



Título: Desarrollo e implementación FrontEnd en aplicativo de generación de certificados laborales

JUAN DIEGO QUINTERO URREA

Informe de práctica para optar al título de Ingeniero de Sistemas

Asesor Interno

Diana Margot Lopez Herrera, Magíster en Ingeniería Industrial

Asesor Externo

Johana Saldarriaga Naranjo, Magíster en gestión y desarrollo de proyectos de software

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería de Sistemas

Medellín

2024

Cita	Quintero Urrea [1]
-------------	--------------------

Referencia	[1]Quintero Urrea Juan Diego(2024). Desarrollo e implementación de FrontEnd en aplicativo de generación de certificados laborales. Proyecto de grado, Ingeniería de Sistemas Universidad de Antioquia, Medellín.
-------------------	--



Centro de Documentación Ingeniería (CENDOI)

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

Universidad de Antioquia - www.udea.edu.co

Rector: Jhon Jairo Arboleda Céspedes

Decano/Director: Julio César Saldarriaga Molina

Jefe departamento: Diego José Luis Botía

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

A mi familia y amigos que siempre me apoyaron durante todos estos años en la universidad y por fuera. A los que me ayudaron a crecer como persona y fortalecerme emocionalmente para los retos durante este proceso.

Agradecimientos

A todos los que creyeron y a los que no también. Familia, amigos, profesores y tutores que me acompañaron y dedicaron tiempo para brindarme enseñanzas para lograr los objetivos, y también que me ayudaron a formarme mejor como persona y profesional. A Blankfactor por brindarme esta oportunidad de sumergirme y comprender el entorno laboral actual.

Tabla de contenido

Resumen	7
Abstract	8
Introducción	9
Objetivos	10
A. Objetivo general	10
B. Objetivos específicos	10
Marco teórico	11
Metodología	14
Resultados	17
Diseño de Interfaz de Usuario en Figma:	17
Funcionalidad del Aplicativo Web para Generar Certificados:	18
Código limpio y buenas prácticas:	23
Despliegue en Vercel y AWS Docker:	31
Lecciones aprendidas y buenas prácticas	32
Conclusiones	33
Referencias	34

Lista de figuras

- Figura 1. División de componentes en atomic design
- Figura 2. Ciclo del marco de trabajo Scrum
- Figura 3. Visual Story Mapping. Tabla con características y versiones del producto a realizar.
- Figura 4. Flujo de las ramas con Git Flow
- Figura 5. Paleta de colores utilizada para la aplicación.
- Figura 6. Tipografía utilizada para la aplicación
- Figura 7. Iconografía utilizada para la aplicación
- Figura 8. Diseño en figma con rol de administrador
- Figura 9. Diagrama de flujo del aplicativo
- Figura 10. Vista del Login del aplicativo.
- Figura 11. Integración con google para el inicio de sesión.
- Figura 12. Vista de generación de certificado con botón de generar bloqueado
- Figura 13. Vista de generación de certificado con botón de generar desbloqueado
- Figura 14. Vista de generación de certificado con el pdf.
- Figura 15. Vista de pantalla de carga para la sección de reportes.
- Figura 16. Vista de pantalla con la sección de reportes cargada.
- Figura 17. Componente de calendario para selección de fechas.
- Figura 18. Vista validación rango de fechas.
- Figura 19. Componente de carga(Spinner)
- Figura 20 y 21. Vista componente Navbar y versión responsive
- Figura 22 y 23. Vista componente Sidebar y versión responsive
- Figura 24. Componente Breadcrumbs de navegación
- Figura 25. Vista del componente de formulario dinámicamente para admin.
- Figura 26. Vista componente formulario para empleado
- Figura 27. Vista componente de visualización del certificado generado

Figura 28. Componente “DatePicker” para seleccionar rango de fechas

Figura 29. Componente de autocompletado de lista de empleados

Figura 30. Componente de filtrado de la sección de reportes

Figura 31. Arquitectura realizada en el aplicativo focalizada en el Frontend

Figura 32. Organización de carpetas y componentes en el código del aplicativo según el patrón Atomic Design.

Figura 33. Lista de la cobertura de pruebas unitarias de los componentes.

Siglas, acrónimos y abreviaturas

AWS	Amazon Web Services
DOM	Document Object Model
HTPPS	Hypertext Transfer Protocol Secure
HU	Historia de Usuario
MVP	Producto Mínimo Viable
RTK	React Tool Kit
SPA	Single Page Application
UdeA	Universidad de Antioquia
UI	User Interface
UX	User Experience

Resumen

En este documento, se presenta detalladamente el trabajo desempeñado durante el semestre de práctica profesional en la empresa Blankfactor, ocupando el cargo de desarrollador FrontEnd. La labor se centró en la creación de un aplicativo web diseñado para automatizar el proceso de generación de certificados laborales de los empleados vinculados a la empresa, con un módulo de visualización de reportes asociados por parte del líder de recursos humanos, quien también tiene la capacidad de generar certificados para los empleados.

Por medio de metodologías ágiles como Scrum, se logra tener un proceso acorde al ámbito laboral, con reuniones diarias, en el que se destaca el trabajo en equipo y el potencial de adquirir nuevos conocimientos constantemente.

Se logra crear un diseño óptimo del aplicativo gracias a la herramienta Figma, investigando e implementando nuevas herramientas para el desarrollo de una aplicación web con React enfocada a un problema en un ambiente laboral. Desarrollando un manejo de la autenticación y autorización de acceso y gestión de la información, para crear un aplicativo seguro que permita la seguridad y confidencialidad de los datos.

Finalmente, se implementó la sección de reportes de gran utilidad para el área de recursos humanos y colaborando con la funcionalidad de la generación de los certificados los cuales pueden ser visualizados y descargados desde el aplicativo. Se cumplieron los requerimientos y objetivos del Mínimo Producto Viable (MVP) del proyecto, entregando una solución escalable y de calidad a la empresa. La duración del proyecto abarcó un semestre académico completo.

Palabras clave — Front-End, React, desarrollo web, automatización, Single Page Application, Figma, diseño web, arquitectura monolítica, atomic design, pruebas unitarias.

Abstract

This document, the work carried out during the internship semester at Blankfactor is presented in detail, holding the position of FrontEnd Developer. The focus of the work was on creating a web application designed to automate the process of generating employment certificates for the company's affiliated employees. This included a module for visualizing associated reports by the Human Resources leader, who also has the ability to generate certificates for employees.

Utilizing agile methodologies such as Scrum, the work process aligns with the professional environment, incorporating daily meetings that emphasize teamwork and the continuous potential for acquiring new knowledge.

The optimal design of the application was achieved through the use of the Figma tool, involving research and implementation of new tools for developing a web application with React, specifically addressing a workplace issue. This encompassed the development of authentication and access authorization, as well as information management, ensuring the creation of a secure application that prioritizes the safety and confidentiality of data.

Finally, a highly useful reports section for the Human Resources department was implemented, along with collaborating on the functionality of generating certificates, which can be viewed and downloaded from the application. The requirements and objectives of the Minimum Viable Product (MVP) of the project were successfully met. The project spanned an entire academic semester.

Keywords — Front-End, React, web development, automation, Single Page Application, Figma, web design, monolithic architecture, atomic design, unit testing.

Introducción

En el área de Recursos Humanos de Blankfactor, el proceso de generación de certificados laborales se ve actualmente afectado por retrasos debido a la gestión manual realizada por el líder del área. Este proceso implica la consulta en el aplicativo de Workday para obtener información sobre los empleados que están registrados en esta plataforma, así como un documento de Excel almacenado en Google Drive que contiene información sobre los empleados que dejaron la empresa antes de la implementación de dicho aplicativo. Workday es una plataforma en la nube que alberga datos sobre los empleados de la empresa. El archivo en excel requiere actualización manual mes a mes, lo que genera adicionalmente inconsistencia y reprocesos.

¿Cómo generar alguna solución a la problemática?, esta fue la pregunta que se intenta resolver en esta práctica y que da origen a proponer el desarrollo de una aplicación web que automatice el proceso de generación de certificados laborales. A través de una interfaz amigable, el líder de Recursos Humanos podría definir fácilmente las plantillas de los certificados, mientras que los empleados actuales de la empresa podrían gestionar de manera autónoma sus procesos de certificación.

La práctica académica busca desarrollar esta aplicación en conjunto con un grupo de practicantes. Durante los primeros dos meses se conocieron las instalaciones, políticas que maneja la empresa, también se conoce y analiza la problemática a resolver, y se establecieron las tecnologías y arquitecturas que se manejarían para desarrollar la aplicación. Esto se desarrolla utilizando marcos ágiles para mejorar la eficiencia y garantizar entregas de valor al cliente. El grupo de practicantes a cargo de este desarrollo, establecieron responsabilidades y en mi caso la responsabilidad se enfocó en el diseño e implementación del Front End del aplicativo.

