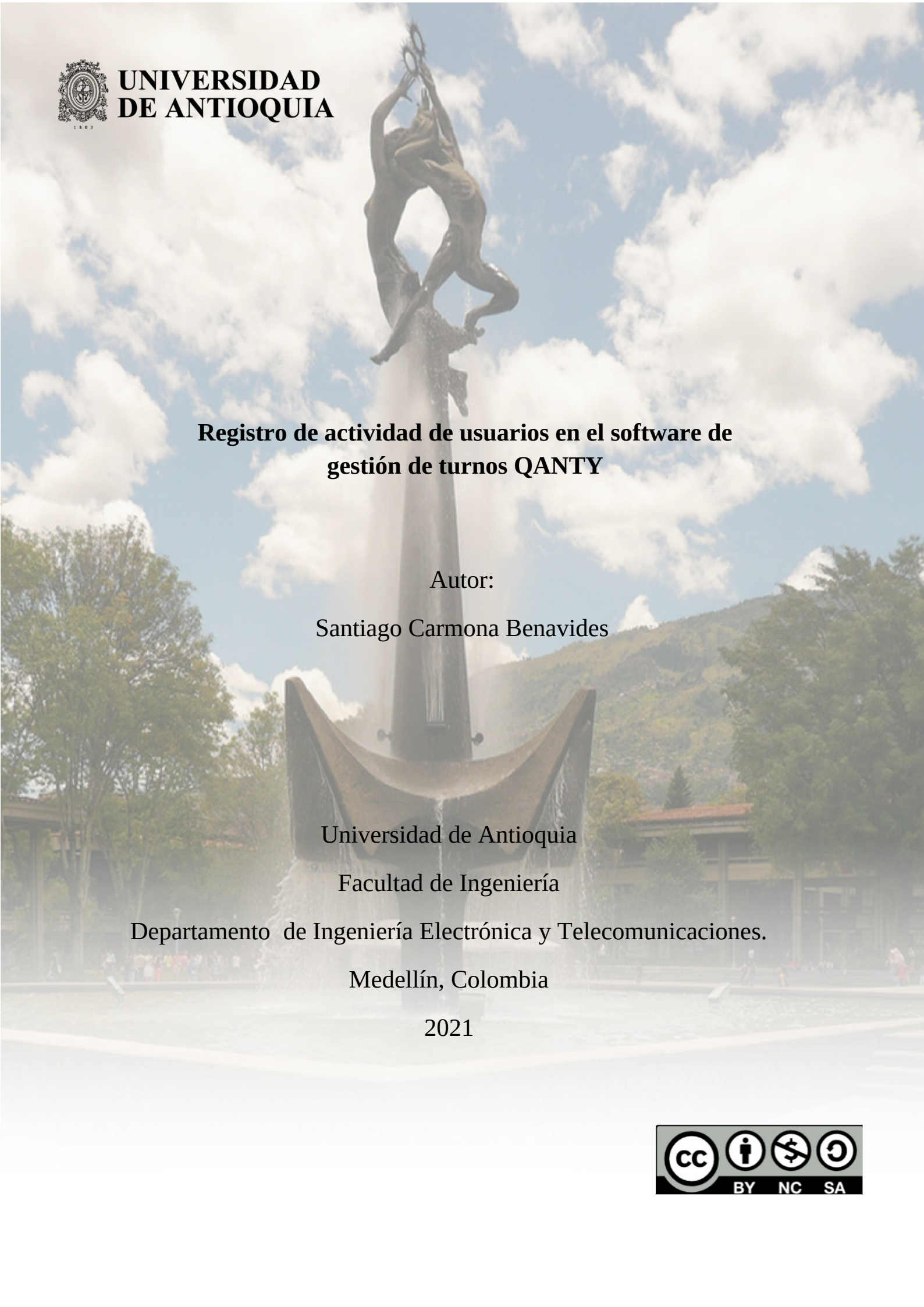

UNIVERSIDAD DE ANTIOQUIA

Registro de actividad de usuarios en el software de gestión de turnos QANTY

Autor:

Santiago Carmona Benavides

Universidad de Antioquia

Facultad de Ingeniería

Departamento de Ingeniería Electrónica y Telecomunicaciones.

Medellín, Colombia

2021

Registro de actividad de usuarios en el software de gestión de turnos QANTY

Autor:
Santiago Carmona Benavides

Informe de práctica como requisito para optar al título de:
Ingeniero Electrónico.

Asesores:

Sebastián Isaza Ramírez
Universidad de Antioquia

Andrés Mauricio Eusse Torres
Dinámica y Desarrollo

Universidad de Antioquia
Facultad de Ingeniería
Departamento de Ingeniería Electrónica y Telecomunicaciones.
Medellín, Colombia
2021.

Resumen

Con el fin de optimizar los tiempos de espera y brindar a los usuarios experiencias agradables, algunas empresas deciden hacer uso de sistemas tecnológicos para la gestión de turnos y filas. Uno de ellos es Qanty, herramienta creada por la empresa Dinámica y Desarrollo, quien ha creado soluciones de este tipo para el mercado Colombiano como lo es la aplicación Filapp y ahora busca incursionar a nivel internacional con Qanty.

El proyecto desarrollado durante el periodo de práctica profesional y enunciado en este informe se enfoca en el lado de los administradores de compañías que hagan uso de la herramienta permitiéndoles tener un registro organizado de las labores realizadas por parte de los asesores encargados de atender a sus clientes.

