



Desarrollo de la interfaz de usuario para la aplicación web “Mapa de Crecimiento”

Mateo Jaramillo Obando

Informe de práctica para optar al título de Ingeniero Electrónico

Asesor

Hernán Felipe García Arias

Universidad de Antioquia
Facultad de Ingeniería
Ingeniería Electrónica
Medellín, Antioquia, Colombia
2023

Cita

(Jaramillo Obando, M., 2023)

Referencia

Jaramillo Obando, M. (2023). *Desarrollo de la interfaz de usuario para la aplicación web “Mapa de Crecimiento”* Semestre de Industria. Universidad de Antioquia, Medellín, Colombia.

Estilo APA 7 (2020)



Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/Director: Julio César Saldarriaga Molina.

Jefe departamento: Augusto Enrique Salazar Jiménez.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Dedicatoria

Dedicado a mis padres, a mis hermanos, a mis sobrinos, a toda mi familia, a mis compañeros de estudio y a todas aquellas personas que confiaron en mí y me apoyaron en algún momento, porque sin ellos, esto no hubiese sido posible.

Agradecimientos

Agradecimientos a mis padres y a toda mi familia por el apoyo incondicional durante todo este tiempo, por los consejos, las palabras de aliento y cada acto que me impulsó a luchar por este sueño, agradecimientos a mis compañeros de estudio que fueron un gran punto de apoyo e inspiración, a mis profesores que me acompañaron en este proceso y al Alma Mater por abrigarme en sus entrañas y brindarme un espacio de crecimiento personal y profesional.

Tabla de contenido

Resumen	7
Abstract	8
Introducción	9
1 Objetivos	11
1.1 Objetivo general	11
1.2 Objetivos específicos	11
2 Marco teórico	12
2.1 Frontend	12
2.1.1 HTML	13
2.1.2 CSS	14
2.1.3 Javascript	14
2.1.3.1 Framework de Javascript	15
2.1.4 Pruebas unitarias	15
2.2 Backend	16
2.2.1 API	16
2.2.1.1 API de Rest	16
2.2.2 HTTP	17
2.3 Código limpio	19
2.4 Diseño atómico	20
2.5 Sistema de control de versiones Git	22
2.5.1 Git-flow	22
3 Metodología	25
3.1 Versionamiento del código y control de flujo	25
3.2 Estándares de desarrollo	25

3.3 Seguridad de la aplicación	25
3.4 Marco de trabajo	26
3.5 Estructura de carpetas del proyecto	47
3.6 Desarrollo de las historias de usuario	48
4 Resultados	58
4.1 Pantalla de Inicio de Sesión	58
4.2 Pantalla Dashboard del Chapter	61
4.3 Pantalla Dashboard del Pragmático	64
4.4 Pantalla Histórico de Valoraciones	69
4.5 Cerrar sesión en la aplicación	71
5 Conclusiones	72
Referencias	74

Siglas, acrónimos y abreviaturas

UdeA	Universidad de Antioquia
Pragmáticos	Empleados de la compañía Pragma S.A.
Chapter	Área de conocimiento de la compañía Pragma S.A.
SPA	Single Page Application
API	Application Programming Interface
URL	Uniform Resource Locator
CLI	Interfaz de Línea de Comandos

Resumen

La compañía PRAGMA S.A. requiere de una aplicación web que le permita a los líderes de las diferentes áreas de conocimiento gestionar los equipos de desarrollo y conocer información detallada de cada uno de los “pragmáticos” (empleados de la compañía), por tanto, en este proyecto se lleva a cabo el desarrollo de la interfaz de usuario o frontend de la aplicación web “Mapa de Crecimiento”. Dicha aplicación tiene como objetivo proporcionar a los líderes las herramientas necesarias para llevar a cabo las labores anteriormente mencionadas y permitirle a los pragmáticos conocer la ruta de aprendizaje que deben seguir para su proceso de crecimiento.

El desarrollo de la interfaz de usuario se realizó siguiendo el marco de trabajo Scrum, aplicando los conceptos de las buenas prácticas en el desarrollo bajo las tecnologías de desarrollo web frontend HTML, CSS y Javascript sobre el popular framework soportado por Google llamado Angular.

Al seguir esta metodología, se obtuvo una interfaz de usuario modular, escalable y mantenible, para la aplicación web "Mapa de Crecimiento", la cual fue integrada con el backend para proporcionarle la lógica necesaria a la aplicación, permitiendo a los líderes de las diferentes áreas de conocimiento acceder a un panel de control donde podrán visualizar información unificada, resumida y detallada de la evolución que han tenido los pragmáticos en cuanto a sus habilidades técnicas y blandas dentro de un área de conocimiento específica, posibilitando agrupar esta información en base a diferentes aspectos tales como evolución en ciertos periodos de tiempo, niveles técnicos, especialidades, entre otros; los líderes también podrán llevar a cabo el proceso de valoración de los pragmáticos con el fin de identificar sus fortalezas, sus brechas, y cómo se encuentran ubicados en base a sus habilidades técnicas para así proporcionarles a éstos una ruta de aprendizaje detallada que les indique el camino a seguir para continuar con su proceso de crecimiento.

Palabras clave: frontend, aplicación web, Javascript, Angular, mapa de crecimiento, código limpio.

Abstract

The company PRAGMA S.A. needs a web application that allows the leaders of the different knowledge areas to manage the development teams and get detailed information about each of the “pragmáticos” (company employees), therefore, the development of the user interface of the web application “Mapa de Crecimiento” is carried out in this project. This application aims to provide leaders with necessary tools to carry out the aforementioned tasks and allow pragmáticos to know the learning path they must follow for their growth process.

The development of the user interface was carried out following the Scrum framework, applying the concepts of good development practices under the frontend web development technologies HTML, CSS and Javascript on the popular framework supported by Google called Angular.

By following this methodology, a modular, scalable and maintainable user interface was obtained for the “Mapa de Crecimiento” web application, which was integrated with the backend to provide the necessary logic to the information displayed, allowing the leaders of the different areas of knowledge to access a control panel where they can view unified, summarized and detailed information on the evolution that pragmáticos have had in terms of their technical and soft skills, making possible to group this information based on different aspects such as evolution in certain periods of time, technical levels, specialties, among others; leaders will also be able to carry out the pragmático assessment process to identify their strengths, weaknesses, and where they stand based on their technical skills in order to give them a detailed learning path that indicates the way to follow to continue with its growth process.

Keywords: frontend, web application, Javascript, Angular, Mapa de Crecimiento, clean code.

Introducción

En la actualidad, cada uno de los pragmáticos pertenecientes a las diferentes áreas de conocimiento, requieren de una evaluación de habilidades técnicas, la cual es llevada a cabo por los líderes del área de conocimiento específica; con los resultados obtenidos en esta valoración, además de categorizar al pragmático entre los diferentes niveles de seniority, el líder debe indicarle, según las fortalezas y brechas de conocimiento encontradas, cuál es la ruta de aprendizaje que este debe seguir para continuar con su proceso de crecimiento y así pasar al siguiente nivel; el proceso anteriormente mencionado se lleva a cabo de una manera muy manual, puesto que la valoración se lleva a cabo usando plantillas poco automatizadas y además los resultados son compartidos con el pragmático por medio del correo electrónico corporativo.

Por otro lado, los líderes de las diferentes áreas de conocimiento llevan a cabo una gestión y control de sus áreas por medio de herramientas ofimáticas, de manera que el proceso de consulta de los integrantes, agrupación y filtro por características tales como especialidad, nivel de seniority, entre otras, resulta engorroso y por tanto, el hecho de designar un pragmático a un proyecto, en base a los requerimientos de este, es arduo y demorado.

Así pues, PRAGMA S.A. propone el desarrollo de una aplicación web llamada “Mapa de Crecimiento”, en la cual los líderes de las diferentes áreas de conocimiento puedan acceder a información unificada y resumida que proporciona una idea de la evolución de los diferentes integrantes del área de conocimiento específica por medio de representaciones gráficas de datos estadísticos, además se le permitirá al líder conocer de manera directa el nivel técnico en que se encuentran cada uno de los pragmáticos, sus especialidades, el proyecto al que pertenece, entre otras características.

Con esta aplicación también se busca que los pragmáticos, puedan saber con precisión cómo se encuentran posicionados en su carrera en cuanto al nivel técnico por medio de un proceso de valoración en el cual se detectan o identifican las brechas y fortalezas que se presentan para luego proporcionarles una ruta de aprendizaje a seguir para continuar con su proceso de crecimiento.

Por consiguiente, en este proyecto se desarrolló la interfaz de usuario (frontend) de la aplicación web “Mapa de Crecimiento”, que por medio de una correcta integración con el backend, el cuál fue desarrollado de manera paralela por otros integrantes del equipo, le proporcionó a los pragmáticos un espacio para determinar dónde están, para dónde van y una idea de como evolucionar sus habilidades técnicas; esto entregándoles de manera automatizada la ruta a seguir para su proceso de crecimiento luego de realizar la valoración de este. También se les proporcionó a los líderes las herramientas necesarias para tener un mejor control de sus respectivas áreas de conocimiento, de manera que se les facilitará llevar a cabo las tareas de gestión del personal, permitiéndoles filtrar y agrupar la información para así llevar a cabo la conformación de equipos de trabajo más eficientes y al nivel de los requerimientos de los diferentes proyectos.

1 Objetivos

1.1 Objetivo general

Desarrollar una interfaz de usuario para la aplicación web “Mapa de Crecimiento” de la compañía PRAGMA S.A, haciendo uso de las tecnologías de desarrollo web HTML, CSS y Javascript, para que los líderes de las áreas de conocimiento cuenten con las herramientas que les faciliten la gestión de los diferentes equipos y que los pragmáticos conozcan la ruta de aprendizaje requerida para continuar con su proceso de crecimiento.

1.2 Objetivos específicos

- Implementar los diferentes componentes visuales de la interfaz de usuario por medio de las tecnologías de desarrollo web HTML, CSS y Javascript, de manera que estos sean reusables y tengan una correcta conexión y comunicación con el backend.
- Integrar todos los componentes de la interfaz de usuario en las diferentes páginas que se tendrán utilizando técnicas de adaptación, para así construir la aplicación y proporcionarle la funcionalidad requerida.
- Realizar pruebas funcionales y de renderizado, tanto de manera individual como integral de los diferentes componentes visuales desarrollados, utilizando herramientas de análisis y prueba de código, buscando así, garantizar que se cumpla con el correcto flujo de la información y la buena experiencia de usuario en la navegación por la aplicación.

2 Marco teórico

El desarrollo web se considera como el proceso de creación y mantenimiento de páginas o aplicaciones web. Esto puede implicar un gran campo de acciones o tareas a desempeñar, desde la codificación y el diseño del sitio que da lugar a la apariencia, hasta la gestión de contenidos y la administración del servidor web que es determinante para el buen funcionamiento de las aplicaciones web. Para llevar a cabo estas tareas, se debe utilizar diferentes lenguajes para el marcado y estilizado del contenido, como lo son HTML y CSS, y también otros lenguajes de programación como PHP y JavaScript para definir la funcionalidad y la lógica de la aplicación web; todos estos en conjunto ayudan a escribir programas que permiten que el sitio o la aplicación web sea práctica, dinámica y ágil.

Un desarrollador web además de ser el encargado de desarrollar soluciones de software basadas en la web, lo que se resume en la creación de aplicaciones web, debe llevar a cabo la planificación del diseño y la arquitectura de la aplicación, seleccionando las tecnologías, librerías y patrones de diseño a emplear que mejor se adapten a las necesidades o requisitos de la aplicación para lograr que esta sea mantenible y escalable, ya que se debe tener en cuenta el posible desarrollo posterior, el mantenimiento y la posterior optimización de las aplicaciones web. Los desarrolladores web deben garantizar que el sitio web cumpla los requisitos del sistema, las especificaciones técnicas y las necesidades del cliente, tanto para la parte visible para los usuarios (frontend) como para la utilizada por los administradores del sitio (backend)(Tekla, 2022).

2.1 Frontend

El “frontend” es básicamente la rama del desarrollo web que se encarga de todo lo que el usuario puede ver en el momento en que ingresa a algún sitio web, como lo es el estilo de tipografía, los colores de los elementos, los efectos visuales, la adaptación a las distintas pantallas, los movimientos y las animaciones de los elementos, etc.

Por tanto, el desarrollo frontend es aquel que se encarga de los aspectos visuales, funcionales y de la usabilidad de los diferentes componentes del sitio, de manera que en este desarrollo se debe tener en cuenta factores como la experiencia de usuario relacionada a la interactividad y la navegación del sitio y la interfaz de usuario que trata aspectos tales como el menú de navegación, colores, estilos de letras, tamaño de fuente, botones, entre otros. En tal sentido, este desarrollo se encarga de las interacciones directas entre el usuario y la aplicación web y es por esto que se conoce como “desarrollo del lado del cliente” (Seguro, N, 2021).

En este proceso se utilizan diferentes tecnologías como lo son HTML, CSS y algún framework del lenguaje de programación Javascript tales como Angular o Vue JS.

El frontend a su vez se podría segmentar entre los diseñadores Web/UI/UX¹, que se encargan fundamentalmente del diseño y el comportamiento de las interfaces de usuario, y los desarrolladores web cuyo trabajo consiste en plasmar la idea presentada por el diseñador UI/UX, con elementos propios de un sitio web, para dotar de interactividad y lógica a la interfaz y hacer comunicaciones con los servidores.

Esto último, que parece muy simple, en realidad es cada vez más complicado debido a que las aplicaciones web de hoy en día deben comportarse como cualquier otra aplicación tradicional de escritorio o dispositivo móvil, por tanto es necesario que conserven esas cualidades de inmediatez, agilidad, estabilidad, entre otras capacidades que estas poseen. Por ello, en la actualidad ser desarrollador web frontend implica dominar un montón de conceptos, tecnologías, APIs, bibliotecas y herramientas (Alarcón, J.M, 2021).

2.1.1 HTML

El Lenguaje de Marcado de Hipertexto (HTML) no es un lenguaje de programación sino que es un lenguaje de marcado que define la estructura del

¹ Entiéndase UI como Interfaz de Usuario y UX como Experiencia de Usuario

contenido de un sitio web y consiste en una serie de elementos que se utilizan para encerrar diferentes partes del contenido y lograr que estos se vean o comporten de una determinada manera. Las etiquetas que cubren o encierran un contenido, permiten cambiar el estilo de la tipografía, cambiando a cursiva, a negrilla, aumentar o reducir o aumentar el tamaño o incluso, pueden hacer de una palabra o una imagen un hipervínculo a otro sitio, estas y otras características es posible determinar por medio de las etiquetas de encierre que ofrece HTML.

2.1.2 CSS

El CSS, también conocido como “hojas de estilo en cascada”, se encarga de darle estilo al sitio web. Aunque no es considerado un lenguaje de programación, es por medio de código CSS que se puede lograr que un sitio web se vea de una manera específica, buscando que se vea bien, se sienta bien y sea congruente a lo largo del contenido, esto se logra modificando aspectos de diseño tales como la forma, tamaño, colores, tipografías de textos, comportamiento ante cambios en el tamaño de la pantalla en la que se está visualizando el contenido, transiciones, animaciones, etc.

2.1.3 JavaScript

JavaScript es un lenguaje de programación interpretado que se encarga del comportamiento de una página web y de la interactividad con el usuario, llevando a cabo tareas tales como la validación de los datos enviados por el usuario en un formulario, creación de menús, botones y otros elementos interactivos, asegurándose de que el sitio web se comporta como se desea, haciendo posible la creación de páginas web dinámicas en las que el usuario acceda a información personalizada, convirtiéndolas en páginas interactivas y reactivas.

2.1.3.1 Framework Javascript

En el ámbito del desarrollo del software, el término “JavaScript Framework” significa una biblioteca que proporciona a los desarrolladores plantillas preconstruidas y código JavaScript pre-escrito para tareas de programación estándar, y es esta la razón por la cual se han convertido en una parte vital y esencial del desarrollo web frontend moderno, ya que proporcionan a los desarrolladores herramientas probadas y testeadas para la creación de aplicaciones web interactivas y escalables, aceleran el flujo de trabajo de desarrollo y permite aplicar las mejores prácticas de una manera fácil y fluida.

Uno de los frameworks JavaScript más poderoso, eficiente, robusto y además de código abierto es Angular, que extiende el HTML a la aplicación e interpreta los atributos para realizar el enlace de datos. Google utiliza este framework y está implementado para usarlo en el desarrollo de SPAs.

2.1.4 Pruebas unitarias

Las pruebas unitarias o unit testing son una forma de comprobar que un segmento o fragmento de código funciona correctamente. Estas consisten en aislar una parte del código y comprobar su correcto funcionamiento. Dicho en otras palabras, son pequeños tests que validan el comportamiento de un objeto y la lógica del mismo.

Estas pruebas suelen realizarse durante la fase de desarrollo de aplicaciones de software. Normalmente las llevan a cabo los desarrolladores y la realización de estas permiten lo siguiente:

- Demostrar que la lógica del código está en buen estado y que funcionará en todos los casos.

- Aumentar la legibilidad del código y ayudar a los desarrolladores a entender el código base, lo que facilita hacer cambios más rápidamente.
- Utilizarlas como documentación del proyecto
- La calidad final del código mejorará ya que, al estar realizando pruebas de manera continua, al finalizar el código será limpio y de calidad.

2.2 Backend

El backend es la parte o rama del desarrollo web encargada de que toda la lógica de una página tenga sentido o que funcione, ya que los elementos visuales son incapaces de funcionar por sí mismos. Entre todas las funciones que tiene el backend se destacan las siguientes:

- Acceder a la información que se solicita a través de la aplicación por parte del usuario: En este punto, el backend debe ser capaz de acceder a las bases de datos, encontrar la información precisa que el usuario requiere y además acceder a esta de manera segura.
- Combinar la información encontrada y transformarla de manera que esta resulte útil para el usuario.
- Devolver la información al usuario: El backend debe enviar la información resultante de vuelta al usuario empleando de por medio una API.

2.2.1 API

Las API son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos.

Entre todos los tipos de API que existen, se destacan las API de REST.

2.2.1.1 API de REST

Estas son las API más populares y flexibles que se encuentran en la web actualmente. El cliente envía las solicitudes al servidor como datos, de

manera que el servidor utiliza esta entrada del cliente para iniciar funciones internas y así devolver los datos de salida al cliente.

REST significa transferencia de estado representacional y esta define un conjunto de funciones como GET, PUT, DELETE, etc. que los clientes pueden utilizar para interactuar con los diferentes datos que puedan haber en el servidor, permitiendo llevar a cabo acciones como obtener algún dato o recurso o incluso agregar, editar o eliminar datos. Los clientes y los servidores intercambian datos mediante el protocolo de transferencia de hipertexto HTTP.

