



**UNIVERSIDAD
DE ANTIOQUIA**

**MIGRACIÓN DE INTERFAZ DE USUARIO PARA EL
SOFTWARE QANTY**

Wilder Andrés Arias Murillo

Universidad de Antioquia

Facultad de Ingeniería, Departamento de ingeniería electrónica.

Medellín, Colombia

2020



Migración de interfaz de usuario para el software Qanty

Autor.

Wilder Andrés Arias Murillo

Informe de práctica como requisito para optar al título de:
Ingeniero electrónico

Asesores.

Sebastián Isaza Ramírez
Universidad de Antioquia

Andrés Mauricio Eusse
Dinámica y Desarrollo

Universidad de Antioquia
Facultad de Ingeniería, Departamento de Ingeniería Electrónica.
Medellín, Colombia
2020

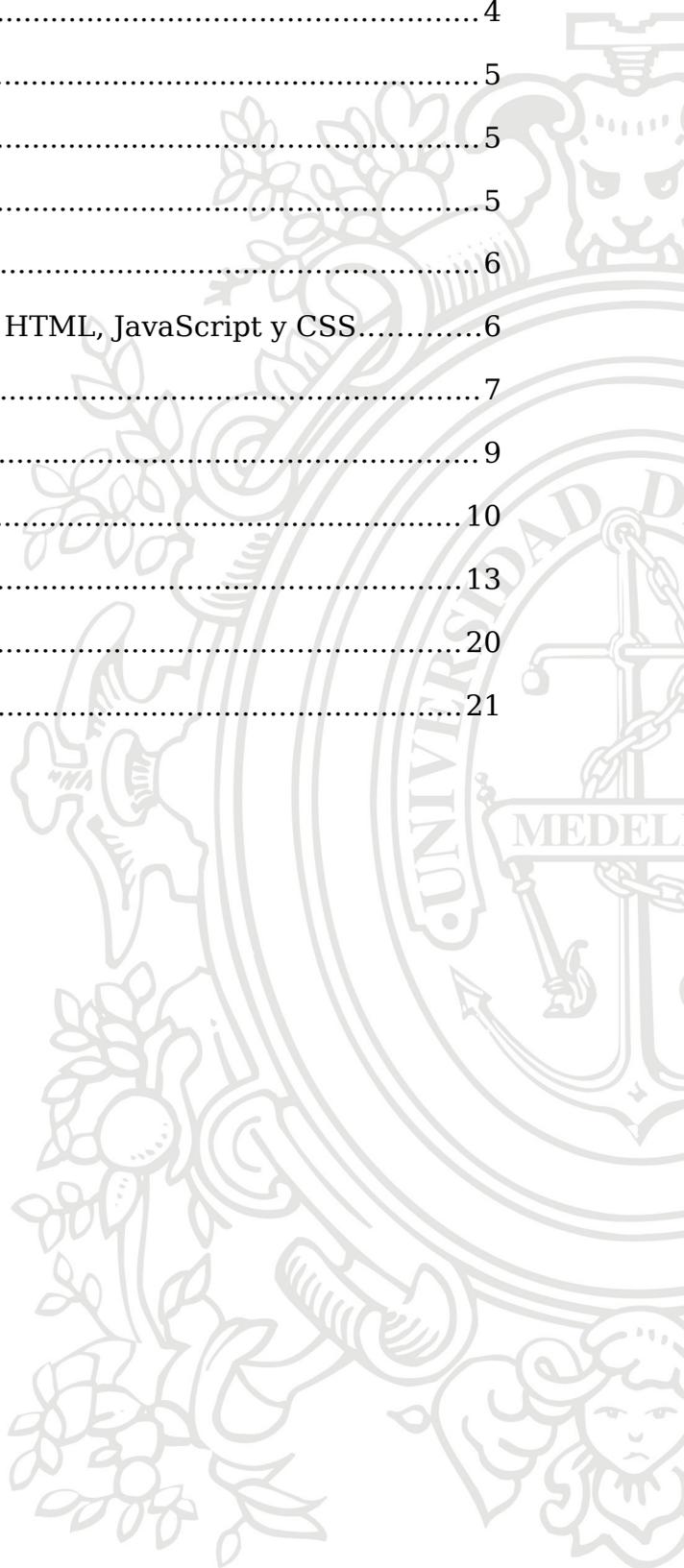
Resumen

En algún momento de nuestras vidas hemos tenido que pasar tiempo esperando para ser atendidos y recibir la prestación de un servicio, tal vez en la fila de un restaurante, en un banco, al ir a comprar, u otros lugares frecuentes del día a día; y tener la sensación de que estamos desperdiciando tiempo valioso de nuestras vidas. Normalmente como clientes no queremos esperar mucho tiempo para poder ser atendidos, y queremos tener la sensación de que la línea de espera se encuentra en movimiento, esto lleva a las organizaciones a buscar estrategias para mejorar la atención de sus usuarios, y a empresas como Dinámica y Desarrollo a brindar soluciones para el manejo de filas de espera.

Al final de los seis meses de la duración de la práctica empresarial se logró entregar un software con el 90% desarrollado, funcional y probado, manteniendo la funcionalidad original de Qanty y añadiendo algunas nuevas funcionalidades como generación y lectura de códigos QR, cambios en la paleta de colores de la interfaz y ubicación de algunos elementos; la migración de Qanty se realizó empleando el framework Vue.js como tecnología central. El código fuente se entregó de tal manera que cumpliera con los requerimientos exigidos al inicio del proyecto por parte de la empresa Dinámica y Desarrollo en cuestiones de comentarios, lineamientos en la estructura que se debía llevar en los scripts, manejo de errores en el software ya existente y de aquellos que surgieron en la migración del software Qanty.

Tabla de contenido

I. Introducción.....	4
II. Objetivos.....	5
Objetivo general.....	5
Objetivos específicos.....	5
III. Marco Teórico.....	6
Los tres pilares de la programación web: HTML, JavaScript y CSS.....	6
Vue.js y sus bibliotecas centrales.....	7
Back-end.....	9
IV. Metodología.....	10
V. Resultados y análisis.....	13
VI. Conclusiones.....	20
VII. Referencias Bibliográficas.....	21



I. Introducción

La calidad en las empresas se mide a partir de una variedad de factores que abarcan desde las instalaciones físicas de la compañía, los métodos de trabajos empleados, los recursos humanos, y hasta la atención a los usuarios; este último es el que jugó un papel importante en el desarrollo del proyecto empresarial. Es común que los clientes o usuarios cuando van a recibir un servicio no quieren esperar mucho tiempo para ser atendidos, o tener la sensación de estar estancados en una fila, esto lleva a las empresas a buscar estrategias para mejorar la experiencia de espera de sus clientes.

