

Ingeniería de Software

Entrega 2

Esteban Álvarez Zuluaga

Jacobo Montes Castaño

Carlos Restrepo Giraldo

Samuel Lenis Mira

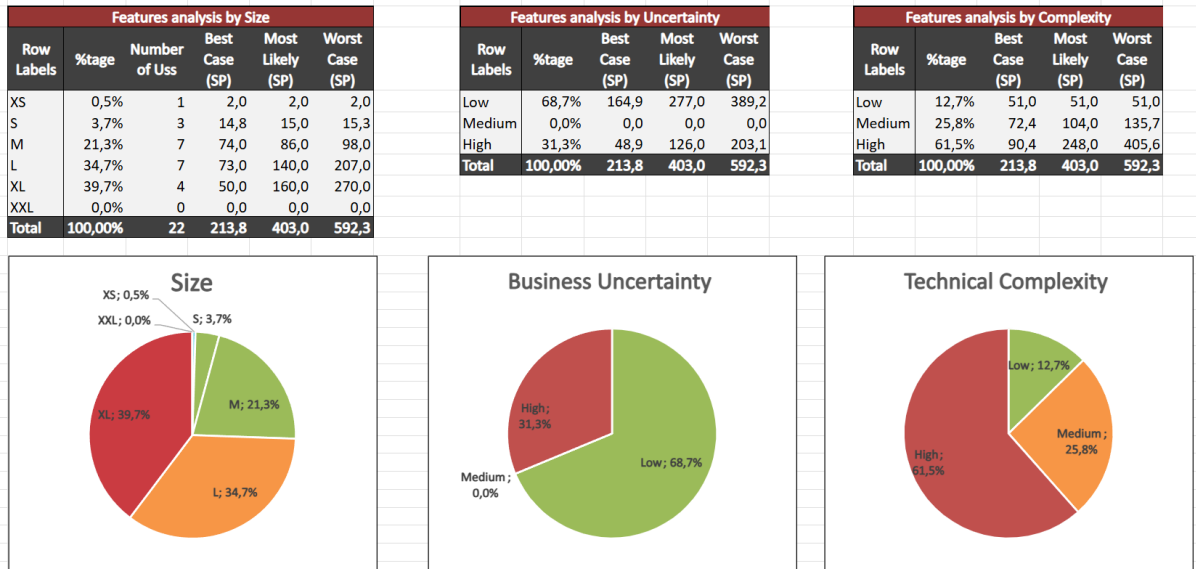
Universidad EAFIT

Ingeniería de Sistemas

10/2/2024

Cambios entregaEntrega2_Template_HistoriasUsuario y Cono de la Incertidumbre

Graficas Composición Sprints

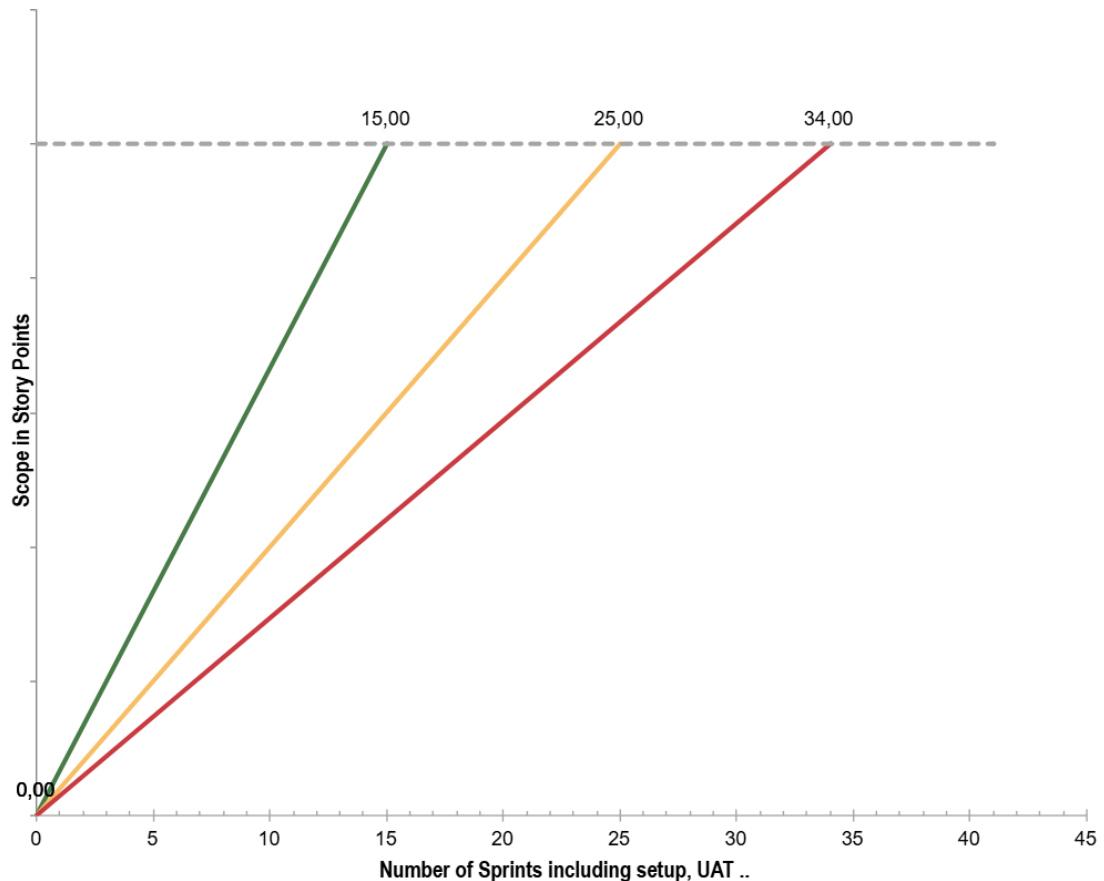


En esta gráfica podemos ver que nuestro proyecto cuenta con requisitos ~~muy~~ importantes y de gran tamaño con alta complejidad interna ~~y~~ es posible, que se puedan obtener nuevos requisitos en base a ellos. Aun así, la mayoría son requisitos de tamaño M y ~~L~~ los cuales están dentro del rango optimo de tamaño de requisito.

En general ~~se~~ tiene un bajo riesgo empresarial ~~y~~ que la gran mayoría de requisitos son fundamentales para el funcionamiento de la aplicación ~~y~~ solo un 30% pueden tener incertidumbre en su implementación de parte de los stakeholders.

Se ~~ve~~ también una complejidad técnica generalmente alta, pero no son un porcentaje alto los extremadamente difíciles. ~~Vemos~~ que entre mayor talla mayor complejidad, sin embargo, ~~también~~ hay complejidad en las tallas menores, en realidad los requisitos ~~son complejos~~, y algunos tienen bastante incertidumbre y como equipo debemos evaluar bien qué requisitos ~~vamos~~ a desarrollar en relación incertidumbre-complejidad.

