



**Desarrollo de una aplicación web para la administración de tareas de los empleados de las
compañías con un modelo Paas usando AWS**

Santiago Cárdenas Duque

Informe de práctica para optar al título de Ingeniero de Sistemas

Asesora

Ana Lucía Pérez Patiño, Doctor (PhD) en Ingeniería de sistemas

Universidad de Antioquia
Facultad de Ingeniería
Ingeniería de Sistemas
Medellín, Antioquia, Colombia
2021

Cita	Cárdenas Duque [1]
Referencia Estilo IEEE (2020)	[1] S. Cárdenas Duque, “Desarrollo de una aplicación web para la administración de tareas de los empleados de las compañías con un modelo Paas usando AWS”, Trabajo de grado profesional, Ingeniería de Sistemas, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2021.



Biblioteca Carlos Gaviria Díaz

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

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/Director: Jesús Francisco Vargas Bonilla.

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.

TABLA DE CONTENIDO

RESUMEN	5
ABSTRACT	6
I. INTRODUCCIÓN	7
II. OBJETIVOS	8
III. MARCO TEÓRICO	9
IV. METODOLOGÍA	12
VI. RESULTADOS	13
REFERENCIAS	22

LISTA DE FIGURAS

Fig. 1. Arquitectura de la aplicación Incplan	144
Fig. 2. Zona de disponibilidad.....	14
Fig. 3. Red privada	15
Fig. 4. Servicios de Aws: Cognito, Cloudfront y S3.....	16
Fig. 5. Servicios de AWS: Aurora, Parameter store, Cloudwatch, Glacier, SES.....	17
Fig. 6. Landing page sitio web incplan	18
Fig. 7. Creación de work item	18
Fig. 8. Asignación de work item	19
Fig. 9. Global search	19
Fig. 10. Pantalla de información de usuario.....	¡Error! Marcador no definido.0
Fig. 11. Pantalla de compra de servicios	¡Error! Marcador no definido.0

RESUMEN

Este proyecto se enfocó en el desarrollo de una aplicación web con el fin de actualizar un sistema de administración de tareas para los empleados de las empresas, que abarcara tantas funcionalidades requeridas como fuera posible para dar cumplimiento a las tareas necesarias para cada uno de sus empleados, asimismo como la administración de la plataforma con sus diferentes roles. De esta manera se planteó la tecnología a implementar de forma que usáramos un framework para aplicaciones web modernas, con los últimos estándares de seguridad y con una plataforma como AWS que nos proporcionara servicios para conseguir los recursos necesarios con mayor flexibilidad, agilidad y disponibilidad.

ABSTRACT

This project focused on the development of a web application in order to update a task management system for company employees, which will cover as many functionalities required as possible to fulfill the necessary tasks for each of its employees, as well as the administration of the platform with its different roles. In this way, the technology to be implemented was proposed so that we would use a framework for modern web applications, with the latest security standards and with a platform such as AWS that provides us with services to obtain the necessary resources with greater flexibility, agility and availability.

I. INTRODUCCIÓN

El desarrollo de aplicaciones web tiene gran demanda debido a sus múltiples usos y su fácil acceso desde cualquier lugar del mundo y la flexibilidad para la creación de funcionalidades que permiten al usuario ejecutar.

Por consiguiente, el objetivo del desarrollo de esta aplicación es crear, reutilizar y asignar tareas a cada empleado con su respectiva compañía con el fin de tener una distribución equitativa y organizada de las actividades que cada uno debe realizar. Además, tener registro de cada actividad con el fin de poder realizar auditoría general a las actividades con respecto a sus empleados y generar recordatorios también para el desarrollo de las mismas.

El trabajo se abordó teniendo en cuenta los distintos roles de los que se hacen uso, separando así las funcionalidades de cada actor participante y asegurando el cumplimiento de sus necesidades dentro de la aplicación. De esta forma se cuenta con empleados; los cuales son los usuarios que harán uso de las tareas ya creadas y podrán tomar cada una de ellas para su realización y crear recordatorios para la misma. Luego están los administradores, que serán las personas encargadas de crear, editar o eliminar tareas y usuarios. Por último el manager, el cual será el encargado de la compañía y de sus empleados.

II. OBJETIVOS

A. *Objetivo general*

Construir una aplicación web que provee servicios para la administración de tareas de los empleados en las compañías basada en 4 roles: administrador, manager, empleado y cliente

B. *Objetivos específicos*

- Obtener los requerimientos necesarios para el desarrollo de la aplicación con el cliente por medio de reuniones.
- Analizar los requerimientos del sistema con el fin de obtener sus limitaciones, problemas y cambios existentes.
- Diseñar el software basado en los requisitos y análisis del sistema para realizar el diseño lógico y físico de la aplicación, tales como: estructura de la base de datos, del backend, frontend y diagramas de flujo.
- Desarrollar la implementación del diseño de software tanto del backend como del frontend con el lenguaje typescript y Java.
- Realizar pruebas unitarias tanto en Back como en Front por cada
- desarrollo implementado.
- Integrar componentes de la aplicación web con medios proporcionados tanto del backend como del frontend por medio de pipelines para ambiente de qa y desarrollo.
- Realizar el mantenimiento del software por medio de cambios requeridos por el cliente y solución de problemas.