A la fecha se han entregado el código que permite: registrar las acciones que se realizan tanto asesores como administradores, ver en tiempo real cual es el estado actual de todos los módulos de la compañía (queriendo decir esto que se conoce el estado del asesor que se encuentra trabajando en dicho módulo) además de crear, editar y eliminar motivos de pausa.

Introducción

Hoy en día existen una amplia cantidad de compañías que brindan atención a un gran número de clientes o usuarios cada día, esto genera la necesidad de contar con un sistema para el control de turnos por medio del cual se pueda organizar el orden de atención, logrando prestar el servicio de forma eficiente a todas las personas que lo requieran. Inicialmente los sistemas de gestión de turnos se limitaban a organizar las personas según el orden de llegada a la sede en la cual esperaban ser atendidas, al encontrarse con la gran limitante de ser necesario el desplazamiento del usuario hasta la sede para poder contar con un turno. La empresa Dinámica y Desarrollo S.A.S ha incursionado en la creación de sistemas para la gestión de turnos que han tenido gran acogida al contar con mayor flexibilidad y permitir la asignación de estos a los usuarios sin la necesidad de encontrarse presentes en el lugar de atención, permitiendo de esta manera que las personas puedan aprovechar en otras actividades todo el tiempo que usualmente se perdía al interior de una sede esperando a ser atendidas y las empresas tengan un flujo controlado de usuarios acorde con la capacidad de atención. Por parte de las compañías se tienen 2 roles principales que hacen uso del sistema de gestión de turnos Qanty los cuales son: Asesor que entre sus funciones puede crear y atender turnos o citas y Administrador quien puede, además de desarrollar tareas de asesor, realizar diferentes configuraciones de la herramienta de software para la compañía o sede donde trabaja.

El uso que se le da a Qanty por parte de los usuarios con estos 2 roles ya mencionados generan gran cantidad de información que puede llegar a ser muy útil para la compañía y al momento de iniciar con este proyecto de prácticas dicha información no se estaba recopilando, fue por esto que se vio la necesidad de registrar las actividades que desempeñan los usuarios de esta herramienta con el fin de ofrecer la posibilidad de analizar estadísticamente la eficiencia en la atención a los turnos, evaluar en tiempo real el trabajo realizado por los asesores y llevar un control de las configuraciones realizadas por parte de los administradores teniendo el registro de quien ha hecho cada uno de los cambios.

Para guardar dicha información se capturan las peticiones que se hacen desde el front-end hacia el back-end, después se analiza cuales peticiones deben ser registradas en la base de datos y luego, para cada una de estas se analiza que información se va a guardar y se define la estructura con que está se va a escribir, con esta información se pueden generar diversos tipos de reportes dependiendo de la forma en que se filtren los datos guardados. Además de esto se ha creado panel de visualización en tiempo real del estado de módulos y asesores permitiendo así a los administradores de las compañías tener claro en todo momento el flujo de trabajo que tiene cada una de sus sedes. Todo esto mediante el uso de herramientas para desarrollo web como lo son HTML, CSS, JavaScript, el servicio de Firebase, la framework VueJs y Bootstrap.

Objetivos

General

Complementar el software Qanty con funcionalidades que permitan a los asesores entrar en estado de pausa y en general registrar las acciones desarrolladas por los usuarios mediante el desarrollo de códigos front-end y back-end en un entorno conjunto de VueJs, Bootstrap y Firebase para permitir tener un inventario de las acciones realizadas en la aplicación.

Específicos

- Agregar al código de Qanty la capacidad de guardar en base de datos información de las acciones realizadas en el front-end detectando las peticiones que se hacen al back-end y tomando datos que puedan ser útiles para luego generar reportes y estadísticas de atención a usuarios.

- Realizar el desarrollo necesario para que el asesor entre en estado de pausa creando el botón, la función y la ventana modal que permitan seleccionar el motivo y también una rutina que analice cuánto tiempo se lleva en este estado y ejecute las acciones necesarias. Esto con el fin de que el asesor pueda ausentarse de sus labores durante cierto tiempo de acuerdo al motivo seleccionado.

- Crear una rutina que se encargue de calcular en todo momento el estado de los asesores. Para esto se analiza si están atendiendo a alguien, hay turnos pendientes o se entró en pausa. Con el estado se muestra a los administradores en tiempo real que están haciendo los asesores y se guarda información estadística de atención a usuarios.

- Desarrollar un panel de visualización en tiempo real mediante lecturas al estado ya definido, organizando a manera de lista todos los asesores, diferenciando por colores su estado y mostrando cuánto tiempo llevan en el estado actual. Así el administrador sabrá en cualquier momento: el flujo de usuarios de la sede y sí el número de asesores con los que cuenta es el adecuado.

- Añadir la opción de generar reportes de uso por parte de asesores y administradores, para la generación de estos se usará la información que ya se ha guardado. Dichos reportes permiten el análisis de la eficiencia en atención por parte de los asesores y llevar un control de los cambios de configuración realizados por los administradores.

- Complementar el panel existente de edición para la compañía y sus sedes para poder crear, editar y eliminar los motivos de pausa, esto se hará editando archivos ya existentes dentro del proyecto para agregar un formulario de creación, una ventana de edición y un botón que permita eliminar. También se deben hacer cambios en el back-end para guardar los motivos en la base de datos. Todo esto con el fin de que el administrador pueda variar según sea necesario los motivos de pausa para los asesores.