Grafica Cantidad de Sprints



Project timeline	No of Sprints in Build Phase	Total no of Sprint including (setup, UAT ..)
Best Case	13,00	15,00
Most Likely	23,00	25,00
Worst Case	32,00	34,00
Threshold	41,00	41,00

Nuestro proyecto en la gráfica fue estimado en 18 sprints dentro del cono de incertidumbre debido a la complejidad y extensión de muchos de los requisitos que debemos implementar.

se reducen

Sin embargo, es probable que ~~no necesitemos todos esos sprints~~ ya que el tiempo que tenemos para desarrollar el proyecto es limitado, y con el avance de los sprints ~~iremos~~ *se han* completando varios requisitos a la vez y afinando nuestras estimaciones.

se describen las
se
A medida que ~~desarrollemos muchas más~~ funcionalidades y reduzcamos la incertidumbre sobre estas historias de usuario habrá ~~más~~ claridad sobre las tareas, *se* podremos optimizar el trabajo y reducir el tiempo ~~necesario~~ para completar el proyecto.

se considera
comprender
En general, ~~consideramos~~ que esto nos puede guiar a ~~darnos cuenta de~~ la complejidad de ~~nuestro~~ proyecto, y que posiblemente subestimemos su dificultad. ~~Consideramos que posiblemente cada requisito tenga demasiada complejidad, y pueda necesitar dividirse. Aun así, consideramos que podemos llevar a cabo los requisitos en menor tiempo del que el sistema propone.~~

https://eafit-my.sharepoint.com/:x/g/personal/ealvarez_eafit_edu_co/ETi-G6BgobZLorj0YrsZsXkBVybYsSS67VZenwJeMz5zdg?e=qDo4So

Entrega2_Template_HistoriasUsuario y Cono de la Incertidumbre

Cambios Historias de Usuario Primera Entrega

Debido a la naturaleza dinámica del desarrollo, se debieron hacer cambios a las historias de usuario para mejor reflejar el proyecto y sus funcionalidades. Específicamente en el motor de automatización, descubrimos que existe un mayor nivel de complejidad del que teníamos previsto, y como tal más funcionalidad que no se tuvieron en cuenta previamente.

De esta forma, se añadió la siguiente historia de usuario al backlog.

1. **RF-23:** Yo como cliente ~~quiero~~ *debo* que el sistema sea capaz de interactuar con una gran cantidad de formularios, ~~determinando~~ *de* que campo corresponde a que dato de usuario.
 - **Responsable:** Esteban Alvarez Zuluaga
 - **Condiciones de aceptación:**
 - La detección de campos funciona de manera coherente en un 50% de los casos.

Adicionalmente, se eliminaron ciertas historias de usuario que se determinó que causarían fricción en la utilización apropiada del producto.

- **RF-4:** Yo como cliente quiero poder confirmar utilizando mis credenciales si deseo que el sistema llene el formulario, para evitar que otros actores accedan a mis datos personales.
 - Se eliminó, debido a que el objetivo es lograr que el usuario gaste el menor tiempo posible en llenar el formulario, y confirmar en cada llenado implica un gasto alto de tiempo. Adicionalmente, el usuario ya estaría autenticado entonces no habría problemas de seguridad

Aspectos generales de la entrega

La entrega actual se enfocará en aquellos componentes necesarios para comenzar la implementación del programa AutoForm de manera que cumpla con los requisitos de utilización definidos en la entrega anterior.

Para esto, se implementarán varios diagramas y mecanismos de planificación que permitan generar una idea de la arquitectura a ser utilizada en el proyecto y su implementación técnica. Adicionalmente, se implementaron algunos componentes del proyecto.

La visión del proyecto sigue siendo la misma, y los objetivos a cumplir no se han modificado, aunque si se ha modificado la manera en que planeamos lograrlo, y los requisitos e historias de usuario que se harán.

Se ha comenzado a implementar el proyecto y consideramos para su arquitectura un modelo de servidor y extensión que estén en constante comunicación. Esto permite que la extensión web que lleva a cabo el llenado de formularios sea liviana y los datos sensibles sean solo manipulados por el servidor, bajo autenticación adecuada. Como tal, todos los procesos de ingreso de información a la base de datos serán llevados a cabo en la página web de apoyo que controla el servidor.

En cada vista arquitectónica 4+1 se incluyen los diagramas nuevos de la entrega, así como los llevados a cabo en el primer sprint.

Evaluación Sprint anterior

Utilizamos la técnica más utilizada en la metodología Scrum que se basa en unas preguntas claves para determinar el desempeño pasado y determinar lo que se debe hacer mejor en el futuro.

¿Que hicimos bien?

Como equipo consideramos un buen desempeño, pues logramos realizar lo necesario con un alto nivel de calidad. Ciertos aspectos de la colaboración fueron efectivos, como la distribución de los diagramas.

Además, logramos un buen manejo de las historias de usuario desde GitHub, y logramos guiarnos en la creación del producto mediante los diagramas realizados.

¿Que no hicimos bien?

Aunque la coordinación permitió una distribución apropiada de las tareas y el logro de los objetivos, hubo grandes falencias en este frente, pues hubo tareas comunitarias difíciles de

completar. Esto se debió a una falta de comunicación entre los miembros, y generó problemas en la etapa final del sprint.

¿Qué debemos mejorar?

Debemos incrementar la cooperación en el equipo y planear las cosas con tiempo para lograr coordinarnos para repartir de manera equitativa el trabajo. Adicionalmente, consideramos que para ciertos requisitos fundamentales es importante que trabajemos todos de manera presencial.

Planificación Sprint actual

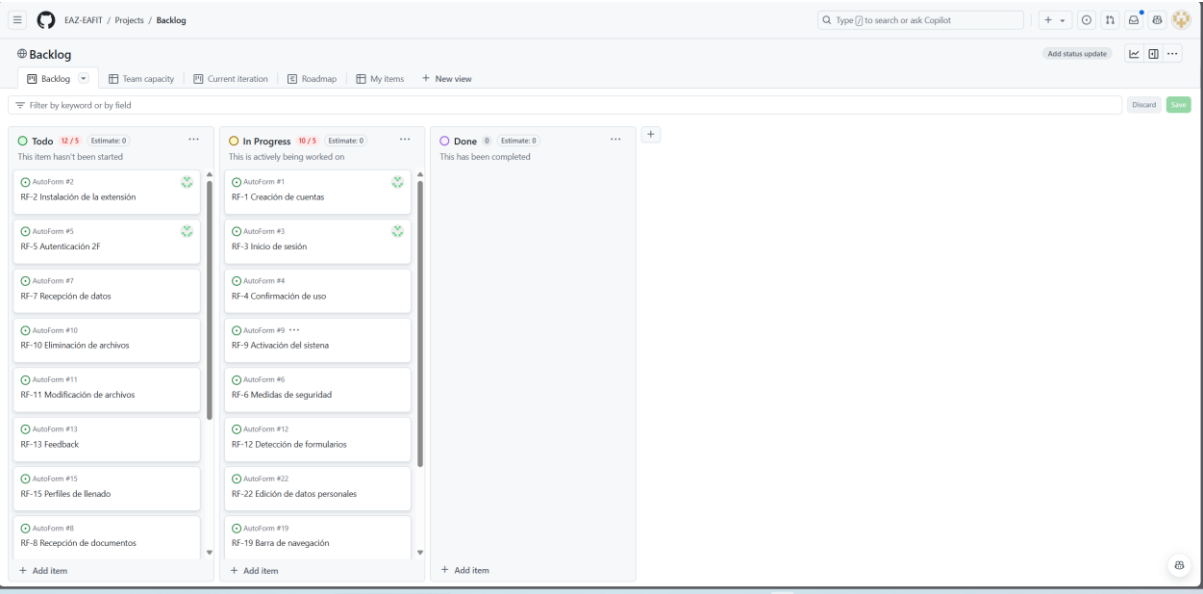
Para el sprint actual, se determinó que se debía implementar la funcionalidad básica del programa que permitiera llevar a cabo el llenado de formularios. Se considero fundamental implementar el módulo de autenticación y manejo de los datos, así como el módulo de detección y llenado de formularios.

Aunque el manejo de archivos es una parte importante del programa, debido a la complejidad de los otros módulos y de lograr una interconexión efectiva entre ellos, se determinó implementar este aspecto en el siguiente sprint, y priorizar un producto mínimo viable completo.