La principal característica de la API de REST es la ausencia de estado. La ausencia de estado significa que los servidores no guardan los datos del cliente entre las solicitudes. Las solicitudes de los clientes al servidor son similares a las URL que se escriben en el navegador para visitar un sitio web. En la siguiente figura se presenta la manera en que se lleva a cabo la comunicación entre un cliente y un servidor utilizando la arquitectura del modelo API REST.

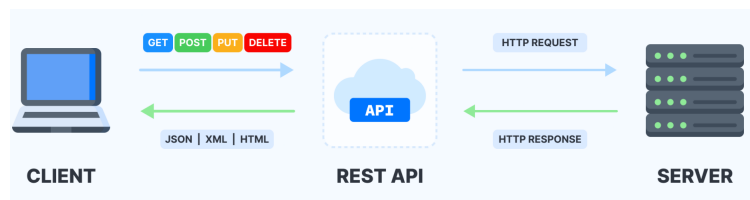


Figura 1. Arquitectura API REST.

2.2.2 HTTP

El Protocolo de Transferencia de HiperTexto (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP.

HTTP se basa en sencillas operaciones de solicitud/respuesta, donde un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud; el servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Este protocolo establece una serie de métodos de petición que se utilizan para indicar la acción que desea que se lleve a cabo sobre el recurso específico. Los métodos o verbos más comunes son los siguientes:

- GET: Empleado para obtener una representación de un recurso específico.
- POST: Se utiliza para enviar información y que esta sea procesada en un recurso específico.
- PUT: Utilizado para actualizar contenido pero también para crearlo.
- DELETE: Sirve para eliminar un recurso especificado en la petición.

Para cada petición el servidor devuelve un código de respuesta con la información del recurso que haya sido solicitado, este código indica lo que ha pasado con la petición y el resto del contenido de la respuesta dependerá del valor de este código. Los códigos de respuesta son identificados por números de 3 dígitos que a su vez están agrupados en base al primer dígito a la izquierda de la siguiente manera:

- 1xx: respuestas informativas, la petición se está procesando
- 2xx: operación exitosa, la petición ha sido procesada correctamente
- 3xx: respuesta de redirección, indica que el cliente necesita realizar más acciones para finalizar la petición
- 4xx: indica que hubo un error causado por el cliente, el servidor recibe la solicitud pero esta no se puede llevar a cabo

- 5xx: error ocasionado por el servidor, se da cuando la solicitud correspondiente está temporalmente deshabilitada o es imposible llevar a cabo.

2.3 Código Limpio

Dentro del ámbito del desarrollo de software, hay una serie de buenas prácticas que se deben tener en cuenta para garantizar la mantenibilidad y escalabilidad de los proyectos; estas buenas prácticas son una serie de lineamientos, principios o técnicas de desarrollo que permiten escribir lo que se conoce como “código limpio”. Se considera que un código es limpio cuando es fácil de leer y entender, por tanto, si un código o programa resuelve los problemas sin agregar complejidad innecesaria, permitiendo que el mantenimiento o las adaptaciones, por algún cambio de requerimiento, sean tareas más sencillas, entonces estamos hablando de “clean code” o “código limpio” (Morales, K.I, 2021). A continuación se describen algunas de las características más importantes a tener en cuenta para lograr escribir un código limpio:

1. Los nombres que se usen para las variables y las funciones deben ser claros, descriptivos y significativos para ayudar a que la lectura del código sea más fluida y favorecer un rápido entendimiento.
2. Una regla primordial es que las funciones siempre deben ser lo más pequeñas posibles y tener una única responsabilidad, con el fin de facilitar su comprensión y evitar que sean más propensas a los cambios.
3. Es recomendable utilizar objetos inmutables porque vuelve al código más predecible, ya que se eliminan los efectos secundarios. Así nos aseguramos de que los objetos no se modifiquen en lugares inesperados. Esto los vuelve mucho más seguros de usar en aplicaciones concurrentes.
4. Tratar de hacer el código lo más simple y sencillo posible, evitando cualquier complejidad innecesaria que dificulte la comprensión del código, sin embargo, teniendo en cuenta que la legibilidad del código siempre es más importante que su concisión.

5. Evitar repetir sin motivo segmentos de código, por tanto, se debe buscar crear funciones que realicen tareas específicas que se lleven a cabo de manera repetitiva dentro del código.

2.4 Atomic Design o Diseño Atómico.

Atomic Design no es nada más que un sistema de diseño o una metodología empleada para el diseño de interfaces de productos digitales, la cual se basa en la filosofía de la construcción de interfaces de usuario por medio del desarrollo de componentes que permitan construir cada una de las pantallas de la aplicación web.

Cuando desarrollamos estos componentes, estamos creando un diseño modular que nos permite sistematizar esos elementos para que puedan ser replicados y utilizados cuantas veces sean necesarias y además hacerlo de diferentes maneras, pudiendo incluso, modificar determinadas características.

Con esta metodología lo que se busca es descomponer una interfaz web en unidades pequeñas, y reutilizar estos componentes a través de la composición para generar elementos más grandes y potentes que compongan todo el diseño.

La metodología establece que a la hora de diseñar un producto nos encontramos con cinco niveles de detalle que tenemos que tener en cuenta: átomos, moléculas, organismos, plantillas y páginas. La relación entre cada uno de estos niveles de componentes es el siguiente:

- a. Átomos: Son la unidad básica, unidades indivisibles que tienen sentido por sí mismas.
- b. Moléculas: Están conformadas por grupos de dos o más átomos y cumplen solamente una función concreta.
- c. Organismos: Son moléculas funcionando juntas como una unidad que pueden formar desde organismos muy pequeños a organismos muy grandes.
- d. Plantillas: consisten en grupos de organismos colocados de manera conjunta para formar páginas. Estas permiten definir la estructura que tendrá el contenido de la página, estableciendo la disposición de los organismos ya creados. Es aquí donde comenzamos a ver la coherencia del contenido y el diseño, junto a temas de

estructura como la retícula. La función principal de estas plantillas es dar contexto a las pequeñas unidades, jerarquizarlas, dar orden a la lectura y disponer los contenidos. En definitiva comenzar a crear lo que conocemos como wireframes.

- e. Páginas: Las páginas o pantallas son el entregable final, la representación diseñada de toda la interfaz del producto. En el diseño de las páginas cobran vida finalmente todos los componentes que se han creado ya que se unen y se comunican para llevar a cabo una lógica determinada.

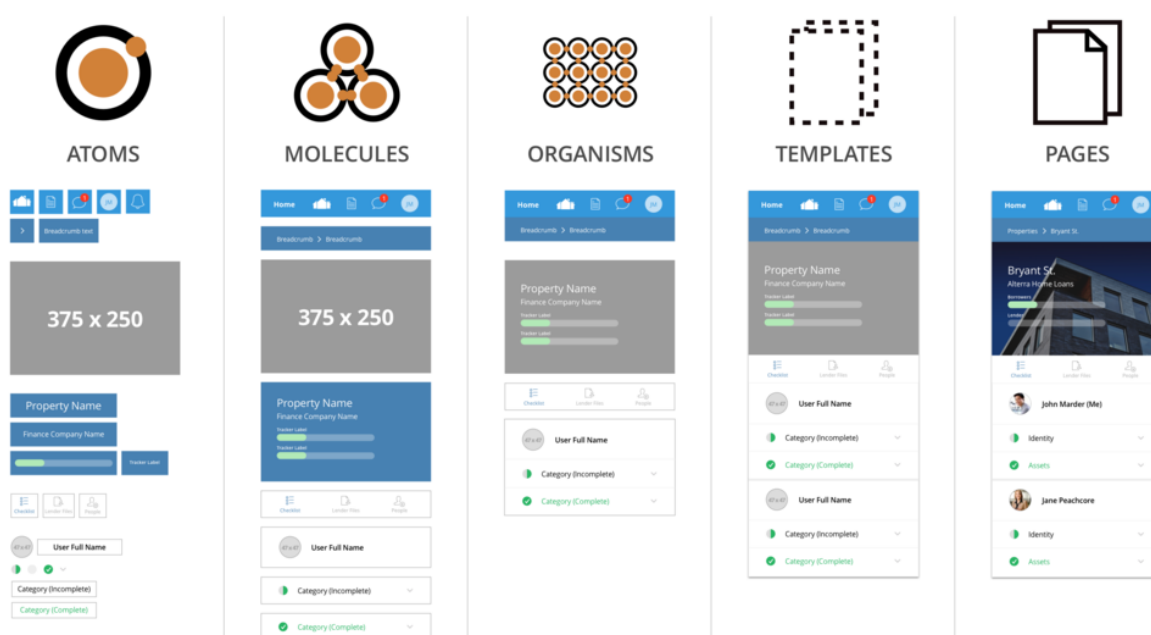


Figura 2. Ejemplo de aplicación de Atomic Design para una interfaz de usuario.

En la figura anterior (Figura 2) se puede observar como en una interfaz de usuario, los elementos más básicos como botones, íconos, figuras, entre otros, se denominan átomos, de manera que la unión de estos átomos permiten formar las moléculas que representan componentes más complejos dentro de la interfaz de usuario como un menú de navegación, una tabla, o un bloque de texto con íconos y botones como un buscador, una tarjeta o card con imagen y descripción. A su vez, la unión de estas moléculas conforman los organismos, los cuales se pueden ver como el header o encabezado de la aplicación que normalmente contiene el menú de navegación, el logo de la aplicación y el cuadro de búsqueda, secciones con contenido como por ejemplo un conjunto de tarjetas

de productos organizadas de determinada manera. Por otro lado, las templates o plantillas son las encargadas de definir la estructura en la cual se mostrará el contenido y finalmente, la unión de todos los átomos, moléculas y partículas bajo la estructura definida por la plantilla conforman lo que se denomina como páginas.

2.5 Sistema de control de versiones Git

Git es un software que permite el mantenimiento de versiones de aplicaciones cuando estas poseen gran cantidad de ficheros o archivos de código fuente; esta lo que permite es llevar un registro de las modificaciones que ha sufrido el proyecto relacionadas a agregar, eliminar o editar ficheros; este registro es de gran utilidad ya que posibilita coordinar el trabajo que varias personas puedan realizar sobre el proyecto. Una de las principales características de Git es la posibilidad de la creación de ramas, las ramas en Git son una división del estado del código, lo cual permite crear nuevos caminos a favor de la evolución del código; cuando se desea añadir una nueva funcionalidad o solucionar un error en la aplicación, lo que se hace es generar una nueva rama para alojar estos cambios. Esto hace que resulte más complicado que el código inestable se fusione con el código base principal, y te da la oportunidad de limpiar tu historial futuro antes de fusionarlo con la rama principal.

La generación de estas ramas se hace de manera estratégica para garantizar un correcto flujo de trabajo a la hora de generar incrementales para una aplicación de manera que la rama principal siempre esté lo más limpia y estable posible al procurar por la mitigación de errores en esta.

2.5.1 Git-Flow

Git-Flow es un flujo de trabajo basado en Git que brinda un mayor control y organización en el proceso de integración continua en el desarrollo de un proyecto. Este permite que los integrantes de un proyecto puedan desarrollar diferentes características o funcionalidades de la aplicación de manera paralela sin interrumpir o interferir entre ellos. Git-flow propone una serie de ramas y nombres

para las mismas de manera que sea fácil entender la funcionalidad de cada una de ellas (Lopez, P, 2021).

2.5.1.1 Rama master:

Es aquella rama que contiene el código de producción, es una principal, la cual no permite recibir código directamente sino utilizando una rama de por medio, a excepción de los hotfix que son cambios que necesitan hacerse de manera urgente para corregir algún error que se tenga en producción.

2.5.1.2 Rama Hotfix:

Son ramas que se utilizan para corregir errores críticos encontrados en producción. Nacen siempre desde la rama master y se fusionan con la rama master y develop, con el objetivo de poner el hotfix en producción y que también esté disponible para nuevos evolutivos en la rama de desarrollo.

2.5.1.3 Rama QA:

En esta rama, el grupo de analista de calidad se encarga de probar toda la aplicación, con el objetivo de verificar el correcto funcionamiento de cada una de las funcionalidades desarrolladas por el equipo de desarrollo.

2.5.1.4 Rama Develop:

También conocida como rama de desarrollo, es a partir de la cual saldrán las diferentes ramas en las que se desarrollarán las nuevas características o funcionalidades de la aplicación, esta rama siempre contiene los últimos cambios desarrollados para la próxima versión de la aplicación, ya que es allí donde se unen o se fusionan los desarrollos que se han llevado a cabo por los participantes del proyecto.

2.5.1.5 Rama Feature:

Son las ramas en las que los desarrolladores trabajan, ya que allí llevarán a cabo el desarrollo de de las nuevas funcionalidades de la aplicación, una vez el desarrollo está terminado, se fusiona esta rama con la rama Develop y se elimina luego de hacer esta operación.

2.5.1.6 Rama Bugfix:

Son ramas que nacen a partir de la rama develop y se utilizan para corregir errores o solucionar problemas de la aplicación que aún no han llegado a producción.

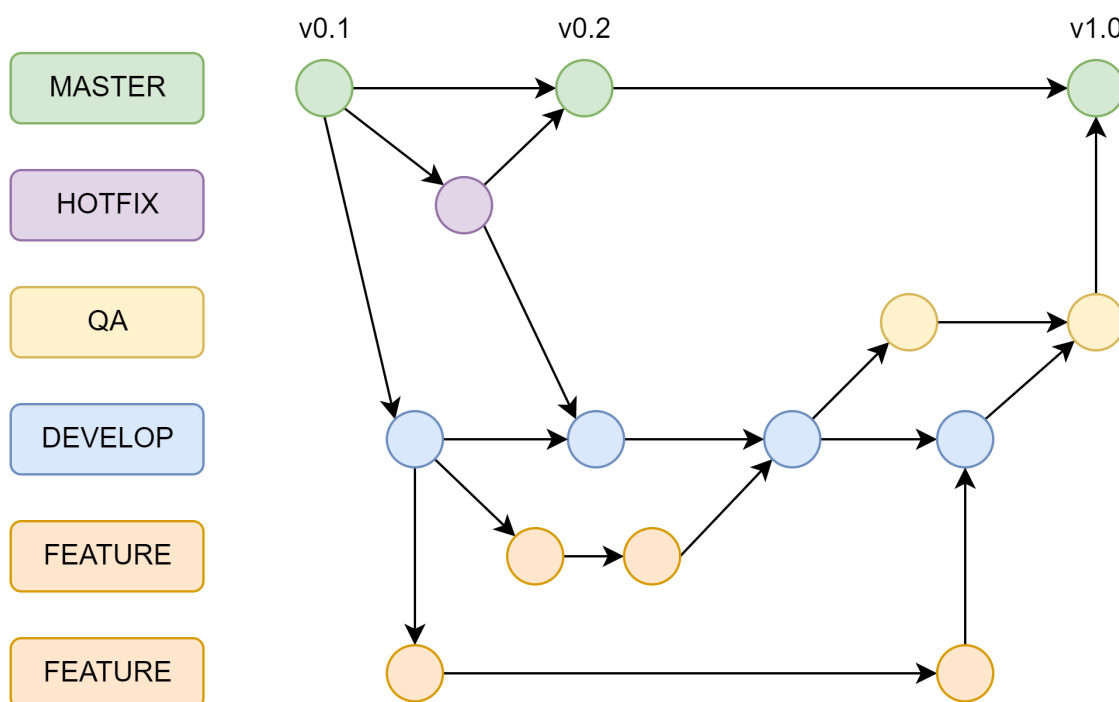


Figura 3. Representación de control de versiones utilizando Git-Flow.

3 Metodología

A continuación se especificará de manera detallada todo el proceso llevado a cabo para el desarrollo de la interfaz de usuario para la aplicación web “Mapa de Crecimiento”.

3.1 Versionamiento del código y control de flujo.

Para comenzar con el desarrollo de la aplicación, primero se decidió utilizar el sistema de control de versiones Git implementado en la plataforma web Gitlab, además, se optó por emplear el modelo de creación de ramas “Git flow” para garantizar la estabilidad y reducción de errores en la rama master que será aquella que contendrá el código que estará en producción.

3.2 Estándares de desarrollo.

Se decidió seguir los principios de código limpio, que involucran una convención de nomenclatura clara, detallada y concisa, funciones con responsabilidad única, tipado de variables para minimizar errores en el procesado de los datos, no repetir código y realizar pruebas unitarias limpias e independientes; por otra parte se realizó una modularización en el enrutamiento para permitir la carga diferida o lazy loading de los módulos de la aplicación y así garantizar su carga o inicialización sólo en el momento en que sea necesario.

3.3 Seguridad de la aplicación.

Se emplearon los “Guards” de Angular para proporcionar la lógica de protección de rutas, de manera que la carga y el acceso a un módulo de la aplicación solo se pudiera hacer cuando el usuario tuviese una sesión iniciada, además, la aplicación fue desarrollada de tal manera que estuviese compuesta de múltiples módulos que bajo la estrategia de lazy loading o carga perezosa, sólo se cargan cuando se requieran, lo cual proporciona mayor nivel de seguridad a la aplicación.

3.4 Marco de trabajo.

El desarrollo de software se llevó a cabo siguiendo el marco de trabajo Scrum, en la cual segmenta el producto o proyecto en funcionalidades que se van desarrollando e implementando por medio de la realización de tareas también conocidas como historias de usuario, cuyo desarrollo aporta un valor significativo al proyecto generando poco a poco un incremento del mismo y que finalmente, todas en conjunto conforman el producto con todas las funcionalidades requeridas. Para la etapa de desarrollo del proyecto bajo este marco de trabajo, se ejecutaron ciclos temporales de dos semanas, en las cuales se desarrollaron alrededor de 3 historias de usuario. A continuación se presentan las historias de usuario desarrolladas en este proyecto con su respectiva explicación funcional y requisitos o criterios de aceptación necesarios para la entrega e integración de cada una de estas con las demás para ir conformando la aplicación Mapa de Crecimiento.

3.4.1 Inicio de sesión en la aplicación.

Yo como líder de chapter o embajador necesito iniciar sesión en la aplicación Mapa de Crecimiento para tener un único usuario dentro de la aplicación.

Criterios de aceptación:

- a. La implementación debe guiarse por el diseño presentado por el equipo de UI.
- b. El acceso debe hacerse con la cuenta de google del usuario.
- c. Al presionar el botón “Inicia sesión con Google” se debe redirigir al sitio de autenticación de google.
- d. Solo se debe permitir el acceso a usuarios con correos pertenecientes al dominio de pragma.