Las líneas de espera se forman debido a un desequilibrio en un momento dado entre la demanda de un servicio y la capacidad del sistema para brindar el servicio. Entendiendo esta problemática, surge un estudio para el manejo de líneas de espera conocido como la teoría de colas, esta teoría estudia los distintos modelos para la gestión de líneas de espera, buscando un nivel óptimo entre el número de los clientes y la capacidad de trabajo del sistema o la empresa, también el tiempo de espera en la cola, como el tiempo de atención al brindar un servicio.

Uno de los mayores desafíos al encontrar un problema, es seleccionar la mejor solución para este. En los últimos años las compañías se han inclinado por soluciones tecnológicas gracias al crecimiento tecnológico que se ha evidenciado en las últimas dos décadas en materia de conectividad, infraestructura informática, teléfonos inteligentes, aplicaciones móviles y webs; además la crisis surgida a partir del coronavirus(COVID-19) no ha sido solamente una amenaza para la salud pública, si no también las perturbaciones a nivel económico y social que ponen en peligro el bienestar de millones de personas, los mercados laborales y a las empresas, que han llegado a la interrupción generalizada del comercio. Estos avances y crisis que han sucedido han llevado a las empresas a re-adaptarse y ha pensar en soluciones tecnológicas para poder continuar ofreciendo sus servicios y productos.

Dinámica y Desarrollo es una empresa formada por un grupo de personas que notaron la necesidad de ofrecer servicios relacionados con los sistemas de atención y líneas de espera en la región antioqueña, y que han ido llevando a otras regiones; uno de sus servicios es el software Qanty, un sistema para el control de filas, que permite manejar las líneas de espera de los clientes con el objetivo de ahorrarles tiempo, además los administradores o dueños de las empresas pueden monitorear en tiempo real el estado de las filas, ver los tiempos de atención, y con esta información tomar decisiones.

El software Qanty es una solución tecnológica que someramente consta de una página web y una aplicación móvil disponible en la tienda Google Play y App Store donde los usuarios pueden consultar las empresas y solicitar turnos de atención. El problema que se abordó durante la práctica empresarial fue la actualización de la aplicación web del software Qanty hacia tecnologías más recientes que permitieran mejorar la experiencia de los usuarios en el sistema, dar soluciones a fallas existentes en el sistema, y en el camino del desarrollo del proyecto surgió la necesidad de atención de usuarios en línea, en parte por la crisis del coronavirus. Esto se logró empleando tecnologías en el desarrollo web como HTML, CSS, JavaScript, bases de datos y la de mayor impacto en este proyecto el framework Vue.js.

II. Objetivos

Objetivo general

Actualizar el front-end del software web Qanty de la empresa Dinámica y Desarrollo manteniendo su funcionalidad actual, y la adición de nuevas funcionalidades mediante la incorporación de frameworks recientes y una nueva estructura de navegación que mejore y facilite la forma cómo los usuarios ven e interactúan con la página web Qanty.

Objetivos específicos

- 1 Rediseñar la interfaz gráfica del software web Qanty empleando algunas técnicas de experiencia de usuario, responsive design y material design para lograr adaptabilidad en diferentes pantallas.
- 2 Depurar el software Qanty de los fallos detectados en la versión actual por medio de pruebas manuales realizadas durante la actualización del software y los reportes de errores en la base de datos para mejorar la estabilidad y rendimiento del software.
- 3 Adicionar nuevas funcionalidades al software como el lector de código QR, atención de usuarios on-line para brindar nuevas líneas de atención y añadir valor agregado con la nueva versión de Qanty.
- 4 Realizar pruebas del software mediante pruebas manuales, y pruebas automatizadas para detectar fallos del sistema y garantizar el correcto funcionamiento.

III. Marco Teórico

Los tres pilares de la programación web: HTML, JavaScript y CSS

HTML o Lenguaje de marcado de hipertexto: es la pieza más básica para la construcción de una página web, y se usa para definir el sentido y estructura del contenido de una página web. Es un estándar a cargo del consorcio World Wide Web(W3C), esta organización se encarga de estandarizar casi todas las tecnologías ligadas a la web.

HTML usa marcado para anotar textos, imágenes y otro contenido para ser mostrado en un navegador web, el marcado se realiza mediante el uso de elementos. Un elemento es la estructura básica de HTML, y se compone de atributos y contenido; los atributos del elemento están contenidos en la etiqueta de inicio y el contenido está entre la etiqueta de inicio y la etiqueta de cierre, la figura 2 ilustra la estructura de un elemento HTML. Hipertexto se refiere a los enlaces que conectan las páginas web entre sí, ya sea dentro de un mismo sitio web o entre diferentes sitios web.

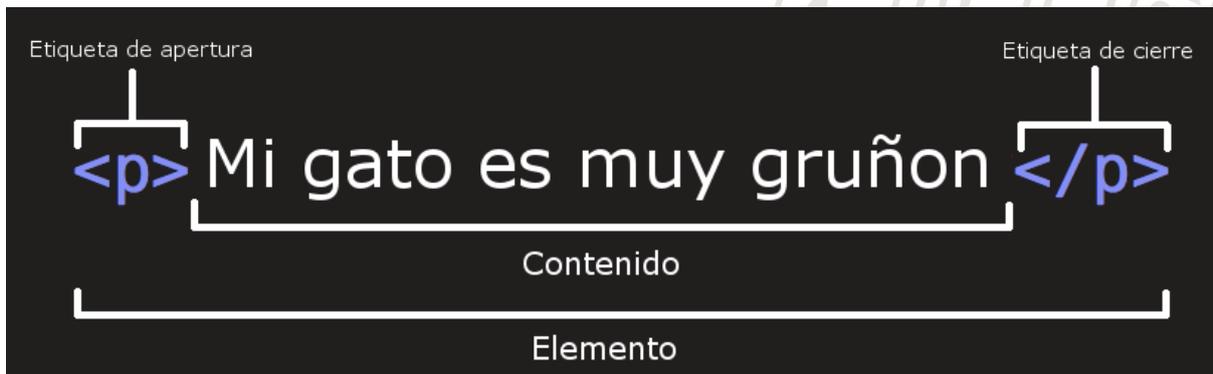


Figura 1. Estructura de un elemento HTML. Recuperado de https://developer.mozilla.org/es/docs/Learn/HTML/Introduccion_a_HTML/iniciar

JavaScript: Es el lenguaje de programación más usado para páginas web, es un lenguaje interpretado o lo que es lo mismo, se compila durante la ejecución de la página web. Se rige bajo un estándar que es ECMAScript, esto se da debido a la guerra tecnológica que se estaba dando a mediados de los 90 entre las compañías que estaban jugando un papel importante en el desarrollo de la web. Actualmente, JavaScript se utiliza principalmente del lado del front-end o cliente, implementado casi en la totalidad de los navegadores permitiendo mejoras en la interfaz de usuario y crear páginas web dinámicas, mediante funciones embebidas en páginas HTML y que interactúan con el Modelo de objetos del documento de la página, DOM, y permiten detectar eventos con la página web

como pulsaciones de teclado y ejecutar tareas localmente en el navegador del usuario[1].