- Verificar el correcto funcionamiento de todo el desarrollo realizado mediante el uso de la aplicación desde perfiles tanto con el rol de administrador como el de asesor con el fin de garantizar a la empresa que todos los códigos entregados cumplen a cabalidad los objetivos para los cuáles fueron creados.

Marco Teórico

En general los proyectos relacionados con el desarrollo web se dividen en 3 componentes principales que son HTML, JavaScript y CSS. Cada una de estas partes hace un aporte importante para lo que al final, en conjunto, se convertirá en una página web.

HTML son las siglas representativas para *HyperText Markup Language*, se traduce como lenguaje de marcas de hipertexto. Estas marcas definen el contenido y estructura de la página web. HTML se encuentra estandarizado y es soportado por W3C, consorcio comercial fundado por Tim Berners-Lee y que se dedica a implementar tecnologías uniformes en el uso y desarrollo de Internet. Este lenguaje hace uso de marcas para etiquetar el contenido que conformará la página web, este puede ser texto o archivos multimedia (imágenes, video, audio etc). Las marcas de HTML incluyen gran variedad de elementos predefinidos que pueden ser títulos, párrafos, botones, tablas, y muchos más, los cuales se distinguen entre si mediante etiquetas que consisten en el nombre del elemento rodeado por "<" y ">". [1]

Figura 1. sintaxis básica de un elemento HTML, tomado de https://www.ampersoundmedia.com/20-etiquetas-html-esenciales-para-traducir-paginas-web/

JavaScript es actualmente uno de los más potentes e importantes lenguajes de programación por tres enfoques claros: es útil, práctico y está disponible en cualquier navegador web. nace como un lenguaje sencillo destinado a añadir algunas características interactivas a las páginas web. Sin embargo, hoy en día ha crecido de manera acelerada y es el lenguaje de programación que se utiliza en casi todos los sitios web en el mundo. Se encuentra disponible principalmente en lado front-end, agregando mayor interactividad a la web, también es posible usar librerías y frameworks como: Jquery, Angular, React y demás, las cuales se encuentran escritas sobre JavaScript y se usan para crear una mejor experiencia de usuario en los sitios web. De igual manera JavaScript se puede utilizar en los servidores web siendo node.js la más recomendable de las opciones para este caso. [2]

CSS del inglés *Cascading Style Sheets*, se puede traducir al español como Hojas de Estilo en Cascada, se trata de una tecnología utilizada para dotar de cualidades visuales y estéticas a una página web, es un código que unido al HTML permite darle forma, color, posición y otras características visuales a una página. Se le denomina estilos en cascada porque se aplican de arriba a abajo siguiendo un patrón de herencia y en el caso de existir alguna ambigüedad, se siguen una serie de normas para resolverla. [3]

Para todas las actividades planteadas para el desarrollo de este proyecto se tomaron en cuenta 3 necesidades básicas: el almacenamiento de información en base de datos, organización de paneles en front-end agradables a la vista y la recopilación y manejo de información. Para esto se usaron las siguientes herramientas cuyo funcionamiento se basa en los 3 componentes básicos del desarrollo web anteriormente mencionados:

Firestore es una base de datos NoSQL dentro de la cual se pueden almacenar datos entre combinaciones de documentos, colecciones y subcolecciones a los cuáles se puede acceder directamente desde la aplicación web de qanty. Firestore hace parte de la plataforma Firebase creada por Google la cual permite desarrollar aplicaciones de gran calidad y se encuentra alojada en la nube, este entorno se puede conectar con servicios de Playstore, autenticación, GoogleAds, crash reports, analytics, cloud functions y otros más, todos estos con respaldo por parte de Google. [4]

Otro ejemplo del uso que se da a Firebase a parte de la base de datos Firestore es el servicio de **Cloud Functions**, framework sin servidores que permite ejecutar automáticamente el código de back-end en respuesta a solicitudes realizadas desde el front-end. El código de las funciones se almacena en la nube de Google y se ejecuta en un entorno administrado haciendo que no sea necesario administrar ni escalar servidores propios [5]. El llamado a estas funciones se hace mediante solicitudes HTTPS realizadas desde el front-end de Qanty, por ejemplo, todo el proceso de recopilación de datos correspondientes a las actividades realizadas por administradores y asesores se hizo mediante la lectura de todas estas solicitudes HTTPS en el momento que llegan al back-end.

VueJS es un framework de código abierto para JavaScript creada por Evan You, la cual permite crear interfaces de usuario reactivas ante cambios en los datos del proyecto, se encuentra enfocada en la capa de visualización y es fácil de integrar con otras librerías. [6] Una característica interesante a parte de su reactividad es la librería **Vuex** que se usa para la gestión del estado (State Management) de aplicaciones Vue.js. Sirve como un almacén centralizado para los componentes de una aplicación [7]. Vuex se utiliza dentro del proyecto de Qanty para almacenar variables globales a las cuales se puede acceder desde cualquier parte del proyecto según sea necesario.

Un aspecto importante del uso que se le da a VueJS dentro de Qanty es la posibilidad que brinda para crear elementos re utilizables. Un ejemplo de esto es la ventana modal desarrollada para la creación y edición de los motivos de pausa, esa misma ventana se usa en todos lo niveles que sea posible configurar dichos motivos, es decir, durante la edición de motivos de pausa globales, para una sede, una zona o un solo módulo. Así como este caso particular son muchos mas lo elementos que se reutilizan en diferentes partes de Qanty mostrando la versatilidad y el alcance de la framework utilizada para el proyecto.