Se usó Figma como principal herramienta para el diseño de las interfaces a desarrollar. ReactJs fue usado para el desarrollo del código gracias a su componentización que nos permite realizar un aplicativo reutilizable y escalable. AWS para su despliegue en la nube y diferentes herramientas de desarrollo ágil como Jira y Confluence. Se logra obtener un aplicativo funcional de acuerdo a los requerimientos especificados en los objetivos, el cual permite generar certificados laborales de forma automática por medio de un formulario dinámico que con unos campos determinados genera un pdf con el certificado, teniendo la posibilidad de descargarlo. Esto solo lo pueden realizar las personas vinculadas a la empresa y el líder de recursos humanos tiene la opción de generar certificados de cualquier empleado vinculado en la empresa. También el líder de recursos humanos tiene la opción de visualizar un historial de todos los reportes que han sido generados, con la posibilidad de filtrarlos por una fecha determinada y/o el nombre o ID del empleado.

Se logró entregar a la empresa un activo, un aplicativo mantenible y escalable para los futuros desarrolladores que retomen el aplicativo.

Objetivos

A. Objetivo general

Desarrollar y diseñar el FrontEnd de una aplicación web que logre automatizar el proceso de generación de certificados laborales en el área de Recursos Humanos de Blankfactor. Eliminando la gestión manual actual y reduciendo significativamente el tiempo dedicado a esta tarea.

B. Objetivos específicos

- Diseñar mockups del aplicativo respetando los lineamientos de diseño de la empresa.
- Diseñar una interfaz fácil de usar que permita al responsable de Recursos Humanos cargar y gestionar las plantillas necesarias para la generación de certificados.
- Integrar la funcionalidad de autorización para lograr un flujo del aplicativo para administrador o líder de recursos humanos y otro diferente para empleados.
- Colaborar en la implementación de funcionalidades para la generación y visualización de certificados laborales destinados a los empleados..
- Diseñar un sistema de acceso al aplicativo donde solo los vinculados a la empresa puedan ingresar para generar sus certificados.
- Procurar que la aplicación cumpla con los estándares de seguridad y privacidad de datos, protegiendo la información confidencial de los empleados y cumpliendo con las regulaciones aplicables.
- Desarrollar un módulo de reportes que permita ver la información/histórico de los certificados emitidos con la posibilidad de poder buscar y filtrar para mayor flexibilidad a la hora de necesitar certificados en una fecha o con un nombre en específico.
- Mantener un código limpio, mantenible y escalable para el manejo de buenas prácticas y que sea de fácil legibilidad para los futuros desarrolladores del proyecto.

Marco teórico

La gestión eficiente de los recursos humanos es esencial para el buen funcionamiento de cualquier organización. En el área de Recursos Humanos de Blankfactor, se identifica una problemática específica relacionada con la generación de certificados laborales, la cual se aborda a través de la propuesta de desarrollo de una aplicación web.

La automatización de procesos en el área de Recursos Humanos ha ganado relevancia en la optimización de tareas y la mejora de la eficiencia organizacional. En este contexto, la generación de certificados laborales es una actividad crítica que puede beneficiarse de soluciones tecnológicas para agilizar y mejorar su ejecución. (Torres P, 2019)

Para ser más claros, un certificado laboral es una carta de constancia de trabajo. Generalmente, se crea con el membrete o en papelería oficial de una organización, para demostrar que la persona solicitante es un empleado actual o anterior de ella. Comúnmente, se solicita este documento para verificar el estatus laboral, título del cargo y salario del ciudadano. Muchas veces, el certificado laboral incluye el historial de empleo, los domicilios de los trabajos anteriores, el crecimiento salarial y un resumen del desempeño ocupacional. (Indeed, 2023)

Un paso para lograr darle solución a esta problemática es crear el diseño de la interfaz de usuario para el aplicativo, para esto debemos conocer las herramientas necesarias para realizarlo, en este caso Figma. Figma Design sirve para crear, compartir y probar diseños de sitios web, aplicaciones móviles y otros productos y experiencias digitales. Es una herramienta popular para diseñadores, gestores de productos, escritores y desarrolladores, y ayuda a cualquier persona involucrada en el proceso de diseño a contribuir, dar su opinión y tomar mejores decisiones, más rápido. (Figma, 2023)

Luego debe implementarse el diseño desarrollado, en este caso se realizaría una aplicación web. El desarrollo web juega un papel crucial en la automatización de procesos debido a varios factores que contribuyen a mejorar la eficiencia, la accesibilidad y la escalabilidad de los procesos empresariales. Las aplicaciones web permiten el acceso desde cualquier lugar con conexión a Internet, lo que facilita la distribución y la disponibilidad para los usuarios, independientemente de su ubicación física.

Las tecnologías de desarrollo web, como Typescript y Frameworks modernos (React, Angular, Vue), permiten el desarrollo rápido y ágil de aplicaciones. Esto es beneficioso para la automatización de procesos, ya que facilita la adaptación rápida a cambios en los requisitos comerciales. (Williams, D. E., & Miller, P. A, 2019)

Un mejor manejo del código, a la hora de dar tipado a la información y hacer más legible el código, y más segura la aplicación, implementamos TypeScript, que es un lenguaje de programación fuertemente tipado que se basa en JavaScript, proporcionándole mejores herramientas a cualquier escala. (TypeScript, s.f.)

Para el proceso de desarrollo del FrontEnd es muy importante conocer profundamente las herramientas a usar. La librería React, React JS es una biblioteca de código abierto que se utiliza

para crear interfaces de usuario explícitamente para aplicaciones de una sola página. ReactJS permite a los ingenieros de software crear inmensas aplicaciones web que pueden usar datos y cambiar después de un tiempo sin recargar la página. (Aggarwal, S., 2018).

Durante este proceso, el seguir buenas prácticas para una mejor legibilidad, mantenibilidad y escabilidad del aplicativo es fundamental, implementar una metodología de organización de componentes, Atomic design es una metodología compuesta por cinco etapas distintas que trabajan juntas para crear sistemas de diseño de interfaces de forma más deliberada y jerárquica. Las cinco etapas del diseño atómico son:

1. Átomos
2. Moléculas
3. Organismos
4. Plantillas
5. Páginas

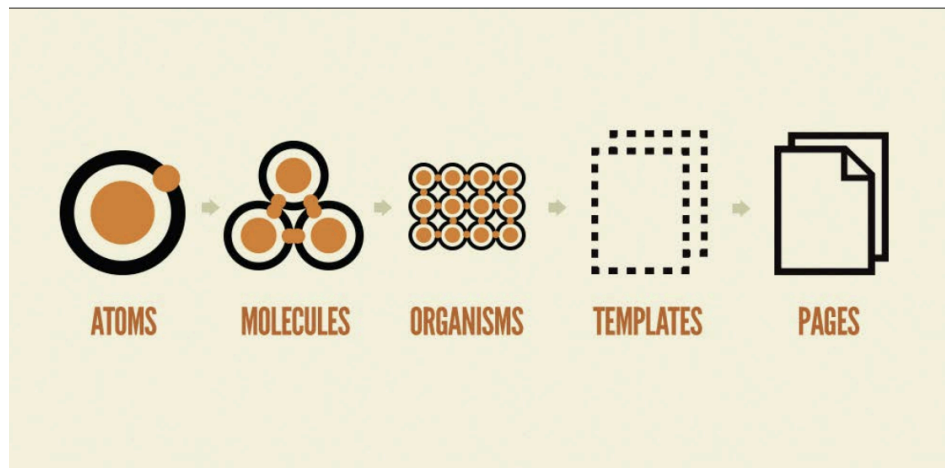


Figura 1 (Frost, B., 2016)

El manejo de la información que circula en el aplicativo se realizó por medio de Redux, el cual nos permite seguir buenas prácticas evitando manejos incorrectos de información dentro del aplicativo. Redux es un contenedor predecible del estado de aplicaciones JavaScript. Ayuda a escribir aplicaciones que se comportan de manera consistente, corren en distintos ambientes (cliente, servidor y nativo), y son fáciles de probar. Además de eso, provee una gran experiencia de desarrollo, gracias a edición en vivo combinado con un depurador sobre una línea de tiempo. (Official Redux Documentation., 2023)

En estos aplicativos empresariales, la seguridad y privacidad de los datos son elementos críticos en el desarrollo de la aplicación. La automatización del proceso de generación de

certificados debe cumplir con estándares de seguridad para proteger la información confidencial de los empleados. Esto incluye la aplicación de medidas de encriptación, acceso controlado y el cumplimiento de regulaciones pertinentes. (El Emam, K., & Neri, E., 2017)

Para mantener la seguridad del envío de información entre el backend y el frontend existe una herramienta llamada JSON Web Token (JWT) , el cual es un estándar abierto (RFC 7519) que define una forma compacta y autocontenida de transmitir información de forma segura entre las partes como un objeto JSON. Esta información puede ser verificada y fiable porque está firmada digitalmente. Los JWT pueden firmarse utilizando un secreto (con el algoritmo HMAC) o un par de claves pública/privada. (JWT.io., s.f.)