CSS: CSS del inglés Cascading Style Sheets o en español Hojas de Estilo en Cascada es el lenguaje de estilos utilizado para establecer el diseño visual de los elementos una página web mencionados anteriormente, como por ejemplo cambiar la fuente, el color, el tamaño, añadir animaciones e incluso la ubicación del elemento dentro del documento. CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este. La separación del contenido y el formato hace posible presentar el mismo documento en diferentes estilos como por ejemplo para diferentes tamaños de pantalla, esto fue de gran utilidad para lograr el objetivo que Qanty fuera adaptable a diferentes pantallas. Al igual que HTML es un estándar a cargo del consorcio WWW.

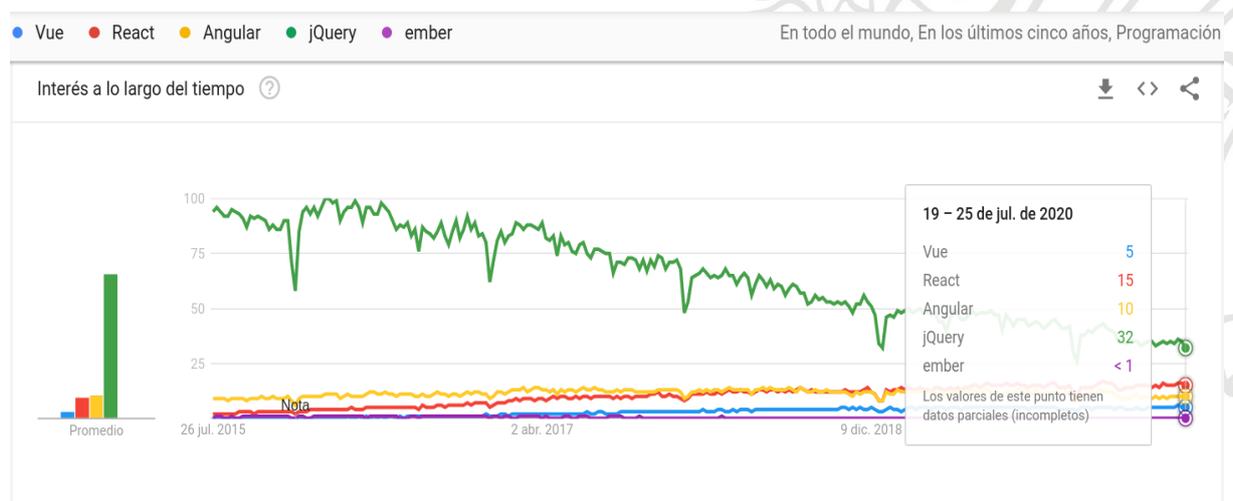
Vue.js y sus bibliotecas centrales

Vue.js: La Tecnología central en el desarrollo del proyecto fue Vue.js, un framework de código abierto de JavaScript, creado por el ingeniero Evan You luego de haber trabajado en Google usando AngularJS. Evan You tuvo la idea de tomar las partes que realmente consideraba importantes de Angular y otros frameworks, y con estas construir una tecnología nueva pero liviana. La primera versión de Vue fue lanzada en el año 2014, y durante el poco tiempo que lleva ha obtenido un crecimiento en su comunidad que ha logrado llevar el framework a estar entre los favoritos a la hora de desarrollar páginas web. La gráfica 1 tomada de la plataforma Google Trends nos muestra el interés sobre los frameworks más usados para desarrollo front-end a lo largo de los últimos cinco años; adicional la gráfica 2 tomada del sitio de preguntas y respuestas para programadores profesionales Stack Overflow, indica el comportamiento del porcentaje de preguntas realizadas sobre los frameworks más usados para el desarrollo front-end desde el año 2009, y se evidencia la curva de crecimiento que esta tomado Vue.

Se eligió Vue sobre frameworks como React o Angular, por tres razones principales: primero, Vue es un framework desarrollado con la idea de poder ser adaptable a proyectos ya desarrollados, lo que hizo muy cómodo su adaptación en el proyecto Qanty; segundo, El equipo de Vue ha tomado e implementado lo mejor que tienen otros frameworks como React y Angular que llevan más tiempo en el mercado y han desechado aquellas cosas que no aportan mayor valor, dejando abierta la puerta de añadir estas funcionalidades a nuestro proyecto mediante software de terceros según nuestras necesidades; tercero, La curva de aprendizaje para Vue es notablemente más asequible comparada con Angular y React, esto es muy importante debido al tiempo que se tenía para el desarrollo

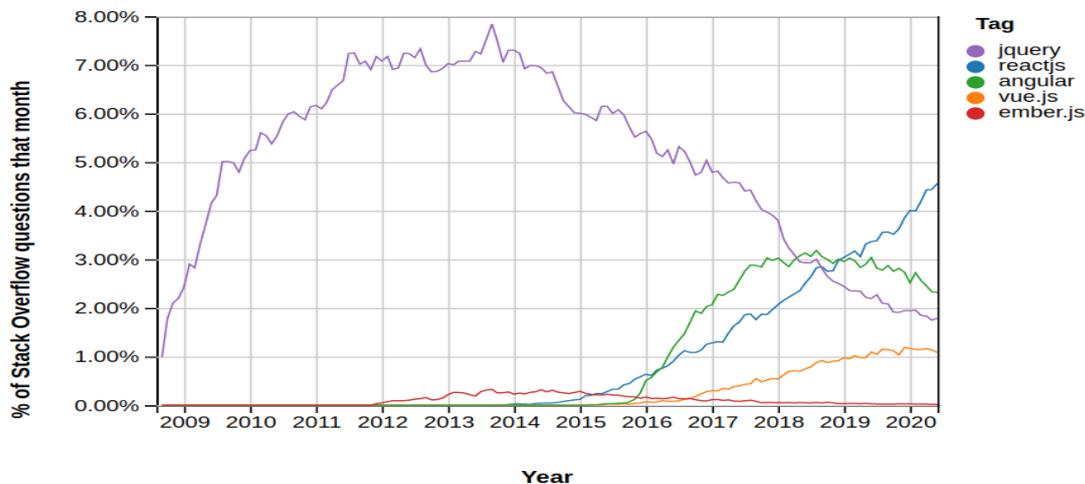
del proyecto, Vue nos ofrece una sintaxis que no resulta compleja, y es muy intuitiva para los desarrolladores.