III. MARCO TEÓRICO

Las aplicaciones web son un software en un servidor al cual los usuarios pueden acceder por medio de un dispositivo móvil a través del internet o de una intranet. Su facilidad de uso las ha hecho populares ya que no son independientes del sistema operativo y no requieren instalación.

Muchos de los servicios que usan las aplicaciones web están realizando sus implementaciones en la nube ya que el costo del desarrollo ha disminuido considerablemente y su rapidez, fiabilidad y confiabilidad ha incrementado brindando una mayor confianza a los usuarios.

1.1 Pila tecnológica

La pila tecnológica es un ecosistema de datos que incluye las herramientas, los marcos y las bibliotecas subyacentes que se utilizan para crear y ejecutar una aplicación.

1.2 Herramientas de la pila tecnológica

- Frontend: Angular versión 9. Framework con lenguaje de programación Typescript.
- Backend: Java versión 14, Amazon Amplify con Cognito y S3.
- Base de datos: Aurora.

2.1 Servicios en la nube

2.1.1 Definición

El sitio de Red Hat propone la siguiente definición de servicios en la nube:

Los servicios de nube son infraestructuras, plataformas o sistemas de software que los proveedores externos alojan y que se ponen a disposición de los usuarios a través de Internet. Estos facilitan el flujo de datos de los usuarios a través de Internet, desde los clientes de front-end hasta los sistemas de los proveedores, y viceversa [6].

Existen diferentes servicios en la nube para casi cualquier recurso que requiere el desarrollo de una aplicación web, desde los servidores, bases de datos, redes, y almacenamiento de archivos, hasta interfaces gráficas lo cual supone una gran ventaja tanto en el momento de su creación como en el de su ejecución.

2.1.2 Ventajas de servicios en la nube

Las ventajas de tener servicios en la nube son varias, entre ellas tenemos las siguientes:

- Puede pagar solo cuando consume recursos informáticos y solo paga por la cantidad que consume en lugar de tener que invertir mucho en centros de datos y servidores.
- Los proveedores como AWS pueden lograr mayores economías de escala, lo que se traduce en menores precios de pago por uso.
- Se puede acceder a tanta o poca capacidad como necesite, y escalar hacia arriba y hacia abajo según sea necesario con solo unos minutos de aviso.

- Los nuevos recursos de TI están a solo un clic de distancia, lo que significa que reduce el tiempo para poner esos recursos a disposición de sus desarrolladores de semanas a solo minutos.
- Permite a los usuarios concentrarse en sus propios clientes, en lugar de en el trabajo pesado de almacenar, apilar y alimentar servidores.
- Se puede proporcionar una latencia más baja y una mejor experiencia para sus clientes a un costo mínimo.

Estas ventajas nos dan una idea de la agilidad que pueden tener los usuarios al tener servicios en la nube, el tiempo y costo que pueden llegar a tener ya que todos estos recursos están a la mano y su implementación no tiene mucho esfuerzo. Además de la seguridad que brinda ya que al tener varios centros de cómputo en diferentes regiones del planeta la experiencia para los clientes es mucho mejor por su confiabilidad, eficacia y seguridad.

2.1.3 Servicios en la nube requeridos para el desarrollo de la aplicación

Para el desarrollo de esta aplicación se hace uso de algunos recursos tales como: Amazon Cognito, Aurora, Lambdas, SES, Load Balancer, Amazon S3, Fargate, Application Load Balancer, CloudFront, VPC, CloudWatch, Glacier, Parameter store, Nat gateway.

- Amazon Cognito: Es un servicio para el manejo de sesiones y roles de usuario para las aplicaciones. Permite gestionar accesos mediante la creación de grupos. Además, los usuarios pueden iniciar sesión mediante proveedores de identidades de redes sociales, como Google, Facebook y Amazon.
- Aurora: Una base de datos relacional compatible con MySQL y PostgreSQL creada para la nube. Combina el rendimiento y la disponibilidad de las bases de datos empresariales tradicionales con la simplicidad y la rentabilidad de las bases de datos de código abierto.
- AWS Lambda: Es un servicio informático sin servidor que permite ejecutar código sin aprovisionar ni administrar servidores, crear una lógica de escalado de clústeres basada en la carga de trabajo, mantener integraciones de eventos o administrar tiempos de ejecución. En nuestro caso usamos el lenguaje de programación Nodejs para ejecutar un Lambda PostConfirmation el cual el asigna el rol al usuario una vez se ha registrado perteneciendo este a un grupo.
- SES: Es un servicio de correo electrónico rentable, flexible y escalable que permite enviar correos desde cualquier aplicación.
- Elastic Load Balancer: Distribuye automáticamente el tráfico de aplicaciones entrante a través de varios destinos, tales como las instancias de Amazon EC2, los contenedores, las direcciones IP, las funciones Lambda y los dispositivos virtuales.
- S3: es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento.
- Fargate: Es un motor informático sin servidor que funciona tanto con Amazon Elastic Container Service (ECS) como con Amazon Elastic Kubernetes Service (EKS). Fargate elimina la necesidad de aprovisionar y administrar servidores, le permite especificar y pagar recursos por aplicación y mejora la seguridad mediante el aislamiento de aplicaciones por diseño.