El marco de trabajo Scrum el cual es adecuado para empresas en las que el desarrollo de los productos se realiza en entornos que se caracterizan por tener incertidumbre, autoorganización, control moderado y transmisión del conocimiento. Scrum al ser un marco de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo, en Scrum llamados “Sprints”. (Trigás Gallego, M., 2012)

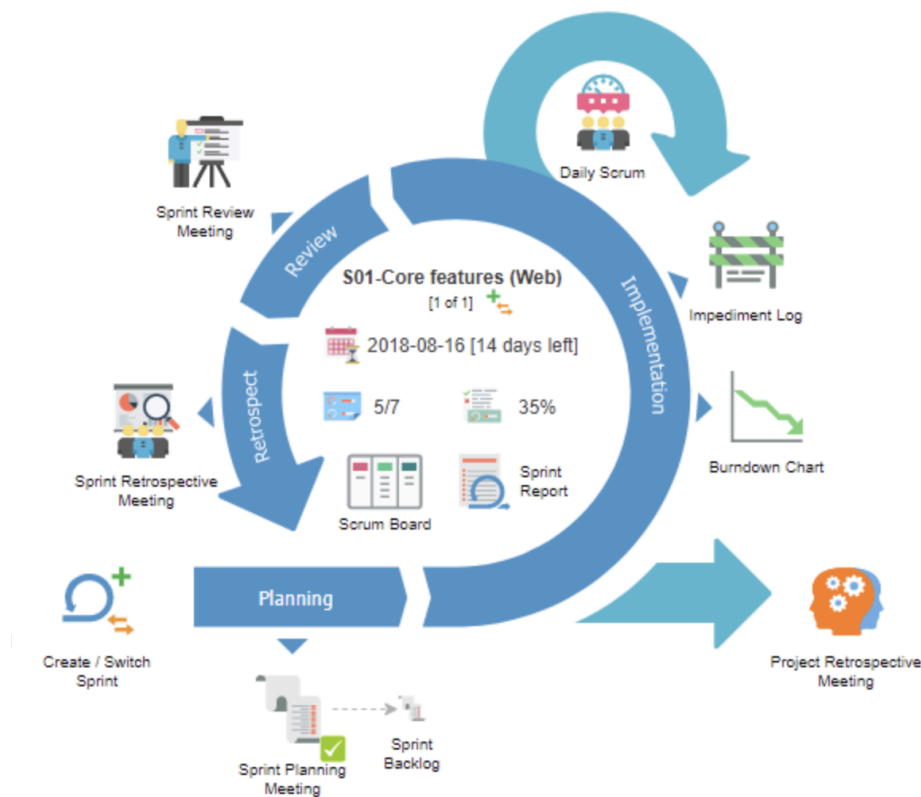


Figura 2 (Visual-paradigm., s.f.)

Para poder desplegar de una manera rápida y eficiente el aplicativo para desarrollo y que la persona encargada en hacer el testing pueda poder probarlo de la mejor manera, se utilizó la

plataforma llamada Vercel, el cual proporciona las herramientas, los flujos de trabajo y la infraestructura que necesita para crear y desplegar sus aplicaciones web más rápidamente, sin necesidad de configuración adicional. Vercel es compatible con los marcos de front-end más populares y su infraestructura escalable y segura se distribuye globalmente para servir el contenido desde centros de datos cercanos a sus usuarios para lograr velocidades óptimas. (Vercel., s.f.)

Metodología

El proyecto se desarrolló bajo una serie de actividades que integran el marco ágil Scrum y diversas metodologías; a continuación se describen los pasos que se siguieron. En primer lugar, se realizó la fase de investigación y entendimiento, donde se entra a conocer la empresa, su estructura, procesos y cultura, también se identificaron las áreas de oportunidad y los desafíos existentes en la empresa. Se definió un alcance del proyecto estableciendo los objetivos claros y específicos, y delimitando el alcance del trabajo de grado y la solución a la propuesta.



Figura 3

Para un manejo ágil y adaptativo del desarrollo del aplicativo se decidió implementar el marco de trabajo ágil Scrum un enfoque que se adapta a la naturaleza iterativa del desarrollo realizando en cada iteración las actividades de análisis, diseño, codificación y pruebas.

En la primera iteración se conformó el equipo de trabajo, tres compañeros en Backend, dos en Frontend y una en QA; y con la ayuda y guianza de tres tutores por equipo. Siendo un

equipo de desarrollo Scrum, tenemos un Product Owner al cual le entregaremos el aplicativo, un Product Manager que nos ayudó para la comunicación con el cliente, y un Team Lead/Scrum master

Planificación con Scrum:

- Se creó un backlog inicial en JIRA con las funcionalidades y tareas identificadas para el proyecto.
- Se dividió el proyecto en sprints de 2 semanas y establecer prioridades para cada sprint.

Cada Sprint tuvo sus respectivas reuniones para implementar este marco de trabajo de la siguiente manera:

- **Planning** al inicio para definir las HUs y tareas a realizar y a entregar para el Sprint
- **Refinement** a mitad de sprint para organizar sprint backlog y tareas.
- **Review** para mostrar los objetivos alcanzados del Sprint
- **Retrospective** para como equipo mirar que se puede mejorar, cómo se puede mejorar y que se puede mantener o seguir haciendo.
- **Dailys** reuniones cortas para centrarse en los obstáculos que pueden impedir el progreso.

Se designan los primeros Sprint para el diseño y arquitecturas a usar en el Front-end del aplicativo.

Finalmente se realizaron pruebas de integración

Se desplegó el aplicativo y se liberó la documentación.

La capacitación se realizó en dos vías: la técnica, para que el producto pueda seguir escalando en alcance y la de uso para que los usuarios conozcan el producto.

Se implementaron herramientas como Slack, Jira, Confluence para optimizar la colaboración y la gestión del proyecto. Slack como una herramienta informal para mantener a los miembros del equipo en sintonía y permitir una colaboración fluida. Además, se empleará JIRA para describir con precisión los problemas y realizar un seguimiento detallado del tiempo de trabajo en cada tarea.

Para la documentación y el almacenamiento de información se empleó Confluence, brindando un espacio centralizado para la creación y el acceso de todo el equipo a los recursos del proyecto. En general, esta combinación de metodología Scrum, Slack, reuniones de Google, JIRA y Confluence asegura un enfoque coherente y colaborativo en todas las etapas del proyecto.

Teniendo en cuenta el desarrollo del proyecto con metodologías ágiles, se necesitó un flujo de trabajo ágil en github para poder estar adaptados a cambios o nuevas versiones, siguiendo esto para mantener un código ordenado y limpio, se implementó Gitflow de la siguiente manera:

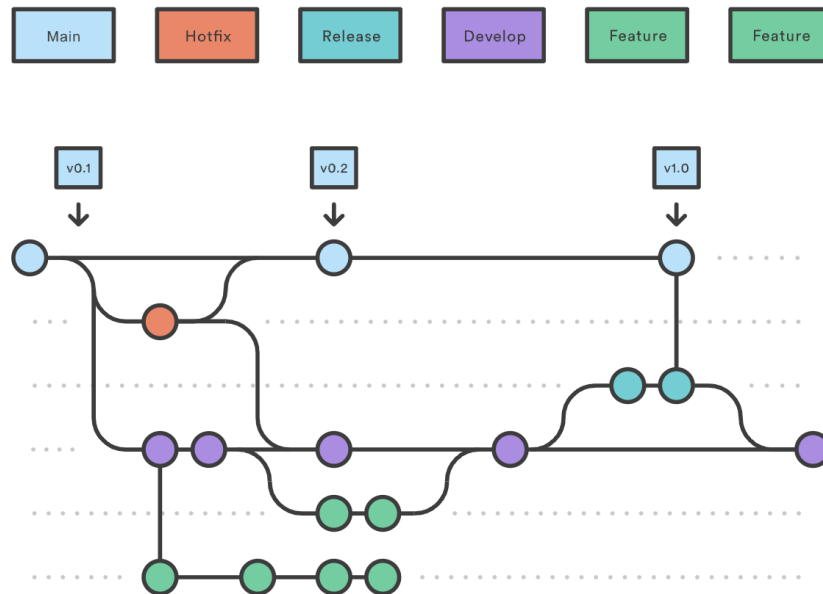


Figura 4 (Atlassian., s.f.)

Se creó una rama principal “main”, siendo la versión del proyecto lista para producción. Otra rama de “QA” para tener una versión del código preparada para las respectivas pruebas realizadas por los encargados de asegurar la calidad del producto. Una rama “Dev” donde se actualiza con las características que realizan los desarrolladores durante el proceso por medio de ramas llamadas “features”, y así mantener un orden de versionamiento del proyecto a lo largo del ciclo de desarrollo. También si surgen bugs las ramas de features tomarán el nombre de “Hotfix”.