Bootstrap es un conjunto de herramientas de código abierto desarrollado por Twitter donde se tiene una combinación de CSS y JavaScript que permiten estilizar los elementos que se encuentran en una página HTML con el fin de ofrecerle al usuario una experiencia de navegación agradable, es de fácil integración con VueJS [8]. Es la herramienta usada para dar estilo a todo el proyecto definiendo tamaño, color y posición de todos los elementos en general.

Metodología

Primer acercamiento y conocimiento del proyecto

Al momento de iniciar este proyecto de prácticas se tomaron unos días de acondicionamiento con el entorno laboral de la empresa tomando la oportunidad de conocer el flujo de trabajo, el equipo de desarrollo que se encontraba trabajando en el momento en Qanty, los procedimientos de asignación de tareas y entregas de código en los repositorios de desarrollo. Aparte de esto se tuvo una aproximación inicial con todo el proyecto, conociendo su estructura, como se divide en varios componentes, las notaciones usadas para nombrar funciones, variables, archivos, etc.

La primera funcionalidad que fue agregada al proyecto consiste en una barra de búsqueda que permite filtrar entre todas las oficinas, zonas y módulos mostrando únicamente los que coinciden con el termino ingresado para la búsqueda y ordenando los resultados alfabéticamente, ya que en su momento aparecían las opciones en desorden, puesto que provienen de un objeto de JavaScript y estos no ordenan los componentes de la misma manera cada que se crean. Este desarrollo fue de gran ayuda para conocer la distribución de archivos del proyecto y como introducción al desarrollo con las herramientas usadas por parte de la empresa como son VueJs y Bootstrap.

Recopilar datos de las acciones de los usuarios

Pasando a temas directamente relacionados con el proyecto de prácticas empezamos a trabajar con el back-end de Qanty. Analizando el flujo de las peticiones realizadas desde el front hacia el back se recopiló toda la información que llega para cada una de las acciones que se pueden realizar en la herramienta, desde la creación y atención de turnos que son actividades de asesor, hasta la creación y edición de zonas, módulos, usuarios y demás labores de configuración que se llevan a cabo por parte de los administradores, teniendo estos datos para cada una de las acciones los pasos a seguir fueron:

- Definir cuales acciones se iban a registrar en base de datos y cuales no
- Decidir qué información se iba a guardar para cada acción registrada
- Establecer la estructura con la cual se guarda la información para cada registro
- Seleccionar la colección de la base de datos donde se guardaría toda esta información

Luego del anterior análisis se crearon 2 funciones en el back-end, una encargada de registrar actividades de asesor y otra para las de los administradores, cada una de estas escribe la información deseada en la base de datos de manera separada en 2 colecciones diferentes, las cuales serán consultadas según sea necesario dependiendo del tipo de reporte que se solicite desde el front-end.

Rutina para calcular estado de los asesores

Siguiendo con los objetivos planteados se debía calcular el estado de los asesores en todo momento y registrar los cambios de estado en la base de datos para poder contar con un panel de visualización en tiempo real. Para esto trabajamos sobre los archivos que componen la interfaz en la cual el asesor atiende a los usuarios, allí analizando la cantidad de turnos que se encuentran en espera o el hecho de que un asesor se encuentre atendiendo a un cliente podemos definir si este se encuentra en estado de reposo, disponible o productivo.

Un problema que apareció en este punto es el hecho de que debíamos estar pendientes de varios datos al mismo tiempo para poder calcular el estado real y dado que algunos se actualizan más rápido que otros, ocurrían cambios inesperados de estado, por ejemplo: si un asesor se encontraba disponible y solo había un turno en espera para ser atendido, en el momento que se llamaba dicho turno, el nuevo estado debía ser productivo, pero al llegar a cero el contador de turnos en espera se activaba el estado de reposo antes de detectar que el estado era productivo porque se estaba atendiendo a un usuario. Para resolver este tipo de situaciones se creó una función que espera hasta que se termine la ejecución de las funciones que están cargando y se actualicen las variables para poder calcular el estado real a escribir en la base de datos.

Motivos de pausa

Otro estado en el que se pueden encontrar los asesores es el de pausa, para esto fue necesario realizar el montaje completo en los paneles de edición para las configuraciones de la compañía, oficinas, zonas y módulos con el fin de que en cada uno de estos niveles fuera posible configurar motivos de pausa definiendo un nombre para identificarlos y unos límites de tiempo duro y suave teniendo en cuenta la posibilidad de pasar a otro estado denominado como ‘pausa extendida’. También se define un campo de herencia al momento de editar los motivos de pausa brindando la posibilidad de contar con los mismos motivos que ya se encuentran establecidos para un nivel de mayor jerarquía.

Teniendo ya en la base de datos la información correspondiente a los motivos de pausa y la herencia de estos, fue necesario crear una función en el back-end que definiera para cada uno de los módulos cuales eran sus motivos de pausa correspondientes realizando un análisis de jerarquía y herencia, siendo los niveles de jerarquía de mayor a menor: la compañía, las oficinas, las zonas y los módulos. Para determinar cuales son los motivos de pausa que se deben cargar en cada uno de los módulos es necesario analizar si la herencia se encuentra activada en el módulo, en la zona, y la sede en la cual se encuentra ubicado, a parte del análisis de la herencia, si alguna de estas opciones no tiene motivos definidos se cargan los motivos globales de la compañía.

