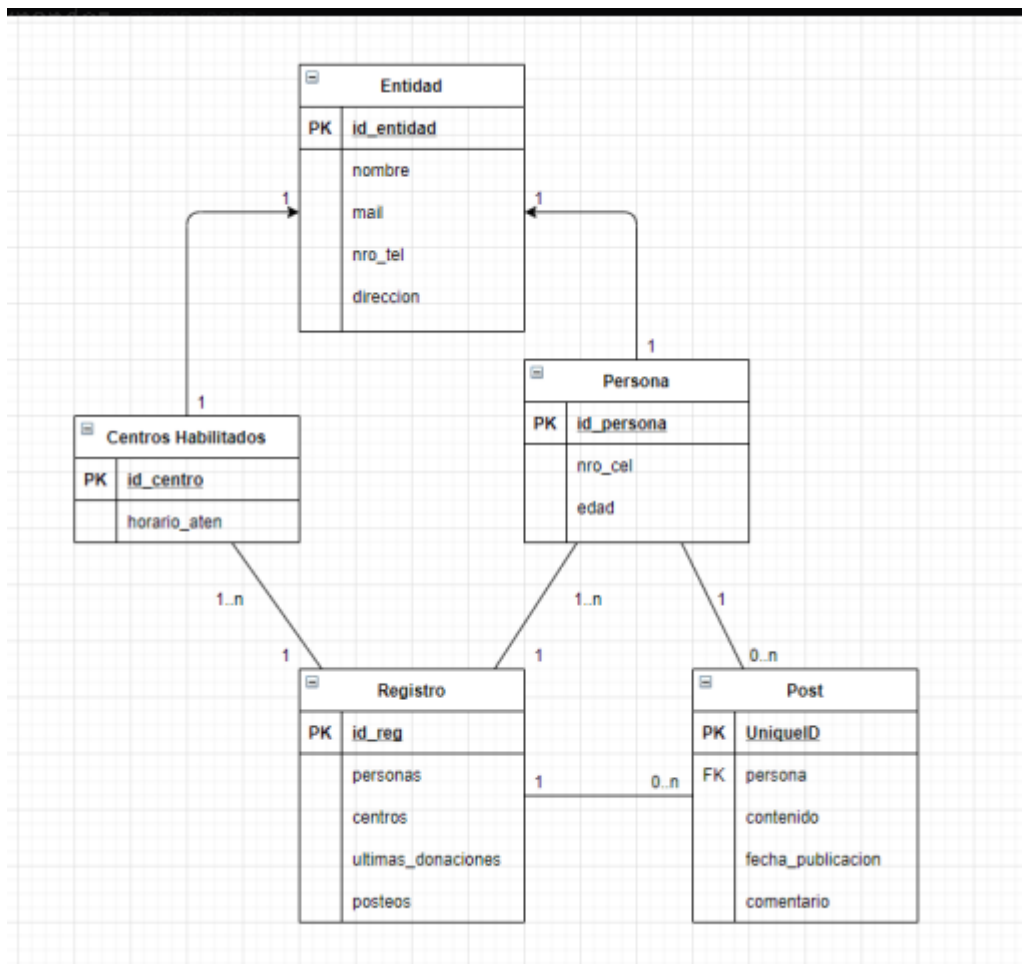


Fig 18: Diagrama de Entidad Relación.

## CAPÍTULO V

### 5.1 Resultados

#### Plan de Prueba

Id. Nombre: **Usuario Inicia Sesión.**

Versiones del caso de prueba: 1

Id caso de uso: Iniciar Sesión.