Resultados

Diseño de Interfaz de Usuario en Figma:

Se creó un diseño en Figma que sigue los lineamientos de la empresa en cuanto a colores, logotipos, tipografía, referencias de tamaños, entre otros; para garantizar coherencia visual y una experiencia de usuario consistente a la empresa.

Este diseño tiene integrado dos flujos de la aplicación, uno para el administrador o líder de recursos humanos el cual tiene la funcionalidad de generar certificados de los empleados y adicional ver una barra lateral o “sidebar” para navegar al histórico de reportes si desea visualizarlos; y otro para los empleados que solo podrán generar su certificado.

La interfaz es amigable, intuitiva y cumple con los estándares de usabilidad.

A lo largo de este camino se logra entender y comprender acerca de la creación de una Interfaz de Usuario(UI) y de una buena Experiencia de Usuario(UX) para desarrollo de aplicaciones web.

01. Colors



Figura 5

02. Typography

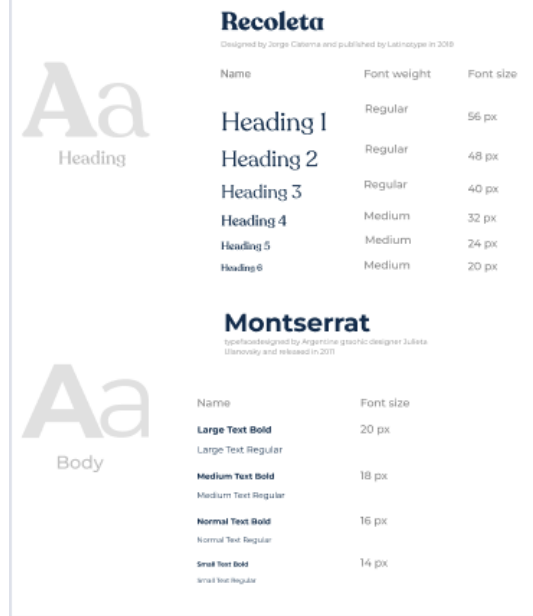


Figura 6

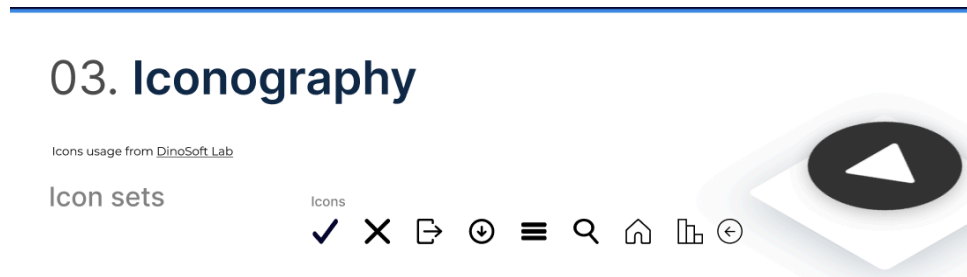


Figura 7

Se tuvo en cuenta los lineamientos internos de la empresa para el uso de los colores, iconografía, tipografía.

Funcionalidad del Aplicativo Web para Generar Certificados:

Se implementó un aplicativo web que permite al líder de Recursos Humanos generar certificados para ella y otros empleados. También que cada empleado pueda ingresar y generar su propio certificado. Los certificados generados pueden ser descargados en formato digital para su uso inmediato.

Todo esto con un formulario dinámico que permite mostrar solo unos campos los cuales de acuerdo a lo seleccionado se renderizan diferentes campos. Gracias a esta implementación esta funcionalidad se abre para poder ser utilizada de diferentes maneras de acuerdo a los filtros elegidos, mostrando diferentes tipos de certificados a generar.

Se diseñó e implementó una página de reportes que proporciona un historial completo de los certificados generados que solo podrá ver el líder de recursos humanos. Con la posibilidad de filtrar los reportes por fecha, nombre o ID para una fácil búsqueda.

Para la creación de este componente de reportes se integró con los diferentes filtros de búsqueda en este caso un selector de fecha, para lograr seleccionar las determinadas fechas y tener un rango de búsqueda, y también integrarlo con un campo de autocompletado donde se podrá buscar a cualquier empleado por el nombre o por el ID de workday para su búsqueda.

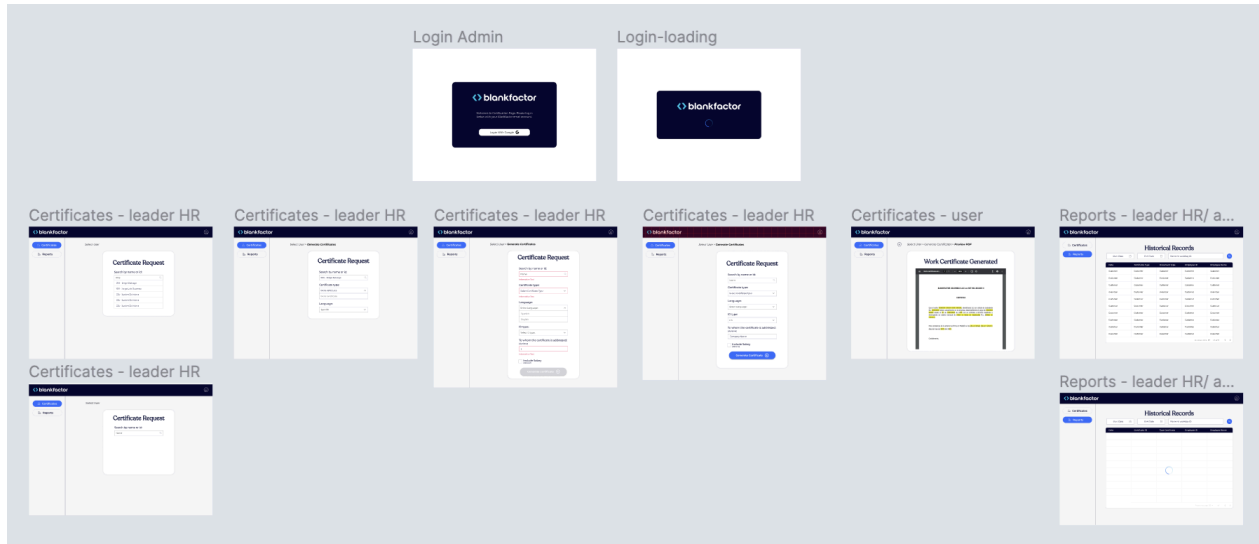


Figura 8

Para mostrar un poco flujo del aplicativo se realizó este diagrama de flujo (Figura 9) donde el líder de recursos humanos puede iniciar sesión, se verifica la vinculación y rol en la empresa, luego puede generar algún certificado, visualizarlo y/o descargarlo, adicional puede visualizar los reportes, para luego poder cerrar sesión. Y por el lado del empleado podrá hacer lo anterior exceptuando la visualización de reportes y generando solo su certificado.

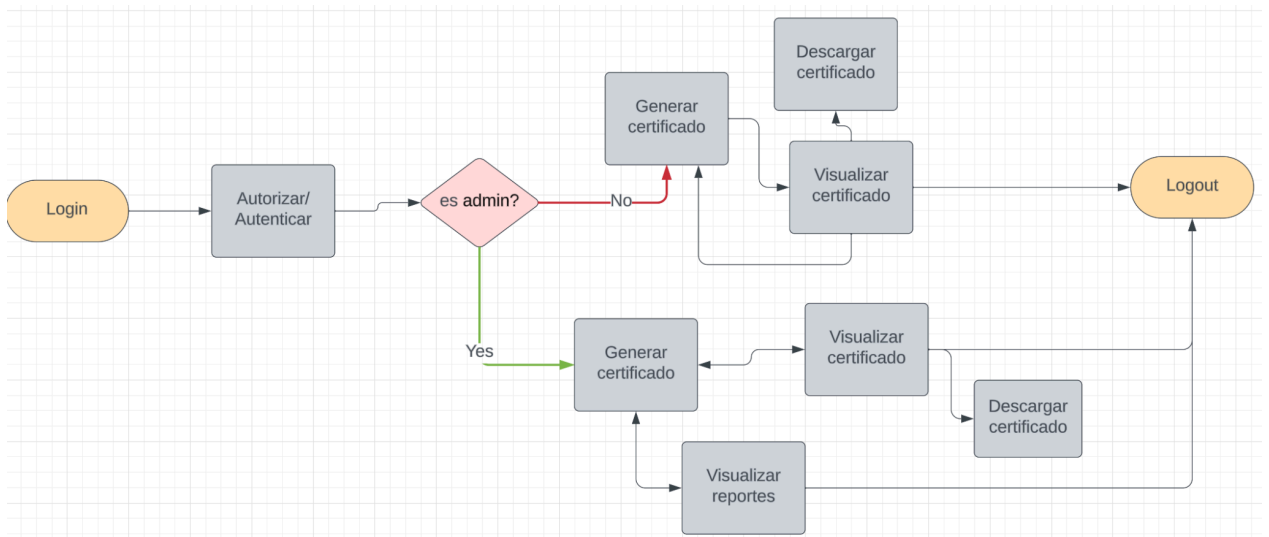


Figura 9

Para la pantalla de Login (Figura 10), se llevó a cabo una integración con Google para implementar la autenticación a través de esta plataforma como se muestra en la Figura 11. En conjunto con JSON Web Tokens (JWT), la información que el backend necesitaba enviar al frontend fue cifrada. Posteriormente, mediante un decodificador en el lado del cliente, descriptamos el token para gestionar la información necesaria en diversas funcionalidades. Mejorando la seguridad del aplicativo se integra el protocolo HTTPS para tener una forma más segura de enviar datos entre el servidor y el navegador web.