Las historias de usuario que se determinaron a implementar en el sprint son:

- ✓ **RF-9:** Yo como cliente quiero que al encontrar un formulario web, pueda activar el sistema que llevará a cabo el llenado automático para ahorrar el tiempo que gastaría llenándolo.
- ✓ **RF-12:** Yo como cliente quiero que el sistema detecte automáticamente cuando hay formularios que hay que rellenar, para no tener que gastar tiempo abriendo el sistema cada vez que quiera activarlo. *reducir*
- X **RF-1:** Yo como cliente quiero crear una cuenta nueva con usuario y contraseña encriptadas para lograr la seguridad de mis datos y persistencia en el tiempo. *No es RF es RNF es seguridad*
- X **RF-3:** Yo como cliente quiero poder ingresar a la plataforma web haciendo uso de mis credenciales para asegurar la seguridad y persistencia de mis datos. *RNF disponibilidad*
- X **RF-23:** Yo como cliente quiero que el sistema sea capaz de interactuar con una gran cantidad de formularios, determinando que campo corresponde a que dato de usuario. *RNF disponibilidad*
- ✓ **RF-19:** Yo como cliente quiero que la página web de AutoForm tenga una barra de navegación para fácilmente acceder a todos los servicios prestados.
- +o- **RF-20:** Yo como cliente quiero que la página web de AutoForm tenga una página de about para saber que puede proveerme el producto. *no lo pide el cliente*
- **RF-14:** Yo como administrador quiero poder eliminar usuarios de la plataforma para evitar que haya usuarios que no cumplan con el código de conducta de la empresa

Github Backlog



[Backlog · Backlog \(github.com\)](#)

Aspectos estructurales y arquitectónicos de la solución

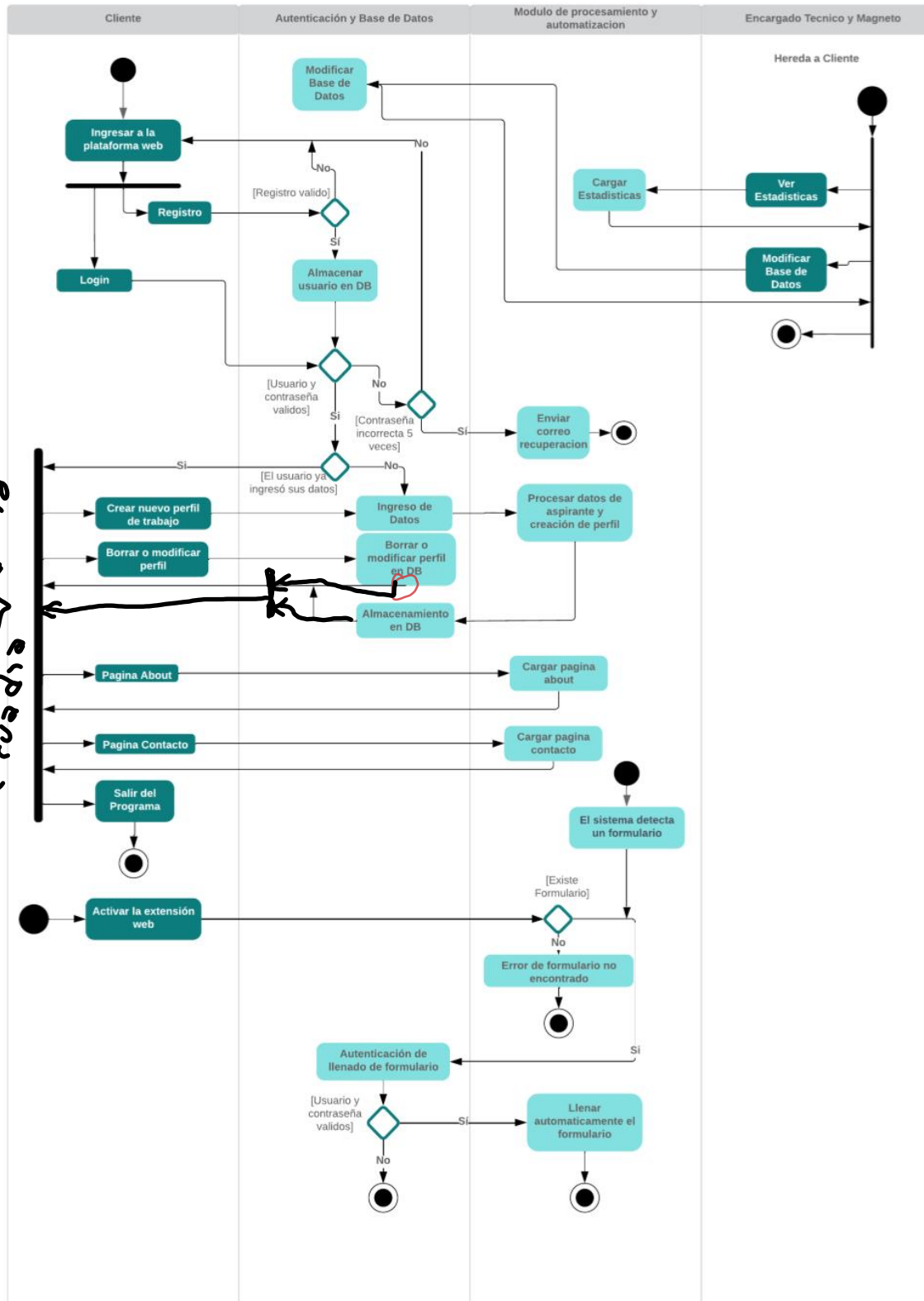
Estilos Arquitectónicos Usados

Tipo Aplicación:	Extensión web y aplicación web
Estilo Arquitectónico:	<p>Los principales estilos arquitectónicos que se utilizarán serán el de Client/Server, MVC (Model, View, Controller) y Component-based.</p> <p>Client/Server</p> <p>Se utilizará una arquitectura client/server debido a que el sistema se compone de dos módulos, el cliente (extensión web) y el servidores (servidor Django de python).</p>

	<p>La extensión web interactúa con el usuario y con la página web donde esté el formulario laboral. Utiliza la información que recibe de estas fuentes para comunicarse con el servidor y recibir nueva información que utiliza para llenar los formularios.</p> <p>El servidor se encarga de llevar a cabo el procesamiento, validación y almacenamiento de los datos, asegurando la seguridad del proceso.</p> <p>MVC (Model, View, Controller)</p> <p>El framework de desarrollo web Django implementa el MVC como arquitectura. En nuestro caso, se utiliza el backend de Django para procesar aquellas solicitudes de la extensión, así como para mostrar la página suplementaria donde se puede ingresar a la plataforma y modificar el perfil.</p> <p>Esta página suplementaria se implementa por medio de la arquitectura MVC. En primer lugar se tienen los Modelos, los cuales hacen referencia a datos y base de datos, y que en Django se implementa como una base de datos relacional integrada. Por otro lado, se tiene las vistas las cuales corresponden a la interfaz de usuario de la página web de apoyo. Por último se tiene el controlador, el cual se encarga de todo el procesamiento necesario para llevar a cabo las vistas como about, login, signup, modificar datos, etc.</p> <p>Component Based</p> <p>El sistema también posee elementos de arquitectura basada en componentes, específicamente en la arquitectura interna de cada uno de los módulos.</p> <p>En el módulo de extensión, se separan las funcionalidades por componentes que corresponden a un archivo de JavaScript, por ejemplo, relegando a un componente la comunicación con el backend, otro al llenado de formularios, etc.</p> <p>En el servidor de Django, se tienen componentes diferenciados para la comunicación con el servidor y para el control de la página de apoyo.</p>
Lenguaje programación	JavaScript, Python, SQL, HTML, CSS
Aspectos técnicos	Se utilizará el framework de Django de python, el cual utiliza una arquitectura MVC (Model View Controller para la página

	<p>web. Este framework posee la ventaja de poder fácilmente implementar sistemas de autenticación, así como una base de datos sqlite3 integrada. Este módulo de Django servirá para recibir y enviar mensajes a la extensión web por medio de HTTP, así como para la página web de apoyo. La extensión web, estará implementada en JavaScript y enviará requests al backend de Django para recibir la información relevante. Entre los componentes de la extensión web, se comunicarán por medio de las APIs de chrome de comunicación.</p>
Frameworks	Django web framework

Diagrama de Actividades



Diagramas de Secuencias

Diagrama de secuencias que muestra el proceso de autenticación por parte de los usuarios y cómo interactúa cada componente del sistema en dicho proceso.