- Cloudfront: Es un servicio rápido de red de entrega de contenido (CDN) que distribuye a clientes datos, videos, aplicaciones y API globalmente de forma segura, con baja latencia, altas velocidades de transferencia y dentro de un entorno fácil para desarrolladores.
- Glacier: Amazon S3 Glacier y S3 Glacier Deep Archive son clases de almacenamiento en la nube en Amazon S3 seguro, duradero y de muy bajo costo para archivar datos y realizar copias de seguridad a largo plazo. Están diseñadas para ofrecer una durabilidad del 99,999999999 % y ofrecen capacidades de conformidad y seguridad integrales que pueden ayudar a cumplir los requisitos normativos más estrictos.
- Parameter store: Proporciona almacenamiento jerárquico seguro para la gestión de datos de configuración y la gestión de secretos. Puede almacenar datos como contraseñas, cadenas de bases de datos, ID de imagen de máquina de Amazon (AMI) y códigos de licencia como valores de parámetro. Puede almacenar valores como texto sin formato o datos cifrados.

2.1.3 Proveedores de servicios en la nube

Los proveedores de nube son empresas que ofrecen entornos de TI los cuales pueden prestar servicios en línea que incluyen la infraestructura como servicio, las plataformas como servicio y el software como servicio. Algunos de ellos son:

- Microsoft Azure
- Amazon
- Google Cloud
- IBM

IV. METODOLOGÍA

- Metodología de trabajo: Se trabajó mediante el marco de desarrollo ágil llamado Scrum; basado en sus ceremonias y principios ágiles, se realizaron reuniones al principio del sprint (tiempo estimado de entrega de funcionalidades) para definir el alcance de lo que se logrará en cuanto a entregas funcionales, reuniones diarias para discutir sobre el día a día de los avances en las tareas, reuniones de revisión para realizar entregas de valor e incrementos en la aplicación y reuniones de refinamiento para discutir sobre las tareas que se trabajarán en un futuro.
- Metodología de desarrollo: Con el fin de lograr los objetivos planteados se realizaron las siguientes actividades:

Etapa de fundamentación:

- Consultar qué es Angular
- Explorar qué son y cómo funcionan los estados en Angular [2]
- Consultar las diferentes formas de manejar los estados en la aplicación[2]
- Consultar administración de grupos con Cognito de AWS
- Explorar las pruebas unitarias con Angular y con Java.

Etapa de construcción:

- Iniciar con la construcción del aplicativo web mediante el cli de Angular.
- Diseñar y construir el frontend de la aplicación
- Crear componentes de la aplicación
- Configurar componentes de la aplicación
- Consumir los microservicios necesarios desde el frontend
- Configurar los accesos de la aplicación
- Definir el ciclo de vida de la aplicación (estados y pasos)
- Integrar datos en las vistas y la forma como se visualizan
- Consumir API de AWS
- Diseñar experiencia de usuario
- Desarrollar modelos requeridos por el backend
- Consumir microservicios en aplicación web desde cada vista
- Integrar datos con la aplicación en cada vista

Etapa de validación:

- Validar funcionalidades de la aplicación web desde los distintos escenarios de uso.
- Validar compatibilidad de la aplicación con plataformas y distintas versiones móviles.
- Realizar pruebas unitarias
- Validar con el cliente los requerimientos demandados

VI. RESULTADOS

Se logra como objetivo principal la construcción y el despliegue de una aplicación completamente funcional, con el fin de lograr una mejor gestión de las actividades requeridas para los empleados de las diferentes empresas que adquieren el servicio, así como la distribución de datos de interés para ser suministrada a los clientes de la aplicación y/o de la empresa en cuestión.

En la aplicación se pueden encontrar diferentes funcionalidades dependiendo del rol que tenga el usuario. Para el rol de administrador se pueden crear plantillas que pueden servir para un uso futuro las cuales estarán compuestas por varias tareas que se le asignarán a un empleado para que las desarrolle. Además, podrá gestionar los usuarios y ver los indicadores de rendimiento, así también como cambiar el rol del usuario, crear usuarios, crearle empresas a los clientes, cambiar datos de las empresas de los clientes y asignar tareas a los empleados, visualizar las plantillas, los servicios agregados por cada empresa y modificarlos, visualizar los recordatorios, las tareas asignadas a cada usuario así también como las tareas realizadas, entre otros.