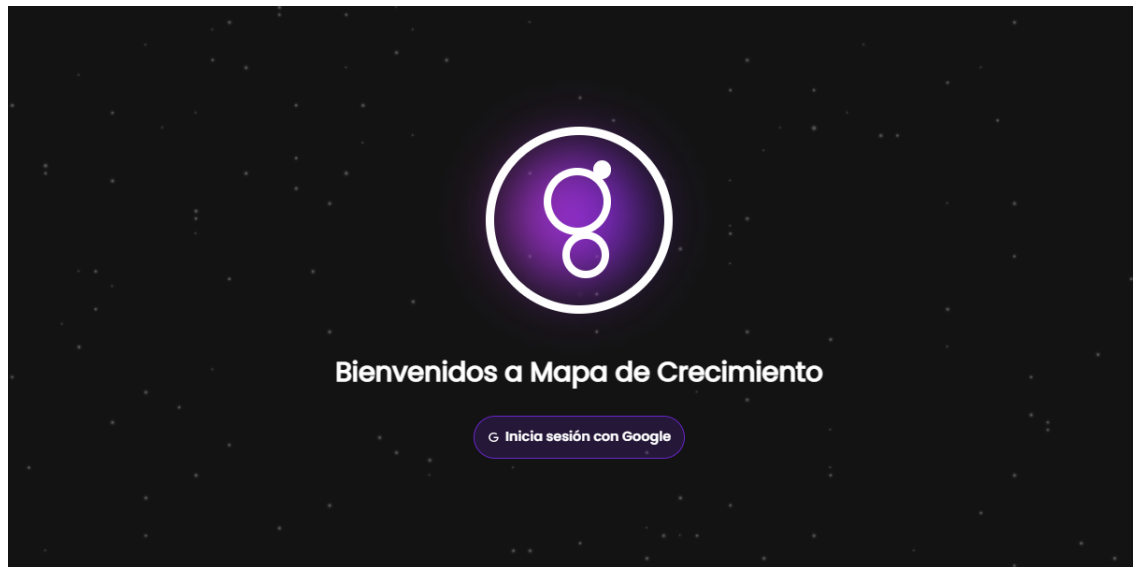


Figura 4. Diseño de la pantalla de inicio de sesión de la aplicación presentado por el equipo de UI.

3.4.2 Header de la aplicación.

Yo como líder de chapter o embajador necesito que la aplicación tenga un header en el que se pueda visualizar un menú de navegación global a toda la aplicación y que también contenga información relacionada al usuario que tenga su sesión iniciada para poder navegar entre las páginas de la aplicación y llevar a cabo las tareas deseadas.

Criterios de aceptación:

- La implementación debe guiarse por el diseño presentado por el equipo de UI.
- Que sea posible navegar entre las diferentes páginas
- Se debe poder reconocer en qué página se encuentra como se muestra en el diseño
- El logo de la derecha debe contener la imagen del usuario que tenga la sesión iniciada



Figura 5. Diseño del header de la aplicación presentado por el equipo de UI.

3.4.3 Card con resumen de información de seniority y gráfica de dona del chapter.

Yo como líder de chapter o embajador necesito conocer información resumida y representada de manera gráfica sobre la evolución de los diferentes niveles de seniority en el chapter.

Criterios de aceptación:

- La implementación debe guiarse por el diseño presentado por el equipo de UI.
- La información mostrada debe corresponder a la evolución de los niveles de seniority en el mes actual.
- La información representada en la gráfica tipo dona debe representar el resumen de la cantidad de pragmáticos en el chapter y su distribución en cada uno de los niveles de seniority.



Figura 6. Diseño de card presentado por el equipo de UI.

3.4.4 Brechas del chapter.

Yo como líder de chapter o embajador necesito poder visualizar las brechas más comunes en mi chapter para tener control tomar decisiones con operaciones y levantar alertas de necesidades de habilitación.

Criterios de aceptación:

- a. La implementación debe guiarse por la imagen definida por el equipo UI.
- b. Se deben analizar todas las habilidades técnicas que se tienen definidas en la matriz de valoración de los pragmáticos.
- c. Las brechas que se visualizarán, serán todas aquellas habilidades del chapter que en promedio tengan una calificación entre 1 y 2.



Figura 7. Diseño de tarjeta de Brechas del chapter presentado por el equipo de UI.

3.4.5 Diagrama de barras con información de evolución de nivel de seniority en los diferentes meses.

Yo como líder de chapter o embajador necesito poder visualizar información agrupada por meses sobre la evolución de los pragmáticos en los diferentes niveles de seniority.

Criterios de aceptación

- a. La implementación debe guiarse por el diseño presentado por el equipo de UI.
- b. Se debe poder mostrar la cantidad de pragmáticos que fueron categorizados en determinado nivel de seniority en el mes específico.

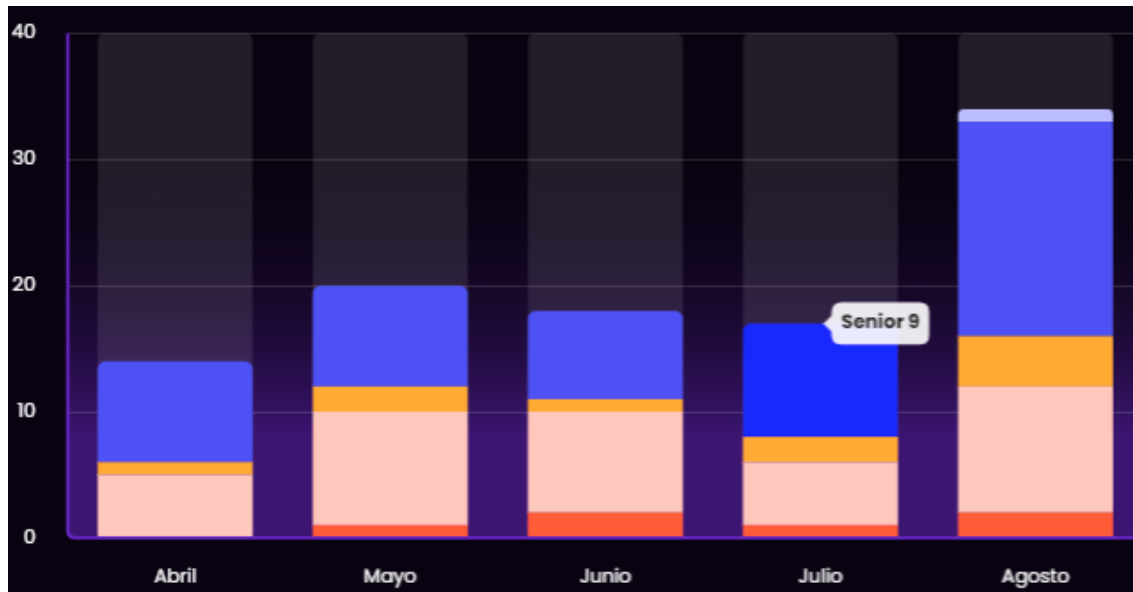


Figura 8. Diseño de diagrama de barras presentado por el equipo de UI.

3.4.6 Lista de pragmáticos en el chapter.

Yo como líder de chapter o embajador necesito ver un listado de los pragmáticos para poder acceder fácilmente a la información unificada y prioritaria de los pragmáticos de mi chapter.

Criterios de aceptación:

- a. La implementación debe guiarse por el diseño presentado por el equipo de UI.
- b. Los pragmáticos deben estar ordenados por próximas valoraciones, las valoraciones vencidas deben aparecer primero y luego las próximas valoraciones en base a la cercanía a la fecha en que se ingresa a la aplicación.
- c. Las columnas que deben estar en la tabla son: Pragmaticos, Cargo nómina Seniority, Marca/Proyecto, Próxima valoración.
- d. Tomar la fecha de comparación de un Ntp global.
- e. La lista de pragmáticos debe tener un paginador.
- f. Se deben mostrar por defecto un total de 25 pragmáticos por página.

Pragmáticos	Cargo nómina	Seniority	Marca/proyecto	Próxima valoración
Andreina Yajaira Francesa Serrano	Backend Java Full Stack	Senior L3	Comfama - lorem ipsum	Valoración vencida
Andreina Yajaira Francesa Serrano	Backend Java Full Stack	Senior L3	Comfama - lorem ipsum	Valoración vencida
Andreina Yajaira Francesa Serrano	Backend Java Full Stack	Senior L3	Comfama - lorem ipsum	Valoración vencida
Andreina Yajaira Francesa Serrano	Backend Java Full Stack	Senior L3	Comfama - lorem ipsum	Hoy
Andreina Yajaira Francesa Serrano	Backend Java Full Stack	Senior L3	Comfama - lorem ipsum	Mañana
Andreina Yajaira Francesa Serrano	Backend Java Full Stack	Senior L3	Comfama - lorem ipsum	21 / Jun / 2022
Andreina Yajaira Francesa Serrano	Backend Java Full Stack	Senior L3	Comfama - lorem ipsum	35 / Jun / 2022

Figura 9. Diseño de lista de pragmáticos presentado por el equipo de UI.

3.4.7 Buscador de pragmáticos en el chapter.

Yo como líder de chapter o embajador necesito buscar pragmáticos pertenecientes a mí chapter para poder acceder fácilmente a la información unificada y prioritaria de uno o varios pragmáticos en el chapter.

Criterios de aceptación:

- La implementación debe guiarse por el diseño presentado por el equipo de UI.
- Se debe poder buscar por: Nombre, proyecto, seniority y cargo en nómina.
- La acción del buscador se activa en el evento "onchange" del input, es decir, mientras escribo, se van buscando los ítems. Desaparecen las opciones de la lista de pragmáticos que no hacen match con la búsqueda.
- Si no hay resultados para la búsqueda del usuario, debe aparecer el mensaje "No se encontraron resultados para '<palabraQueBuscaElUsuario>'".

- e. Cuando se empieza a escribir, debe aparecer una X para cancelar la búsqueda y eliminar el texto si lo requiere el usuario.

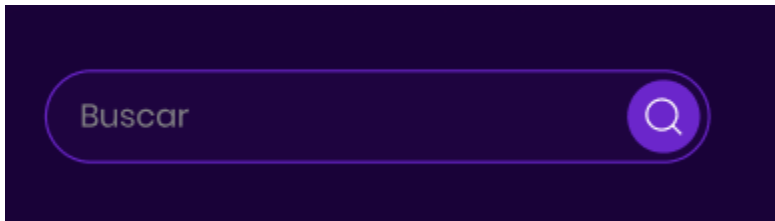


Figura 10. Diseño de buscador de pragmáticos presentado por el equipo de UI.

3.4.8 Filtro en la información de niveles de seniority.

Yo como líder de chapter o embajador necesito filtrar la distribución de seniority del chapter para tomar decisiones con operaciones y levantar alertas de necesidades de habilitación.

Criterios de aceptación:

- a. La implementación debe guiarse por el diseño presentado por el equipo de UI.
- b. Filtros de la izquierda al seleccionar los filtros de seniority se presentan sólo los datos de los niveles seleccionados.
- c. Filtro superior maneja el rango de tiempo debe mostrar la grafica va desde 1-12 meses.
- d. Al cargarse el sitio debe mostrar todos los niveles y en rango de tiempo por defecto 6 meses (todos los checkbox des-seleccionados).
- e. Debe agregarse un checkbox para ver todos.
- f. Puedo elegir varios niveles de seniority a la vez.
- g. Debe estar seleccionado por lo menos un filtro de seniority.
- h. Al dar click en el mes se debe desplegar en la parte inferior la información detallada, tal y como se ve en la imagen se deben desplegar los datos según los filtros aplicados.
- i. Si en un mes hay 0 datos se muestran 0 datos (0 junior, 0 trainee, etc.)

- j. Por defecto debe cargarse el mes actual con la información correspondiente.

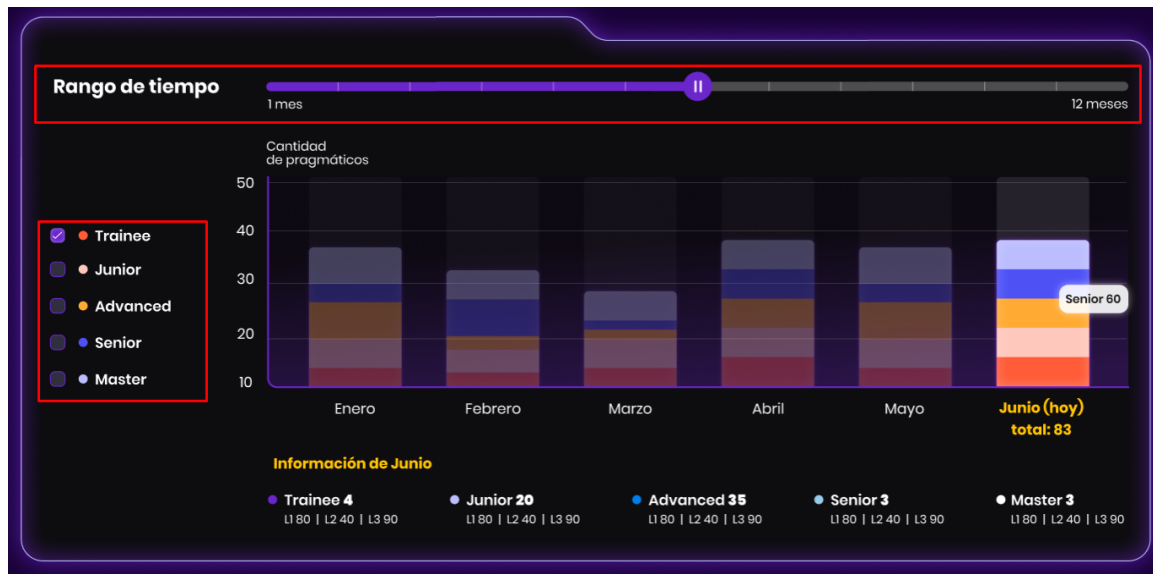


Figura 11. Diseño de filtros presentado por el equipo de UI.

3.4.9 Filtro en la columna “Próxima valoración” en la lista de pragmáticos en el dashboard del chapter.

Yo como líder de chapter o embajador necesito poder filtrar la información de los pragmáticos del chapter para acceder fácilmente a la información unificada y prioritaria de uno o varios pragmáticos en el chapter.

Criterios de aceptación:

- La implementación debe guiarse por la imagen definida por el equipo de UI.
- Se deben resaltar las fechas tal y como se muestra en la imagen, vencidas, hoy y mañana con los colores indicados.
- Las fechas vencidas deben tener un tooltip que al seleccionarse se muestra que día venció la valoración, tal como lo indica la imagen.
- Los datos deben ser dinámicos.

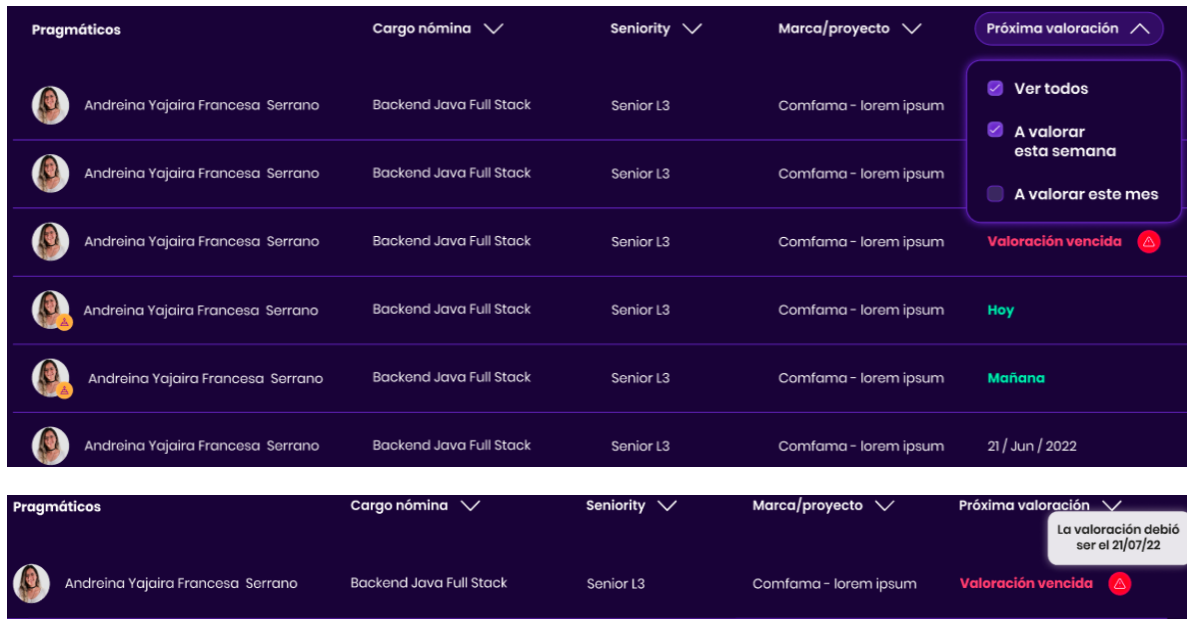


Figura 12. Diseño de filtro de columna “Próxima valoración” presentado por el equipo de UI.

3.4.10 Filtro en la columna “Cargo nómina” en la lista de pragmáticos en el dashboard del chapter.

Yo como líder de chapter o embajador necesito poder filtrar la información de los pragmáticos del chapter para acceder fácilmente a la información unificada y prioritaria de uno o varios pragmáticos en el chapter.

Criterios de aceptación:

- La implementación debe guiarse por la imagen definida por el equipo de UI.
- Se debe poder filtrar la lista de pragmáticos en base al cargo de nómina que tienen los pragmáticos.

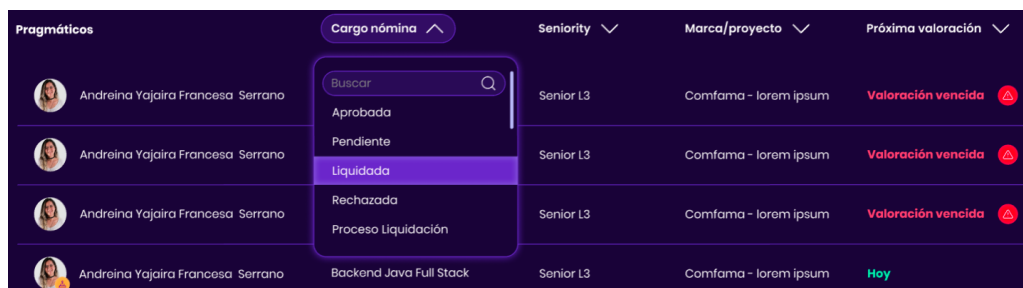


Figura 13. Diseño de filtro de columna “Cargo nómina” presentado por el equipo de UI.