Una de las características importantes y que fue muy utilizada en la migración del software Qanty es el trabajo con componentes de Vue. Un componente, en términos simples, es un elemento (como los que se trabaja en HTML) en el que se encapsula código reutilizable; la estructura básica de un componente se basa en tres partes: elementos HTML, código JavaScript, y estilos de CSS. Otra de las características más potentes de Vue es que es un framework reactivo, es decir, es un sistema que reacciona a los cambios. Por ejemplo, si el valor de alguna variable cambia, podemos reaccionar a este cambio alterando la vista de la página, esto sin la necesidad de recargar la página. Cada componente lleva un seguimiento de sus dependencias reactivas, de esta manera el sistema de Vue sabe cuándo debe renderizar y qué componentes renderizar [2].



Gráfica 1. Interés sobre frameworks front-end más usados durante los últimos cinco años. Recuperado de <https://trends.google.com/trends/explore?cat=31&date=today%205-y&q=Vue,React,Angular,%2Fm%2F0268gyp,ember>

Vue Router: es el enrutador oficial de Vue que nos permite construir aplicaciones de una sola página de una manera más eficiente. En el proyecto se utilizó algunas de sus funcionalidades y entre ellas el trazado de rutas o vistas anidadas, con esto logramos enlazar componentes con diferentes URLs. La gran ventaja de esto es que no tenemos que recargar nuevamente nuestra página, sino que el sistema de Vue solo tendrá que renderizar la página con los nuevos componentes, aunque el usuario tendrá la sensación de que la página se carga completamente.



Gráfica 2. Porcentaje de preguntas realizadas sobre los frameworks más usados para el desarrollo front-end. Recuperado de <https://insights.stackoverflow.com/trends?tags=reactjs%2Cvue.js%2Cangular%2Cember.js%2Cjquery>

Vuex: Es una librería oficial para Vue.js que sirve como un almacén de datos centralizado para todos los componentes dentro de una aplicación. Vuex se empleó en el desarrollo del proyecto para manejar los estados que eran comunes a múltiples componentes, y también con la idea, que cuando alguno de los componentes modificara un estado este cambio se transmite a los demás componentes que tenían dependencia del estado modificado esto sacando provecho del poder del sistema reactivo de Vue.js. Un estado se actualiza mediante las mutaciones que se realizan en respuesta a una acción invocada por algún componente[3]; una demostración de esta secuencia de pasos la podemos ver en la figura 2 donde podemos ver de forma genérica el flujo de trabajo de la librería Vuex.

Back-end

Firestore: El back-end del software Qanty se ha desarrollado alrededor de la plataforma Firestore de Google. Firestore soportada por Google actualmente, es un conjunto de herramientas orientados a la creación de aplicaciones de alta calidad, con fácil adaptabilidad al crecimiento de usuarios, y a diferentes entornos como Android, iOS, y la web, la integración de servicios de una manera ágil, además permite la escalabilidad de proyectos en el tiempo. La plataforma Firestore ofrece un sin número de servicios, el software Qanty emplea los servicios Authentication para el registro y autenticación de usuarios, Cloud Firestore es la base de datos, Storage se emplea para almacenar recursos del software como imágenes y reportes, Cloud Functions se emplea para manejar las peticiones a la base de datos.

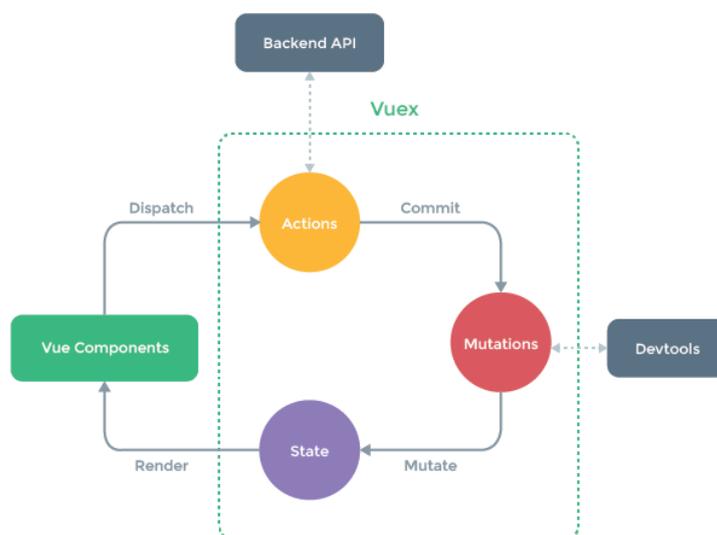


Figura 2. Representación simple del flujo del trabajo de Vuex. Recuperado de <https://vuex.vuejs.org/>

IV. Metodología

En el proceso de desarrollo del proyecto 'Migración de interfaz de usuario para el software Qanty', se realizó mediante prioridades, es decir, había tareas solicitadas por la empresa que debían ser desarrolladas primero que otras con la idea de lograr un producto mínimo viable que pudiera ser llevado a producción.

Nuestro primer paso fue conocer el software Qanty, familiarizarnos con sus interfaces tanto la de dispositivos móviles y la web, esta última siendo la prioridad en el proyecto que desarrollamos; mirar con qué tecnologías se estaba soportando Qanty, los nombres de los archivos y las direcciones donde se ubicaban dentro del proyecto, esto muy importante por el tamaño y la cantidad de archivos que se manejan, y el poder saber dónde había que buscar cuando se detectaba un fallo o se quería revisar algo en específico. En el proceso de conocer Qanty se detecta que el front-end web se encontraba desarrollado con tecnologías HTML, CSS, JavaScript[4], por lo que el siguiente paso fue un proceso de repaso y aprendizaje de estas tecnologías para poder interpretar Qanty desde código fuente y entender el trabajo realizado de los anteriores desarrolladores.

Luego, se inicia el aprendizaje del framework principal elegido para el desarrollo del proyecto, el framework Vue.js[5], este se eligió por las características tan interesantes que ha ido adaptando durante los últimos años. El equipo detrás de

Vue.js liderado por el ingeniero Evan You, ha tomado e implementado lo mejor que tienen otros frameworks y ha desechado aquellas cosas que no aportan mayor valor, dejando abierta la puerta de añadir estas funcionalidades a nuestro proyecto mediante software de terceros según nuestras necesidades. Esto nos permitió tener un mayor control sobre las librerías que utilizamos, y el tamaño del proyecto fue considerablemente menor[6].