Por el lado del cliente, pueden modificar sus datos personales, los datos de su empresa a excepción del nombre, recibir notificaciones que el empleado le envíe de acuerdo a su tarea realizada, visualizar las plantillas, las tareas asignadas y las que ya se han completado.

Por el lado del manager podrá ser notificado por medio de un correo electrónico en caso de que ocurra algún problema con el desarrollo de una tarea.

En cuanto a la búsqueda de la información que tiene la aplicación se dividieron en tabs los cuales tenían su propio filtro dinámico y reactivo que cambia de acuerdo a los atributos del elemento a buscar. Además, estas búsquedas se realizaron paginables de manera que el cliente pudiera tener una mejor experiencia de usuario y facilidad de visualización.

Por otro lado, la aplicación se compuso de tres ambientes: desarrollo, qa y producción, dentro de los cuales se integraron los recursos por medio de terraform con AWS, parameter store para el manejo de credenciales y liquid base para la actualización de la base de datos, todo esto con el fin de integrar los cambios dentro de los diferentes ambientes y no tener fallas por falta de recursos que actualizaron.

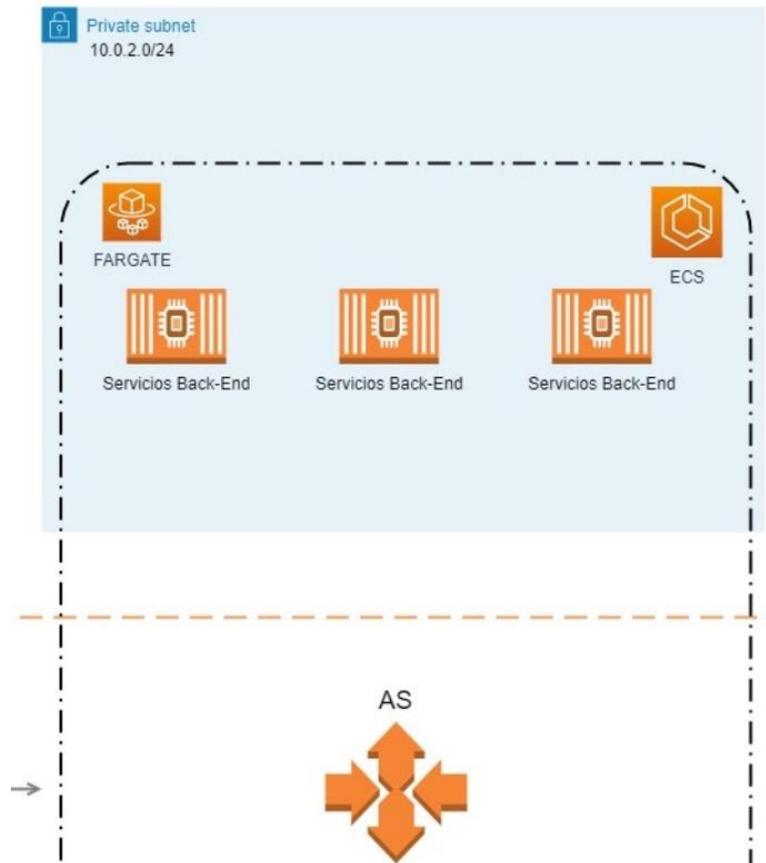


Fig. 3. Red privada

Nota: La red privada agrupa los contenedores los cuales son administrados por el servicio de Auto Scaling de AWS, el cual se encarga de proporcionar escalabilidad horizontal a los contenedores según el tráfico.

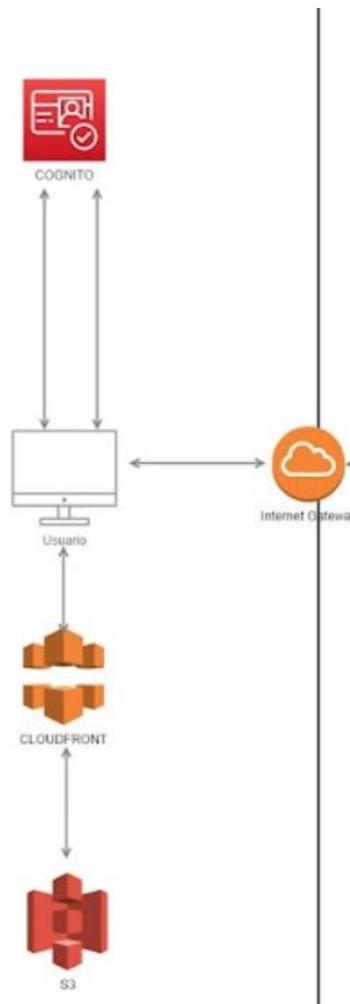


Fig. 4. Servicios de Aws: Cognito, Cloudfront y S3.