3.4.11 Filtro en la columna “Seniority” en la lista de pragmáticos en el dashboard del chapter.

Yo como líder de chapter o embajador necesito poder filtrar la información de los pragmáticos del chapter para acceder fácilmente a la información unificada y prioritaria de uno o varios pragmáticos en el chapter.

Criterios de aceptación:

- La implementación debe guiarse por la imagen definida por el equipo de UI.
- Se debe poder filtrar la lista de pragmáticos en base al nivel de seniority.

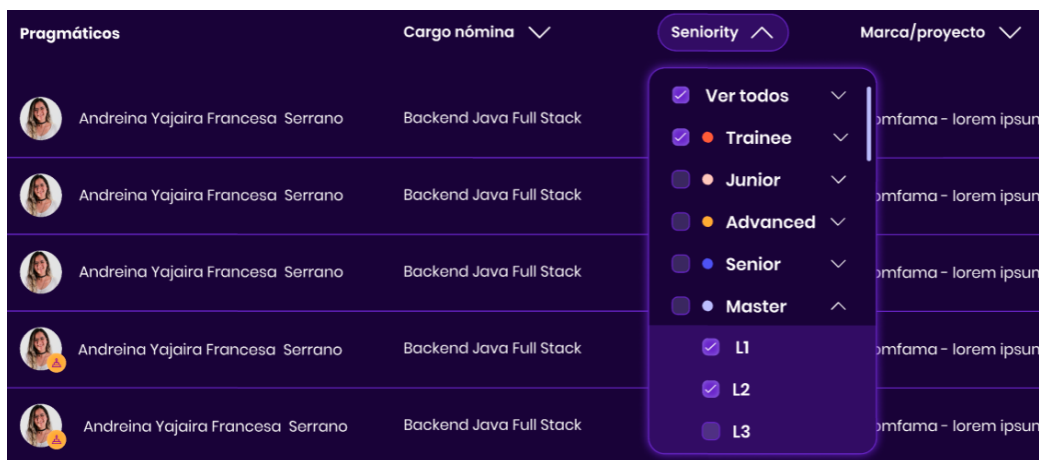


Figura 14. Diseño de filtro de columna “Cargo nómina” presentado por el equipo de UI.

3.4.12 Filtro en la columna “Marca/Proyecto” en la lista de pragmáticos en el dashboard del chapter.

Filtros en Dashboard del chapter (Tabla, Listado) columna “Marca/Proyecto”

Yo como líder de chapter o embajador necesito poder filtrar la información de los pragmáticos del chapter para acceder fácilmente a la información unificada y prioritaria de uno o varios pragmáticos en el chapter.

Criterios de aceptación:

- La implementación debe guiarse por la imagen definida por el equipo de UI.
- Se debe poder filtrar la lista de pragmáticos en base a la marca o proyecto en que se encuentran.

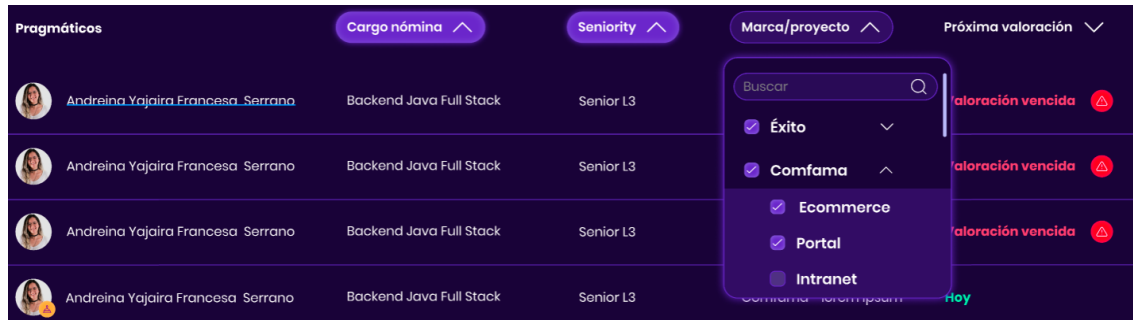


Figura 15. Diseño de filtro de columna “Marca/Proyecto” presentado por el equipo de UI.

3.4.13 Histórico de valoraciones realizadas en el chapter.

Yo como líder de chapter o embajador necesito poder visualizar el historial de valoraciones para conocer los diferentes pragmáticos que han sido valorados y tener un acceso rápido al informe resultante que reportó el valorador en su momento.

Criterios de aceptación:

- La implementación debe guiarse por la imagen definida por el equipo de UI.
- Se debe poder ver la lista de pragmáticos que han sido valorados con las siguientes columnas: Pragmáticos (Nombre del pragmático), Fecha de valoración, Nivel de seniority, Valorador y una última columna que ofrezca la posibilidad de Visualizar el informe que se le realizó al pragmático en la valoración.
- La lista de pragmáticos debe tener un paginador.
- Se deben mostrar por defecto un total de 25 pragmáticos por página.
- Al dar click sobre el texto “Visualizar informe” en la fila de una pragmático específico, se debe abrir una nueva pestaña donde se visualice el informe que se hizo del pragmático en dicha valoración.










Pragmáticos	Fecha de valoración	Nivel de seniority	Valorador	< 1 2 3 ... 250 > 25 por página
 Andreina Yajaira Francesa Serrano	44 / Ago / 2022	Senior L3	Yousef Deivid Alcatrán Zapato	Visualizar informe →
 Andreina Yajaira Francesa Serrano <small>En vacaciones</small>	44 / Ago / 2022	Senior L3	Yousef Deivid Alcatrán Zapato	Visualizar informe →
 Andreina Yajaira Francesa Serrano	44 / Ago / 2022	Senior L3	Yousef Deivid Alcatrán Zapato	Visualizar informe →
 Andreina Yajaira Francesa Serrano	44 / Ago / 2022	Senior L3	Yousef Deivid Alcatrán Zapato	Visualizar informe →
 Andreina Yajaira Francesa Serrano <small>En vacaciones</small>	44 / Ago / 2022	Senior L3	Yousef Deivid Alcatrán Zapato	Visualizar informe →
 Andreina Yajaira Francesa Serrano	44 / Ago / 2022	Senior L3	Yousef Deivid Alcatrán Zapato	Visualizar informe →
 Andreina Yajaira Francesa Serrano	44 / Ago / 2022	Senior L3	Yousef Deivid Alcatrán Zapato	Visualizar informe →
 Andreina Yajaira Francesa Serrano	44 / Ago / 2022	Senior L3	Yousef Deivid Alcatrán Zapato	Visualizar informe →
 Andreina Yajaira Francesa Serrano <small>En vacaciones</small>	44 / Ago / 2022	Senior L3	Yousef Deivid Alcatrán Zapato	Visualizar informe →

Figura 16. Diseño de lista en pantalla de historial de valoraciones realizadas en el chapter.

3.4.14 Buscador de pragmáticos en el historial de valoraciones.

Yo como líder de chapter o embajador necesito buscar pragmáticos dentro del historial de valoraciones para acceder rápidamente a la información relacionada a las valoraciones de los pragmáticos del chapter.

Criterios de aceptación:

- La implementación debe guiarse por el diseño presentado por el equipo de UI.
- Se debe poder buscar por: Nombre, nivel de seniority o valorador.
- La acción del buscador se activa en el evento "onchange" del input, es decir, mientras escribo, se van buscando los ítems. Desaparecen de la lista de pragmáticos las opciones que no hacen match con la búsqueda.
- Si no hay resultados para la búsqueda del usuario, debe aparecer el mensaje "No se encontraron resultados para '<palabraQueBuscaElUsuario>'".
- Cuando se empieza a escribir, debe aparecer una X para cancelar la búsqueda y eliminar el texto si lo requiere el usuario.



Figura 17. Diseño de buscador de pragmáticos en el historial de valoraciones presentado por el equipo de UI.

3.4.15 Cabezote del dashboard del pragmático.

Yo como líder de chapter o embajador necesito obtener fácilmente la información a detalle del pragmático que deseo valorar, para tener claro dónde está, para dónde va, tomar decisiones con operaciones y levantar alertas de necesidades de habilitación.

Criterios de aceptación:

- a. La implementación debe guiarse por la imagen definida por el equipo de UI.
- b. El dashboard del pragmático se debe abrir al dar click en una fila de la lista de pragmáticos del dashboard del chapter.
- c. Al dar click en el botón atrás, vuelve al dashboard del chapter sin los filtros seleccionados.
- d. Si no tiene rol secundario debe decir “No tiene”.
- e. Si no tiene nivel de ingles debe decir “No registra”.
- f. Si no tiene algún dato numérico debe aparecer “No aplica”.
- g. En el botón de volver, debe decir “volver al chapter <Frontend, Backend, Ux>”
- h. Si un correo de pragmático no coincide, enviar alerta.
- i. Si la fecha de valoración está vencida, indicarlo cambiando el color de fondo del recuadro de la parte derecha.

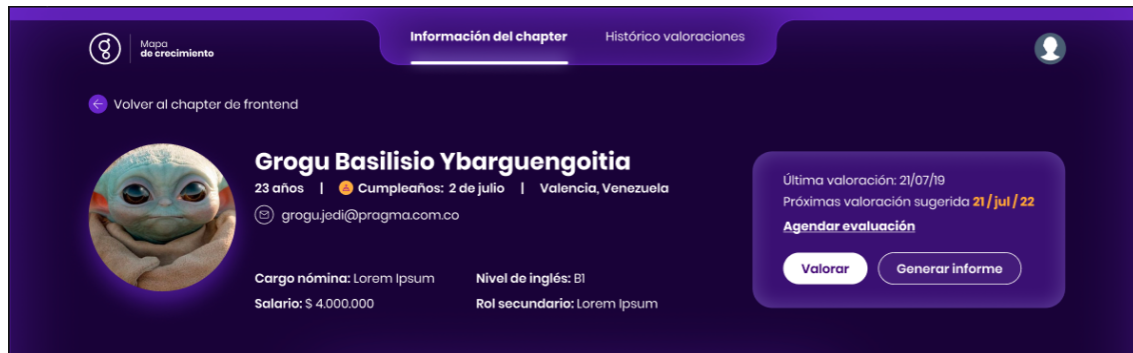


Figura 18. Diseño de cabezote dashboard pragmático presentado por el equipo de UI.

3.4.16 Selección de chapter a visualizar.

Yo como líder de chapter o embajador necesito poder seleccionar los diferentes chapters para poder ver la información de los diferentes chapters a los que puedo pertenecer.

Criterios de aceptación:

- La implementación debe guiarse por la imagen definida por el equipo de UI.
- Al ingresar a la pantalla principal se mostrará el chapter al que el pragmático que inició sesión pertenece.
- El pragmático podrá cambiar el chapter a visualizar por medio de un selector de chapters ubicado al lado del título del chapter.
- Para los pragmáticos que su chapter asociado no se encuentre activo, se les mostrará por defecto el chapter de Frontend.
- El selector es configurable para cargar los chapters activos, inicialmente Frontend y Backend.

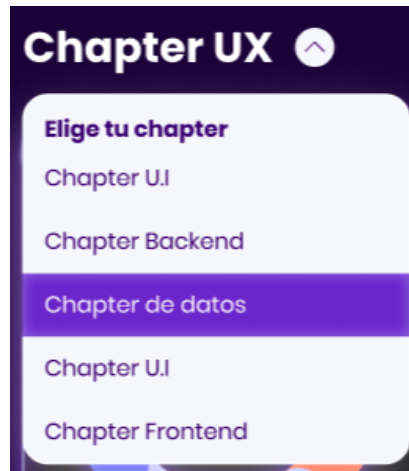


Figura 19. Diseño de selector de chapter presentado por el equipo de UI.

3.4.17 Flujo de valoración del pragmático.

Yo como líder de chapter o embajador necesito poder valorar a los pragmáticos de mi chapter para saber en qué nivel se encuentran, conocer sus fortalezas y brechas y levantar alertas en caso de que sea necesario.

Criterios de aceptación:

- La implementación debe guiarse por la imagen definida por el equipo de UI.
- Al dar en el botón valorar ubicado en el dashboard del pragmático se debe redireccionar a la matriz de valoración del pragmático.
- Se debe presentar un modal con los tips de valoración para el valorador antes de redireccionar al Google Sheet.
- Al presionar el botón “Generar informe” en la matriz de valoración de Google Sheet, se debe guardar los cambios de este documento y almacenar el reporte en base de datos, seguido de esto, debe redireccionar al dashboard del pragmático y notificar el éxito en la generación del reporte con una alerta de tipo Pop-up.



Figura 20. Diseño de modal de tips antes de hacer la valoración presentado por el equipo de UI.

Nombre: Jerry Sanchez
 Proyecto: Talent Experience
 Rol: Analista UX
 Consideración evaluador: Observaciones

	TRAINEE to JUNIOR	JUNIOR to ADVANCED	ADVANCED to SENIOR	SENIOR to MASTER	MASTER to INFINITY	
Diseño centrado en las personas	Revisa los Bases o mapas de flujo de experiencia de usuario centrado en las personas para entender relaciones abstractas y preparar usuarios para su aplicación. (Design Thinking - Flow thinking, define, ideas generation, make, take, deliver - convergencia y divergencia)	Aplica en su cotidianidad metodologías de diseño centrado en las personas para la resolución de los problemas de diseño a los que se enfrenta. (Design thinking, Coble diamond, lean.)	Analiza problemas de diseño complejos y propone soluciones creativas, aplicando la metodología de diseño centrado en el usuario que más funcione en el contexto. Es capaz de enseñar algunos de las metodologías.	De acuerdo al contexto, selecciona y ejecuta artefactos de diseño centrados en las personas (Customer Journey Map, Empathy Map y Buyer Persona) conoce el objetivo de ellos y sabe cómo impactan su proceso de diseño. Identifica la diferencia entre UX, CX, Service Design y Product design.	Analiza problemas de diseño complejos y propone soluciones creativas, aplicando la metodología de diseño centrado en el usuario que más funcione en el contexto. Es capaz de enseñar algunos de las metodologías.	Analiza problemas de diseño complejos y propone soluciones creativas, aplicando la metodología de diseño centrado en el usuario que más funcione en el contexto. Es capaz de enseñar algunos de las metodologías.
Metodologías	Trabaja eficientemente con equipos de desarrollo ágiles/iterativos de trabajo ágil (Scrum, Kanban)	Es capaz de apoyar al PO en la construcción de HI con criterios de aceptación claros.	Ejecuta ejercicios de priorización con metodologías y frameworks existentes.	Construye frameworks de trabajo personalizados, integrando diferentes metodologías de diseño según la necesidad específica de cada proyecto.	Construye artefactos orientados a una visión más holística y estratégica del producto de forma autónoma (service blueprint, product vision board).	
Usabilidad	Construye experiencias sencillas teniendo en cuenta los principios básicos de usabilidad (heurísticas de尼尔森)	Realiza análisis de usabilidad de productos digitales para identificar problemas de diseño, además construye y repone sus resultados. (heurísticas, pruebas con usuarios, mapeo cognitivo, mapas de calor)	Ejecuta de forma autónoma ejercicios de usabilidad de diferentes niveles (heurísticas, test no moderados, mapas de calor, A/B test, mapeo cognitivo, think aloud test) comunica los hallazgos y proporciona recomendaciones para mejorar los diseños de la aplicación de los usuarios de modo iterativo.	Construye artefactos orientados a una visión más holística y estratégica del producto de forma autónoma (service blueprint, product vision board).	Define el alcance de la implementación de parámetros de accesibilidad en proyectos nuevos	
Preventa				elementos de Interacción. Así como estimación de tiempos y creación de células de desarrollo del proyecto.		
Mentoría					Acompaña el proceso de mentores Trainee a Advanced. Participa de comunidades de conocimiento para la exposición de habilidades que se crean desde el chapter recibiendo a roles avanzados y senior en procesos de liderazgo.	

Guardar cambios **Generar informe**

Guardado con éxito

Figura 21. Diseño de matriz de valoración en Google Sheet presentado por el equipo de UI.

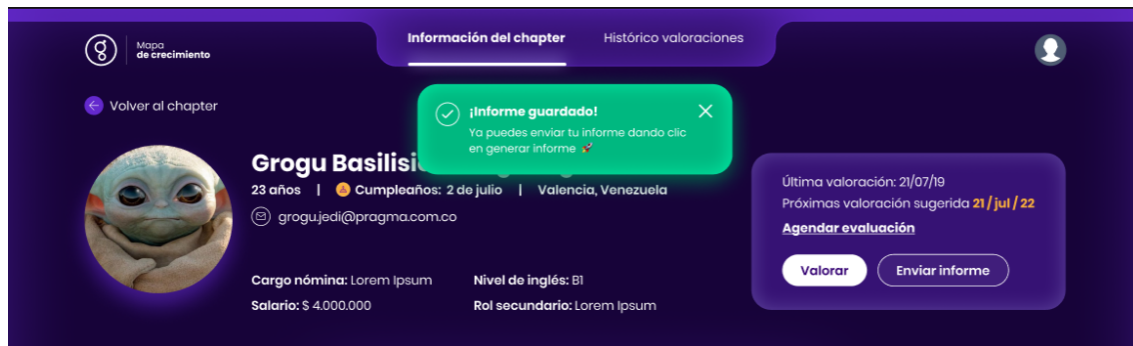


Figura 22. Diseño de alerta Pop-up para indicar éxito en generación de reporte presentado por el equipo de UI.

3.4.18 Envío automático de plan carrera.

Yo como líder de chapter o embajador necesito, después de haber valorado un pragmático, poder enviarle a este y a todos los interesados, el resultado y plan carrera para que continúen su ruta de habilitación y puedan alcanzar el próximo nivel.

Criterios de aceptación:

- La implementación debe guiarse por la imagen definida por el equipo de UI.
- En el dashboard del pragmático habrá un botón que diga "Enviar informe".
- Si el pragmático no tiene ninguna evaluación reciente o vencida debe primero hacer la valoración y luego se habilitará el botón de enviar informe.
- Si la última valoración del pragmático vence en menos de 7 días debe aparecer desactivado el botón de enviar informe.
- Al dar click en el botón "Enviar informe" en el dashboard del pragmático se debe abrir un modal para confirmar el resultado de la valoración y los destinatarios para enviar el resultado.
- Cuando desee añadir más remitentes se debe listar los correos que coinciden para elegir.

- g. Al dar click al botón de enviar informe, se debe enviar el correo al pragmático y a cada uno de los remitentes agregados, además, se debe mostrar una notificación de tipo Pop-up que indique éxito en el envío del correo en el dashboard del pragmático.
- h. Se debe enviar un correo con la plantilla descrita en el diseño.
- i. El correo enviado debe tener adjunto el excel del plan carrera convertido al formato pdf.
- j. El correo es una plantilla que tiene unos campos personalizados, los cuales deben cambiar de manera dinámica con los datos de cada pragmático, los datos a cambiar son: Asunto, Nombre, Chapter al que pertenece, y nivel de seniority en el que quedó.