*Tabla 20: Plan de Prueba del caso de uso: Usuario inicia sesión*

<b>Caso de Prueba</b>	<b>Objetivo</b>	<b>Datos de Entrada</b>	<b>resultado Esperado</b>	<b>Resultado Objetivo</b>
1	Verificar si el sitio tiene una sesión iniciada.	Abrir el sitio.	No existe una sesión abierta, solicitar el ingreso de un usuario y contraseña	OK
2	Verificar si existe una sesión iniciada y consulta si desea el cierre de sesión.	Abrir el sitio y la respuesta es "NO".	Existe una sesión abierta, despliega un mensaje consultando el cierre de la sesión, recibe la respuesta "NO", y deja la sesión en el estado inicial.	OK
3	Verificar si existe una sesión iniciada y consulta si desea el cierre de sesión.	Abrir el sitio y la respuesta es "SI".	Existe una sesión abierta, despliega un mensaje consultando el cierre de sesión, recibe la respuesta "SI", cierra sesión abierta y solicita el ingreso del usuario y contraseña.	OK
4	Solicitar el ingreso de	Usuario y	Los datos	OK

	usuario y contraseña, y verificar si son correctos.	contraseña válidos para el sistema Usuario: juan Contraseña: 6itati123 Usuario: Pablo Contraseña: ***** Usuario:dbymatt Contraseña: ***** Usuario: iankelebold Contraseña: *****	ingresados son correctos, inicia la sesión, se modifica el estado y queda pendiente de confirmación.	
5	Solicitar el ingreso de usuario y contraseña y verificar si son correctos.	Usuario y contraseña inválidos para el sistema, "*****"	Los datos ingresados son incorrectos, se muestra un mensaje informando que no son correctos, y solicita que ingrese nuevamente los datos.	OK
6	Solicitar el ingreso de usuario y contraseña y verificar si son correctos	Usuario y contraseña inválidos para el sistema, "*****"	Los datos ingresados son incorrectos, se muestra un mensaje informando que no son correctos, y solicita que ingrese nuevamente los datos.	OK



### CURSO NORMALES:

Camino: 1-4.

Camino: 2.

Camino: 3-4.

Camino:1-5.

Camino:1-6.

### **Plan de Prueba**

Id. Nombre: **Usuario accede al contenido**

Versiones del caso de prueba: 1

Id caso de uso: Acceder al contenido.

*Tabla 21: Plan de Prueba del caso de uso: Usuario accede al contenido.*

<b>Caso de Prueba</b>	<b>Objetivo</b>	<b>Datos de Entrada</b>	<b>resultado Esperado</b>	<b>Resultado Objetivo</b>
1	Seleccionar el contenido a visualizar.	Dar un clic en la contenido a visualizar.	Visualizar las vistas de los temples.	OK
2	Seleccionar banner para navegar.	Dar un clic en la pestaña que desee visualizar.	Visualiza un mensaje: "write content"	OK

Curso Normal:

Camino: 1-2-3.

Camino: 2.

Camino:3.

### Plan de Prueba

Id. Nombre: **Usuario Genera Contenido**

Versiones del caso de prueba: 1

Id caso de uso: Generar Contenido.