Nota: Integración del frontend el cual está montado en S3 y tiene una conexión directa con Cognito el cual se encarga de verificar la identidad de los usuarios para el acceso a la aplicación. Además, el servicio de Cloudfront se encarga de entregar el contenido, haciendo de proxy y almacenando en caché datos web en las ubicaciones más cercana a los usuarios.



Fig. 5. Servicios de AWS: Aurora, Parameter store, Cloudwatch, Glacier, SES.

Nota: Servicio de base de datos (Aurora), el servicio de Cloudwatch el cual nos permite ver los logs de los servicios que usamos como: Lambda, EC2, cargas de archivos, entre otros. También tenemos el parameter store, el cual se encarga de almacenar las variables como por ejemplo de los nombres de las lambdas, nombres de los roles, credenciales del servicio de correo electrónico, entre otros.

Imágenes de la aplicación desarrollada

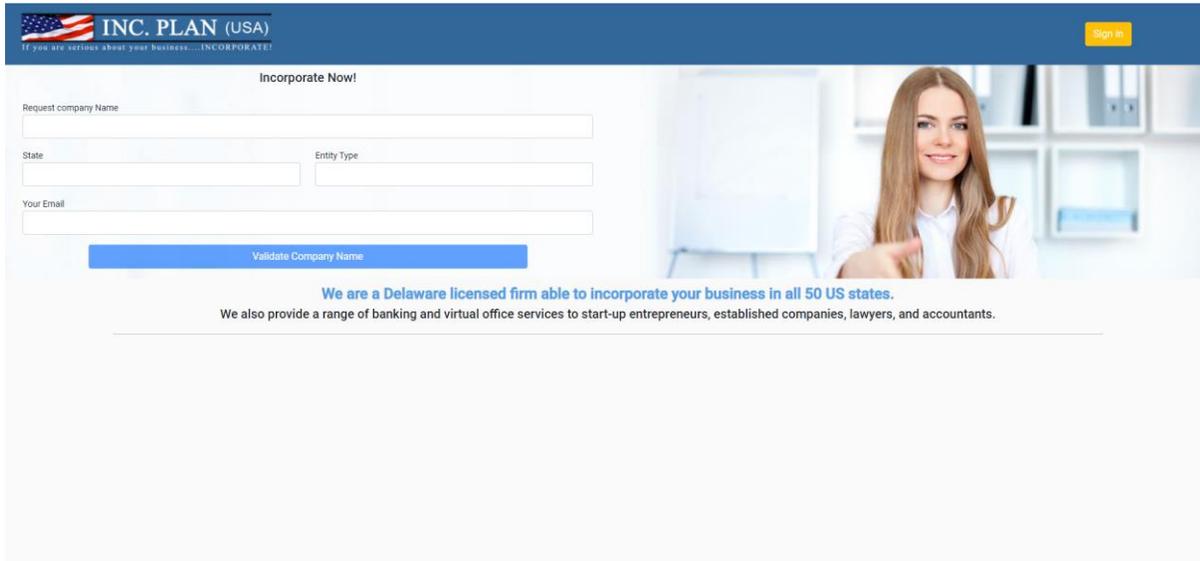


Fig. 6. Landing page

Nota: En esta pantalla podemos crear un work item el cual está seguido de dos pasos más: una información general sobre la tarea donde se selecciona el paso a crear (Step) y la otra pantalla que tendrá un contenido diferente de acuerdo al paso seleccionado.