Generar informe de valoración ✕

Confirma el nuevo nivel de seniority del pragmático
Recuerda que esta información está en el encabezado del Google Sheets que acabas de enviar.

Elige el seniority **Elige el nivel**

Selecciona seniority ▼ L1 L2 L3

El mapa de crecimiento será enviado a:

Al dar clic en **enviar informe** el pragmático recibirá en su correo el mapa de crecimiento y tú recibirás una copia 📧

Figura 23. Modal para enviar el informe de valoración presentado por el equipo de UI.

Generar informe de valoración [X]

Confirma el nuevo nivel de seniority del pragmático
Recuerda que esta información está en el encabezado del Google Sheets que acabas de enviar.

Elige el seniority **Elige el nivel**

Selecciona seniority [v] L1 L2 L3

El mapa de crecimiento será enviado a:

Al dar clic en **enviar informe** el pragmático recibirá en su correo el mapa de crecimiento y tú recibirás una copia 📧

Figura 24. Confirmación de nivel de seniority del pragmático luego de hacer la valoración.

Generar informe de valoración [X]

Confirma el nuevo nivel de seniority del pragmático
Recuerda que esta información está en el encabezado del Google Sheets que acabas de enviar.

Elige el seniority **Elige el nivel**

Advanced [v] L1 L2 L3

El mapa de crecimiento será enviado a:

Figura 25. Selección de destinatarios o remitentes para enviar el informe luego de valorar al pragmático.

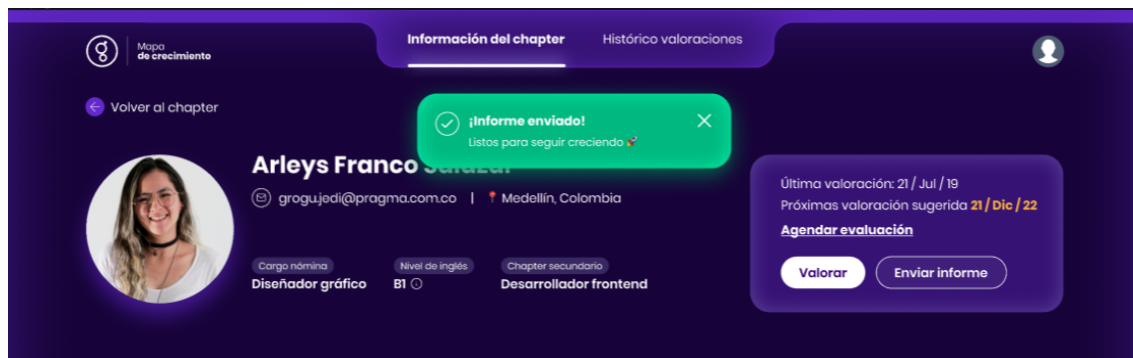
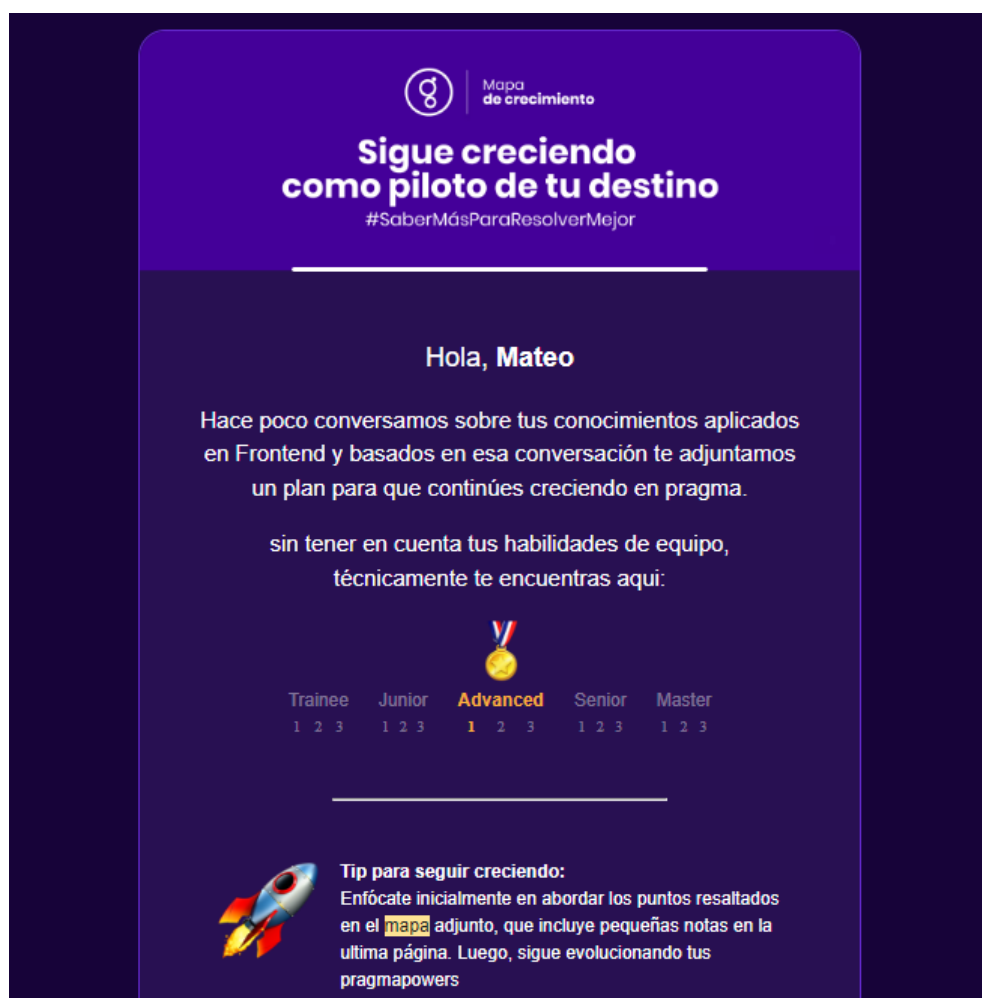


Figura 26. Alerta Pop-up para indicar éxito en el envío del informe al pragmático y demás remitentes.



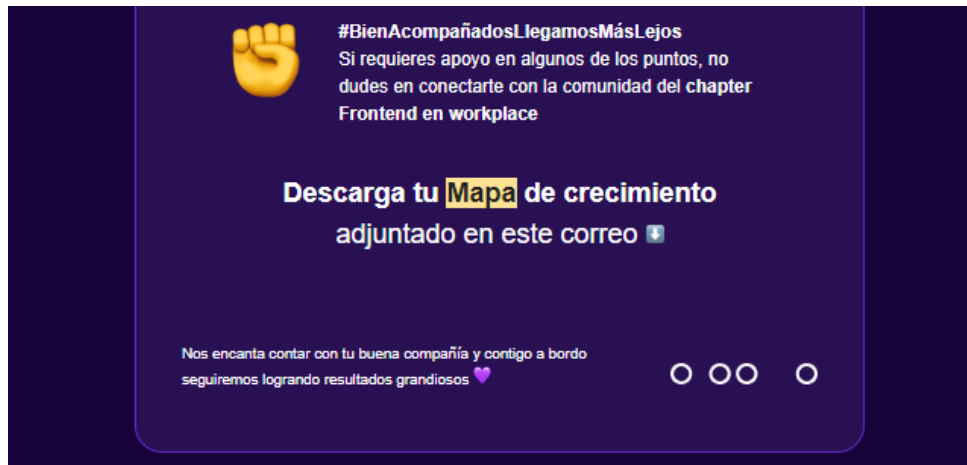


Figura 27. Plantilla de correo que se envía al pragmático y a los demás remitentes.

3.4.19 Cerrar sesión en la aplicación.

Yo como líder de chapter o embajador necesito, poder cerrar sesión en la aplicación para evitar que otras personas utilicen mi cuenta.

Criterios de aceptación:

- La implementación debe guiarse por la imagen definida por el equipo de UI.
- Al dar click en la imagen de perfil se debe desplegar el botón que permita cerrar sesión.
- Al dar click en el botón “Cerrar sesión”, la sesión debe cerrarse y se debe redirigir a la pantalla de inicio de sesión.

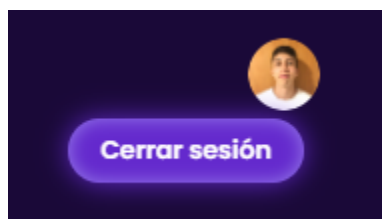


Figura 28. Botón de cerrar sesión presentado por el equipo de UI.

3.5 Estructura de carpetas del proyecto.

Teniendo en cuenta la estructura proporcionada por la CLI de Angular a la hora de crear una aplicación, y orientado en la metodología “Atomic Design” se definió la siguiente estructura de carpetas para el proyecto:

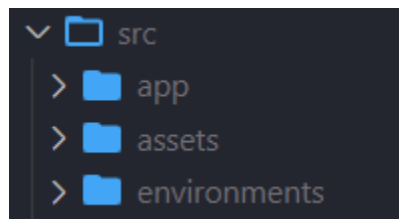


Figura 29. Estructura de carpetas principal del proyecto.

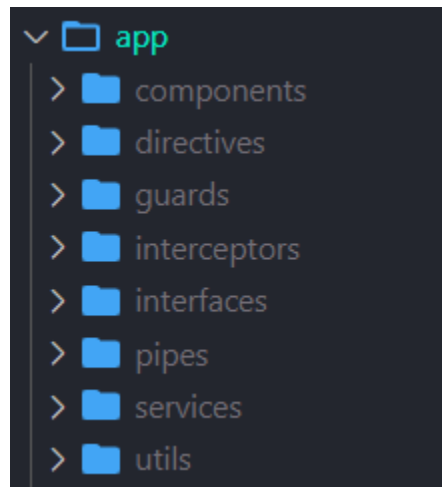


Figura 30. Estructura interna de la carpeta “app” del proyecto.

En la carpeta “app” se incluyen todos los componentes y demás elementos que aportarán lógica y seguridad a la aplicación como lo son los “Guards”, interceptores, interfaces, servicios, entre otros.

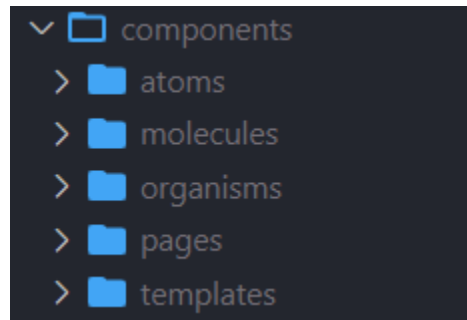


Figura 31. Estructura interna de la carpeta “components” siguiendo el sistema de diseño atómico propuesto para los componentes.

3.6 Desarrollo de las historias de usuario.

Este desarrollo se llevó a cabo siguiendo el orden de las historias de usuario mencionadas en la sección 3.4.

Debido a que toda la aplicación y los datos que en ella se presentarán están alojados en AWS, para el inicio de sesión en la aplicación se realizó una integración entre Google como proveedor de identidad para el servicio de autenticación de AWS llamado AWS Cognito; esta integración se conoce como “Identidad Federada” y lo que permite es facilitar el inicio de sesión pero también asegurarse de que solo puedan acceder a la aplicación y aquel grupo de usuarios pertenecientes a la compañía, de manera que el flujo de este proceso es el siguiente:

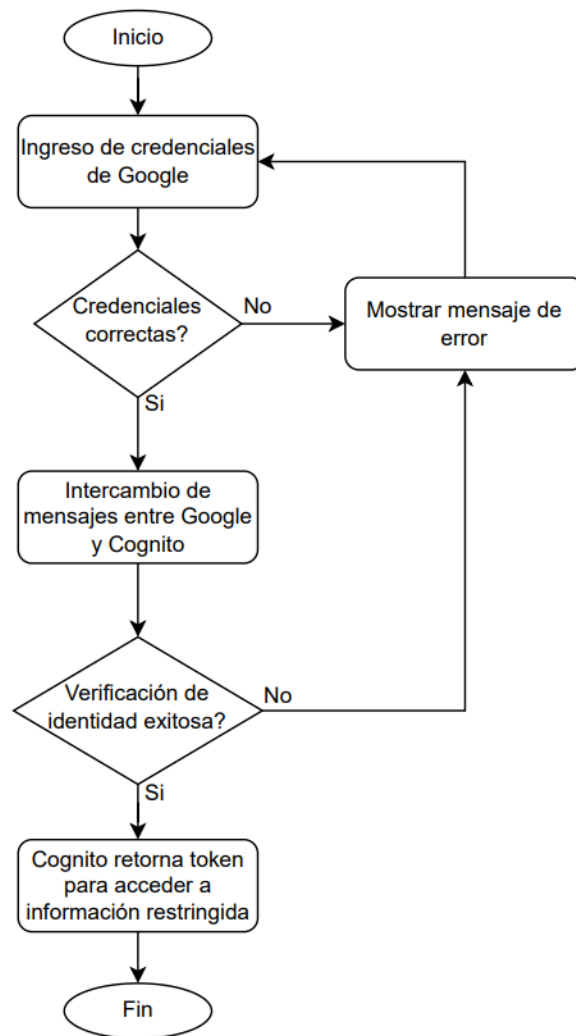


Figura 32. Diagrama de flujo para el inicio de sesión en la aplicación Mapa de Crecimiento

Una vez que el usuario haya iniciado sesión satisfactoriamente es dirigido a la pantalla principal de la aplicación en la cual se le presentará el dashboard del chapter que está conformada por los componentes mencionados desde la historia de usuario de la sección 3.4.2 hasta la 3.4.12. Para las historias de usuario de la sección 3.4.2 hasta la 3.4.6 el flujo es muy similar, por tanto se muestra de manera generalizada en la siguiente figura:

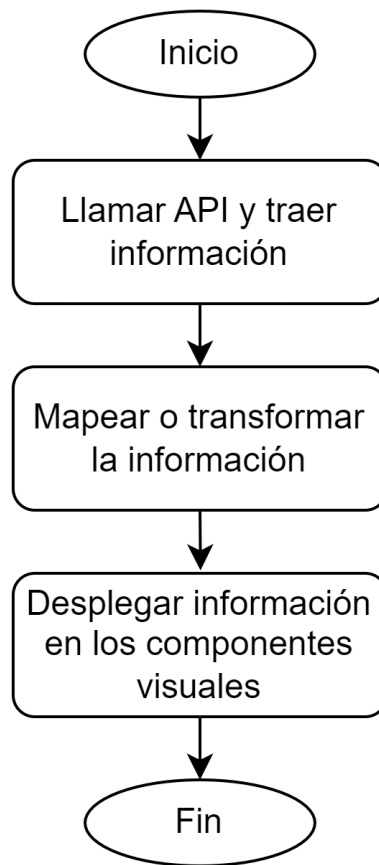


Figura 33. Diagrama de flujo para el despliegue de la información en los componentes visuales.

Por otro lado, las historias de usuario desde la sección 3.4.7 hasta la 3.4.12, en las cuales lo que se hace es filtrar la información presentada anteriormente, el flujo cambia, puesto que es necesario que la interfaz esté pendiente de las interacciones del usuario con los componentes para llevar a cabo el filtrado correspondiente de la información, tal flujo se muestra en la Figura 34.

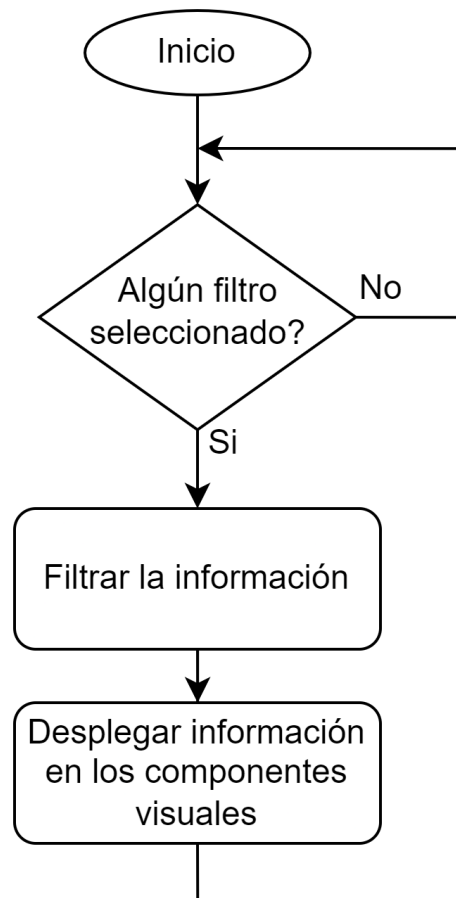


Figura 34. Diagrama de flujo para filtrar y desplegar la información en los componentes visuales.

Para la historia de usuario de la sección 3.4.7, la lógica para la acción de búsqueda es diferente puesto que aunque se filtra la información, este filtrado se lleva a cabo en el backend de la aplicación, y se logra pasándole como parámetro a la petición al API el texto a buscar dentro de la lista de pragmáticos, además de esto, se debe implementó el buscador con una estrategia de “debounce” para reducir la cantidad de peticiones que se le hace al servidor. Dicha lógica se muestra en la Figura 35.

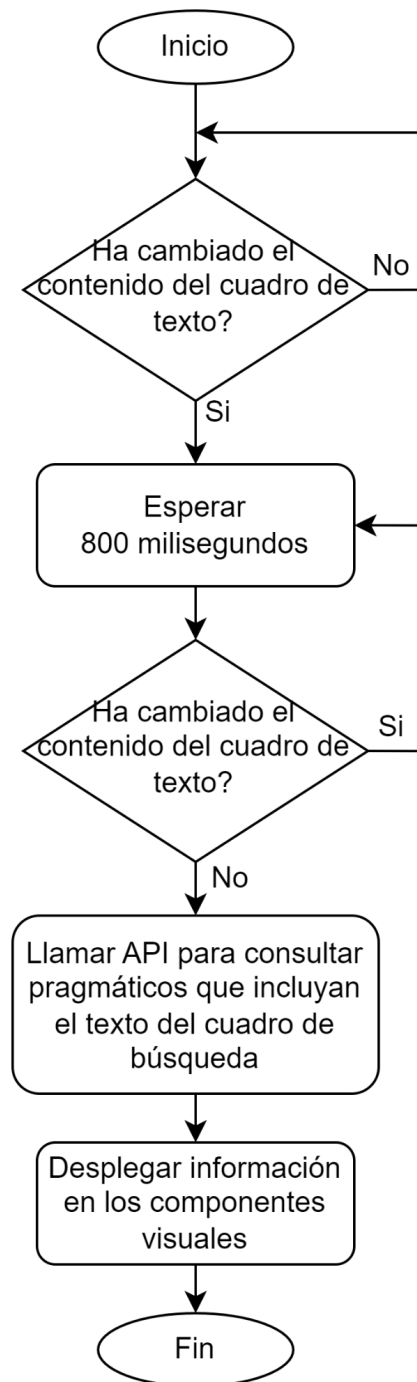


Figura 35. Diagrama de flujo del componente buscador de pragmáticos de la sección 3.4.7.