*Tabla 22: Plan de Prueba del caso de uso: Generar contenido.*

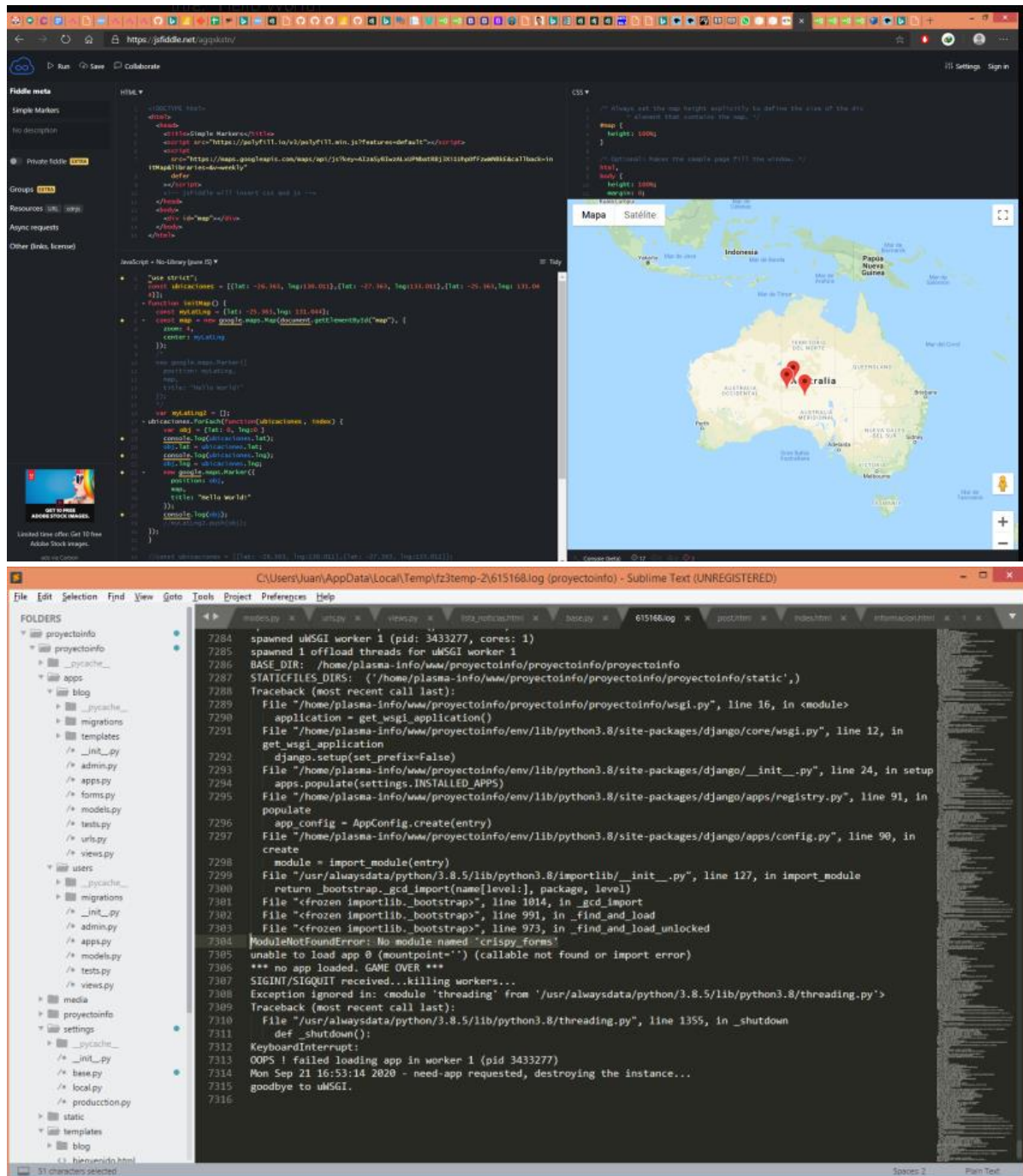
<b>Caso de Prueba</b>	<b>Objetivo</b>	<b>Datos de Entrada</b>	<b>resultado Esperado</b>	<b>Resultado Objetivo</b>
1	Seleccionar el tipo de noticia a realizar.	Dar un click en Anexar la misma.	Se despliega en una lista con las opciones disponibles.	OK
2	Seleccionar el tipo de noticia a realizar.	Dar un click en Anexar.	Se despliega la lo que anexo.	OK

Curso Normal:

Camino: 1

Camino:2

## CAPTURAS DE LA IMPLEMENTACION DEL PROYECTO



Sublime Text (UNREGISTERED) - C:\Users\Juan\AppData\Local\Temp\fb2temp-2\base.py (projectinfo) -

```
31 ALLOWED_HOSTS = ['plasma-info.alwaysdata.net', 'localhost',]
32
33
34 # Application definition
35
36 INSTALLED_APPS = [
37     'django.contrib.admin',
38     'django.contrib.auth',
39     'django.contrib.contenttypes',
40     'django.contrib.sessions',
41     'django.contrib.messages',
42     'django.contrib.staticfiles',
43     'projectinfo.apps.blog',
44     'projectinfo.apps.users',
45     #'geoposition',
46     'crispy_forms',
47     'import_export',
48     'ckeditor',
49 ]
50
51 CRISPY_TEMPLATE_PACK = 'bootstrap4'
52
53 MIDDLEWARE = [
54     'django.middleware.security.SecurityMiddleware',
55     'django.contrib.sessions.middleware.SessionMiddleware',
56     'django.middleware.common.CommonMiddleware',
57     'django.middleware.csrf.CsrfViewMiddleware',
58     'django.contrib.auth.middleware.AuthenticationMiddleware',
59     'django.contrib.messages.middleware.MessageMiddleware',
60     'django.middleware.clickjacking.XFrameOptionsMiddleware',
61 ]
```