Luego de definir los motivos de pausa correspondientes a cada módulo, se creó en la interfaz de asesor del front-end un botón con el cual los empleados pueden entrar en estado de pausa, dicho botón despliega una ventana en la cual se encuentra un listado de los motivos de pausa que están disponibles y al seleccionar uno de estos se muestra otra ventana en la cual se le informa al asesor cual es el tiempo que se tiene definido para el motivo seleccionado y cuanto falta para que dicho tiempo se cumpla. En caso de terminar la cuenta, si el asesor no ha retornado a sus actividades saliendo del estado de pausa se entra en pausa extendida, cada que se activa cualquiera de estos 2 estados se actualiza en la base de datos para tener dicha información disponible en la tabla de tiempo real.

Panel de visualización en tiempo real

A pesar de que el proyecto ya contaba con una versión inicial de panel de visualización en tiempo real, su funcionamiento se quedaba corto en ocasiones, por lo cual se creó una versión mejorada, para la cual fue necesario crear una nueva colección en la base de datos en la cual se actualiza el estado de cada uno de los módulos de la compañía, siendo el estado cada módulo correspondiente al del asesor que se encuentra atendiendo dentro de él. Desde el front-end se crea un snapshot (copia instantánea) de dicha colección el cual detecta todos los cambios que allí ocurren y con base en estos cambios se van actualizando los estados de los módulos y asesores en el nuevo panel de tiempo real.

Reportes detallados

Un objetivo más que se planteó para el proyecto fue la generación de reportes detallados de usuarios y administradores utilizando la información recopilada y almacenada en base de datos, esta tarea no se ha terminado completamente dado que se están estudiando todos los posibles tipos de reportes que se pueden crear a partir de la información que se guarda y esto conlleva a realizar algunos cambios en la estructura de la base de datos buscando crear un algoritmo robusto y que sea tanto eficiente como escalable a futuro con respecto a los cambios que puedan realizar los administradores de las compañías.

Pruebas y documentación de código

Cabe destacar que durante el desarrollo de cada una de las funcionalidades aquí mencionadas se realizaron pruebas necesarias para garantizar que se cumpliera con los parámetros de funcionamiento planteados por parte de la empresa sin que se genere ningún tipo de errores ni afectaciones con el funcionamiento que tenía el software de Qanty hasta ese momento, dichas pruebas se realizaron desde diferentes perfiles con el fin de realizar actividades tanto de asesor como de administrador.

Todo el código nuevo se iba entregando en repositorios ubicados en GitHub, allí se recibía realimentación por parte del equipo de desarrollo dando sugerencias de cambios a realizar o mencionando posibles errores que se encontraban con el fin de hacer las correcciones correspondientes antes de llevar los nuevos desarrollos a la versión oficial del proyecto en su rama máster.

Todo el código creado durante este proyecto de prácticas es auto-documentado, dando nombres claros a todos los archivos, variables y funciones que se crearon, con el fin de que estos indiquen su finalidad y en general todo el código fuera de fácil entendimiento para cualquier persona del equipo que se encuentre trabajando en Qanty, en algunos casos particulares cuando se consideraba necesario se dejaron algunos comentarios indicando el funcionamiento de partes especificas del código.

Resultados y análisis

Primer acercamiento y conocimiento del proyecto

A continuación, en las figuras 2 y 3 se muestran algunas capturas de qanty donde se aprecia el funcionamiento de la barra de búsqueda creada durante el proceso de adaptación al proyecto y al desarrollo con herramientas nuevas como son VueJs y Bootstrap, todo esto fue de gran ayuda para comprender la distribución de los archivos, la sintaxis para nombrar variables, y en general conocer el funcionamiento de Qanty. Una mejora que se le podría realizar a esta funcionalidad es que no sea sensible a las tildes al momento de buscar ya que por ejemplo si se busca una sede escribiendo la palabra 'Bogota' no mostrará ningún resultado porque dicha sede incluye una tilde en su nombre, está guardada y debe buscarse como 'Bogotá'.

Figura 2. Barra de búsqueda y resultados en orden alfabético.

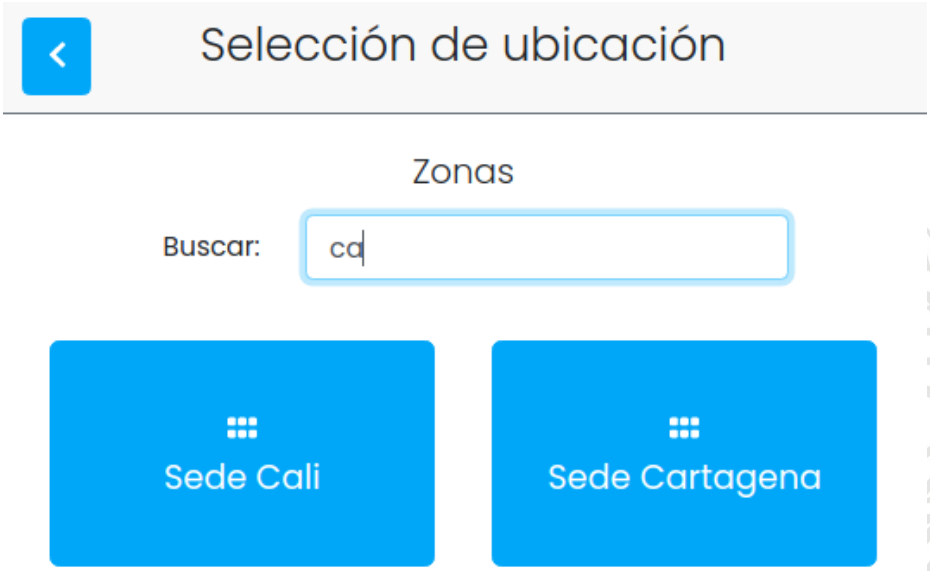

Figura 3. Resultados filtrados al escribir un término de búsqueda.