Pasado alrededor de un mes y medio realizando los pasos mencionados anteriormente, se inició el proceso de migración del software Qanty y se logran realizar las siguientes actividades durante el desarrollo del proyecto:

Front-end:

- Interfaz de Loguear: Formulario para validar que un usuario es válido para ingresar a Qanty.
- Sección de asesor: En esta sección los asesores pueden realizar tareas como la consulta de agenda de citas disponibles o los turnos por llamar, creación de turnos, consulta del estado de órdenes o pedidos, y ver los tiempos de atención o espera de los clientes.
- Historial de pedidos: En esta sección se pueden consultar los pedidos de ciertas fechas, y tener un rastreo del estado de los pedidos, como también información del cliente que realizó el pedido, fecha de entrega o recogida de estos.
- Sección de infraestructura: Esta sección es visible para los usuarios que cumplan con el rol de administrador, y se encarga de la administración de las sedes. Por esta sección los administradores pueden crear nuevos usuarios y brindarles permisos, crear y administrar las filas de atención, definir los horarios de atención, logos para las impresiones de turnos, configurar las pantallas de llamado de turnos, etc.
- Sección de reportes: Sección disponible para los administradores, donde pueden generar reportes sobre el detallado de citas, detallado de órdenes, detallado de turnos, etc.
- Interfaz llamado de turnos: Se implementa las vistas para la notificación del llamado de turnos en las pantallas o televisores las cuales informan a los clientes el módulo y el turno para su atención.
- Lector código QR: Cuando el usuario crea un turno desde la página web o la aplicación móvil se genera un código QR que se envía al correo del

cliente para ser escaneado cuando asista a la sede o a la cita virtual para notificar su llegada.

Back-end:

- Reportes temporales en base de datos: Se generan los reportes desde el back-end, y no desde el navegador, y se almacenan estos reportes de forma temporal en las bases de datos.

Bugs en la versión anterior de Qanty:

- Fechas de creación y recogida de órdenes: Se corrigió que las fechas de creación y recogida de órdenes, se estaban almacenando de forma trocada, y por ende, se estaba generando información errónea en los reportes.
- Fechas con la norma ISO 8601: Estandarizar el almacenamiento de horas y fechas según la norma ISO 8601, sin offset del tiempo coordinado UTC.
- Consulta de pedidos sobre solo un día: Solo se podía realizar consulta de pedidos sobre solo un día, se modificó para consultar sobre más de un día.

Bugs en la versión actual de Qanty:

- Cero adicional en la hora de recogida de pedidos: Se estaba mostrando un cero adicional al lado derecho de la hora de recogida de pedidos.
- El excel que se exporta desde la tabla pintada en el front no queda con los cambios hechos sobre la disposición de la misma tabla.
- Las tablas de reportes lanzan errores cuando se cambia el idioma luego de haber hecho alguna consulta
- Reinicio de eventos de pantallas: Se detectó que los eventos diarios de turnos no se estaban reiniciando.
- Buscador para sedes: Cuando la lista de sedes es demasiado grande no hay una manera de un buscador para facilitar la selección por parte de los usuarios.

Durante el tiempo de desarrollo del proyecto se debía registrar los avances en la herramienta Github, cumpliendo con las pautas dadas al inicio del proceso; y por este canal la empresa, revisaba el trabajo que se estaba haciendo, y dejaba consignado allí los comentarios o sugerencias respectivos. Cada que se realizaba alguna entrega o avance, se debía realizar pruebas del trabajo hecho que garantizara el correcto funcionamiento, estas pruebas las realizaba

manualmente, rastreando la ejecución de la página web empleando el inspector de los navegadores Google Chrome y Firefox web, usando muchas de sus funcionalidades como *console* para mostrar mensajes, *Network* para rastrear las descargas y peticiones de la página web, *Vue Devtools* lo empleamos mucho para observar los datos y componentes en tiempo real, etc.

Adicional a esto, se realizaron reuniones con el equipo del proyecto o directamente con el director de investigación y desarrollo de Dinámica y Desarrollo, para dar reportes de avances, realizar exposiciones sobre algún tema, y en muchas ocasiones solicitar mayor información sobre los requerimientos para poder realizar cierta tarea.

V. Resultados y análisis

La herramienta Github, encargada de registrar y controlar los cambios realizados durante el proyecto, tiene una función que nos informa qué porcentaje de código se ha desarrollado en los diferentes lenguajes de programación o frameworks. La figura 3.a muestra la información del proyecto Qanty al inicio del proyecto donde estaba implementado en un 100% con JavaScript puro y la figura 3.b la información del proyecto Qanty a la terminación del contrato de aprendizaje, donde se logró llevar a Vue.js un 95.8%, y otros frameworks de diseño como Bootstrap, BV ocuparon un 0.6%, y JavaScript puro, paso de ser el 100% a solo ocupar un 3.6% del proyecto Qanty.

Languages



a

Languages



b

Figura 3. Porcentaje de código de cada lenguaje empleado en Qanty.

Se obtiene un sistema con una nueva paleta de colores, elegido por el equipo de trabajo, con la idea de no emplear los valores por defecto que ofrecen los navegadores, y de esta manera dar una imagen distintiva a Qanty; en la tabla 1 se detalla la paleta de colores. Las traducciones o idiomas de Qanty(español,

inglés y portugués) se centralizaron de manera que todo el sistema emplee las mismas traducciones.

El software Qanty es un sistema responsive, es decir, se logró que este pudiera adaptarse de forma automática a diferentes tamaños de pantallas y reaccionar a cambios en el tamaño de la ventana del navegador, es decir, es completamente adaptable(Responsive). Esto se logró en mayor medida al uso de componentes como b-row combinado con b-col, que nos permitió ver la ventana como filas y columnas ocupando porcentajes de la ventana; y algunas propiedades de CSS. Si observamos las figuras 4, 5 y 6 observamos como es el comportamiento de todas las vistas en general en Qanty cuando cambian el tamaño disponible para la vista, lo primero que salta a la vista es el cambio en la barra de navegación, cuando la ventana o pantalla son grandes se muestra la totalidad de pestañas junto con el logo y la información del usuario, figuras 4 y 5, al volverse más pequeña la pantalla se ocultan las pestañas y la información del usuario, y se muestra un icono de menú por donde se podrá ver esta información, figura 6. Otro comportamiento observable es la redistribución de los elementos en la tabla de 'Filas de oficina', donde cada elemento pasa a ocupar una fila.