Autenticación/registro usuarios

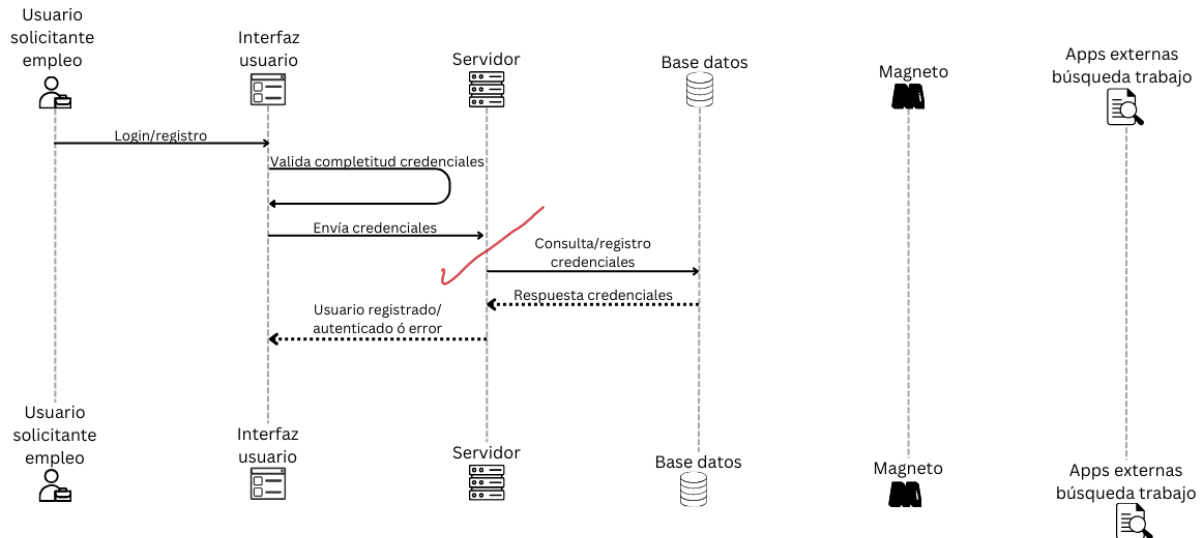


Diagrama de secuencias que muestra el proceso de ingreso de hojas de vida por parte de los usuarios y cómo interactúa cada componente del sistema en dicho proceso.

Ingreso y procesamiento hoja de vida

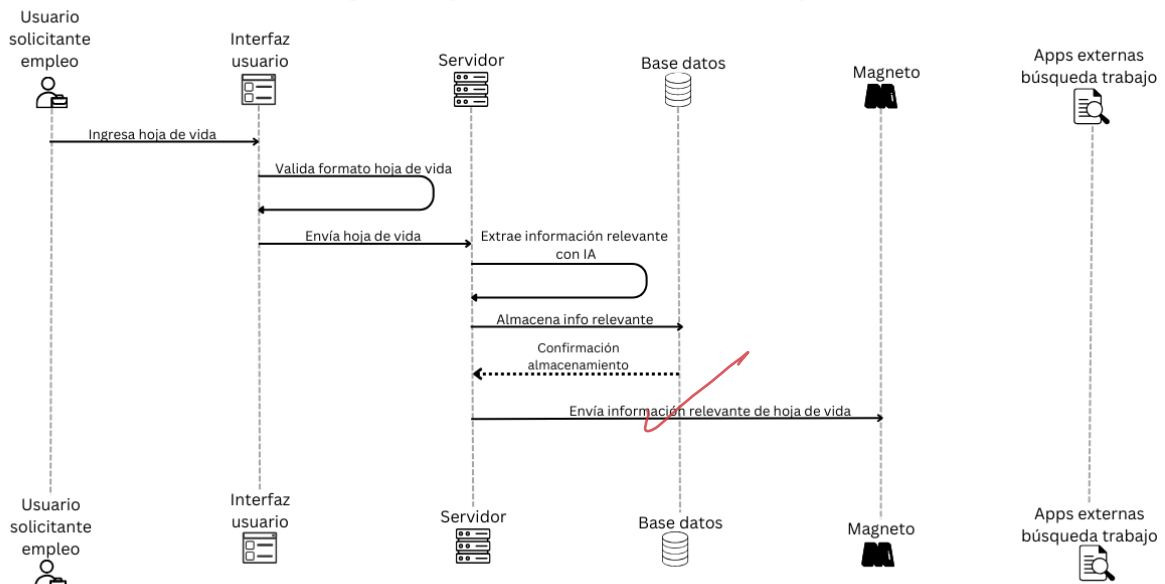
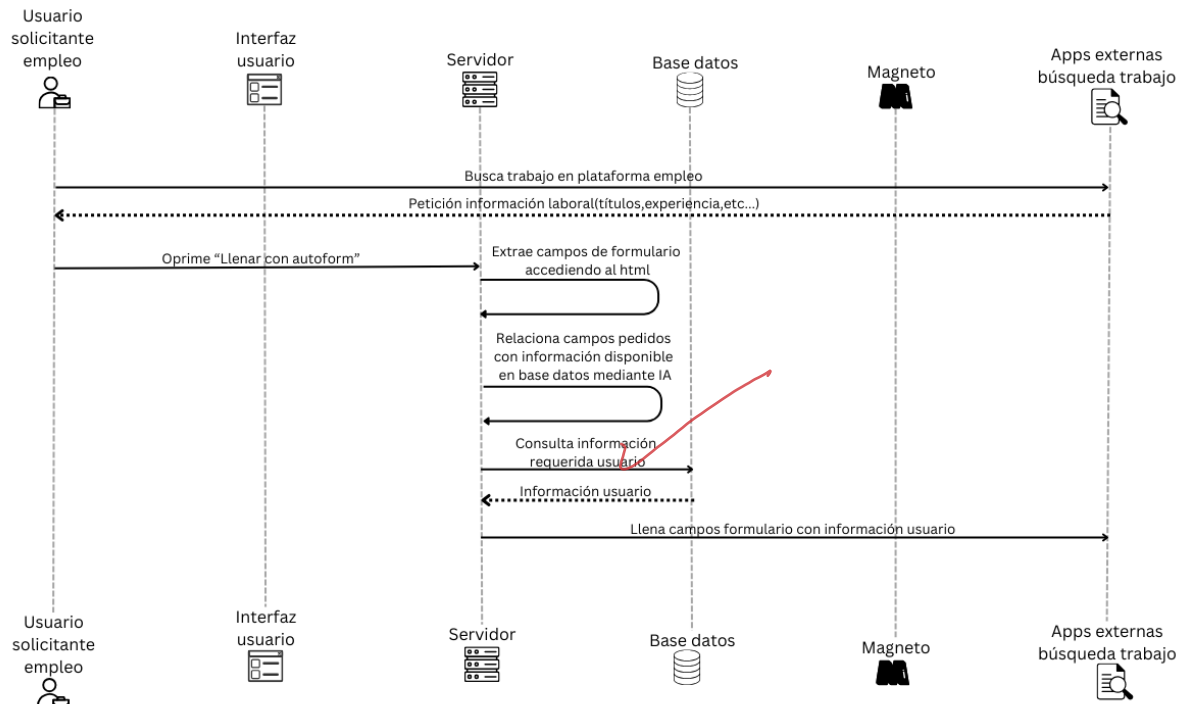


Diagrama de secuencias que muestra el proceso de llenado automático de formularios y cómo interactúa cada componente del sistema en dicho proceso.