Recopilar datos

Las funciones de back-end anteriormente mencionadas escriben exitosamente en la base de datos la información que se definió para cada caso y a partir de la cual se pueden realizar acciones de filtrado por fecha, identificador de usuario y otros tipos de identificadores según sea el caso para generar el informe que se desee. En la figura 4 se muestra un ejemplo de como se guarda información en la base de datos al realizar tareas como editar una sede.

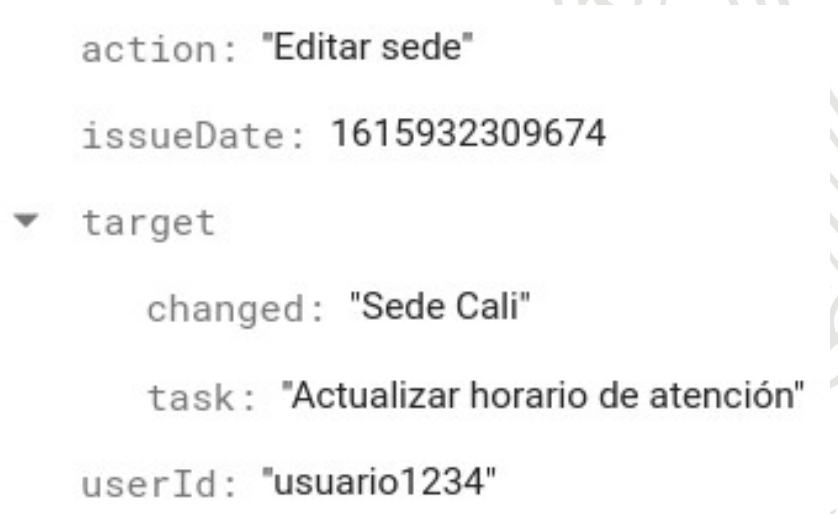

Figura 4. Ejemplo de estructura con la que se guardan los registros en base de datos.

Calcular estado de asesores

Se ha comprobado que la función encargada de calcular en todo momento el estado de los asesores trabaja de manera satisfactoria, dicho estado además de escribirse en base de datos dentro de la colección dedicada para el panel de visualización en tiempo real también se le muestra al asesor en su interfaz de atención a usuarios, como se puede apreciar en la figura 5.

Motivos de pausa

Se entregó el código que permite crear, editar y eliminar motivos de pausa en todos los niveles mencionados que son compañía, sede, zona y módulo. Brindando la posibilidad de heredar los motivos configurados para niveles de jerarquía superior. En la ventana de edición se incluye una breve explicación indicando asuntos como las unidades en que se ingresan los valores de tiempos o que el límite de tiempo suave debe ser menor que el límite de tiempo duro. Además de esto se comprobó que la función del back-end encargada de analizar la herencia para cada nivel si envíe al front-end, para cada módulo, sus motivos de pausa correspondientes de manera satisfactoria.

Figura 5. Interfaz de asesor mostrando el estado en que se encuentra.

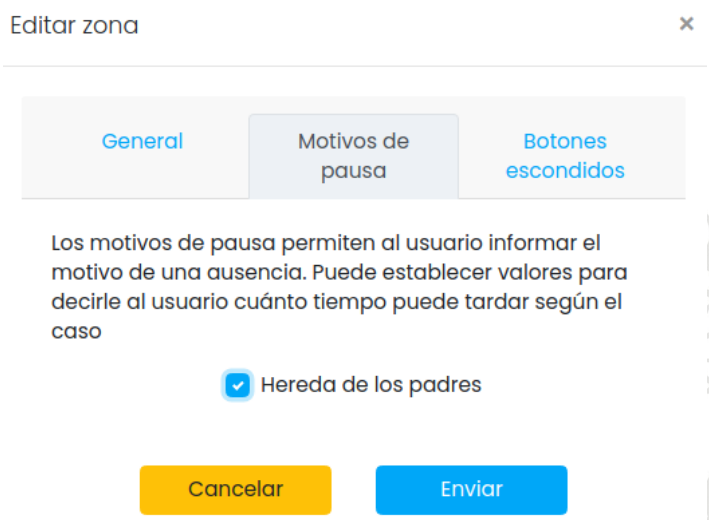

Figura 6. Panel de edición para una zona, incluye motivos de pausa y posibilidad de heredar.