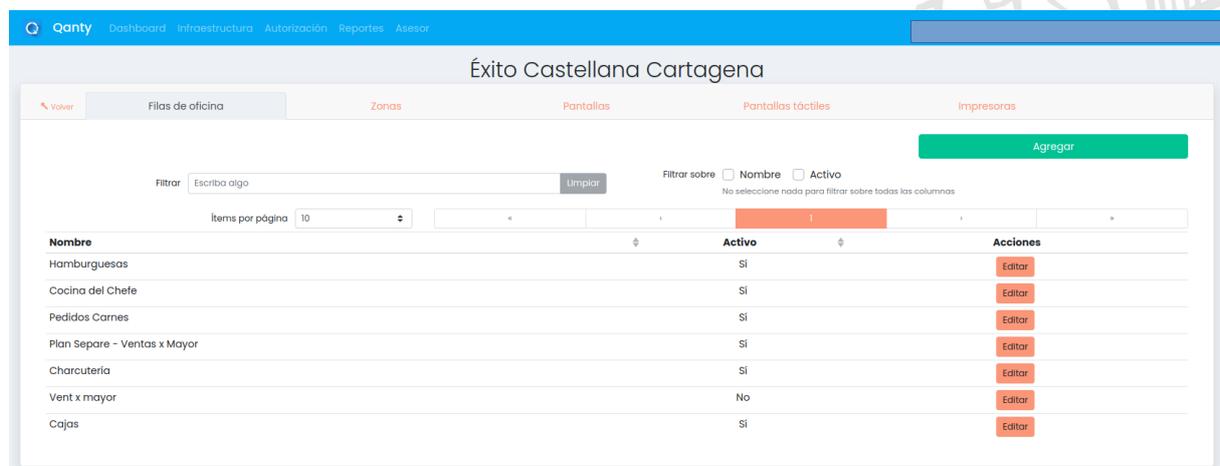
Nombre	Descripción	Color en hexadecimal
body-color	Letras	212529
body-bg	Fondos	edf1f5
primary	Botones, sombras, pestañas.	fb9678
success	Botones secundarios, barra de navegación.	00c292
danger	Mensajes de error.	e46a76
warning	Mensajes de advertencia.	fec107
info	Mensajes informativos.	03a9f3

Tabla 1. Paleta de colores Qanty.

La barra de navegación de Qanty dejó de estar ubicada en la parte izquierda de la pantalla de forma horizontal, para ubicarse en la parte superior de forma

vertical, lo que se gana con este cambio es que las opciones en la mayoría de los casos estuviese visible y aun click, la figura 7 muestra la barra de navegación antigua.

Se implementó lector de código QR, que contiene la información de la cita y los datos del cliente que realizó la solicitud. El código QR y la dirección url para la atención virtual se envía al correo del cliente cuando hace el agendamiento por la página o por las aplicaciones móviles, en las figuras 8, 9 y 10 se puede observar lo descrito.



The screenshot shows the Qanty web application interface. At the top, there is a navigation bar with the Qanty logo and menu items: Dashboard, Infraestructura, Autorización, Reportes, and Asesor. Below this, the page title is 'Éxito Castellana Cartagena'. The main content area has a breadcrumb trail: Volver > Filas de oficina > Zonas > Pantallas > Pantallas táctiles > Impresoras. There is a search bar with the text 'Escribo algo' and a 'Limpiar' button. To the right, there are filter options for 'Nombre' and 'Activo', with a note: 'No seleccione nada para filtrar sobre todas las columnas'. A table with 3 columns is displayed: 'Nombre', 'Activo', and 'Acciones'. The table contains 7 rows of data. A green 'Agregar' button is located at the top right of the table area.

Nombre	Activo	Acciones
Hamburguesas	Si	Editar
Cocina del Chefe	Si	Editar
Pedidos Carnes	Si	Editar
Plan Separe - Ventas x Mayor	Si	Editar
Charcutería	Si	Editar
Vent x mayor	No	Editar
Cajas	Si	Editar

Figura 4. Ventana de tamaño 1679 x 976 px.

Al inicio del proyecto se esperaba implementar en Qanty, la opción de poder realizar videoconferencias para poder atender en línea a los clientes. A la fecha de finalización del contrato de aprendizaje se logró dejar un trabajo iniciado, con la sala de conexión entre el cliente y el asesor implementado, figura 11, pero sin realizar pruebas de correcto funcionamiento.

Se detectaron los siguientes bugs en la versión anterior de Qanty:

- Fechas de creación y recogida de órdenes: Se corrigió que las fechas de creación y recogida de órdenes, se estaban almacenando de forma trocada, y por ende, se estaba generando información errónea en los reportes.
- Fechas con la norma ISO 8601: Estandarizar el almacenamiento de horas y fechas según la norma ISO 8601, sin offset del tiempo coordinado UTC.
- Consulta de pedidos sobre solo un día: Solo se podía realizar consulta de pedidos sobre solo un día, se modificó para consultar sobre más de un día.

Qanty Dashboard Infraestructura Autorización Reportes Asesor

Éxito Castellana Cartagena

Volver Filas de oficina Zonas Pantallas Pantallas táctiles

Impresoras

Agregar

Filtrar Limpiar

Filtrar sobre Nombre Activo
No seleccione nada para filtrar sobre todas las columnas

Ítems por página 10

Nombre	Activo	Acciones
Hamburguesas	Sí	Editar
Cocina del Chefe	Sí	Editar
Pedidos Carnes	Sí	Editar
Plan Separe - Ventas x Mayor	Sí	Editar
Charcutería	Sí	Editar
Vent x mayor	No	Editar
Cajas	Sí	Editar

Figura 5. Ventana de tamaño 805 x 976 px.

Y a la fecha de entrega del proyecto, se presentaban los siguientes bugs algunos sin solucionar de la anterior versión de Qanty y otros que se daban a causa de algunas tareas mal implementadas durante desarrollo del proyecto, como lo son los tres primero de la siguiente lista:

- Cero adicional en la hora de recogida de pedidos: Se estaba mostrando un cero adicional al lado derecho de la hora de recogida de pedidos.
- El excel que se exporta desde la tabla pintada en el front no queda con los cambios hechos sobre la disposición de la misma tabla.
- Las tablas de reportes lanzan errores cuando se cambia el idioma luego de haber hecho alguna consulta
- Reinicio de eventos de pantallas: Se detectó que los eventos diarios de turnos no se estaban reiniciando.

- Buscador para sedes: Cuando la lista de sedes es demasiado grande no hay una manera de un buscador para facilitar la selección por parte de los usuarios.