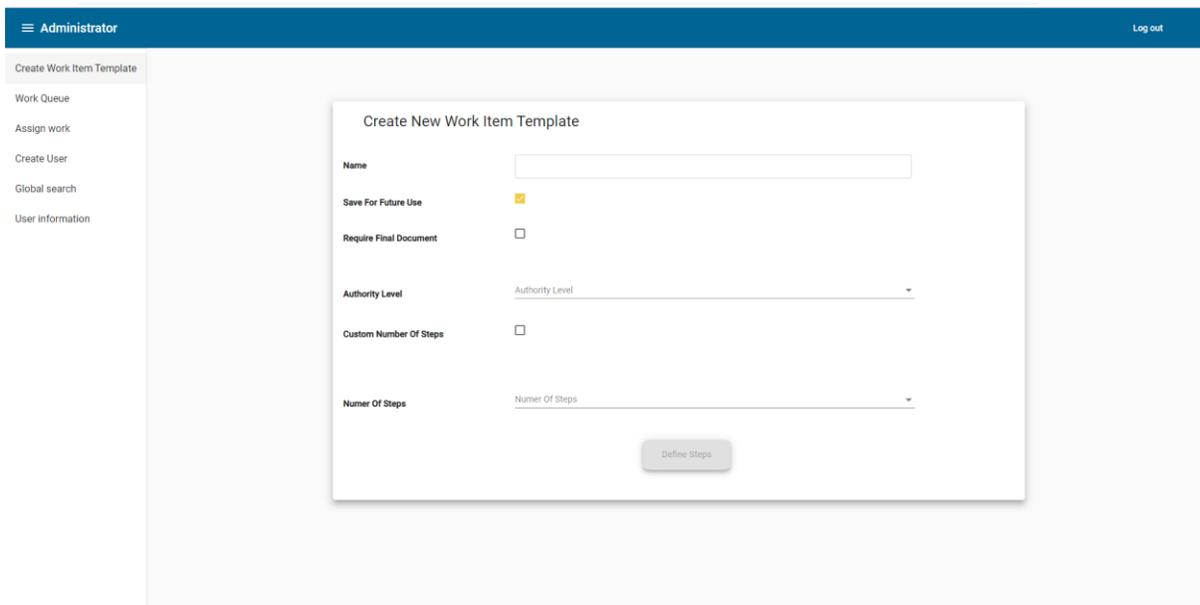


Fig. 7. Creación de work item

Nota: En esta pantalla podemos crear un work item el cual está seguido de dos pasos más: una información general sobre la tarea donde se selecciona el paso a crear (Step) y la otra pantalla que tendrá un contenido diferente de acuerdo al paso seleccionado.

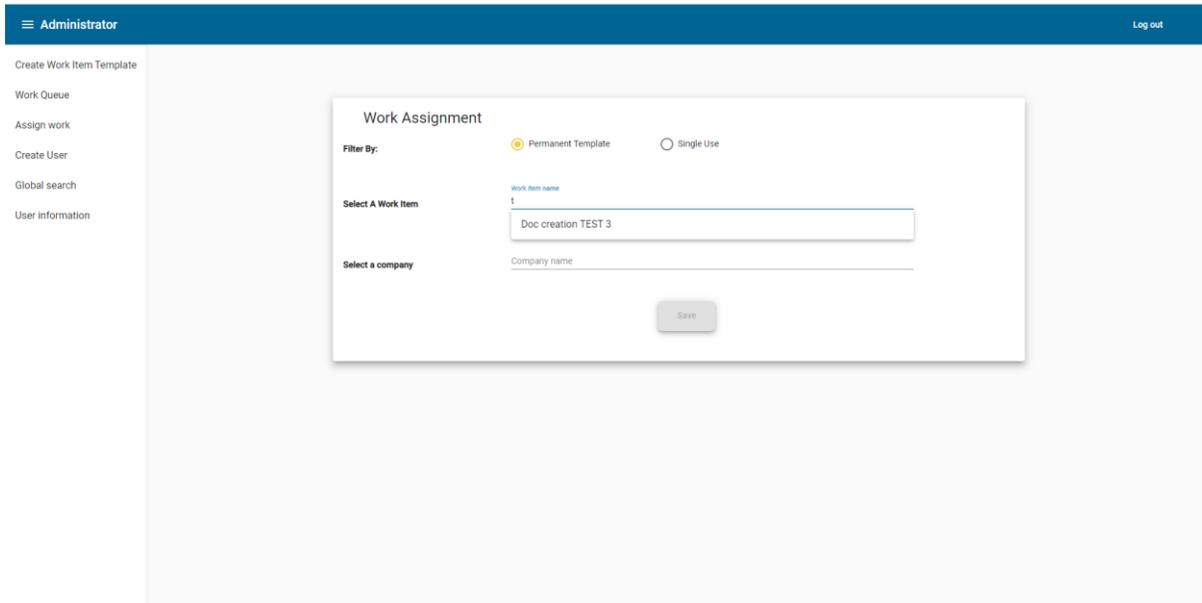


Fig. 8. Asignación de work item

Nota: En esta pantalla se puede seleccionar el work item creado para asignárselo a una empresa.