Como se puede apreciar en este flujo, sólo se hacen peticiones al servidor luego de transcurrir 800 ms y que el contenido del cuadro de búsqueda no haya cambiado, lo que asegura una buena gestión de los recursos.

Para la historia de usuario del dashboard del pragmático, primero se hace la lógica correspondiente en la lista de los pragmáticos para que cuando se dé click sobre una fila que representa un pragmático, se haga el debido redireccionamiento de manera que se pueda acceder a la información del usuario seleccionado en la nueva pantalla, esto se logró agregando el correo del pragmático como parámetro en la url para posteriormente, en el componente dashboard del pragmático poder obtener este parámetro y consultar por medio de una petición a la API, la información del pragmático. La siguiente figura muestra este comportamiento.

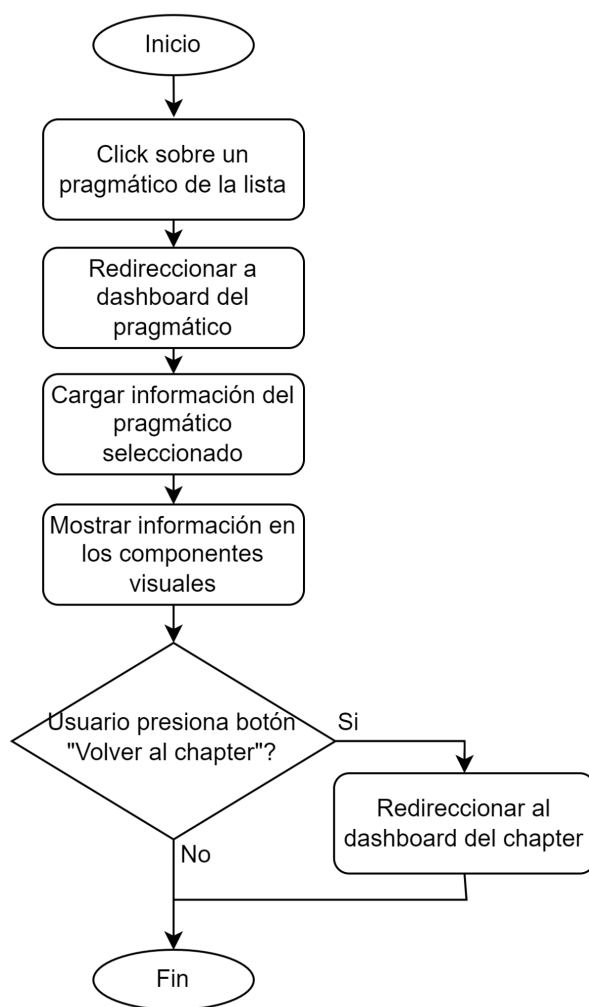


Figura 36. Diagrama de flujo para acceder al dashboard del pragmático.

Ahora para la realización de la historia de usuario de la sección 3.4.16, en la que se selecciona el chapter específico del cual se desea visualizar la información, es necesario

cargar de nuevo toda la información relacionada al chapter, por tanto el diagrama de flujo para este proceso es el siguiente:

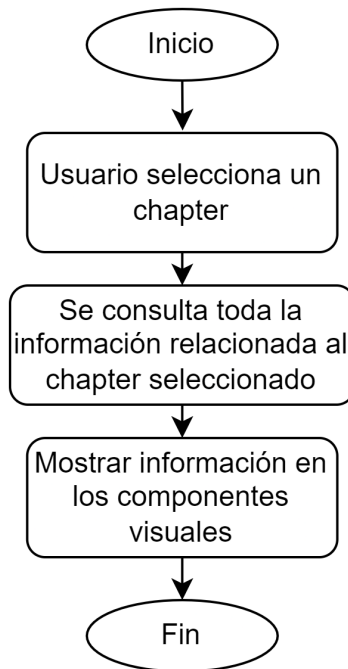


Figura 37. Diagrama de flujo para el selector de chapter

Para la acción de valoración del pragmático mencionada en la sección 3.4.17, se debe estar en el dashboard del pragmático y si el pragmático que se está revisando cumple con las condiciones para hacerse una valoración, el botón “Valorar” estará habilitado, de manera que al darle click, el usuario podrá comenzar el proceso de valoración del pragmático detectando sus fortalezas y debilidades en una conversación sobre conceptos técnicos. El flujo a seguir en este proceso es el que se muestra en la siguiente figura:

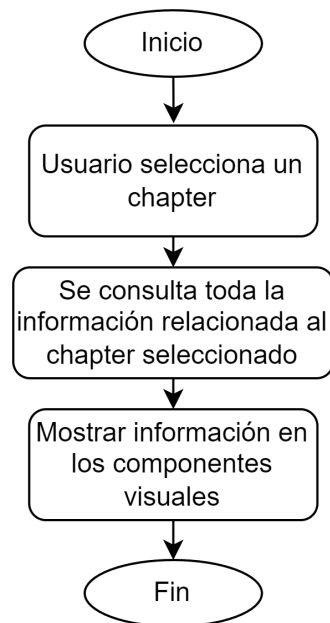


Figura 38. Diagrama de flujo para el proceso de valoración del pragmático.

En la sección 3.4.18 donde se habla del proceso de enviar el plan carrera al pragmático luego de haber hecho la valoración, este comienza cuando el usuario presiona el botón “Enviar informe”, a este se le brindará la oportunidad de seleccionar las personas a las cuales se les enviará este informe ya que pueden haber interesados en esto además del pragmático mismo que ha sido valorado, estos interesados pueden ser los líderes de equipo o líderes técnicos del proyecto en que se encuentra el pragmático. El flujo para dicha acción es el que se muestra en la siguiente figura:

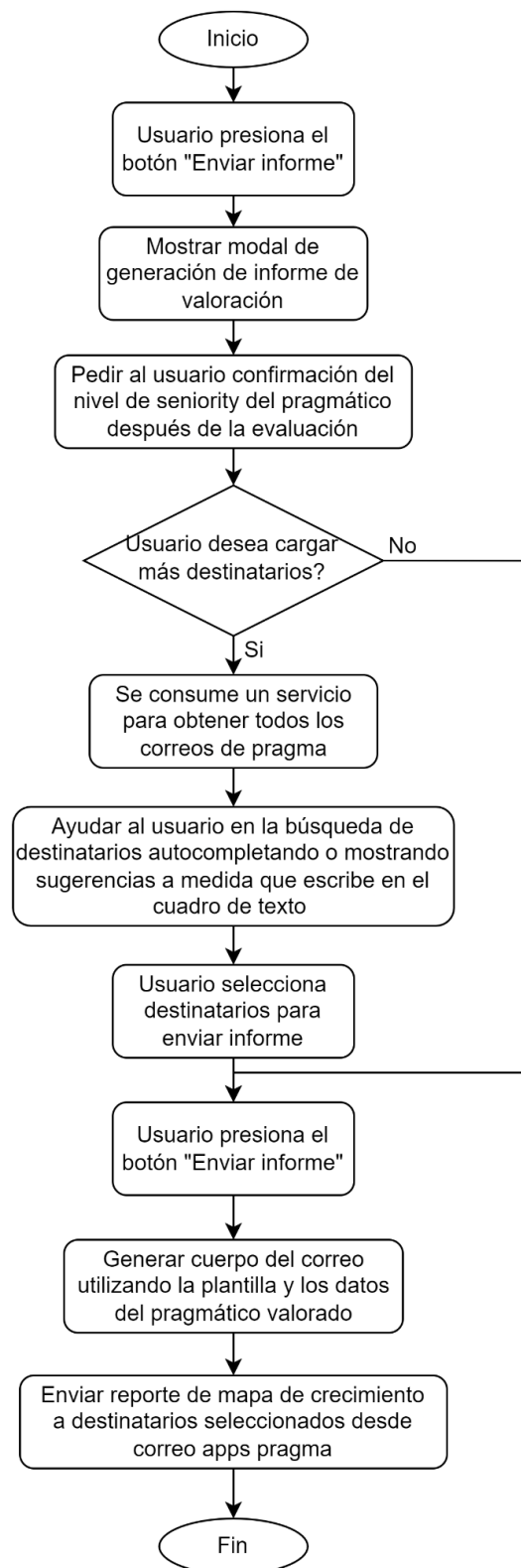


Figura 39. Diagrama de flujo para el envío automático de plan carrera para el pragmático valorado

Para el histórico de valoraciones de la sección 3.4.13 primero es necesario consultar todos los datos de los pragmáticos del chapter que estén relacionados con sus valoraciones ya que esta es la información que se va a presentar. Por otro lado, el buscador funciona de manera idéntica al buscador de pragmáticos de la sección 3.4.7, obteniendo como resultado el siguiente diagrama de flujo:

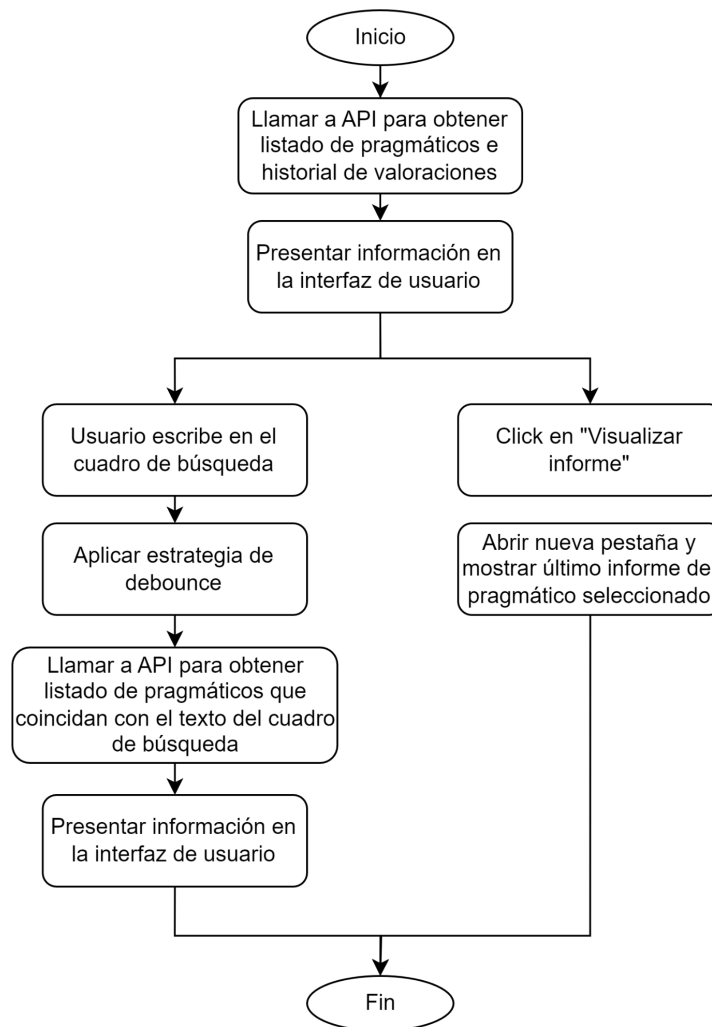


Figura 40. Diagrama de flujo para el histórico de valoraciones.

Para la acción de cerrar sesión la comunicación entre Cognito y Google es básicamente para la eliminación del token de acceso que se había proporcionado al usuario en el proceso de inicio de sesión.

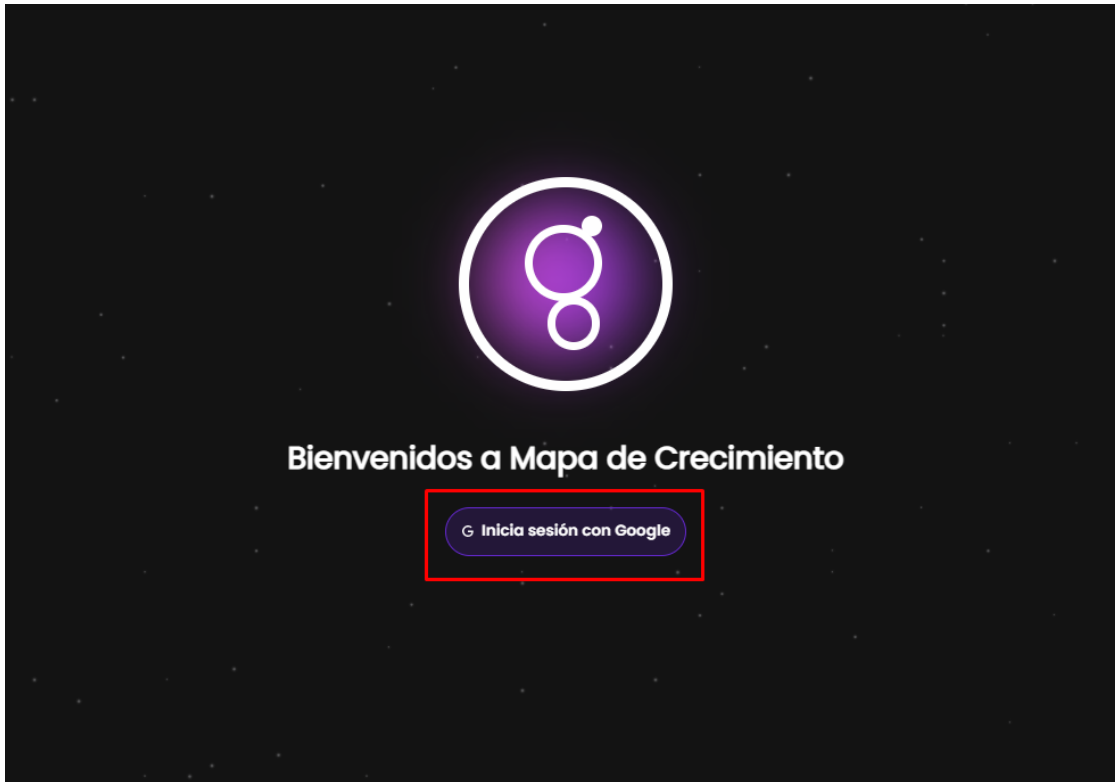
4 Resultados

El resultado obtenido del desarrollo de este proyecto es una interfaz de usuario para la aplicación web “Mapa de Crecimiento” completamente funcional, la cual se comunica con el backend a través de llamadas a una API, permitiendo desplegar la información solicitada y aplicar la lógica y flujo que se estableció en los requisitos del proyecto por parte de la compañía.

A continuación se presentarán los resultados obtenidos por medio de imágenes que representan cada una de los componentes, pantallas desarrolladas y flujo de la aplicación.

4.1 Pantalla de Inicio de Sesión.

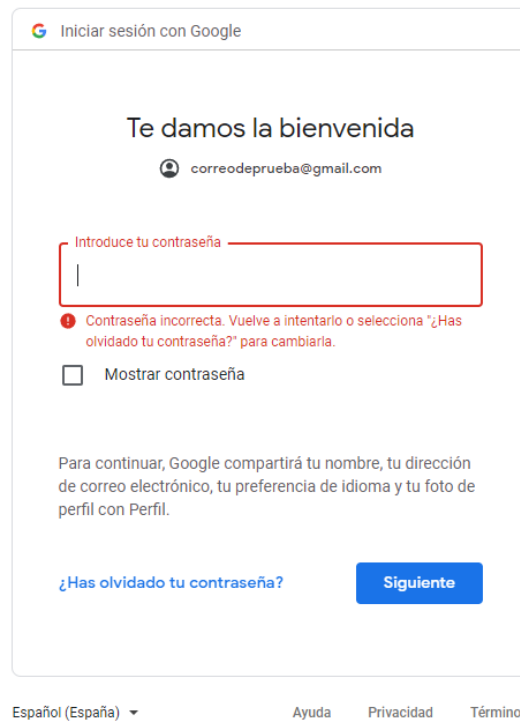
Para el inicio de sesión, como se mencionó anteriormente, se utilizó Google como proveedor de identidad para el servicio de autenticación de AWS llamado Cognito, el cual lleva a cabo esta integración por medio de lo que se conoce como “Identidad federada”. Este proceso de inicio de sesión comienza con la solicitud del ingreso de credenciales de la cuenta de Google del usuario después de haber presionado el botón “Inicia sesión con Google”, esto se muestra en las Figuras 41a) y 41b), además, en la Figura 41c) se muestra el comportamiento cuando se ingresan las credenciales de manera errónea, por otra parte, en las Figura 41d) se evidencia el resultado obtenido al tratar de iniciar sesión con una cuenta de Google que no pertenece a la compañía. En el caso en que la cuenta de Google con la que se está iniciando sesión sea de la compañía, inmediatamente se redirige a la pantalla principal de la aplicación.



a)

A screenshot of the Google login flow. At the top, it says 'Iniciar sesión con Google'. The main heading is 'Iniciar sesión' with a sub-link 'Ir a Perfil'. Below this is a text input field labeled 'Correo electrónico o teléfono' containing a vertical cursor. A link '¿Has olvidado tu correo electrónico?' is positioned below the input field. A paragraph of text explains that Google will share the user's name, email address, language preference, and profile picture with Perfil. At the bottom, there are two buttons: 'Crear cuenta' and 'Siguiente'. The footer contains 'Español (España)' with a dropdown arrow, and links for 'Ayuda', 'Privacidad', and 'Términos'.

b)



Iniciar sesión con Google

Te damos la bienvenida

correodeprueba@gmail.com

Introduce tu contraseña

Contraseña incorrecta. Vuelve a intentarlo o selecciona "¿Has olvidado tu contraseña?" para cambiarla.

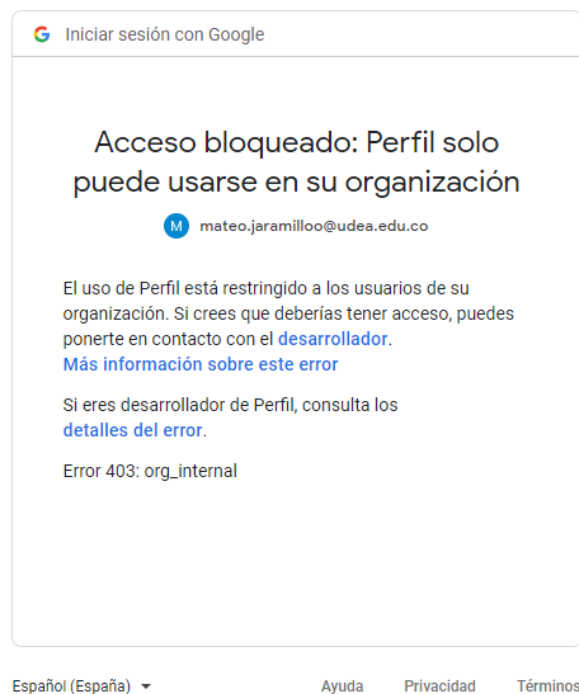
Mostrar contraseña

Para continuar, Google compartirá tu nombre, tu dirección de correo electrónico, tu preferencia de idioma y tu foto de perfil con Perfil.