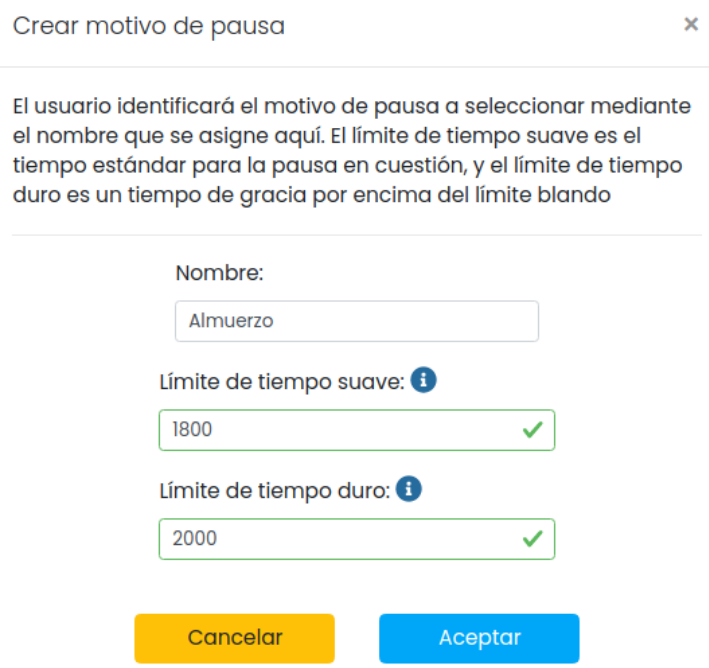

Figura 7. Ventana usada para crear y editar motivos de pausa

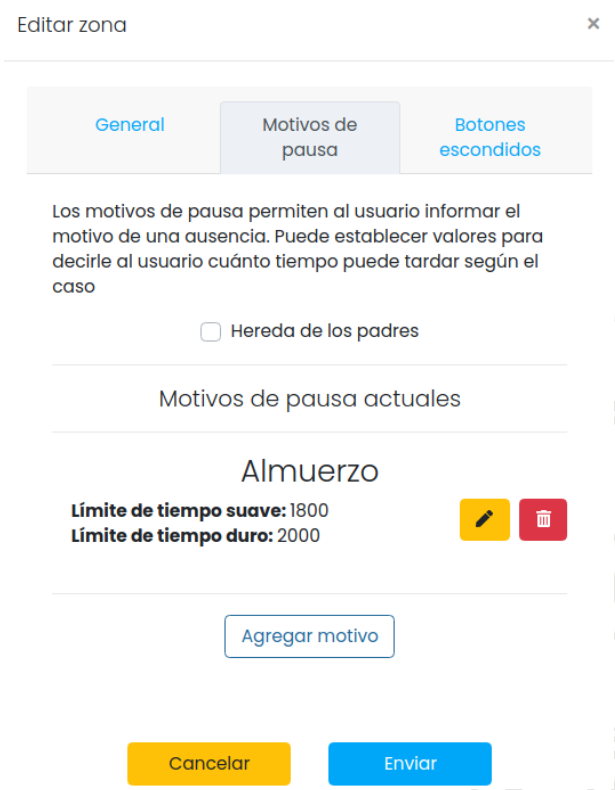

Figura 8. Ventana de configuración con motivos de pausa creados y sin heredar.

Figura 9. Botón de pausa en interfaz de asesor.

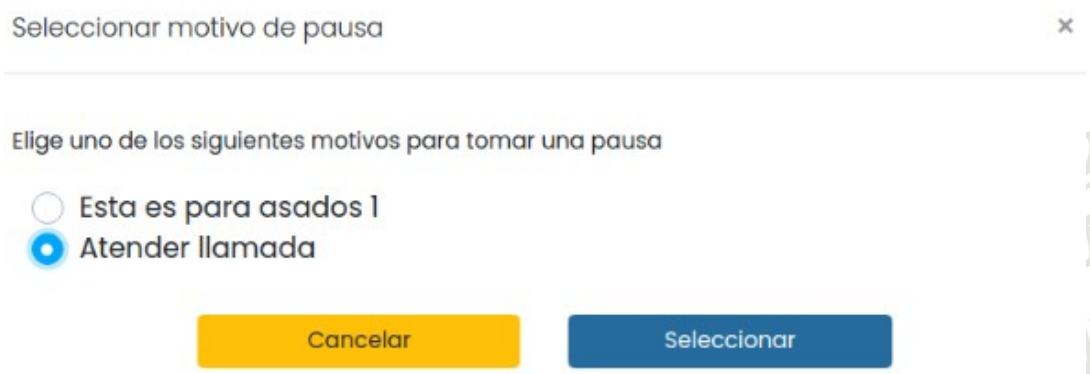

Figura 10. Ventana para seleccionar un motivo de pausa

Figura 11. Ventana que se muestra al asesor indicando información sobre la pausa actual

En las figuras 6, 7 y 8 se presenta el desarrollo que permite la configuración de los motivos de pausa por parte de los administradores. En las figuras 9, 10 y 11 se presentan los cambios realizados en la interfaz de asesor, incluyendo el botón que se usa para entrar en pausa, el menú para seleccionar el motivo y la ventana que muestra información sobre el motivo seleccionado tal como el nombre de la pausa, el tiempo que se tiene establecido para ella y cuánto falta para que dicho tiempo termine, todos los temporizadores funcionan de manera correcta y se muestra la información que se esperaba.

Panel de visualización en tiempo real

Al momento de iniciar la práctica, Qanty ya contaba con una versión inicial del panel para visualizar en tiempo real en qué estado se encontraban los módulos y en caso de estar atendiendo un turno se mostraba el nombre del asesor que estaba realizando tal acción. Dicho panel se quedaba corto en algunos aspectos como el de indicar si un módulo quedaba libre, por ejemplo en la figura 12 se muestra el panel inicial, allí se muestran los módulos de *Carnes 1* y *Pizza 1* con estado disponible, lo cual puede generar confusión con el estado de los asesores ya que un módulo se encuentra disponible si está vacío mientras que un asesor se encuentra disponible si está trabajando y hay turnos que atender.