Llenado automático de formularios



Vista Lógica

Diagrama de Casos de Uso

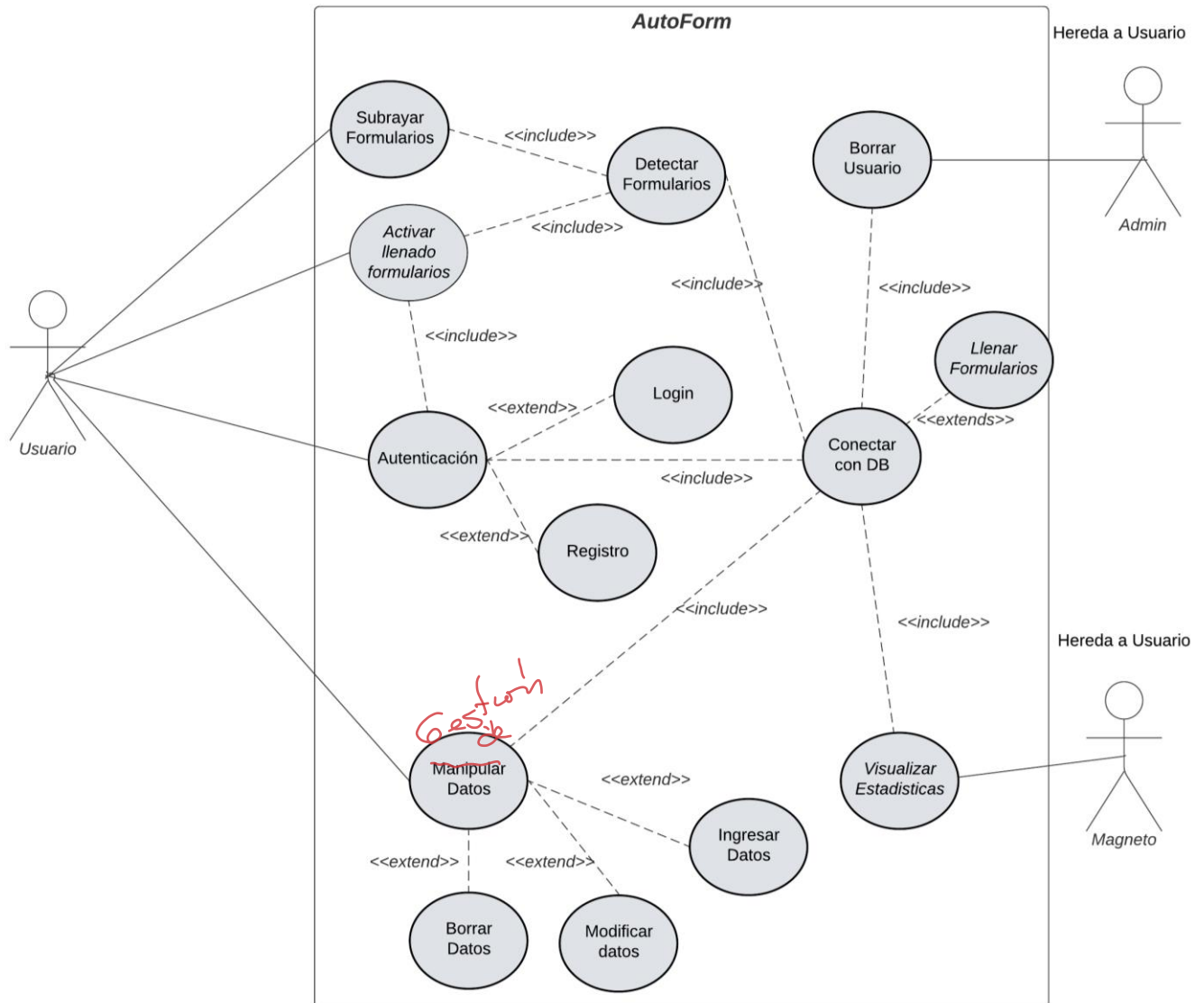


Diagrama de Clases

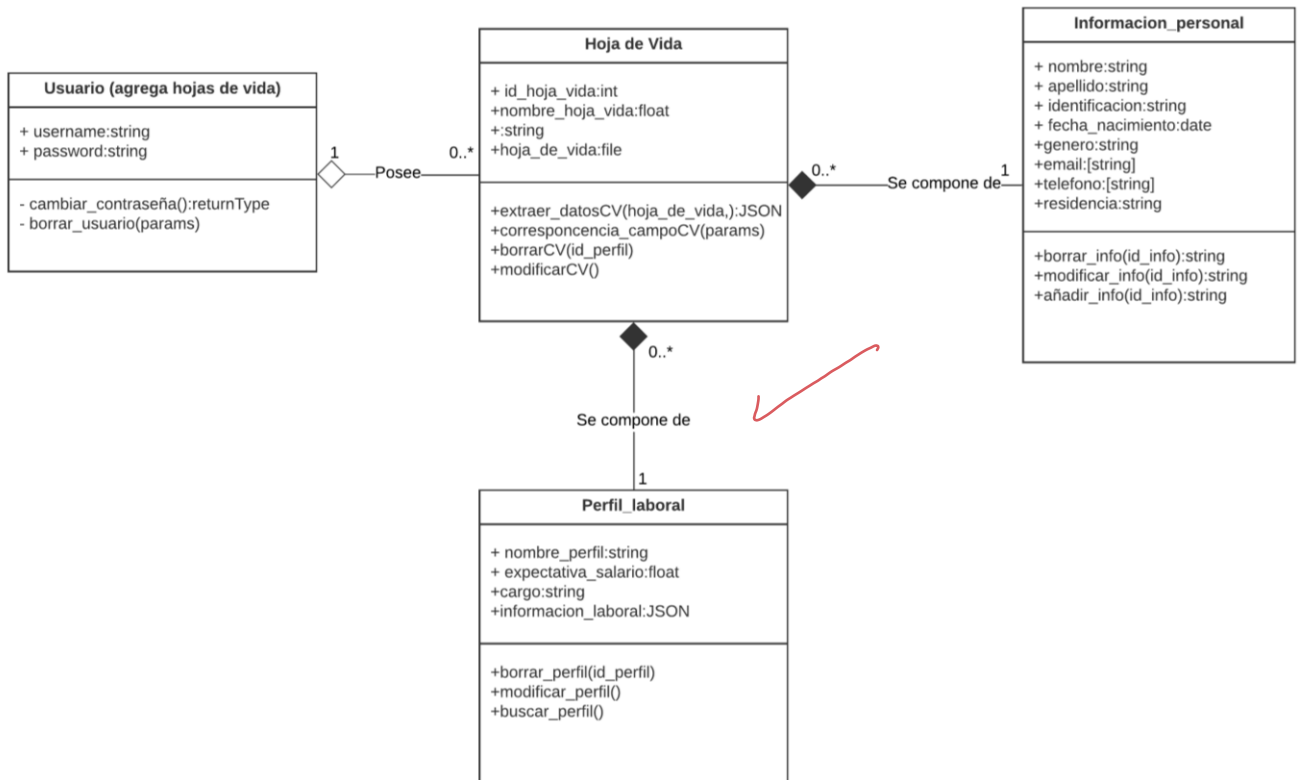