The screenshot shows the Qanty web application interface. At the top, there is a blue header with the Qanty logo and a menu icon. Below the header, the title 'Éxito Castellana Cartagena' is displayed. The main content area features a navigation bar with buttons for 'Volver', 'Filas de oficina', 'Zonas', 'Pantallas', 'Pantallas táctiles', and 'Impresoras'. A green 'Agregar' button is located in the top right corner. Below the navigation bar, there is a search bar labeled 'Filtrar' with the placeholder text 'Escriba algo' and a 'Limpiar' button. Underneath the search bar, there are filter options: 'Filtrar sobre' with checkboxes for 'Nombre' and 'Activo', and a note 'No seleccione nada para filtrar sobre todas las columnas'. A dropdown menu for 'Ítems por página' is set to '10'. Below the dropdown is a pagination bar showing '«', '‹', '1', '›', and '»'. The main content area displays a table with two rows of items:

Nombre	Hamburguesas
Activo	Sí
Acciones	Editar
Nombre	Cocina del Chefe
Activo	Sí
Acciones	Editar

Figura 6. Ventana de tamaño 540 x 976 px.

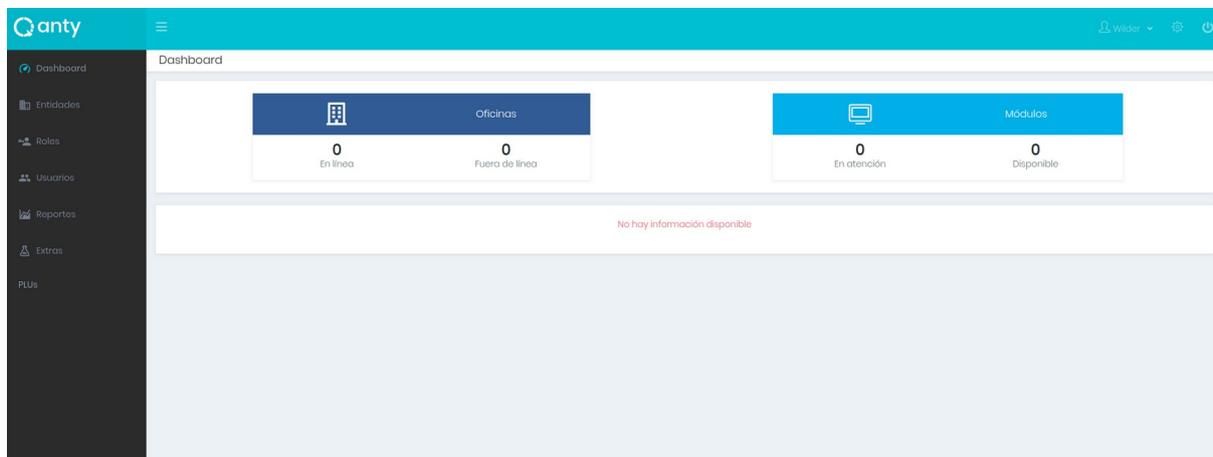


Figura 7. Distribución elementos en la versión anterior de Qanty .

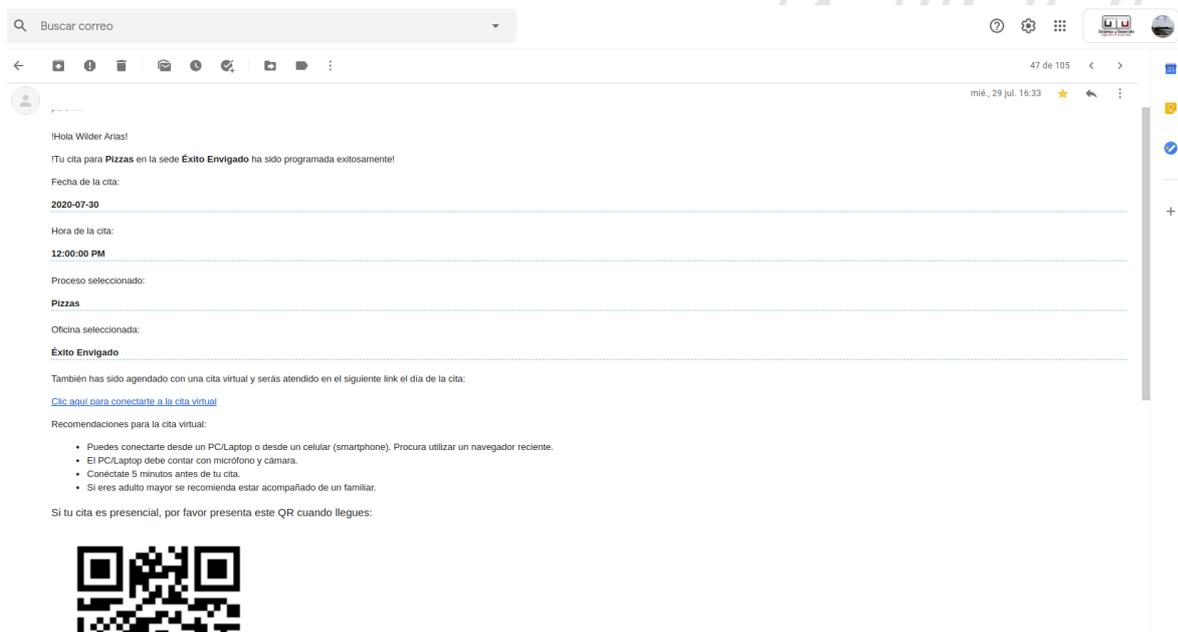


Figura 8. Ejemplo de correo de agendamiento de cita con url para la atención, y código QR.