Otro aspecto que hacía falta en ese panel era tener la posibilidad de saber cuánto tiempo llevaba cada módulo en el estado actual y algo que se prestaba para confusiones era que solo se mostraba el nombre del asesor, de manera que si una sede contaba con varios trabajadores que tuvieran el mismo nombre no podía diferenciar en realidad cual era el estado de cada uno.

Partiendo del panel de la figura 12, analizando sus aspectos a mantener y cuales se debían mejorar se creó un nuevo panel que se muestra en la figura 13. Entre las mejoras realizadas se puede destacar que se muestran todos los módulos que pertenecen a la sede, sin importar que se encuentren vacíos o estén ocupados por asesores realizando su trabajo, se agregó la posibilidad de filtrar las filas que se muestran especificando si se quiere buscar por nombre de asesor, estado o módulo, también se puede filtrar en general por cualquier campo de la tabla. Se agrega el nombre completo del asesor como se encuentre registrado en la compañía, se diferencia el estado disponible de los asesores con los módulos vacíos y se muestra el tiempo que llevan cada módulo y usuario en el estado actual.

Nombre	Estado	usuario	Turno	Fila
Asados 1	Productivo	Santiago	D-1	Asados
Carnes 1	Disponible			
Pizza 1	Disponible			

Figura 12. Versión anterior del panel de tiempo real

Figura 13. Versión actual del panel de tiempo real

Conclusiones

- Se ha cumplido con la mayoría de los objetivos planteados a realizar durante el desarrollo del proyecto de prácticas, quedando sin finalizar la parte de generar reportes detallados de usuarios y administradores, aunque ya se ha trabajado y avanzado en parte de este desarrollo. Para el caso de los demás objetivos como lo son el registro de información en la base de datos, el manejo administrativo de los motivos de pausa, la posibilidad de que los asesores entren en pausa, el cálculo del estado para cada asesor y el panel de visualización en tiempo real los resultados han sido satisfactorios.

- Al momento de trabajar en soluciones de software como Qanty que puede llegar a ser utilizado por diverso tipo de compañías es muy importante tener en cuenta la escalabilidad y robustez de todas las funcionalidades que se tengan pensando en que cada cliente puede darle un uso diferente en cuanto a la cantidad de sedes, zonas y módulos que deseen tener. Otro ejemplo es que cada empresa puede querer manejar de manera diferente la información recopilada para la generación de informes, realizando búsquedas y filtrados diferentes, es por esto y tratando de dar con la mejor solución que no se logró generar los reportes detallados en el plazo establecido para el proyecto de prácticas. Todo el tema de reportes y estadísticas sigue en etapa de desarrollo buscando llegar a un código que cumpla a cabalidad con las necesidades que puedan llegar a tener los clientes.

- Durante la etapa de conocimiento del proyecto qanty se notó la importancia de un código claro y auto-documentado, esto fue de gran ayuda para iniciar con el desarrollo de nuevas funcionalidades, las cuales siguieron con los mismos lineamientos para que en un futuro cuando otra persona llegue a trabajar en el mismo proyecto tenga un acercamiento cómodo y pueda navegar fácilmente entre los diferentes componentes teniendo claridad de que hace cada uno de ellos.

- El tiempo invertido en el desarrollo de este proyecto de prácticas ha sido una gran ayuda como introducción al mundo laboral ya que ha servido para conocer como es el flujo de trabajo en una empresa de tecnología, cuáles son las herramientas que se usan en este momento en el mundo del desarrollo web y otros aspectos que son muy útiles e importantes.

Referencias Bibliográficas

[1] DEVELOPER MOZZILLA [sitio web] HTML: Lenguaje de etiquetas de hipertexto [Consultado: 7 de abril de 2021]. Disponible en: https://devcode.la/blog/que-es-javascript/

[2] DEVCODE [sitio web] ¿Qué es JavaScript? [Consultado: 7 de abril de 2021]. Disponible en: https://devcode.la/blog/que-es-javascript/

[3] LENGUAJECSS [sitio web] ¿Qué es CSS? [Consultado: 9 de abril de 2021]. Disponible en: https://lenguajecss.com/css/introduccion/que-es-css/

[4] FIREBASE [sitio web] cloud firestore [Consultado: 24 de noviembre de 2020]. Disponible en: https://firebase.google.com/docs/firestore?hl=es

[5] FIREBASE [sitio web] Cloud Functions para Firebase [Consultado: 10 de abril de 2020]. Disponible en: https://firebase.google.com/docs/functions?hl=es-419

[6] VUEJS [sitio web] qué es Vue.js [Consultado: 24 de noviembre de 2020]. Disponible en: https://es.vuejs.org/v2/guide/

[7] UN POCO DE JAVA [sitio web] ¿Qué es vuex? [Consultado:10 de abril 2021]. Disponible en: https://unpocodejava.com/2019/05/09/que-es-vuex/

[8] ROCKCONTENT [sitio web] bootstrap, guia para principiantes de que es, porque y como usarlo. [Consultado: 24 de noviembre de 2020]. Disponible en: https://rockcontent.com/es/blog/bootstrap/