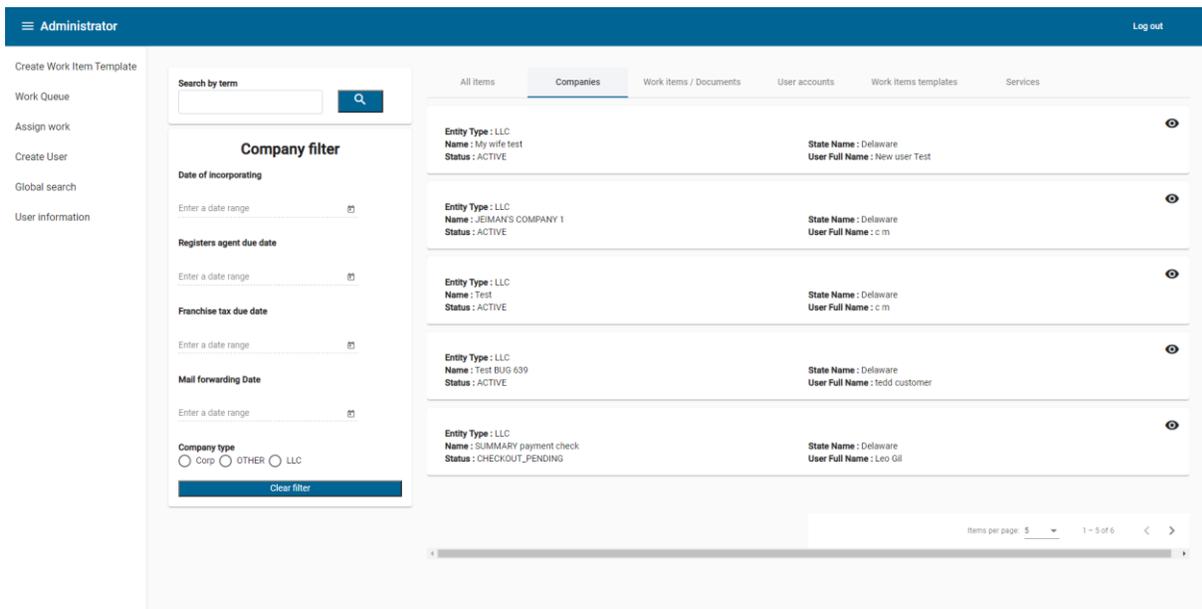


Fig. 9. Global search

Nota: En esta parte se pueden apreciar varias tabs las cuales tendrán un filtro diferente de acuerdo a la información que contiene cada tab, este caso se seleccionó compañía la cual tiene filtros como: fecha de incorporación, fecha de vencimiento de los impuestos y el tipo de compañía.

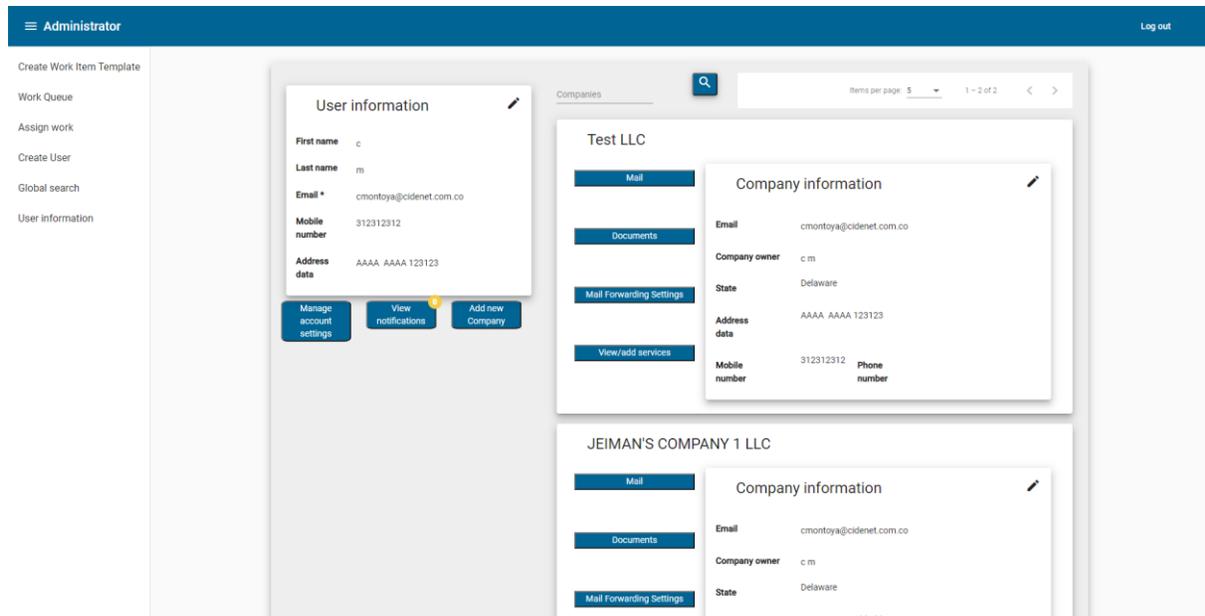


Fig. 10. Pantalla de información de usuario

Nota: En la pantalla del usuario se puede visualizar la información del usuario, la opción de editar la misma. cambiar su contraseña, agregar compañías (para usuarios con rol customer). Además, puede visualizar las compañías relacionadas con ese usuario, así también como la opción de ver los servicios, agregar o eliminar, los documentos relacionados con esa compañía los cuales se consultan en el servicios de Aws de S3.