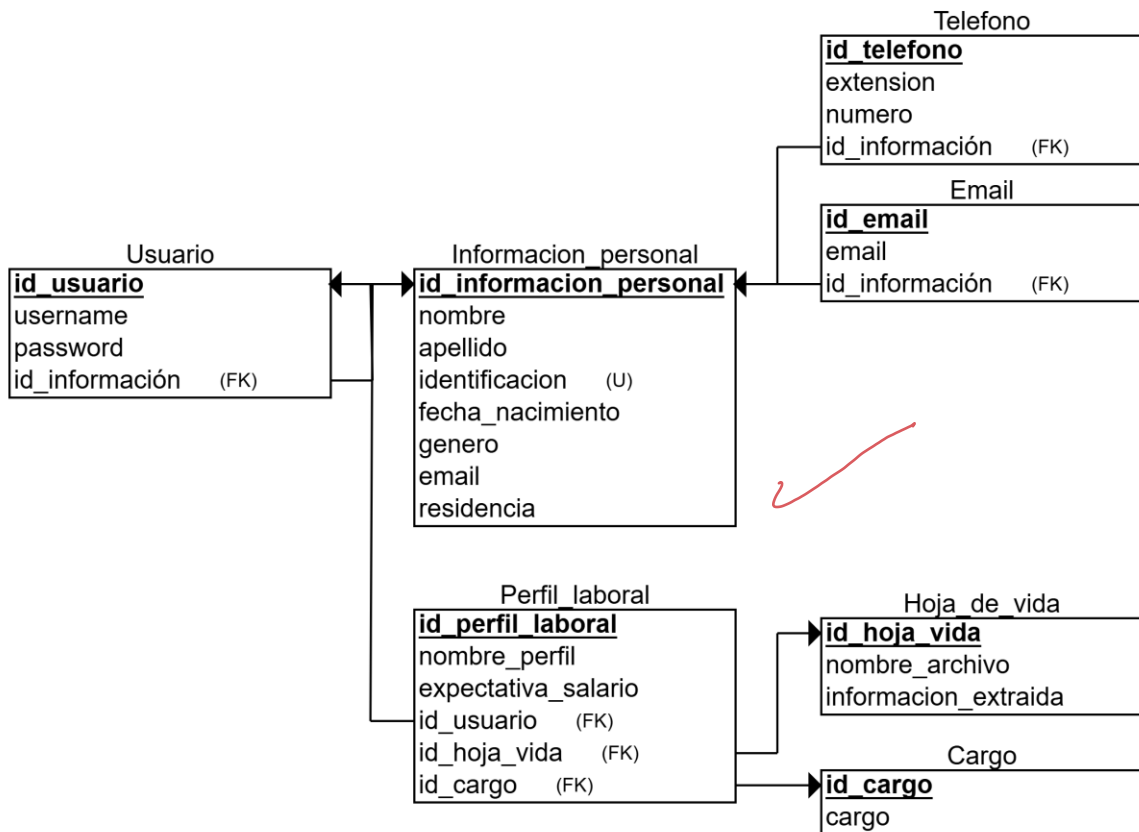
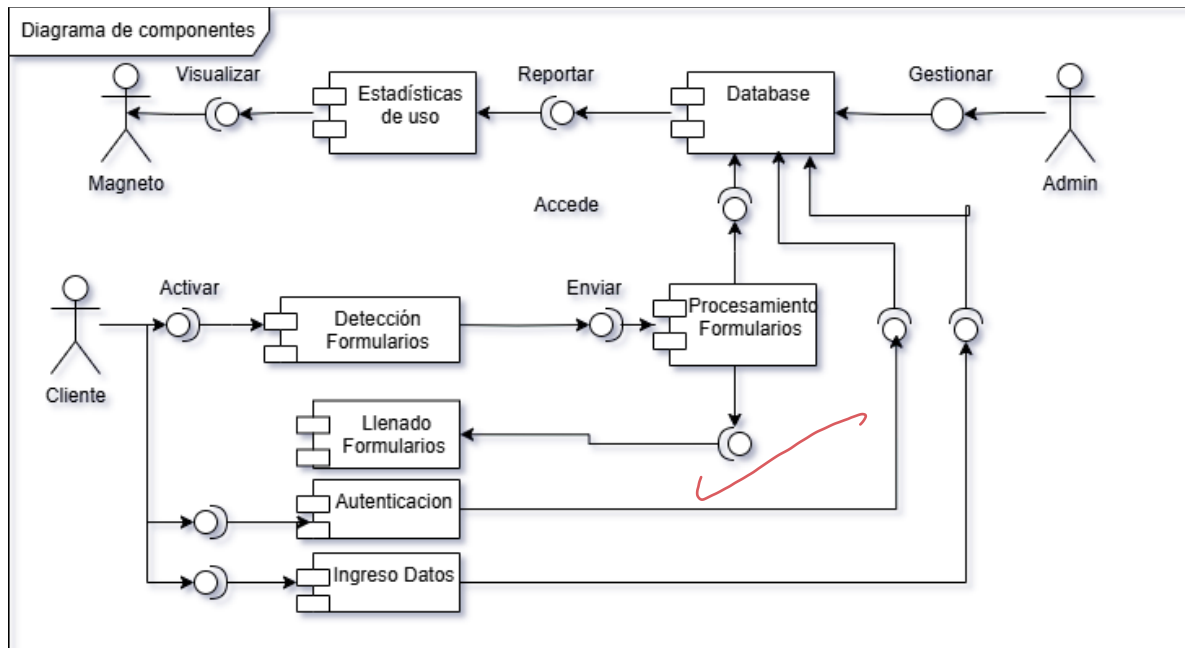


Diagrama Entidad-Relación (Modelo de datos o de persistencia)