SQLite Administrator - db.sqlite3

Base De Datos Tabla Indice Ver Trigger Consulta Datos Ayuda

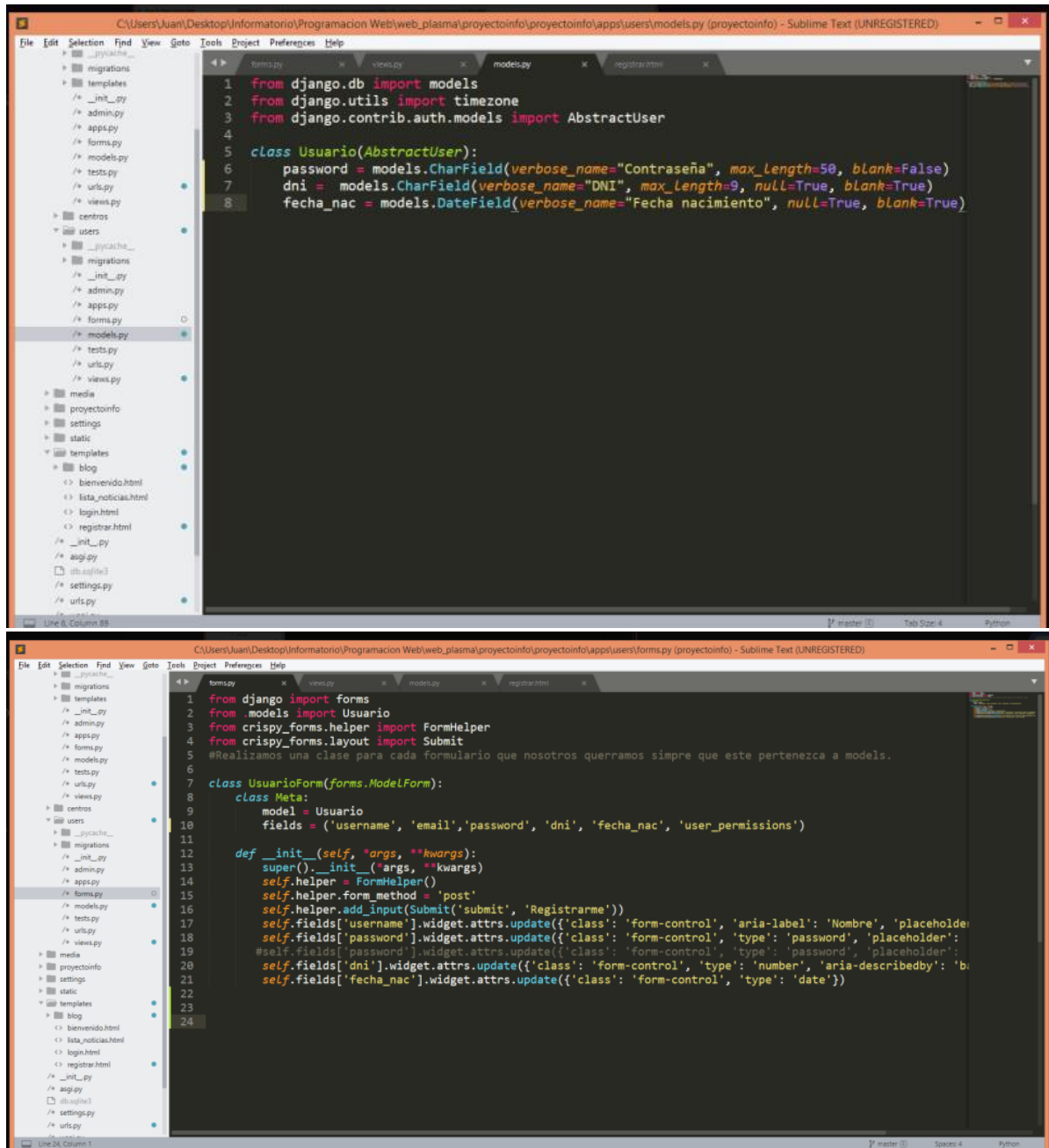
Consulta SQL Resultado Editar Datos

Tabla: users\_usuario

CampoFiltro: Filtro:

id	password	last_log	is_supe	username	first_name	last_name	email	is_staff	is_act	date_joined	dni	fecha_nac
1	pbkdf2_sha256\$18000\$KtTSVCOP287P8fQh5L357H4LET	22/09/2	<input checked="" type="checkbox"/>	juan			jmf798@gmail.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	22/09/202		
2			<input type="checkbox"/>	juanchy			jmf798@hotmail.com	<input type="checkbox"/>	<input checked="" type="checkbox"/>	22/09/202	33000001	16/05/1985
3	clave123		<input type="checkbox"/>	jhon			mail@mail.com	<input type="checkbox"/>	<input checked="" type="checkbox"/>	22/09/202	342343	16/05/1985

Base de Datos Conectada SQLite: 3.5.1 C:\Users\Juan\Desktop\Informatorio\Programacion Web\web\_plasma\projectinfo\projectinfo\db.sqlite3



## Registro

Usuario   
 Contraseña   
 Dirección de email   
 Nombres   
 \* Este campo es obligatorio.  
 Apellidos   
 \* Este campo es obligatorio.  
 DNI   
 Fecha de Nacimiento

Al registrarte, aceptas nuestras Condiciones de uso y Política de privacidad.

¿Ya tienes una cuenta? [Iniciar Sesión](#)

```

1 from django.db import models
2 from django.utils import timezone
3 from django.contrib.auth.models import AbstractUser
4
5 class Usuario(AbstractUser, models.Model):
6     first_name = models.CharField(verbose_name="Nombres", max_length=50, blank=False)
7     last_name = models.CharField(verbose_name="Apellidos", max_length=50, blank=False)
8     password = models.CharField(verbose_name="Contraseña", max_length=50, blank=False)
9     dni = models.CharField(verbose_name="DNI", max_length=9, null=True, blank=True)
10    fecha_nac = models.DateField(verbose_name="Fecha nacimiento", null=True, blank=True)
11
12    class Meta:
13        verbose_name = "Usuario"
14        verbose_name_plural = "Usuarios"
15
16    def __str__(self):
17        return "{} - {} - {}".format(self.username, self.first_name, self.last_name, self.email, self.dni)

```





Welcome

Login

Juntos contra el covid   ¿Donde puedo donar?

Donantes de plasma   ¿Quiénes somos?   Blog



Agregar Lugar



## Lista de lugares

**Hospital Perrando**

perr@andio.com

362444444

9 de julio 1944

1:00 a 23:59

[Ver mas](#)



Welcome

Login

Juntos contra el covid    ¿Donde puedo donar?