[¿Has olvidado tu contraseña?](#) [Siguiente](#)

Español (España) ▾ [Ayuda](#) [Privacidad](#) [Términos](#)

c)



Iniciar sesión con Google

Acceso bloqueado: Perfil solo puede usarse en su organización

mateo.jaramillo@udea.edu.co

El uso de Perfil está restringido a los usuarios de su organización. Si crees que deberías tener acceso, puedes ponerte en contacto con el [desarrollador](#).
[Más información sobre este error](#)

Si eres desarrollador de Perfil, consulta los [detalles del error](#).

Error 403: org_internal

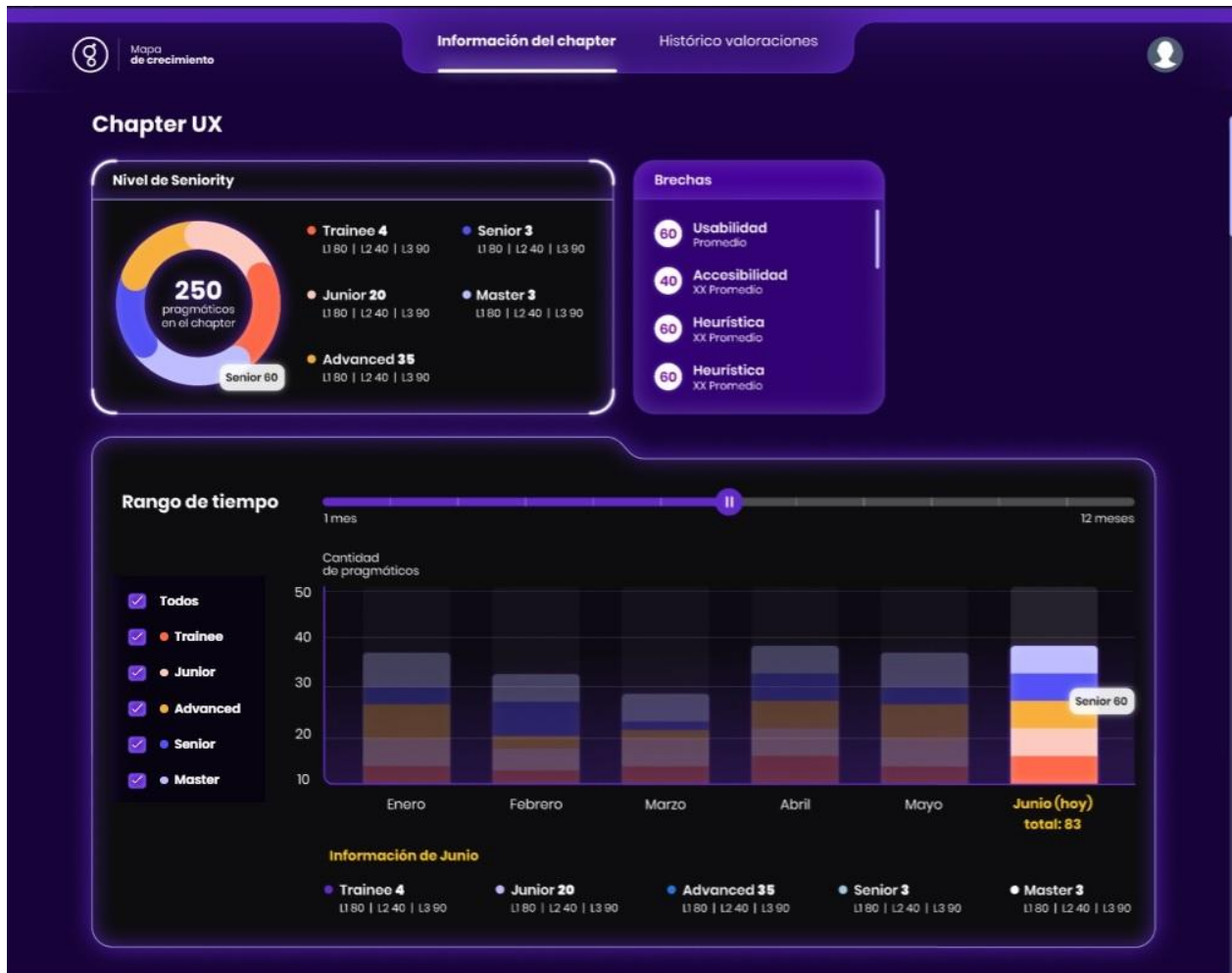
Español (España) ▾ [Ayuda](#) [Privacidad](#) [Términos](#)

Figura 41. a) Pantalla de Inicio de sesión, b) Pantalla de solicitud de credenciales de Google, c) Error en credenciales en el inicio de sesión, d) Acceso bloqueado a las cuentas de Google que no pertenecen a la compañía.

4.2 Pantalla Dashboard del Chapter.

Una vez que las credenciales sean correctas y que se haya concedido el acceso a la aplicación por el hecho de haber ingresado con un correo perteneciente a la compañía, se redirige a la pantalla “Dashboard del Chapter” la cual está conformada por las historias de usuario de las secciones 3.4.2 hasta la 3.4.12, 3.4.16 y 3.4.19 obteniendo así lo que se observa en la Figura 42a).

Al estar en esta pantalla, el usuario podrá filtrar la información presentada de diversas maneras, una de ellas, es en el diagrama de barras, donde el usuario puede seleccionar la cantidad de meses que se van a tener en cuenta y los niveles de seniority que desea ver en dicho diagrama, obteniendo así lo que se muestra en la Figura 42b). Por otro lado, al usuario se le permite buscar entre la lista de pragmáticos pertenecientes al chapter que se encuentra visualizando por medio del buscador, filtrar por nivel de seniority, cargo nómina, marca/proyecto y fecha de valoración tal y como se muestra en las figuras 42c), 42d), 42e) y 42f).



Pragmáticos del chapter

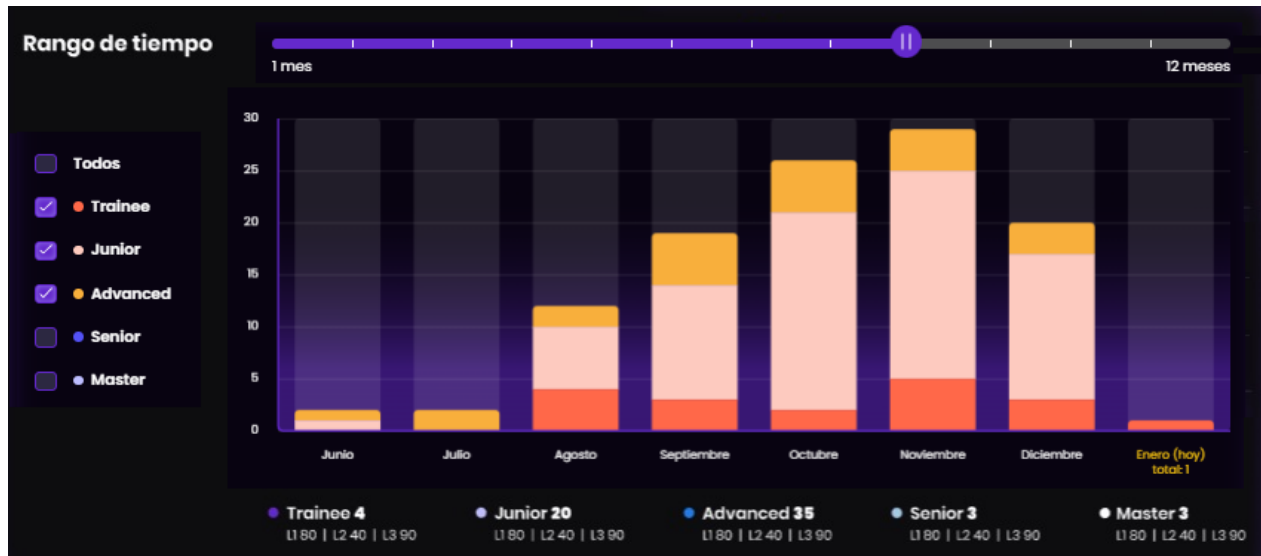
Buscar

< 1 2 3 ... 250 > 25 por página

Pragmáticos	Cargo nómina	Seniority	Marca/proyecto	Próxima valoración
Andrew Chaverra González	Desarrollador de Software	Junior L2	Comfama - lorem ipsum	Valoración vencida
Diego Alonso Álvarez Arcila	Desarrollador de Software	No asignado	Comfama - lorem ipsum	Valoración vencida
Otoniel Felipe Rodríguez Giraldo	Auxiliar de desarrollo	Junior L2	Comfama - lorem ipsum	Valoración vencida
Michael Steven Cuellar Molina	Practicante de desarrollo	Trainee L3	Comfama - lorem ipsum	Hay
Jairo Alberto España Isacas	Desarrollador de Software	Advanced L2	Comfama - lorem ipsum	Mañana
Jeymmy Carolina Daza Gómez	Auxiliar de desarrollo	Junior L2	Comfama - lorem ipsum	21 / Jun / 2022

La valoración debió ser el 21/07/22

a)



b)

Pragmáticos del chapter and X < 1 2 3 ... 250 > 25 por página

Pragmáticos	Cargo nómina	Seniority	Marca/proyecto	Próxima valoración
Andrew Chaverra González	Desarrollador de Software	Junior L2	Comfama - lorem ipsum	Valoración vencida ⚠
Rafael Andres Granados Monroy	Desarrollador de Software	No asignado	Comfama - lorem ipsum	Hay
Francesco André Gómez Osorio	Auxiliar de desarrollo	Junior L2	Comfama - lorem ipsum	Hay
Yaferson Andrés Ríos Martínez	Auxiliar de Desarrollo	Trainee L3	Comfama - lorem ipsum	19/ene/2023
Diego Andrés Vinasco Trujillo	Desarrollador de Software	Advanced L2	Comfama - lorem ipsum	30/ene/2023

La valoración debió ser el 21/07/22

c)

Pragmáticos del chapter rodolfo Q < 1 2 3 ... 250 > 25 por página

Pragmáticos	Cargo nómina	Seniority	Marca/proyecto	Próxima valoración
No se encontraron resultados para 'rodolfo'				

d)

Pragmáticos	Cargo nómina	Seniority	Marca/proyecto	Próxima valoración
Lucas Renato Barrenechea León	Desarrollador de Software	Advanced L3	Comfama - lorem ipsum	Valoración vencida
Diego Alonso Álvarez Arcila	Desarrollador de Software	Advanced L1	Comfama - lorem ipsum	Valoración vencida
Esteban Cadavid Rodríguez	Auxiliar de desarrollo	Advanced L2	Comfama - lorem ipsum	Valoración vencida
Saúl Eduardo Gómez González	Practicante de desarrollo	Advanced L1	Comfama - lorem ipsum	Valoración vencida
David Botero Salazar	Desarrollador de Software	Advanced L2	Comfama - lorem ipsum	Valoración vencida
Jeymmy Carolina Daza Gómez	Auxiliar de desarrollo	Advanced L2	Comfama - lorem ipsum	Hoy

e)

Pragmáticos	Cargo nómina	Seniority	Marca/proyecto	Próxima valoración
Lucas Renato Barrenechea León	Desarrollador de Software	Advanced L3	Comfama - lorem ipsum	Valoración vencida
Diego Alonso Álvarez Arcila	Desarrollador de Software	Advanced L1	Comfama - lorem ipsum	Valoración vencida
Cristian Yamith Burbano Ordoñez	Desarrollador de Software	Advanced L1	Comfama - lorem ipsum	Valoración vencida
Dairo Enrique Cantillo Acevedo	Desarrollador de Software	Advanced L1	Comfama - lorem ipsum	Valoración vencida
David Botero Salazar	Desarrollador de Software	Advanced L2	Comfama - lorem ipsum	Valoración vencida
Yarín Carolina Orcines Manjarres	Desarrollador de Software	Advanced L2	Comfama - lorem ipsum	Valoración vencida

f)

Figura 42. a) Pantalla principal de la aplicación Dashboard del Chapter, b) Filtros en el diagrama de barras, c) Buscador de pragmáticos en el chapter, d) Buscador de pragmáticos en el chapter sin resultados encontrados, e) Filtro por nivel de seniority en listado de pragmáticos del chapter, f) Filtro por cargo nómina y nivel de seniority en listado de pragmáticos del chapter.

4.3 Pantalla Dashboard del Pragmático.

Cuando el usuario está en la pantalla de Dashboard del Chapter, tiene la posibilidad de dar click en cualquiera de los pragmáticos que encuentran en la lista, permitiendo acceder a lo que se conoce como Dashboard del Pragmático, que es el

espacio donde se conoce más información más detallada del pragmático seleccionado y además, en donde se puede llevar a cabo el proceso de valoración del mismo. En la Figura 42 se presenta el Dashboard del Pragmático con la información que se requiere.



Figura 42. Dashboard del pragmático.

Al contar con las respectivas validaciones necesarias para habilitar la opción de valorar al pragmático, se puede acceder al flujo de valoración para posteriormente proceder al envío automático del reporte, donde automáticamente se selecciona el correo del usuario y del pragmático que se ha valorado tal como se muestra en las Figuras 43a), 43b), 43c), 43d), 43e) y 43f)

Antes de iniciar ✕

Te direcionaremos a la matriz de valoración de Google Sheets

No te preocupes por los cambios en el mapa de crecimiento, Google Sheets los **guardará automáticamente** y ten en cuenta los siguientes pasos:

1. Una vez termines tu valoración **regresa al perfil del pragmático**.
2. Ve al botón enviar informe y **agrega el nuevo nivel de seniority**.
3. **Da clic en enviar informe**, el pragmático recibirá en su correo el mapa de crecimiento, además tú y las personas que escojas recibirán una copia.

Tips para tener presente

- La valoración no es una evaluación, es un espacio para que los pragmáticos puedan crecer 🚀.
- Recuerda que el pragmático puede estar nervioso, asegúrate de generar un espacio tranquilo y de confianza 😊.
- Disminuye incertidumbres. Deja claro el objetivo de la valoración e intenta despejar cualquier duda antes de comenzar 🙌.

Continuar

a)

MAPA DE CRECIMIENTO LUX PRAGMA		Nombre	Jenny Sanchez						
		Proyecto	Talent Experience						
		Rol	Analista UX						
		Consideración evaluador							
		Observaciones							
		TRAINEE to JUNIOR	JUNIOR to ADVANCED	ADVANCED to SENIOR	SENIOR to MASTER	MASTER to INFINITY			
		- Graduado o principiante en el rol - Desempeño sobresaliente siguiendo los reglas explícitas	- Autonomía para dirigir problemas similares - Comprensión contextual - Decisiones para situaciones estándar	- Puede resolver nuevos problemas - Comprensión conceptual - Toma de decisiones activa	- Big picture, pensamiento sistémico - Liderazgo de temas importantes - Comprensión profunda de las reglas, teorías, alternativas	- Evaluación, conocimiento tácito y explícito - Visión de lo que es posible, innova			
Diseño centrado en las personas		Revisa los flujos o mapas de flujo para validar de flujo cuando es la persona para conectar relaciones abstractas y propone acciones para su aplicación. (Design Thinking - Flow thinking, define ideas, principios, máximos, Doble Diamond - convergencia y divergencia)	Aplica en su cotidianidad metodologías de diseño centrado en las personas para la resolución de los problemas de diseño a los que se enfrenta. (Design Thinking, Double diamond, Lean,)	Analiza problemas de diseño complejos y propone soluciones acordes, aplicando la metodología de diseño centrado en el usuario que más funcione en el contexto. Es capaz de enseñar algunos de los metodologías.	3				
Metodologías				De acuerdo al contexto, selecciona y ejecuta artefactos de diseño centrados en las personas (Customer Journey Map, Empathy Map y Buyer Persona) conoce el objetivo de éstos y sabe cómo impactan su proceso de diseño.	4				
			Trabaja eficazmente con equipos de desarrollo ágiles y artefactos de trabajo (Scrum, Kanban)	Ejecuta ejercicios de priorización con metodologías y frameworks existentes.	2				
			Es capaz de apoyar al PO en la construcción de HI con criterios de aceptación claros.	3					
			Identifica el concepto de MVP y reconoce su importancia dentro del alcance de los proyectos a los que se enfrenta.	4					
Usabilidad		Construye experiencias sencillas teniendo en cuenta los principios básicos de usabilidad (heurísticas de	Realiza análisis de usabilidad de productos digitales para identificar problemas de diseño, además construye y repone sus resultados. (heurísticas, pruebas con usuarios, mapas cognitivos, mapas de flujo)	Ejecuta de forma autónoma ejercicios de usabilidad de diferentes tipos. (heurísticas, test no moderados, mapas de calor, A/B test, mapas cognitivos, think aloud test) comunica los hallazgos y proporciona recomendaciones para mejorar los diseños de la interfaz de usuario en los productos de desarrollo digital.	4				
				3					
				Define el alcance de la implementación de parámetros de accesibilidad en proyectos nuevos	1				

Preventa	elementos de interacción. Así como estimación de tiempo y creación de células de desarrollo del proyecto.	1
Mentoría	<p>Acompaña el proceso de talento Trainee a Avanzad</p> <p>Participa de comunidades de conocimiento para la exposición de habilidades que se crean desde el chapter.</p> <p>Realiza o roles avanzados y senior en procesos de liderazgo.</p>	1

Guardado con éxito

b)

Mapa de crecimiento

Información del chapter | Histórico valoraciones

Volver al chapter frontend

¡Informe guardado!
Ya puedes enviar tu informe dando clic en generar informe 📄

Última valoración: 11/oct/2022
Próxima valoración sugerida: 08/feb/2023

Valorar | Enviar informe

Sebastian Valencia
sebastian.valencias@pragma.com.co | Bogotá, Colombia

Chapter primario: Backend | Chapter secundario: Frontend

Rol: Desarrollador de software | Nivel de Inglés: 4 (Sin Certificado)

c)

Generar informe de valoración

Confirma el nuevo nivel del seniority del pragmático
Recuerda que esta información está en el encabezado de Google Sheets que acabas de realizar.