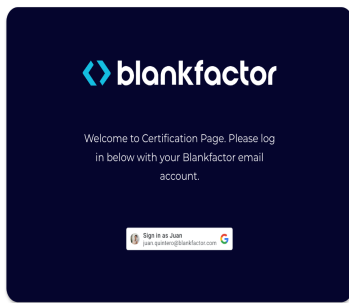


Figura 10

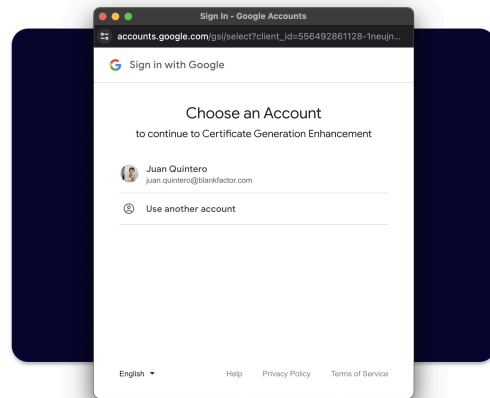


Figura 11

La funcionalidad más importante es la generación de certificados, el cual posee la vista de la figura 12 donde el usuario selecciona el tipo de certificado a generar, el lenguaje con el cual quiere el certificado, el tipo de documento, y dos campos opcionales que en este caso sería a quién irá dirigido el documento, si desea añadir el salario en el documento. El certificado no puede ser generado sin seleccionar los campos obligatorios, tal como se muestra en la figura 12 con el botón de generación inhabilitado.

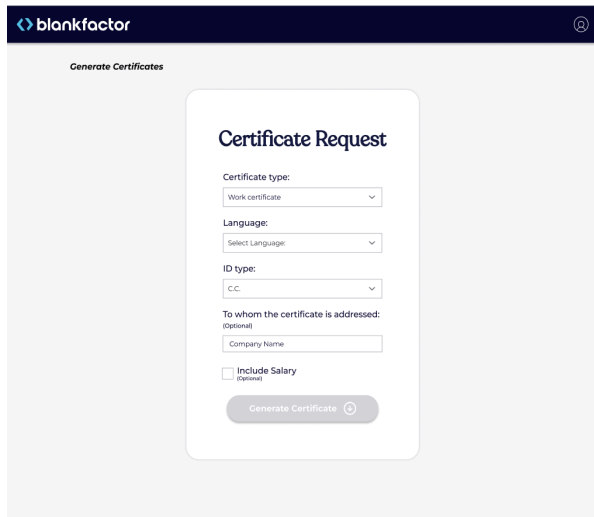


Figura 12

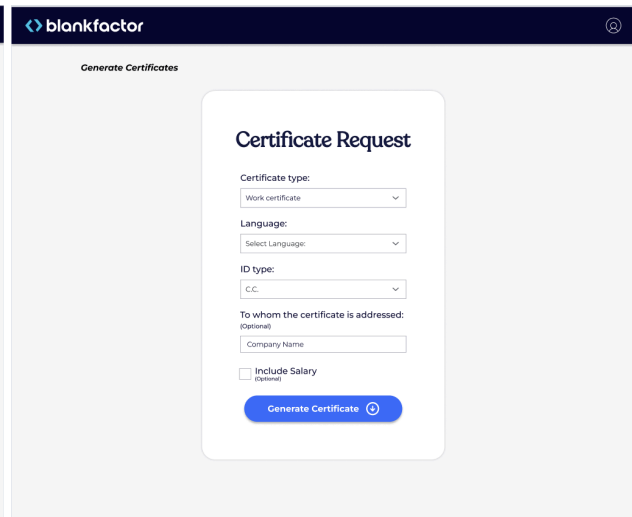


Figura 13

Al generar el certificado se logra visualizarlo en la pantalla en formato de pdf, el cual puede ser descargado si se desea en formato digital. Se tiene en cuenta si el usuario tuvo algún error al diligenciar el formulario, puede devolverse a la pantalla de generación de certificados sin ningún problema con el botón de flecha hacia atrás (figura 14).

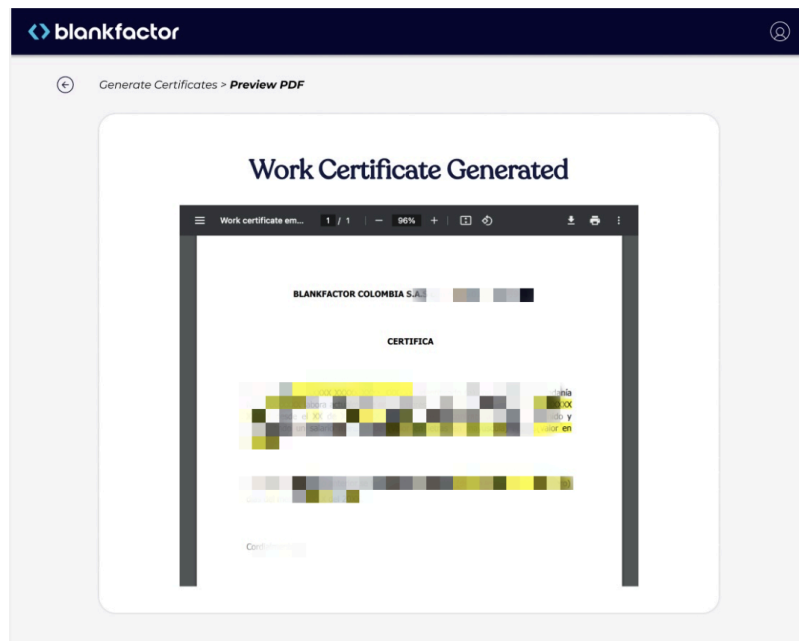


Figura 14

Se desarrolló una pantalla de informes (Figura 16 la información de la imagen está anonimizada visualmente) conforme a las especificaciones del propietario del producto. Esta pantalla incluye los campos críticos que el Líder de Recursos Humanos necesita visualizar, tales

como fecha, tipo de certificado, tipo de documento, número de documento y nombre del empleado, como se ilustra a continuación: La pantalla de informes única para el líder de recursos humanos o administrador, tiene su respectiva carga de la tabla (figura 15) que trae la información del histórico de todos los reportes que se han generado.

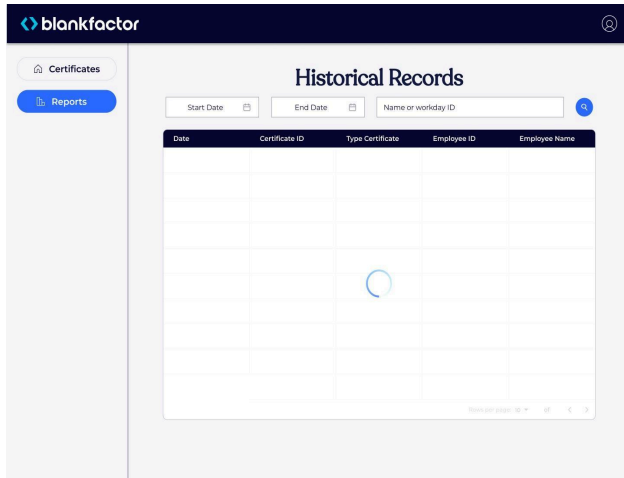


Figura 15

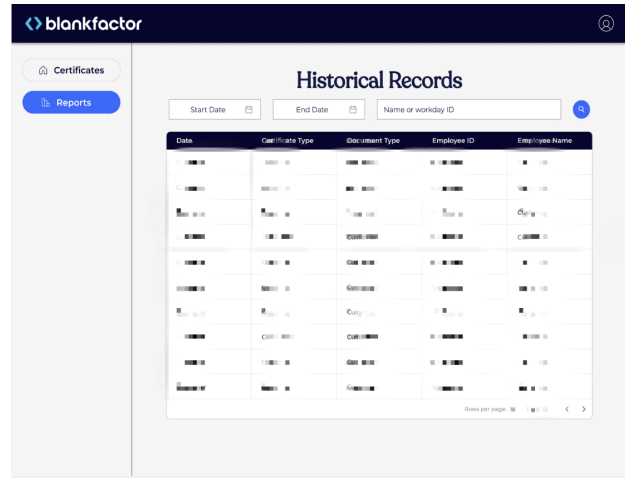


Figura 16

Esta tabla de reportes tiene la opción de filtrar por rango de fechas. Se realizaron las respectivas validaciones importantes en el selector de fechas, para que el usuario no elija una fecha anterior al 2023, o una fecha mayor al día actual (Figura 18); también bloqueando el campo de la fecha final si no se ha seleccionado una inicial. En este caso se desarrolló o usó un tipo de calendario obtenido gracias a la librería de estilos MaterialUI, el cual nos permite seleccionar una fecha en específica y poderla guardar para integrar el rango de fechas con otro componente de calendario (Figura 17).