Donantes de plasma    ¿Quiénes somos?    Blog



Name\*    Mail\*

Laboratorio El Test    lab@test.gob

Phone num\*

0379888888

Location\*

-27.446615053430797,-58.97957146167755



Address\*

güemes 888

Business hours\*

8-16

Guardar

Registro

Usuario    usuario

Contraseña    contraseña    Repetir...    repetir contraseña

Email    nn@mail.com

Nombres    Nombre

Apellidos    Apellido

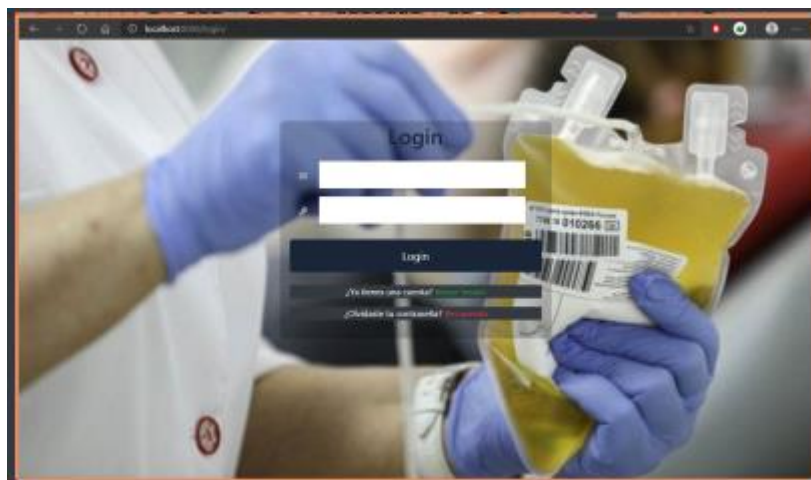
DNI   

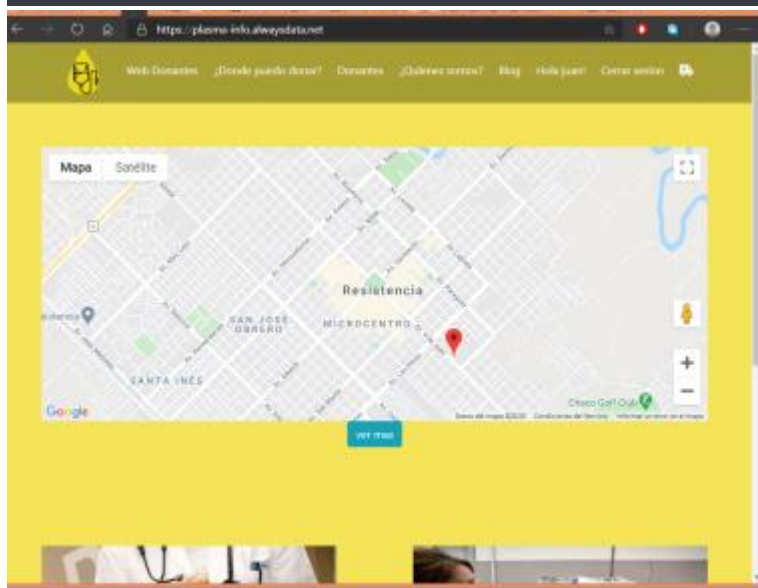
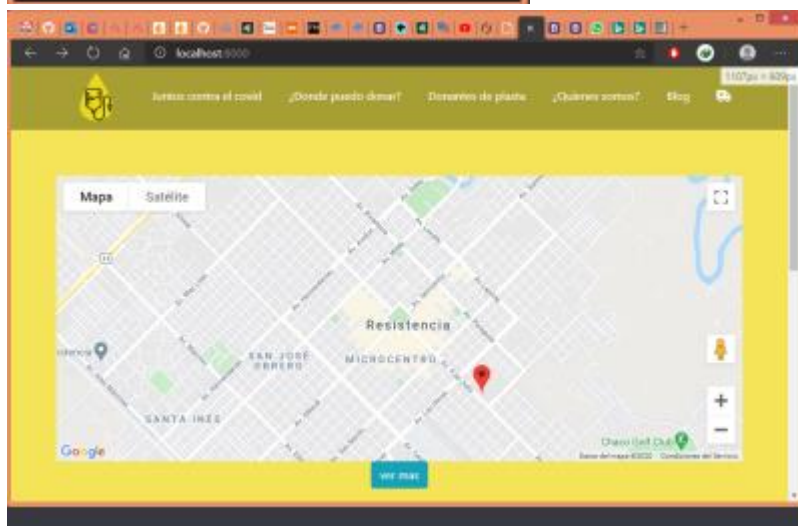
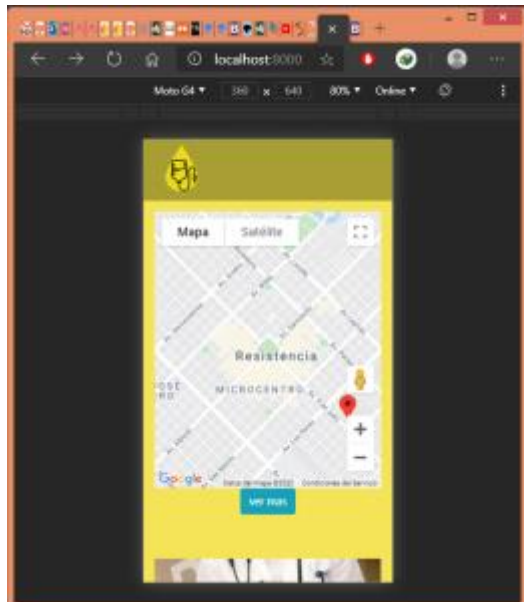
Fecha de Nacimiento   