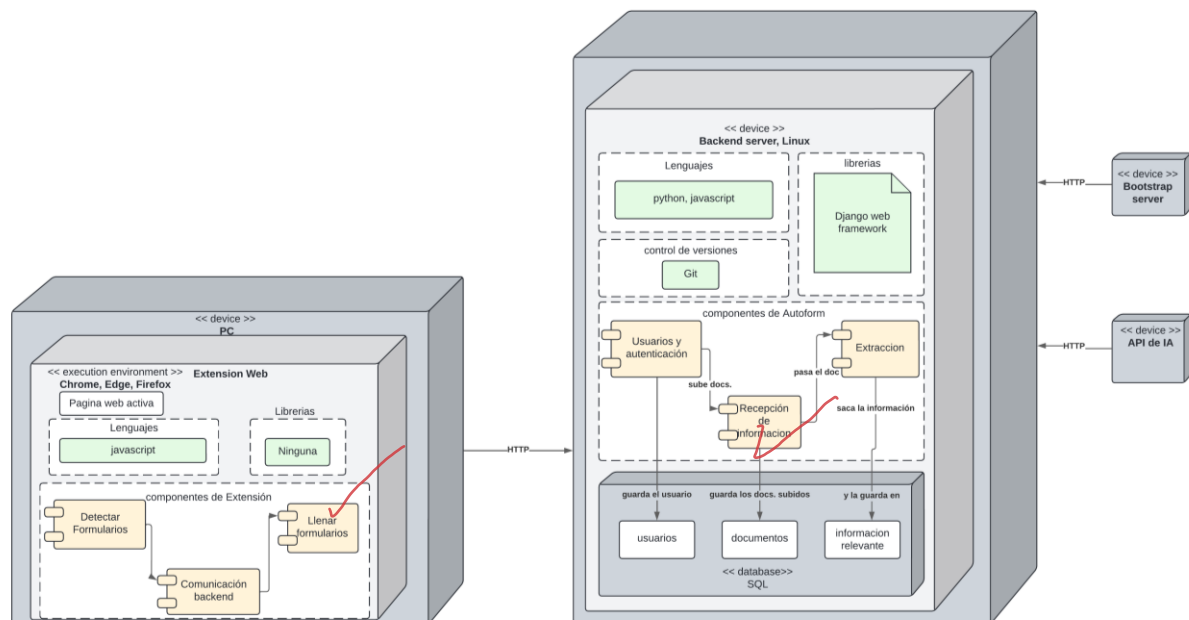
Vista Física

Diagrama de Componentes



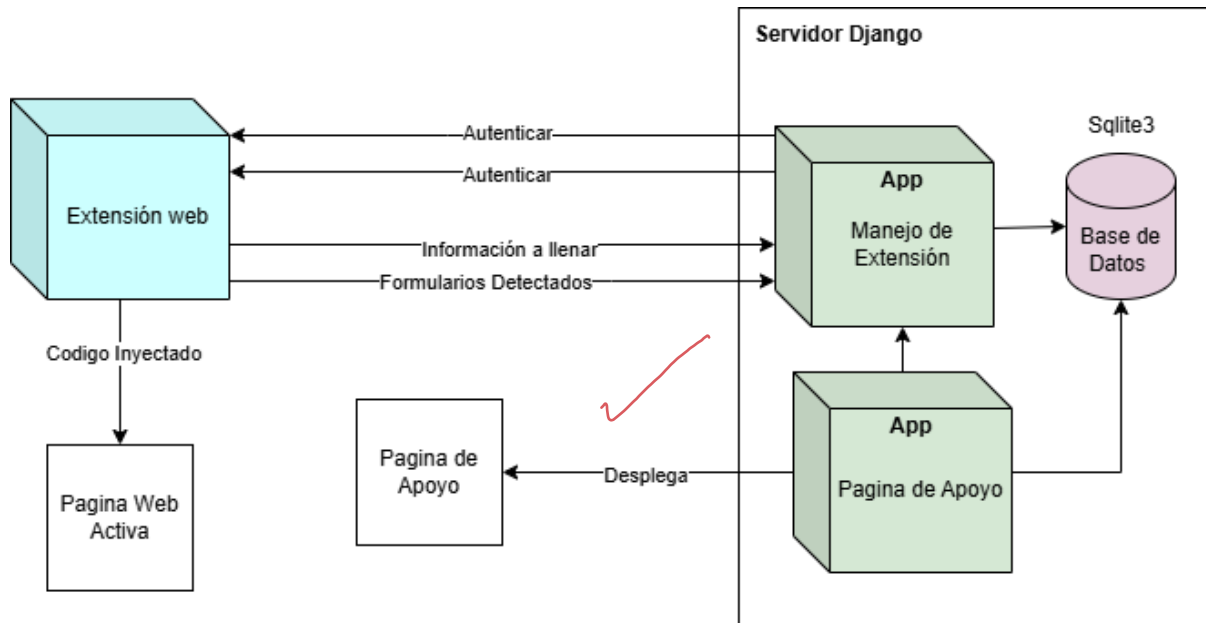
<https://app.diagrams.net/?src=about>

Diagrama de Despliegue



https://lucid.app/lucidchart/5873b04b-58cb-4b1e-931c-cbc607717634/edit?viewport_loc=-1068%2C-619%2C3040%2C1708%2C0_0&invitationId=inv_3e6fb3cf-4274-4c7a-9ce6-db32904703d4

Diagrama de Arquitectura



Avances en cuanto a funcionalidad y demostración

Se implementó la arquitectura de la manera que especifica el diagrama de arquitectura, ya que se implementó una extensión web en JavaScript que se carga al buscador. Una vez se activa, detecta los formularios y por medio de HTTP se comunica con un servidor de Django.

En el servidor, hay dos apps de Django, una para recibir y retornar información a la extensión para el llenado de formularios y la otra para controlar la página de apoyo desde la cual se harán los procesos de autenticación y añadir datos.

Para la autenticación, desde la extensión se verifica con el backend si está autenticado, y en caso de no estarlo le permite al usuario acceder a la página de apoyo donde puede autenticarse.

Estructura de Archivos



El directorio AutoForm corresponde al servidor de Django, Users a la página de apoyo, extension a la app extension y el directorio de extension a la extensión en sí.

Autenticación y Registro

register

ChatGPT

(73) WhatsApp

(12) Empleos de «

Extensiones

Nueva pestaña

http://127.0.0.1:8000/Users/register/

Registrarse

La contraseña debe tener al menos 6 caracteres y no debe ser completamente numérica.

Nombre usuario

Correo electrónico

Contraseña

Confirmar contraseña

Registrarse

¿Ya tienes cuenta? [Iniciar sesión](#)

login

ChatGPT

(73) WhatsApp

(12) Empleos de «

Extensiones

Nueva pestaña

http://127.0.0.1:8000/Users/login/

Iniciar Sesión

Usuario

Tu nombre de usuario

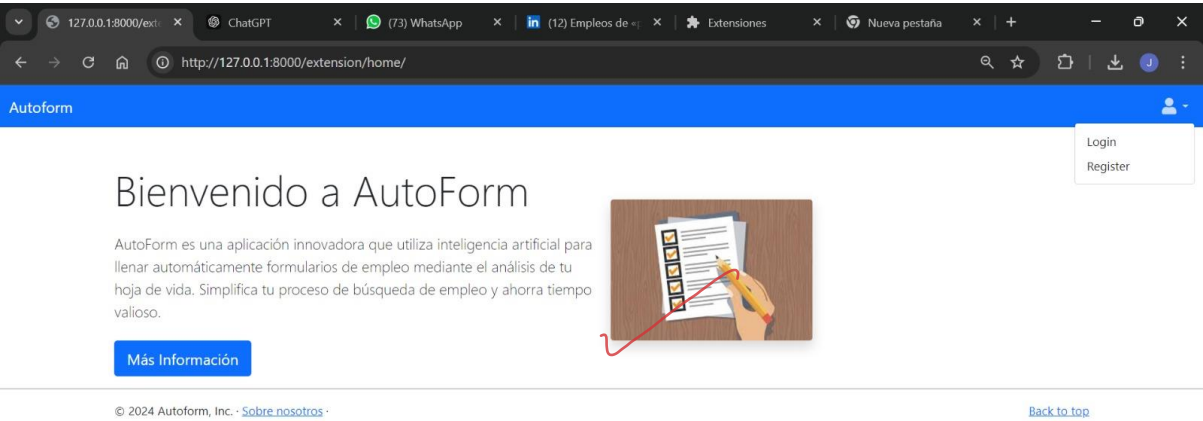
Contraseña

Tu contraseña

Iniciar Sesión

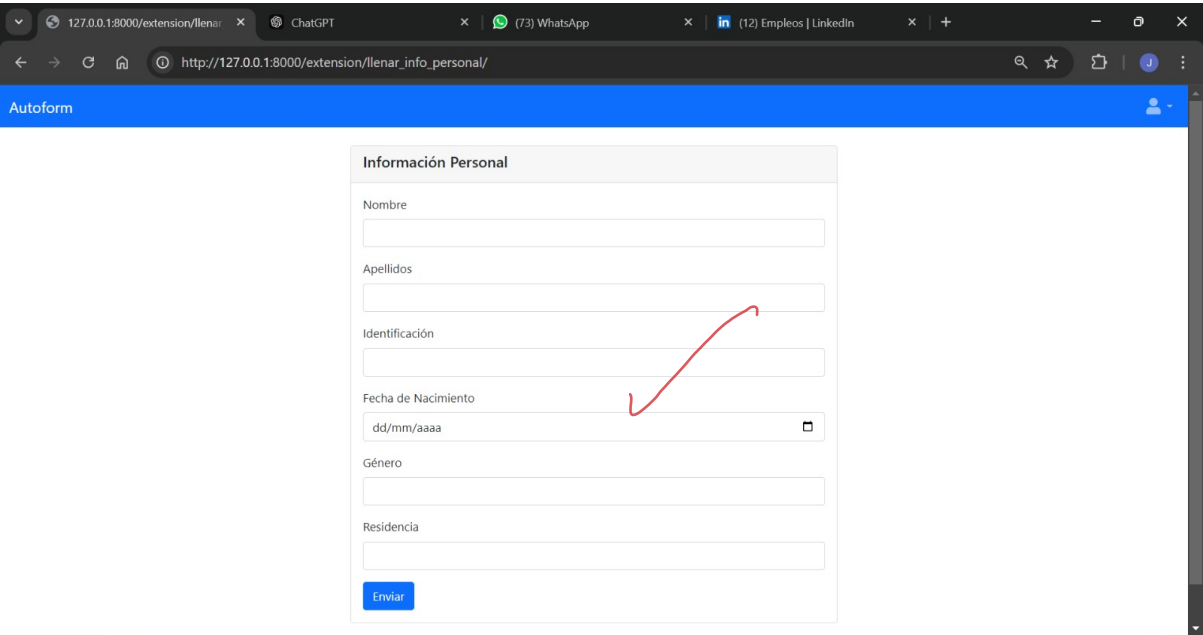
¿No tienes una cuenta? [Crea una ahora](#)