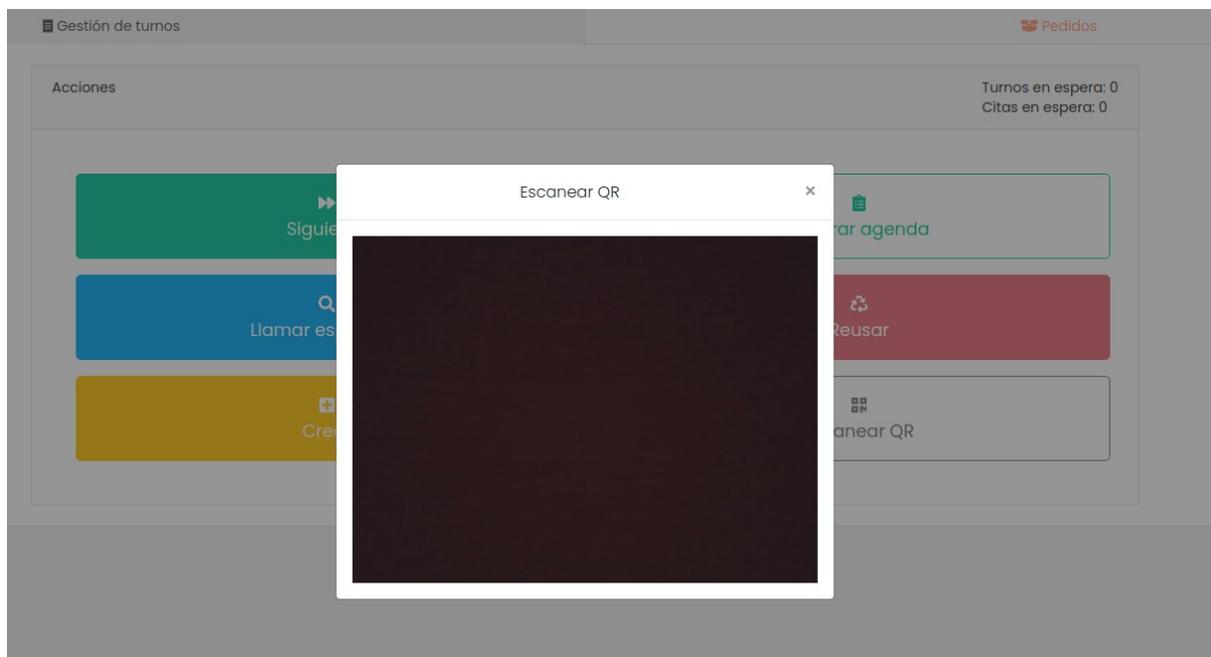


Figura 9. Lector de código QR.

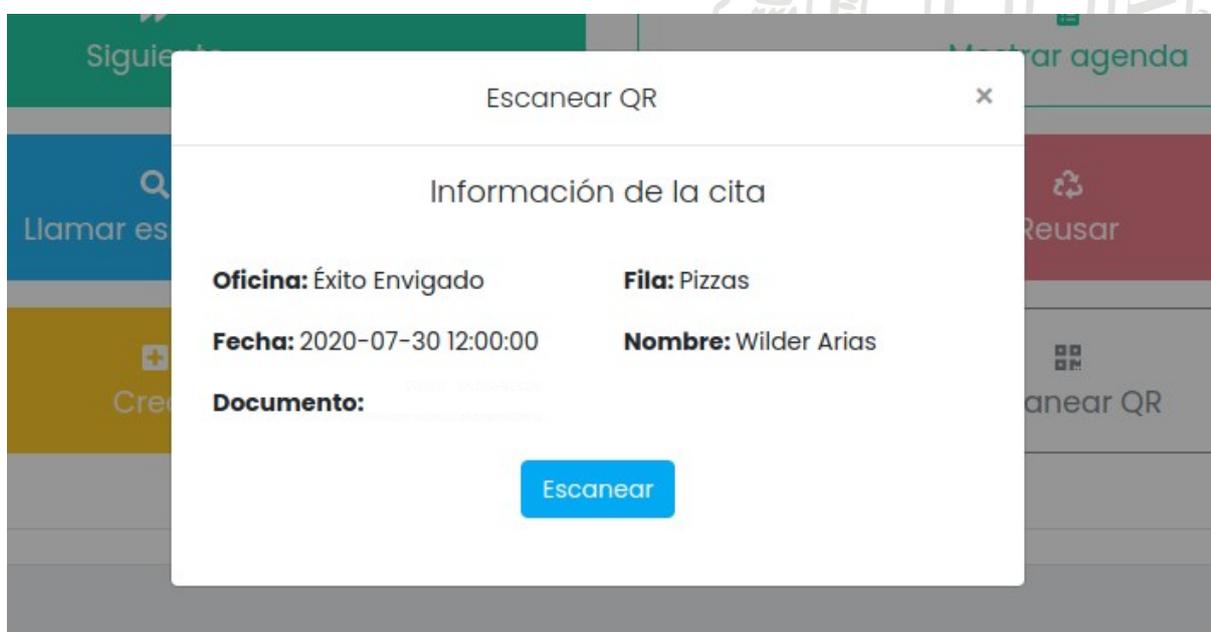


Figura 10. Información dentro del código QR.



Figura 11. Sala de video-llamada cliente-asesor.

VI. Conclusiones

- Se logró una migración alrededor del 90% del front-end antiguo de Qanty con respecto a la meta inicial; se mantuvo la funcionalidad con la que recibimos Qanty, para poder respetar y mantener los compromisos de funcionalidades adquiridos por clientes antes de las actualizaciones.
- La actualización se realizó empleando como tecnología principal el framework Vue.js, por las características tan interesantes descritas durante el informe, pero algo que cabe recalcar, es su facilidad de adaptabilidad a proyectos existentes, como era el caso de Qanty, donde ya teníamos un proyecto desarrollado.
- El proyecto pasa de estar desarrollado 100% en JavaScript puro, a tener un 95.8% de código en Vue.js y un 3.6% de JavaScript puro.
- Se implementó una nueva estructura de navegación más intuitiva que facilita la navegación dentro de la página web; además, una nueva paleta de colores que le dan un distintivo a la imagen de la marca del software Qanty de la empresa Dinámica y Desarrollo.
- Las tareas y funcionalidades desarrolladas durante el proyecto fueron probadas, y se entregaron en un estado funcional cumpliendo con los lineamientos solicitados por la empresa Dinámica y Desarrollo, con su debida documentación y comentarios en la plataforma Github.

- El software cumple con los lineamientos solicitados de Diseño Web Adaptable, con lo que garantizamos que la página web sea adaptable a los diferentes tamaños de los dispositivos o de las ventanas de los navegadores web.
- Aunque se logra dar solución al problema planteado, nos encontramos que no todas las soluciones son perfectas y en nuestro caso al ser Qanty una solución tecnológica que depende de conexiones a una red como Internet, un fallo en este causaría problemas en la prestación del servicio.
- Por último, cabe destacar que cuando se entrega el proyecto al finalizar el periodo del contrato de aprendizaje la compañía ya había realizado cierta migración del software implementado en desarrollo a la línea de producción.

VII. Referencias Bibliográficas

[1]. Tecnología para desarrolladores web. En línea. <https://developer.mozilla.org/es/docs/Web> . Consultado: 10 de agosto 2020

[2]. Introduction, what is Vue.js?. En línea. <https://vuejs.org/v2/guide/index.html> . Consultado: 10 de agosto de 2020

[3]. What is Vuex?. En línea. <https://vuex.vuejs.org/> . Consultado: 10 de agosto de 2020

[4]. Marijin Haverbeke. ELOQUENT JAVASCRIPT: A Modern Introduction to Programming. Tercera edición. 2018. 472 pp. ISBN-13: 978-1-5932-7950-9

[5]. Vue.js & JavaScript courses. En línea. <https://vueschool.io/courses?filter=free-courses>. Consultado: 07 de abril de 2020

[6]. Adam Freeman. Pro Vue.js 2. Primera edición. Londres: Apress, 2018. 746 pp. ISBN-13(Electrónico): 978-1-4842-3805-9