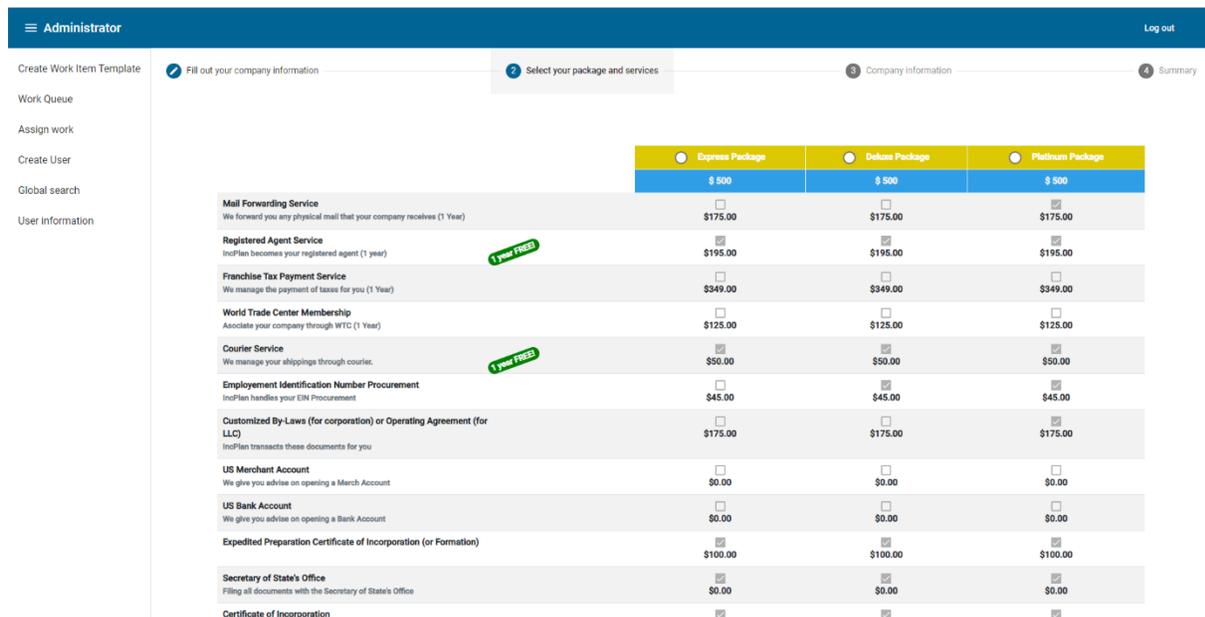


Fig. 11. Pantalla de compra de servicios

Nota: El usuario tiene la posibilidad de seleccionar un paquete y de agregar servicios al mismo. Está compuesto por otros pasos más que son: información del usuario (en caso que no esté completa), información de la compañía y un resumen de la compra con un modal para el ingreso de la información del pago.

XI. CONCLUSIONES

Se trabajó realizando las investigaciones pertinentes para dar cumplimiento con una aplicación con altos estándares de calidad y actualización de servicios con las últimas tecnologías de servicios ofrecidos por la nube.

Dado que el cliente agregó algunas funcionalidades no contempladas al inicio del proyecto no fue posible culminarlo en su totalidad. Sin embargo, se realizaron la mayoría de los requerimientos dentro de los que caben destacar: funcionalidad de diferentes roles, tales como: employee, customer, administrator y manager, seguridad para el acceso a la aplicación, montaje de la aplicación y sus servicios en ambiente de desarrollo y qa, pruebas unitarias del front, integración continua.

Así, teniendo como base las diferentes etapas del desarrollo de software se puede afirmar que se pasó por cada una de ellas durante el desarrollo de la aplicación ya que eran necesarias para tener una dirección y un flujo para el cumplimiento exitoso de cada una de las tareas.

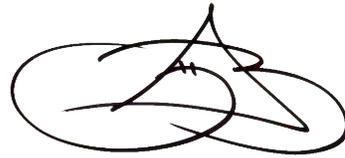
REFERENCIAS

- [1] A. Sargsyan 2020 Marzo 4. “RxJS in Angular:Easy to understand” [En línea]. Disponible en: https://medium.com/@Areg_Sargsyan
- [2] Bhargav Bachina, 2020, Octubre 6. “How To Implement NGRX Store In Angular Apps” [En línea]. Disponible en: <https://medium.com/bb-tutorials-and-thoughts>
- [3] Miro Koczka. 2017, Agosto 21, “3 ways to communicate between Angular components”, [En línea]. Disponible en: <https://medium.com/@mirokoczka>.
- [4] Nixon Augustine. 2020, Junio 26, “Angular Component Interaction Parent-Child Communication” [En línea]. Disponible en: <https://medium.com/@nixonaugustine5>
- [5] “Set up Full stack project” [En línea]. Disponible en: <https://docs.amplify.aws>
- [6] “¿Qué son los servicios en la nube?”, [En línea]. Disponible en: <https://www.redhat.com>

Visto bueno del asesor interno y asesor externo

Como asesor conozco la propuesta y avalo el contenido de la misma.

Arley Fernando Gallego Beltrán
Nombre del asesor interno

A handwritten signature in black ink, consisting of several overlapping loops and a central vertical stroke.

Firma del asesor interno

Ana Lucía Pérez Patiño
Nombre del asesor externo

A handwritten signature in black ink, appearing as a stylized, cursive script.

Firma del asesor externo