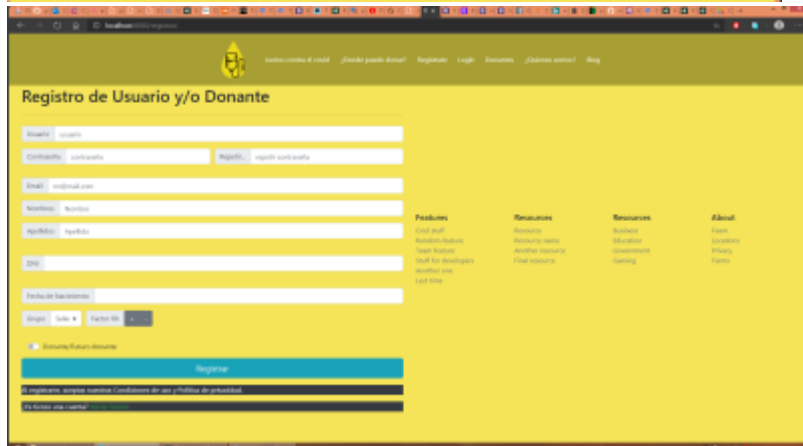
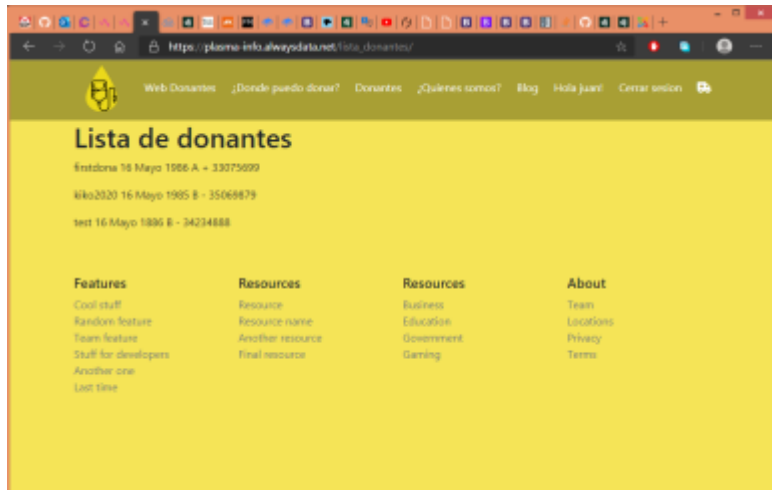
Registrar

Al registrarte, aceptas nuestras Condiciones de uso y Política de privacidad.