Elige el seniority: Trainee | Elige el nivel: L1, L2, L3

El mapa de crecimiento será enviado a:

mateo.jaramillo@pragma.com.co | sebastian.valencias@pragma.com.co

lucas.arboleda@pragma.com.co
lucas.barrenechea@pragma.com.co
lucia.caicedo@pragma.com.co
luis.aristizabal@pragma.com.co
luis.bodera@pragma.com.co

d)

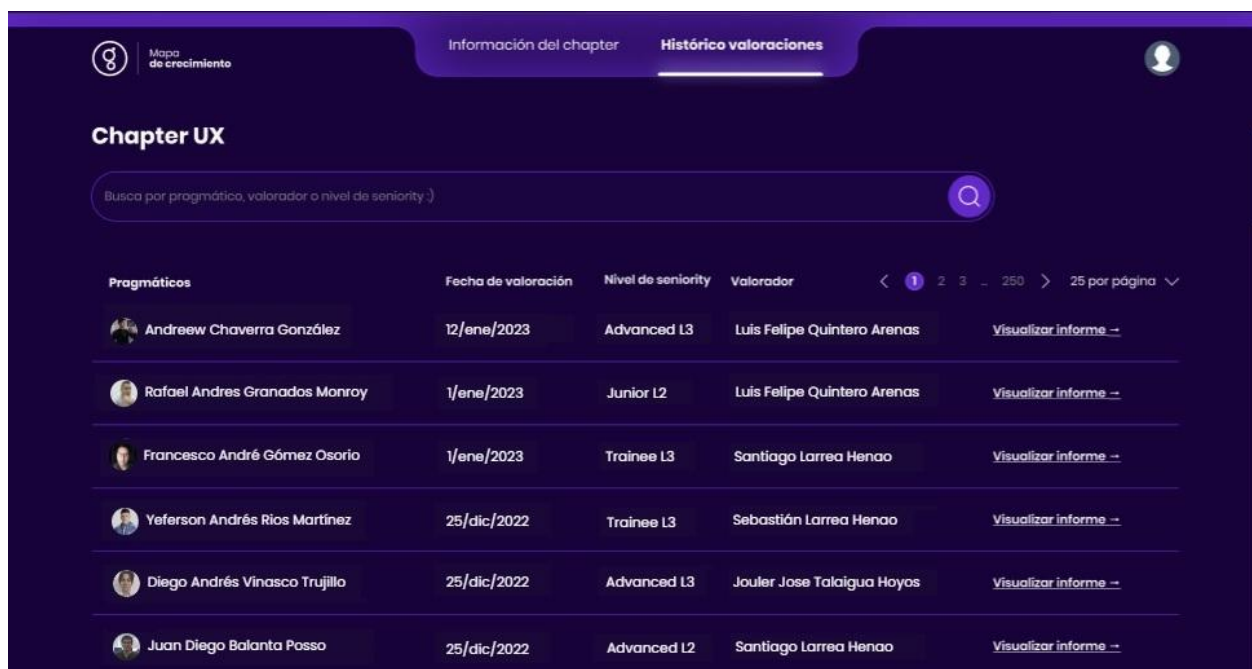
e)







f)

Figura 43. a) Modal de tips antes de iniciar con el proceso de valoración del pragmático, b) Matriz de valoración del pragmático según el chapter que se está analizando, c) Pop-up de confirmación de guardado de informe de valoración, d) Proceso de generar y enviar informe después de la valoración, e) Confirmación de resultados de informe de valoración y agregado de remitentes, f) Confirmación de envío de informe de valoración del pragmático.

4.4 Pantalla Histórico de Valoraciones.

En esta pantalla se permite al usuario visualizar cada una de las valoraciones realizadas en el chapter que se encuentra revisando, pudiendo filtrar por pragmáticos específicos por medio del buscador, conocer quién fue el encargado de la valoración de cada uno de los pragmáticos, la fecha en que se realizó, el resultado y también acceder a la matriz de valoración que se llenó en el proceso de valoración.



Pragmáticos	Fecha de valoración	Nivel de seniority	Valorador	
 Andrew Chaverra González	12/ene/2023	Advanced L3	Luis Felipe Quintero Arenas	Visualizar informe →
 Rafael Andres Granados Monroy	1/ene/2023	Junior L2	Luis Felipe Quintero Arenas	Visualizar informe →
 Francesco André Gómez Osorio	1/ene/2023	Trainee L3	Santiago Larrea Henao	Visualizar informe →
 Yeferson Andrés Rios Martínez	25/dic/2022	Trainee L3	Sebastián Larrea Henao	Visualizar informe →
 Diego Andrés Vinasco Trujillo	25/dic/2022	Advanced L3	Jouler Jose Talaigua Hoyos	Visualizar informe →
 Juan Diego Balanta Posso	25/dic/2022	Advanced L2	Santiago Larrea Henao	Visualizar informe →

a)

Mapa de crecimiento

Información del chapter Histórico valoraciones

Chapter UX

juan

Pragmáticos	Fecha de valoración	Nivel de seniority	Valorador	< 1 2 3 ... 250 > 25 por página
Juan Manuel Baracaldo Marín	10/dic/2022	Advanced L2	Sebastián Larrea Henao	Visualizar informe
Juan Diego Fandiño Urrego	10/dic/2022	Junior L2	Luis Felipe Quintero Arenas	Visualizar informe
Juan Jose Cordoba Ortiz	10/dic/2022	Trainee L3	Santiago Larrea Henao	Visualizar informe
Juan Felipe Londoño Builes	30/nov/2022	Advanced L3	Sebastián Larrea Henao	Visualizar informe
Juan Carlos Vidal Marín	30/nov/2022	Advanced L3	Jouler Jose Talaigua Hoyos	Visualizar informe
Juan Diego Balanta Posso	30/nov/2022	Trainee L3	Santiago Larrea Henao	Visualizar informe

b)

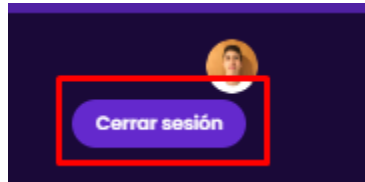
1 / 3 110% + -

pragmática	TRAINEE to JUNIOR	JUNIOR to ADVANCED	ADVANCED to SENIOR	SENIOR to MASTER
Verificación	<ul style="list-style-type: none"> Validado o participado en el rol Completar tareas asignadas de reglas específicas. 	<ul style="list-style-type: none"> Autonomía para elegir problemas similares Compartir conocimiento Definir para situaciones estándar 	<ul style="list-style-type: none"> Puede resolver nuevos problemas Compartir conocimiento Toma de decisiones activa 	<ul style="list-style-type: none"> Big picture, pensamiento sistémico Utilización de nuevas tecnologías Compartir profundo de las reglas, heurísticas, alternativas
Verificación	<ul style="list-style-type: none"> Usa conceptos básicos de Git (Simple User interface) de Git y los aplica en la estrategia de branching definida en el proyecto. Resuelve eficientemente conflictos de empujamiento de código. 	<ul style="list-style-type: none"> Entiende conscientemente la estrategia de branching definida en el proyecto, basado en algún flujo estándar conocido (Gitflow) 	<ul style="list-style-type: none"> Revisa y sugiere (si es requerido). Guía a otros desarrolladores y participa en la revisión de estrategias de branching de un proyecto. 	
Verificación	<ul style="list-style-type: none"> Aplica técnicas de investigación responsive Al menos un sistema de rejilla (Bootstrap, Material) Responde a CSS avanzadas o responsive (Flexbox, Grid) 	<ul style="list-style-type: none"> Aplica conceptos gráficos avanzados de Programación de estilos (Flex) Patrones de nomenclatura (BEM) 	<ul style="list-style-type: none"> Aplica conscientemente conceptos de animaciones y transiciones orientadas a microinteracciones y alineadas con la UI definida en el proyecto. 	
Verificación	<ul style="list-style-type: none"> Aplica conceptos bases en por lo menos un framework o librería sobre su lenguaje de programación primario Implementa librerías de páginas (links o routing virtual del framework/librería) Implementa componentes funcionales de página (menús, paginadores, etc) 	<ul style="list-style-type: none"> Experiencia construyendo soluciones en 1 framework o librería sobre su lenguaje de programación primario Protección de rutas (JWT, Cookies) Usa al menos un patrón de manejo de estados (RxJS, Redux, ContextAPI, etc) 	<ul style="list-style-type: none"> Compara diferentes frameworks/librerías (2 o más) y los usa de manera puntual según contexto de negocio [2] 	<ul style="list-style-type: none"> Compara y aplica conscientemente al menos dos librerías de react/estado web (CSS, SSR y para backend) según el framework/librería y el contexto de negocio
Verificación	<ul style="list-style-type: none"> Compara y sabe cuándo aplicar diferentes estrategias de código Condiciones (if, if else, switch) Resolución (for, for each, while, do while) 	<ul style="list-style-type: none"> Soluciono problemas de código usando técnicas más avanzadas de programación (Algoritmos, Heurísticas, reglas) 	<ul style="list-style-type: none"> Comprende y aplica conscientemente conceptos de patrones orientados a su lenguaje de programación primario (Algoritmos, heurísticas, reglas, paradigmas, etc) 	
Verificación	<ul style="list-style-type: none"> Consumo servicios web (usuario personal, body, autorización), usando los verbos GET y POST 	<ul style="list-style-type: none"> Aplica conscientemente conceptos más profundos de consumo de servicios Interacción de request y response Seguridad y manejo de tokens para consumo de servicios (JWT, tokens de sesión) 	<ul style="list-style-type: none"> Aplica estrategias de manejo de errores y excepciones, mencionando los tipos de los códigos de respuesta HTTP (200, 201, 300, 400, 500) 	<ul style="list-style-type: none"> Conciencia de dónde guardar y leer datos (Local storage, Session storage, Cookies, IndexedDB, etc)
Verificación	<ul style="list-style-type: none"> Resuelve los retos siguientes indicaciones adecuadas y documenta el código cuando necesito encontrar problemas 	<ul style="list-style-type: none"> Cumple con el DAD de una historia de usuario de frontend, haciendo por su cuenta pruebas unitarias (funcionales o de integración) 	<ul style="list-style-type: none"> Revisa el código de otros y da feedback técnico Compara y tiene experiencia aplicando conscientemente conceptos de diferentes principios de código limpio basados en SOLID, DRY, SCAD y KISS [4] 	<ul style="list-style-type: none"> Aplica conscientemente al menos 3 patrones de diseño [“Song of four patterns”] aplicados a frontend
Verificación	<ul style="list-style-type: none"> Nombre adecuadamente las variables, métodos y clases según estándares de Código Limpio 	<ul style="list-style-type: none"> Definición haciendo uso consciente del tipo de variables para mejorar mantenibilidad del proyecto 		
Verificación			<ul style="list-style-type: none"> Comprende el propósito de uso de funcionalidades orientadas a accesibilidad Etiquetas y descripciones adecuadas aria-label, aria-hidden documentación de tipografía [3] 	<ul style="list-style-type: none"> Aplica conscientemente conceptos de estándares a11y para mejorar accesibilidad web
Verificación		<ul style="list-style-type: none"> Reconoce implicaciones del SEO técnico y aplica conscientemente conceptos bases Jerarquía en títulos de página (h1 por página) Atributos en etiquetas y meta etiquetas (alt, name, description, lang) URL descriptiva y válida [5] 	<ul style="list-style-type: none"> Aplica conscientemente estrategias avanzadas de SEO técnico Noticias leg Feedback Skatecamp Fullstackcamp [1] 	<ul style="list-style-type: none"> Aplica conceptos de Core Web Vitals y los usa para tomar decisiones de código que promuevan soluciones en el contexto de negocio
Verificación			<ul style="list-style-type: none"> Conoce las principales características de la PDD (Influencia, abstracción, encapsulamiento) y las aplica para mejorar la calidad general del código. [6] 	
Verificación			<ul style="list-style-type: none"> Comprende los riesgos de vulnerar y minimizar código heredado [3] 	<ul style="list-style-type: none"> Aplica conceptos avanzados de seguridad web Identifica escenarios para reducir ataques tipo XSS o Inyección y sabe cuándo aplicar algunas de ellas
Verificación				<ul style="list-style-type: none"> Comprende los aspectos fundamentales de Cloud Computing y cómo se relaciona con la capa frontend Puede participar activamente en las definiciones de un proyecto/cloud
Verificación				<ul style="list-style-type: none"> Sabe cuándo documentar componentes de código reutilizables para el desarrollador general, usando vocabulario simple y concreto pensado en la experiencia de los desarrolladores (Shinetsu, ReactUI, etc)
Verificación		<ul style="list-style-type: none"> Aplica conceptos para mejorar la mantenibilidad de un proyecto frontend Establece de manera lógica los carpetas y componentes (Clases, design) Aplica estrategias de investigación eficiente para múltiples alternativas (Método first) 	<ul style="list-style-type: none"> Conoce y aplica al menos una técnica profesional de optimización y performance (Lazy loading o session UI) 	<ul style="list-style-type: none"> Entiende cuándo usar micro-frontends y aplica alguna estrategia para solucionar un problema de negocio (Frontend web abstract o non abstract)
Verificación				
Verificación				

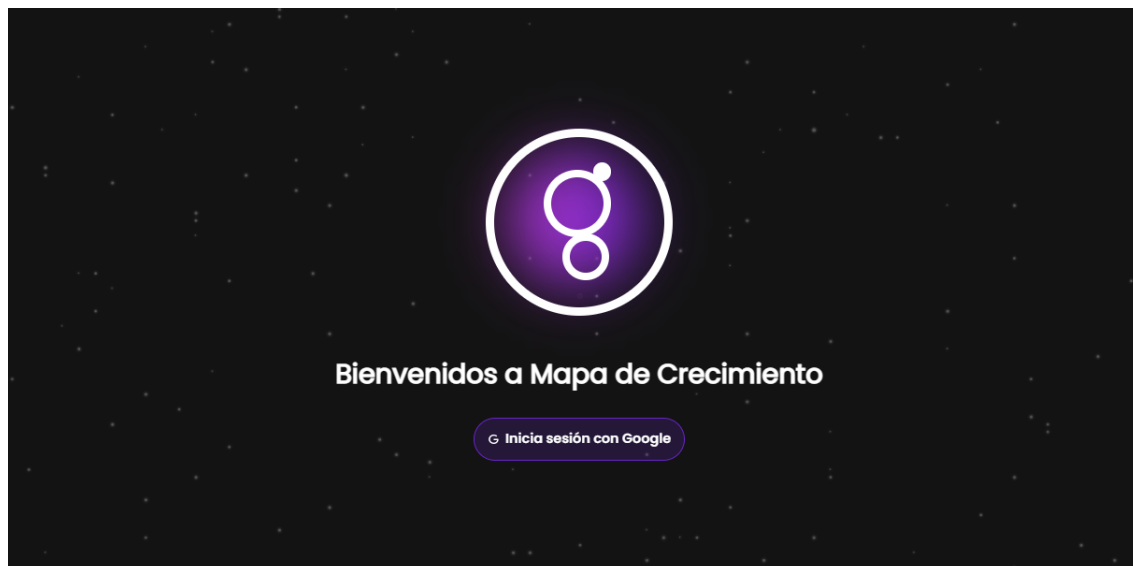
Figura 44. a) Lista histórico de valoraciones del chapter, b) Buscador en el histórico de valoraciones, c) Pestaña nueva que se abre al presionar el botón “Visualizar informe”.

4.5 Cerrar sesión en la aplicación.

Esta acción se logró por medio de una petición de cierre de sesión a Google y por ende a Cognito, para finalmente eliminar toda aquella información que la aplicación almacenó en el navegador como las cookies, token de acceso y demás variables. Eso se puede ver en la Figura 45.



a)



b)

Figura 45. a) Botón para cerrar sesión, b) Redirección a pantalla de Inicio de sesión al presionar el botón "Cerrar sesión"

5 Conclusiones

En este proyecto se desarrolló el frontend para la aplicación web “Mapa de Crecimiento” de la compañía Pragma S.A. la cual proporciona a los líderes de las diferentes áreas de conocimiento una herramienta que les facilita las tareas de gestión y monitoreo de los empleados y los equipos de desarrollo conformados, ya que permite acceder a información unificada y resumida de la evolución del área de conocimiento específica y de cada uno de los integrantes de esta; se le permite además llevar a cabo la valoración y clasificación de los empleados para identificar sus brechas y fortalezas y así proporcionarles la ruta de aprendizaje a seguir para que continúen con su proceso de crecimiento.

Por medio de una comunicación continua con el backend, el cual es la fuente de información de la aplicación y de cierta parte de la lógica de negocio, se le presenta a los líderes la información en la interfaz de usuario, la cual se puede filtrar y agrupar para agilizar las tareas de gestión, control y monitoreo.

Con la información presentada en la aplicación, el líder puede conocer cada uno de los integrantes del área de conocimiento por medio de una tabla que contiene la información relevante del empleado, como nivel técnico en que se encuentra, rol principal, proyecto al cual se encuentra asociado, entre otras.

El uso de la metodología de diseño Atomic Design garantizó generar componentes que tuviesen funcionalidades de responsabilidades únicas, lo cual permitió tener un buen control de la funcionalidad de estos y detectar fácilmente errores que se presentaron en el proceso de desarrollo. Gracias a las cualidades de modularización y reutilización que se proporcionaron a la hora de diseñar y desarrollar los componentes, su integración y unión con los demás, se llevó a cabo de manera sencilla, ágil y ordenada, puesto que se realizó la definición de niveles de jerarquía que facilitaron la comunicación entre estos.

Debido a la modularización que se hizo en el enrutamiento de la aplicación, se logró que los componentes y módulos desarrollados solo fuesen cargados en el momento en que se

requieren, obteniendo así muy buen desempeño y eficiencia en el proceso de carga de las pantallas de la aplicación ya que se redujo notoriamente el tiempo de carga de este.

El código del proyecto se realizó siguiendo los principios más importantes de código limpio, obteniendo como resultado un código muy bien estructurado, legible y fácil de entender, que le proporcionó al proyecto mantenibilidad y escalabilidad para así facilitar futuros desarrollos que permitan extender las funcionalidades de la aplicación.

Referencias

Seguro, N. (2022). ¿Qué es el desarrollo Web y por qué es tan importante?.

<https://us.coderhouse.com/blog/que-es-el-desarrollo-web>

Alarcón. J. M. (2019). ¿Qué necesitas saber en 2020 para ser un buen desarrollador web Front-End? shorturl.at/bcotP

Morales Oviedo, K. I. (2021). ¿Qué es “Clean Code”? Aprendiendo a mejorar tu código con Kotlin

<https://intive.com/es/careers/que-es-clean-code-aprendiendo-a-mejorar-tu-codigo-con-kotlin>

Lopez, P. (2021). Aprende Git de manera sencilla: Git Flow

<https://desarrollowp.com/blog/tutoriales/aprende-git-de-manera-sencilla-git-flow/>