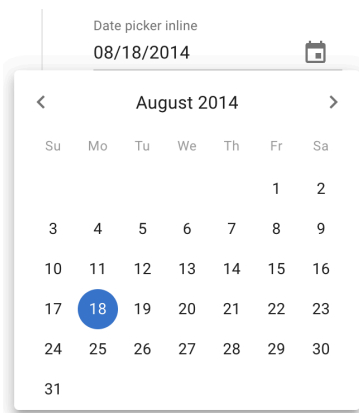


Figura 17

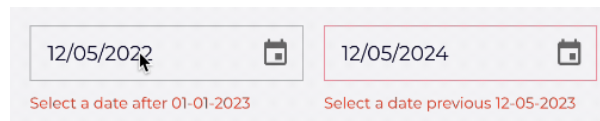


Figura 18

La tabla de reportes cuenta con paginación para mejorar la experiencia de usuario y la eficiencia en la visualización de datos extensos.

Para cada respectiva solicitud al backend se realiza su vista de carga, añadiéndole un Spinner, un componente proporcionado por MaterialUI el cual es un círculo que carga, simulando una carga, e integrándose con los colores y características de la empresa.

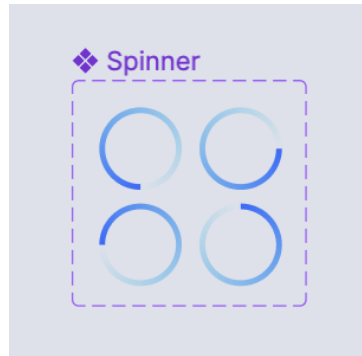


Figura 19

Código limpio y buenas prácticas:

El código del aplicativo web fue desarrollado siguiendo buenas prácticas de programación, gracias a los conocimientos obtenidos en React y Typescript, y con la ayuda de los tutores durante este proceso.

En este sentido para lograr un código escalable y mantenible se seleccionan y dividen los componentes por medio de Figma para lograr comprender mejor cuáles serían los componentes por reutilizar y organizar con el patrón de Atomic Design de la siguiente manera:



Figura 20

En este caso en el Navbar se logró darle el agregado de hacerlo responsive, es decir, que responda adecuadamente a diferentes dispositivos o tamaños de pantallas junto con el Sidebar. Tiene la funcionalidad de poder ver el nombre del usuario ingresado, la imagen de perfil que tenga en google y poder cerrar sesión, y en formato para móvil un botón para abrir o cerrar el sidebar.

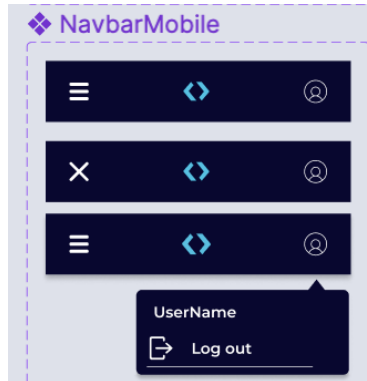


Figura 21

El componente de sidebar de acuerdo a los requerimientos, solo podría ser mostrado para el líder de Recursos Humanos o administrador, cumpliendo sus respectivos diseños para la navegación. Una lista de elementos para navegar en la aplicación, en este caso a Certificados, o Reportes. En su versión móvil trabajará de la misma manera solo que integrado con el Navbar para abrirlo o cerrarlo.

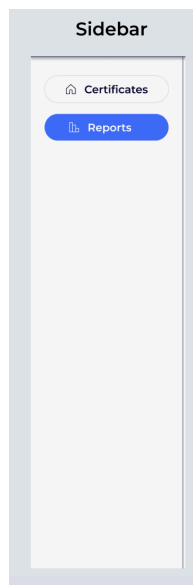


Figura 22

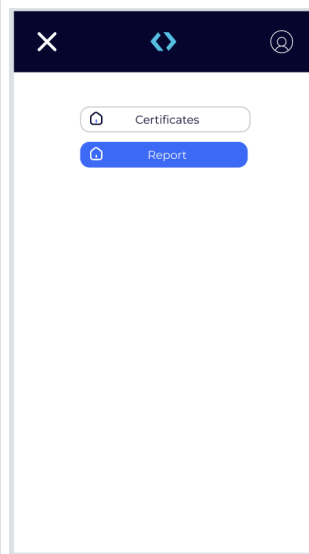


Figura 23

También se realizó un pequeño componente de navegación de la generación de certificados donde el usuario pueda observar en qué sección está ubicado(Figura 24), con la opción escalable de que hayan nuevas funcionalidades para la navegación. Este componente es llamado “breadcrumb” (Figura 18) en el manejo de interfaz de usuario el cual permite seleccionar algún título para navegar a la sección anterior y se va extendiendo si hay más páginas para navegar dependiendo una de la otra.

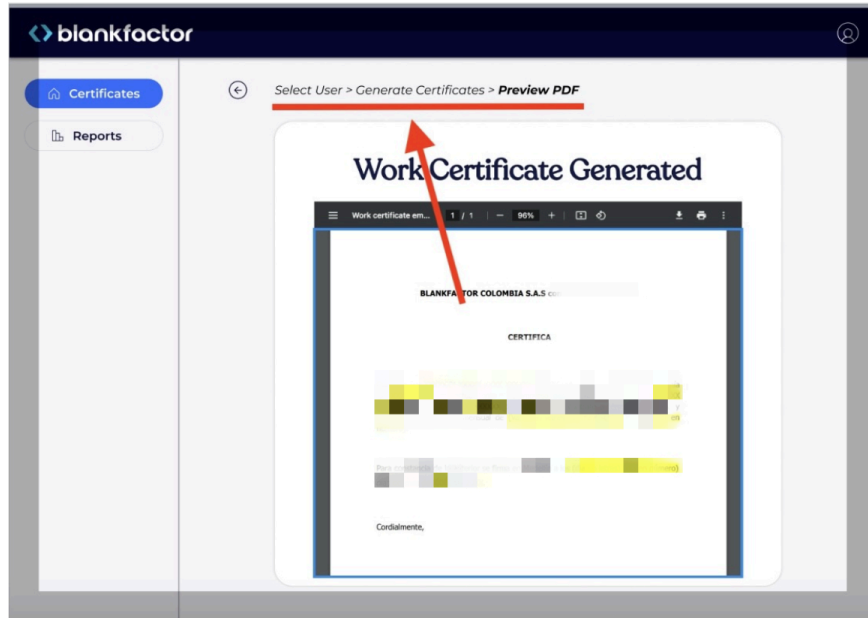


Figura 24

Un componente muy importante para los requerimientos del aplicativo es el formulario de configuración del certificado(Figura 25), el cuál se realizó de manera dinámica colaborando con el diseño e implementación.

En este caso, cuando ingrese el líder de recursos o administrador, podrá generar el certificado de cualquier empleado registrado en Workday gracias al campo de “Search by name or id”.

Con ayuda de herramientas que proporciona React, se puede lograr enviar información sobre el empleado seleccionado al backend y traer algunos datos para mostrar en pantalla los demás campos según el empleado y el tipo de certificado a generar como se muestra en la siguiente figura.

The figure shows two versions of the 'Certificate Request' form. The left version is a simplified view with search and filter options. The right version is the full form with a 'Generate Certificate' button.

Left Screenshot:

- Search by name or id:** Input field with 'Name' and a search icon.
- Certificate type:** Dropdown menu with 'Work certificate' selected.
- Language:** Dropdown menu with 'Spanish' selected.

Right Screenshot:

- Search by name or id:** Input field with 'Name' and a search icon.
- Certificate type:** Dropdown menu with 'Select Certificate Type' selected.
- Language:** Dropdown menu with 'Select Language:' selected.
- ID type:** Dropdown menu with 'C.C' selected.
- To whom the certificate is addressed: (Optional)** Input field with 'Company Name'.
- Include Salary (Optional)**
- Generate Certificate** button with a download icon.

Figura 25

En cuanto a la versión para un empleado sin rol de administrador o líder de recursos, tendría esta vista del componente, sin la opción de seleccionar algún usuario. (Figura 26)

The figure shows the 'Certificate Request' form for an employee without administrative or resource leader roles. It includes the filter and form options but lacks the search field.