¿Ya tienes una cuenta? [Iniciar Sesión](#)







## CAPÍTULO VI

### 6.1 Conclusión

Al finalizar la redacción de este documentos y por consiguiente de la etapa de análisis, tanto los miembros del equipo de análisis como los del equipo de desarrollo concluimos que las dos etapas restantes del ciclo de vida del software: desarrollo/implementación y mantenimiento son posibles y el cumplimiento del cronograma propuesto para las mismas dependerá de los recursos disponibles al momento de iniciar las mismas.

Es importante destacar que encontramos que las estimación del costo de desarrollo realizadas se encuentran dentro de los plazos fijado por el marco institucional en el que se espera implantara el softwares.

Tomando como base lo que hemos logrado con el tiempo establecido por el Informatorio y también además debido al trabajo remoto al que debimos adaptarnos, ya que la situación en la que nos encontramos inmersos por esta pandemia mundial, consideramos que pudimos cumplir con el objetivo.

Como equipo nos propusimos a mejorar este Sitio Web y adaptarlo a las necesidades que vayan surgiendo con el transcurrir de nuestra nueva coteidaneidad.

El hecho de que ningún integrante haya cumplido un rol específico durante el desarrollo del presente nos permitió tener una idea acerca del trabajo que cumple un analista, un programador Backend y un programador FrontEnd. Consideramos que para el próximo proyecto, ya cada uno sabrá qué lugar ocupar y será más óptimo nuestro trabajo como así también las necesidades adicionales que surgieron en la etapa de análisis, serán consideradas para un proyecto posterior.

## ANEXO 1

### Documentos

#### Licencia de MIT

Por la presente se autoriza, de forma gratuita, a cualquier persona que haya obtenido una copia de este software y archivos de documentación asociados (el "Software"), a utilizar el Software sin restricción, incluyendo sin limitación los derechos de usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar, y/o vender copias de este Software, y permitir lo mismo a las personas a las que se les proporcione el Software, de acuerdo con las siguientes condiciones:

El aviso de copyright anterior y este aviso de permiso tendrán que ser incluidos en todas las copias o partes sustanciales del Software.

EL SOFTWARE SE ENTREGA "TAL CUAL", SIN GARANTÍA DE NINGÚN TIPO, YA SEA EXPRESA O IMPLÍCITA, INCLUYENDO, A MODO ENUNCIATIVO, CUALQUIER GARANTÍA DE COMERCIALIZACIÓN, IDONEIDAD PARA UN FIN PARTICULAR Y NO INFRACCIÓN. EN NINGÚN CASO LOS AUTORES O TITULARES DEL COPYRIGHT INCLUIDOS EN ESTE AVISO SERÁN RESPONSABLES DE NINGUNA RECLAMACIÓN, DAÑOS U OTRAS RESPONSABILIDADES, YA SEA EN UN LITIGIO, AGRAVIO O DE OTRO MODO, RESULTANTES DE O EN CONEXIÓN CON EL SOFTWARE, SU USO U OTRO TIPO DE ACCIONES EN EL SOFTWARE. Google Chrome para PC o Mac.

#### **Google Chrome es un navegador web rápido y gratuito.**

Requisitos para su instalación son:

##### **1)Procesador:**

- a)Intel Pentium 4 o superior.
- b)Espacio en disco: 350 MB.
- c)Memoria RAM: 512 MB.

##### **2)Alguno de los sistema operativos de la siguiente lista:**

- a)Windows xp sp2 o superior.
- b)Mac OS X 10.6 y versiones posteriores.
- c)Ubuntu 12.04 y versiones posteriores, Debian 7 y versiones posteriores,
- d)OpenSuSE 12.2 y versiones posteriores, Fedora Linux 17