Pagina About y NavBar

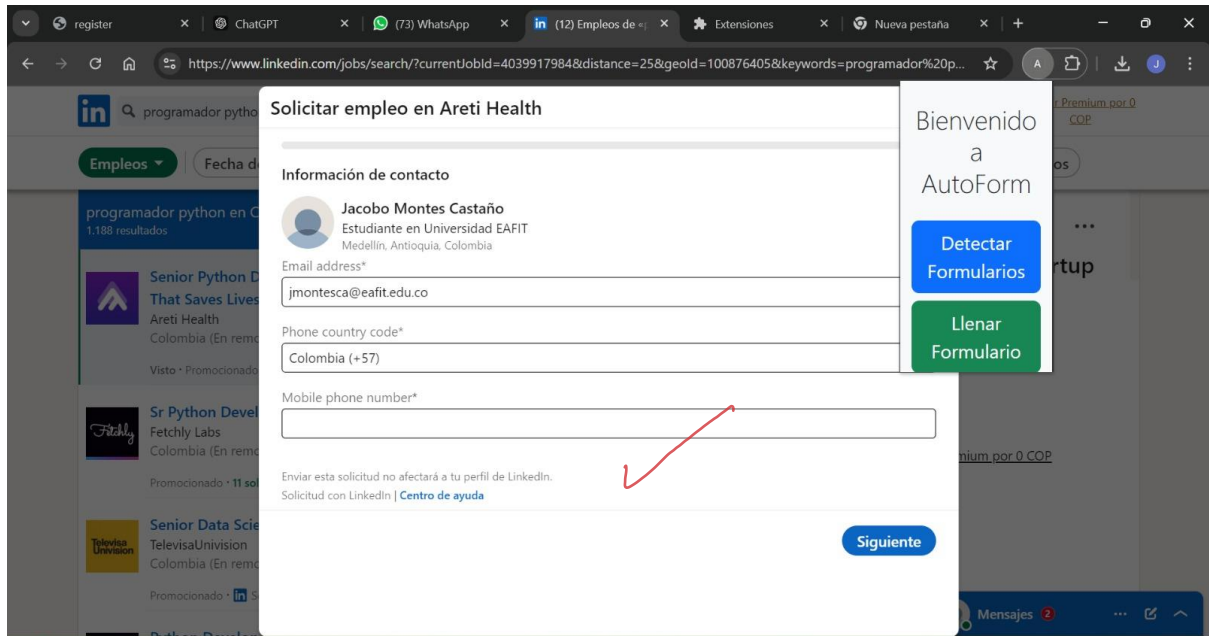


127.0.0.1:8000/extension/home/#

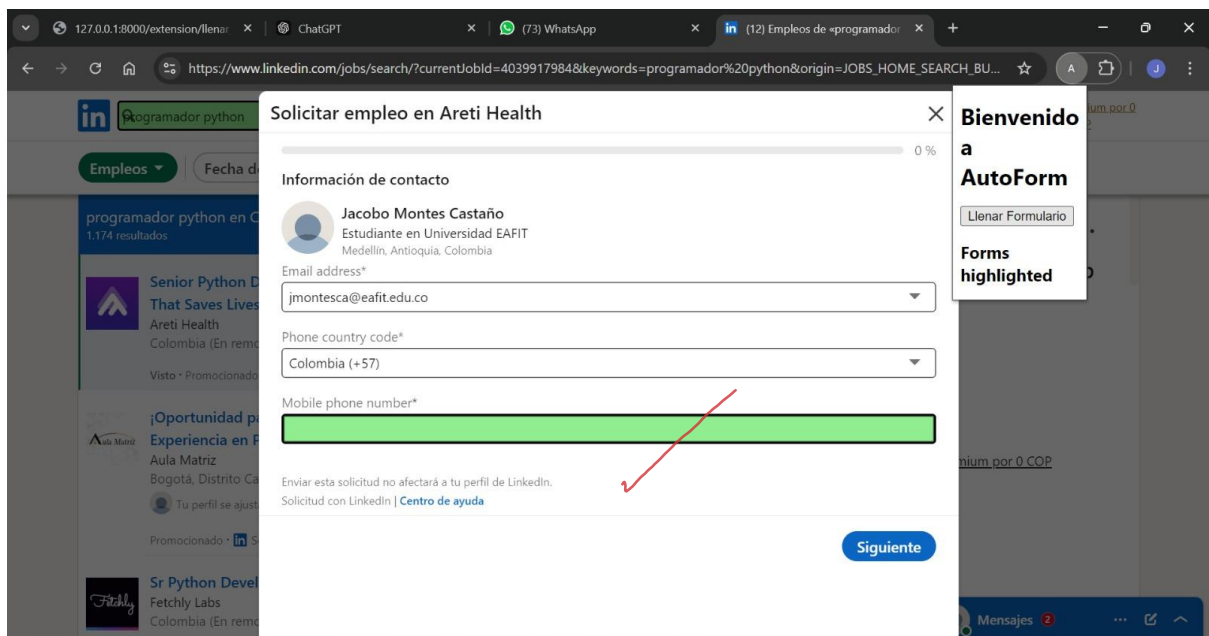
Añadir información personal



Extensión web en formulario de trabajo



Detección de Formularios



Enlace de Github del Proyecto

Conclusiones y lecciones aprendidas

Aprendimos de una gran variedad de diagramas de UML, así como de las diferentes vistas arquitectónicas y como relacionarlas con la implementación física del programa. Mejoramos la coordinación de equipo, aunque todavía hay retos.

Referencias

- Atlassian. (n.d.). Retrospective. Atlassian Team Playbook. Retrieved October 4, 2024, from <https://www.atlassian.com/team-playbook/plays/retrospective>
- Javarevisited. (2023, October 5). 4+1 Architectural View Model in Software. Medium. Retrieved October 4, 2024, from <https://medium.com/javarevisited/4-1-architectural-view-model-in-software-ec407bf27258>
- Microsoft. (n.d.). Patterns & Practices: Application Architecture Guide 2.0. Microsoft Learn. Retrieved October 4, 2024, from [https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ee658117\(v=pandp.10\)](https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ee658117(v=pandp.10))
- Carnegie Mellon University. (n.d.). Introduction to Software Architecture. Retrieved October 4, 2024, from https://www.cs.cmu.edu/afs/cs/project/able/ftp/intro_softarch/intro_softarch.pdf
- FreeCodeCamp. (2023, October 5). El modelo de arquitectura View-Controller Pattern. FreeCodeCamp Español. Retrieved October 4, 2024, from <https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern>
- Sparx Systems. (n.d.). UML 2 Component Diagram Tutorial. Retrieved October 4, 2024, from http://www.sparxsystems.com.ar/resources/tutorial/uml2_componentdiagram.php
- IONOS. (n.d.). Diagrama de Componentes. IONOS Digital Guide. Retrieved October 4, 2024, from <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-componentes>
- Diagramas UML. (n.d.). Diagrama de Clases. Retrieved October 4, 2024, from <https://diagramasuml.com/diagrama-de-clases>
- GeeksforGeeks. (n.d.). Deployment Diagram (Unified Modeling Language). Retrieved October 4, 2024, from <https://www.geeksforgeeks.org/deployment-diagram-unified-modeling-languageuml>