Form Options:

- Certificate type:** Dropdown menu with 'Work certificate' selected.
- Language:** Dropdown menu with 'Select Language:' selected.
- ID type:** Dropdown menu with 'C.C.' selected.
- To whom the certificate is addressed: (Optional)** Input field with 'Company Name'.
- Include Salary (Optional)**
- Generate Certificate** button with a download icon.

Figura 26

Este componente va de la mano con la generación y visualización de certificados, también con la función incorporada de descarga del certificado. (Figura 27)

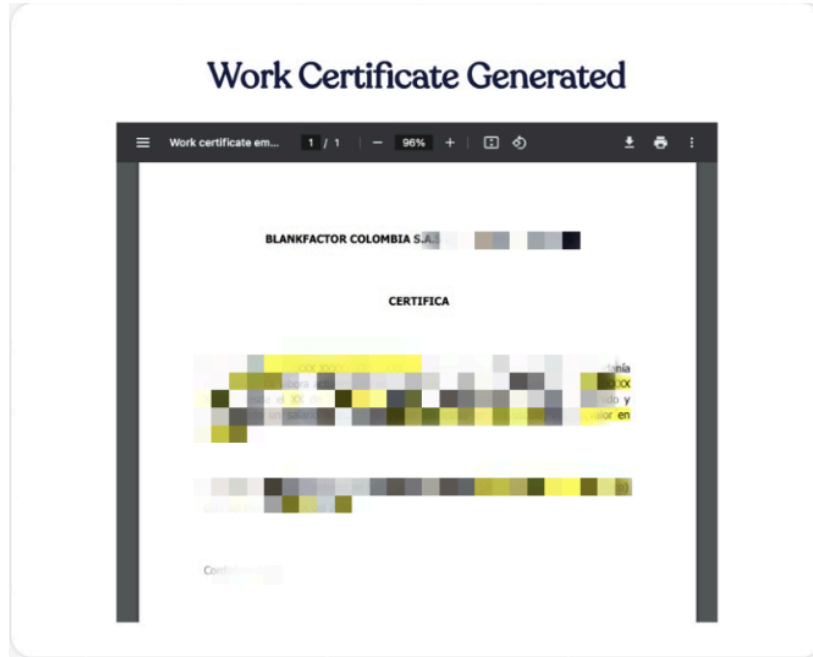


Figura 27

En cuanto a la sección de reportes, un componente complejo de manejar pero finalmente logrando obtener los resultados esperados a la hora de seleccionar un rango de fechas para filtrar en la tabla de reportes. Se implementó con dos calendarios y guardando el rango seleccionado en un estado para luego poder filtrar en la búsqueda. En las fechas se puede seleccionar por día, mes y año. (Figura 28)

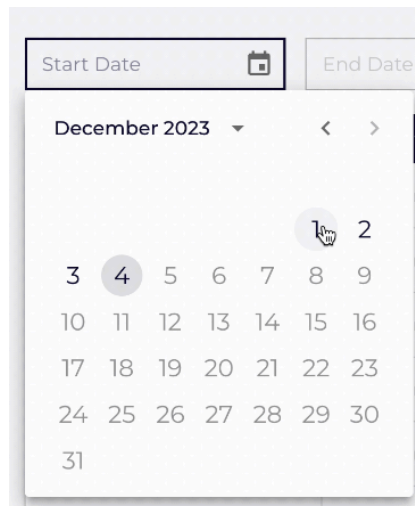


Figura 28

Otro componente que ayuda a hacer el filtro más completo es el del Autocompletado, que permite elegir a una persona de la empresa por workday id o nombre para buscar en la tabla de reportes. (Figura 29)



Figura 29

El filtrado completo de la sección de reportes queda de la siguiente manera (Figura 30) donde se puede filtrar según los requerimientos propuestas, y con la opción de eliminar los filtros para volver a buscar sin necesidad de refrescar la página.

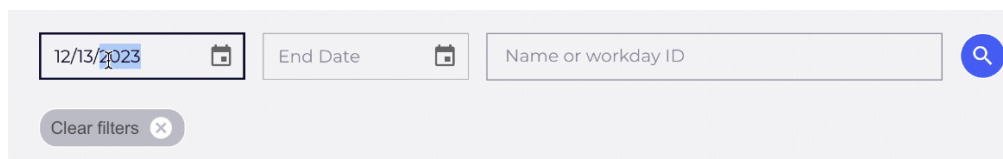


Figura 30

Se implementó una arquitectura modular que facilita la escalabilidad y el mantenimiento futuro. Por medio de una SPA(Single Page Application) nos permite renderizar los componentes encargados de la funcionalidad de la página web, sin necesidad de recargar todo el DOM o la página web.

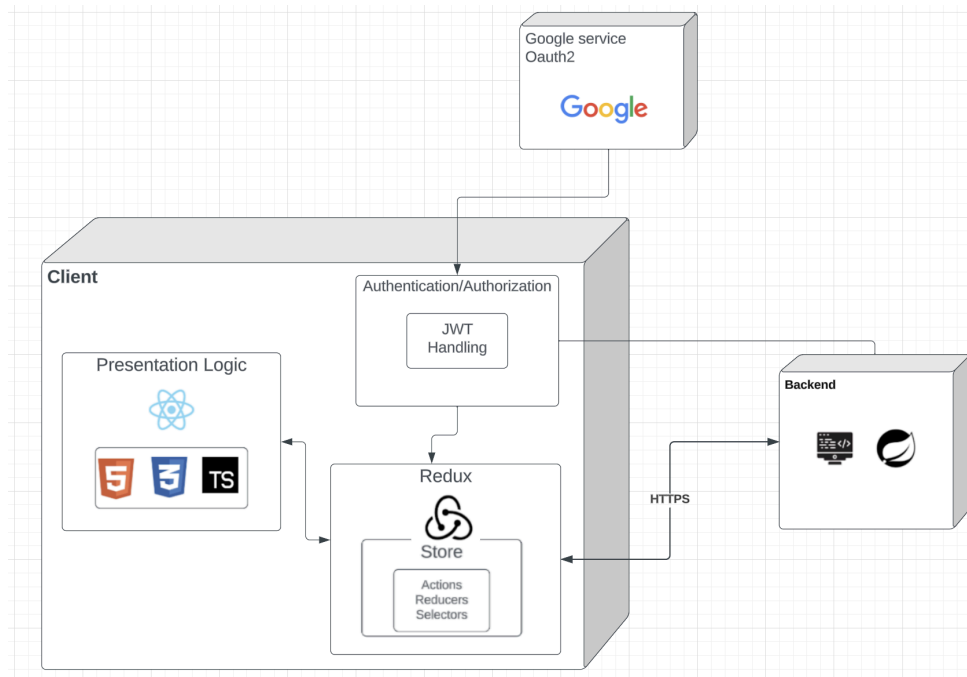


Figura 31

Se crea una arquitectura monolítica la cual nos permite manejar el código del front-end separado del back, dando independencia para mejor legibilidad y mantenibilidad del código por parte del front-end. Al no ser un proyecto robusto no se opta por arquitecturas como microservicios. La comunicación a los servicios del back-end para la obtención, actualización, eliminación y creación de datos, se realiza por medio de peticiones RESTful, a través de peticiones permitidas desde el back-end.

La autenticación se implementó mediante la herramienta de Google OAuth2, fundamental para el proceso de inicio de sesión debido a la vinculación de las cuentas de la empresa con Google. Respecto a la autorización, el backend transmite información y roles mediante un token. Este token es codificado y decodificado a través de JWT, asegurando la seguridad en la transferencia de información entre el backend y el frontend.

Se usa una metodología para la organización del código llamada Atomic design, que gracias a este podemos dividir los componentes en átomos, moléculas, organismos, plantillas y páginas, logrando crear y diseñar componentes más reutilizables y un código más legible para el desarrollo. Esto se realiza de la siguiente manera:

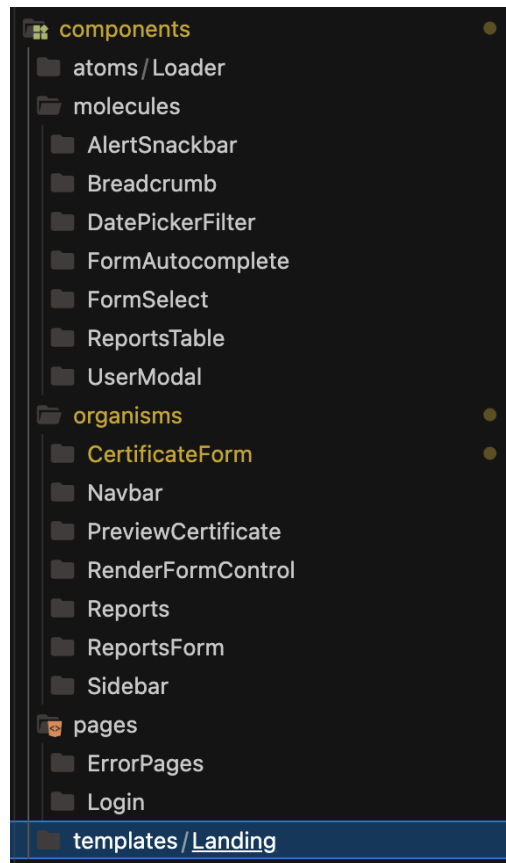


Figura 32

Para mantener la correcta funcionalidad del aplicativo es necesario realizar pruebas al código y a su funcionamiento. Por esto se realizaron las respectivas pruebas unitarias a todos los componentes con una cobertura igual o superior al 80%, un porcentaje recomendado para estas pruebas en específico.

File	% Stmts	% Branch	% Funcs	% Lines
All files	87.12	42.36	58.82	88.23
atoms/Loader	100	100	100	100
index.tsx	100	100	100	100
style.ts	100	100	100	100
molecules/AlertSnackbar	100	100	100	100
helper.ts	100	100	100	100
index.tsx	100	100	100	100
molecules/Breadcumb	81.25	77.77	40	83.87
index.tsx	80	77.77	40	82.75
styles.ts	100	100	100	100
molecules/DatePickerFilter	81.81	100	33.33	90
index.tsx	80	100	33.33	88.88
styles.ts	100	100	100	100
molecules/FormAutocomplete	88.23	53.84	100	88.23
index.tsx	86.66	53.84	100	86.66
styles.ts	100	100	100	100
molecules/FormSelect	81.25	40	75	86.66
index.tsx	78.57	40	75	84.61
styles.ts	100	100	100	100
molecules/UserModal	97.36	50	75	97.36
index.tsx	97.22	50	75	97.22
styles.ts	100	100	100	100
organisms/CertificateForm	81.35	11.11	25	81.35
helper.ts	100	100	100	100
index.tsx	80	11.11	25	80
styles.ts	100	100	100	100
organisms/Navbar	100	75	100	100
index.tsx	100	75	100	100
styles.ts	100	100	100	100
organisms/PreviewCertificate	81.81	36.36	66.66	84.37
index.tsx	80.64	36.36	66.66	83.33
styles.ts	100	100	100	100
organisms/RenderFormControl	80	18.18	40	80
fieldTypes.ts	100	100	100	100
index.tsx	76.19	18.18	40	76.19
styles.ts	100	100	100	100
organisms/Reports	86.76	47.82	62.5	86.76
helper.ts	100	100	100	100
index.tsx	85.93	47.82	62.5	85.93
styles.ts	100	100	100	100
organisms/Sidebar	95.12	40	83.33	95
helper.ts	100	100	100	100
index.tsx	94.28	40	83.33	94.11
styles.ts	100	100	100	100
pages/ErrorPages	92.85	100	50	92.85
index.tsx	91.66	100	50	91.66
styles.ts	100	100	100	100

Figura 33

El código está documentado en Confluence para facilitar la colaboración y futuras actualizaciones. También se documenta la descomposición de los componentes para implementar Atomic Design, y además de esto la configuración del ambiente de desarrollo para hacerlo más legible para quienes tomen de nuevo el proyecto.

Despliegue en Vercel y AWS Docker:

La aplicación se desplegó en Vercel para entornos de desarrollo y pruebas. Siendo un desafío aprender a usar la herramienta de vercel para el despliegue de diferentes versiones del aplicativo, pero logrando el objetivo para que el desarrollo pudiera pasar a pruebas.

En producción, se implementó un despliegue en contenedores Docker en AWS por medio de Github Actions, aprovechando la escalabilidad y la fiabilidad de los servicios de la nube.

Lecciones aprendidas y buenas prácticas

La creación del diseño en Figma, alineado con los lineamientos corporativos, no solo representa un logro estético, sino también un desafío educativo y de crecimiento para el desarrollo FrontEnd. Al adoptar un enfoque cuidadoso hacia la coherencia visual y la experiencia del usuario, se evidencia un compromiso con la calidad y la satisfacción del usuario final.

Aprender y aplicar los lineamientos de diseño relacionados con colores, tamaños, tipografía, etc., no solo fue un ejercicio técnico, sino también una oportunidad de crecimiento personal y profesional para el equipo de desarrollo. La familiaridad con estas prácticas de diseño no solo mejora la estética del aplicativo, sino que también aporta a la evolución de los desarrolladores, permitiéndoles comprender el camino del diseño que a menudo escapa a aquellos con un enfoque puramente programático.

La adopción de herramientas como Figma, Vercel, AWS, Jira desde cero añade una capa adicional de complejidad y aprendizaje. La superación de estos desafíos no solo refleja la capacidad del equipo para adaptarse y aprender nuevas habilidades, sino también destaca la dedicación para lograr un desarrollo correcto de acuerdo a los requerimientos solicitados.

La capacidad de colaborar y comprensión entre el equipo fue esencial para lograr un código limpio y reutilizable, creando un proyecto mantenible y entendible, preparado para integrar futuras características.

La integración de componentes específicos, como la navegación dentro del formulario, selector de fechas, el autocompletado, la tabla con paginación, subraya la atención meticulosa dedicada a la eficiencia y la experiencia del usuario. La elección y aplicación de estos elementos no solo mejoran la usabilidad del aplicativo, sino que también reflejan una comprensión profunda de las necesidades del usuario. La carga de las peticiones al backend junto con la inclusión de un spinner para las solicitudes al backend no solo optimizan el rendimiento, sino que también contribuyen a una experiencia de usuario fluida y sin interrupciones.

Conclusiones

La implementación del aplicativo web ha demostrado ser efectiva al simplificar y automatizar el proceso de generación de certificados laborales. La tecnología ha contribuido de manera destacada a la eficiencia y agilidad en la gestión de documentos laborales, lo que sugiere un impacto positivo en la operación diaria de Recursos Humanos.

La adopción de metodologías ágiles, especialmente Scrum, ha sido clave para el éxito del proyecto. La flexibilidad y adaptabilidad de estas metodologías no solo permitieron una respuesta ágil a los cambios, sino que también enfatizaron la importancia del trabajo en equipo y la colaboración continua, aspectos fundamentales para el desarrollo efectivo de software.

La implementación de medidas robustas de autenticación y autorización ha garantizado la seguridad del aplicativo web, subrayando el compromiso con la integridad y confidencialidad de los datos sensibles. Este enfoque en la seguridad es esencial, especialmente en aplicativos que manejan información laboral confidencial.

La duración del proyecto a lo largo de un semestre académico completo destaca la persistencia y dedicación del equipo ante desafíos extensos. Superar estos desafíos a lo largo de un período prolongado refleja una determinación excepcional para alcanzar los objetivos establecidos.

La experiencia acumulada durante el proyecto ha proporcionado valiosas lecciones aprendidas, identificando áreas de mejora tanto en procesos como en productos. Este enfoque reflexivo y de mejora continua es esencial para el desarrollo profesional y fortalecimiento de habilidades, tanto técnicas como blandas.

El proyecto ha destacado la importancia de las habilidades blandas, como la colaboración, la comunicación efectiva y la adaptabilidad. Estas habilidades son fundamentales para el éxito en proyectos empresariales y contribuyen al desarrollo integral de un profesional.

Referencias

Torres, P. (2019). Automatización de procesos en el área de Recursos Humanos. *Revista de Ciencias Sociales*, 25(3), 37-46.

Indeed. (23 de Agosto de 2023). Certificado laboral: ¿Qué es y cómo se hace uno? *Orientación Profesional*.

<https://www.indeed.com/orientacion-profesional/cv-cartas-presentacion/certificado-laboral-que-es>

Figma. (2023). Figma Documentation. Recuperado de

<https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma->

Williams, D. E., & Miller, P. A. (2019). Agile Web Development: Accelerating Process Automation. *Journal of Software Engineering and Applications*, 12(6), 307-322.

TypeScript. (s.f.). TypeScript: JavaScript that scales. Recuperado de

<https://www.typescriptlang.org/>

Aggarwal, S. (2018). Modern Web-Development using ReactJS. *International Journal of Recent Research Aspects*.

Frost, B. (2016). *Atomic design*. Pittsburgh: Brad Frost.

Official Redux Documentation. (2023, 20 de diciembre). <https://es.redux.js.org/>

El Emam, K., & Neri, E. (2017). Privacy and security in the era of digital health: What should translational researchers know and do about it? *American Journal of Translational Research*, 9(3), 1107-1119.

JWT.io. (s.f.). Introduction to JSON Web Tokens. Recuperado de <https://jwt.io/introduction>

Trigás Gallego, M. (2012). *Metodología scrum*.

Vercel. (s.f.). Getting Started with Vercel. Recuperado de

<https://vercel.com/docs/getting-started-with-vercel> (21 de diciembre de 2023).

Visual-paradigm. (s.f.). What is Scrum.

<https://www.visual-paradigm.com/features/scrums-process-canvas>. Recuperado el 20 de noviembre de 2023.

Atlassian. (s.f.). Comparing workflows: Gitflow Workflow. Recuperado de

<https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow> (20 de noviembre de 2023).