EXPERIENCIA DE USUARIO SEGURA Y PRIVADA

Ingeniería civil informática

Ace Fermandois Maack
Blueng Ingenieros
Enero de 2023

Contenido

1 Resumen ejecutivo

1.1 Español

El proyecto llevado a cabo por Blueng Ingenieros para la empresa Yo Me Controlo (YMC) se centró en la mejora su página web de medicina preventiva, con un enfoque particular en el front-end. La meta principal era hacer la plataforma más amigable para el usuario sin comprometer la privacidad de los datos.

El objetivo principal era garantizar que la página fuera amigable para el usuario, principalmente en el proceso de recopilación de información médica mediante encuestas. Se destacó la importancia de una interfaz fácil de entender para evitar malentendidos y asegurar la obtención de información precisa.

La metodología utilizada incluyó el uso de ReactJS y JavaScript para lograr la responsividad, así como la herramienta PageSpeed de Google para evaluar el rendimiento del sitio. Del mismo modo, se establecieron medidas de desempeño, considerando estadísticas que indican la relación entre la velocidad de carga y la satisfacción del usuario.

En la sección de propuestas de solución, se exploraron diferentes enfoques, desde el uso de formularios externos hasta el desarrollo de una aplicación móvil. La solución elegida involucra mejoras en la encuesta en la página, asegurando que los datos del usuario se utilicen solo para el algoritmo de exámenes y que la información esté disponible solo cuando el usuario lo permita.

En cuanto a la evaluación económica, se realizaron análisis optimistas y pesimistas, considerando el valor de las órdenes médicas y los exámenes recomendados. La viabilidad del proyecto se estableció al menos 30 nuevos usuarios por año. Además, Se abordaron riesgos potenciales mediante una matriz de riesgos, identificando amenazas como la inyección de código malicioso, componentes vulnerables y desactualizados, y posibles fallos en la página.

La implementación del proyecto incluyó el uso de React Bootstrap inicialmente para agilizar el desarrollo, aunque posteriormente se optó por componentes puramente desarrollados en ReactJS para una mayor personalización. Se detalló la implementación de la autenticación de dos factores y la funcionalidad de búsqueda de usuarios. Los resultados mostraron una calificación del 88% en PageSpeed, con tiempos de carga considerados rápidos.

En conclusión, el proyecto logró satisfactoriamente los objetivos establecidos, mejorando la interfaz de usuario, asegurando la privacidad de los datos, y proporcionando una experiencia eficiente para los usuarios. Se destacó la importancia de mantener y refactorizar el código para futuras mejoras.

1.2 Inglés

The project carried out by Blueng Ingenieros for the company Yo Me Controlo (YMC) focused on improving its preventive medicine website, with focus on the front-end. The main goal was to make the platform more user-friendly without compromising data privacy.

The main goal was to ensure that the page was user-friendly, especially in the process of collecting medical information through surveys. The importance of an easy-to-understand interface was emphasized to avoid misunderstandings and ensure the accurate collection of information.

The methodology used included the use of ReactJS and JavaScript to achieve responsiveness, as well as Google's PageSpeed tool to evaluate site performance. Likewise, performance measures were established, considering statistics that indicate the relationship between loading speed and user satisfaction.

In the section about proposed solutions, different approaches were explored, from the use of external forms to the development of a mobile application. The chosen solution involves improvements to the on-page survey, ensuring that user data is used only for the exam algorithm and that information is available only when the user allows it.

Regarding the economic evaluation, optimistic and pessimistic analyses were carried out, considering the value of medical orders and recommended examinations. The viability of the project was established at least 30 new users per year. Additionally, potential risks were addressed using a risk matrix, identifying threats such as malicious code injection, vulnerable and outdated components, and potential page crashes.

The project implementation included the initial use of React Bootstrap to speed-up development, although later, purely ReactJS-developed components were chosen for greater customization. The implementation of two-factor authentication and user search functionality was detailed. The results showed a PageSpeed rating of 88%, with loading times considered fast.

In conclusion, the project successfully achieved the established objectives, improving the user interface, ensuring data privacy, and providing an efficient experience for users. The importance of maintaining and refactoring the code for future improvements was emphasized.

2 Introducción

La empresa Yo Me Controlo (YMC) es una empresa de medicina preventiva, la cual se usa para identificar condiciones crónicas como la diabetes o problemas cardíacos. En este contexto, es esencial llevar a cabo pruebas de detección, además de adoptar prácticas saludables, como llevar una alimentación adecuada, realizar actividad física y no fumar (Xercavins Comas 2014).

Blueng Ingenieros, empresa consultora de software, fue contratada por YMC para generarles una página web que tenga el algoritmo necesario para que pacientes puedan acceder a sus servicios en línea. YMC ya tenía una página, pero esta estaba en WordPress, así que solicitaron una nueva de cero, con nuevo *front-end*[1], nuevo *back-end*[2] y nueva base de datos, el enfoque de este proyecto es en el lado *front-end*.

Para acceder al servicio, es necesario que los usuarios contesten algunas preguntas relacionadas con su historial médico y el de sus familiares. Es importante estas preguntas estén bien formuladas, y que la interfaz de usuario sea fácil de entender, con el fin de garantizar que las respuestas reflejen de manera precisa la realidad. Esto contribuirá a que el diagnóstico se realice con un mínimo margen de error.

Es importante la obtención del historial médico ya que brinda un mejor entendimiento del paciente para poder diagnosticar sus enfermedades de mejor manera. Un ejemplo relevante de esto se observa en el caso del cáncer genético, que, aunque no es hereditario en sí, hereda los cambios genéticos. Un 10% de los casos de cáncer son causado por cambio genético heredado. Por esto, capturar bien el historial médico del paciente es de mayor importancia ya que si no se encuentra puede llegar a ser letal.

Por eso, surge un problema con la interfaz de usuario que no es tan fácil de usar. Al depender demasiado de que los usuarios entiendan los términos, hay margen para errores. Pero, si hay alguna confusión, especialmente en cosas importantes como el historial médico, se podría llegar a errores graves. Es crucial tener una interfaz amigable y comprensible para evitar malentendidos y asegurarse de obtener información precisa.

Es importante resolver problemas como estos, sobre la interfaz de usuario, ya que influye en la experiencia de usuario. Contar con un *front-end* bien diseñado y mejorado es fundamental, ya que hace que el sitio sea intuitivo y fácil de navegar, proporcionando así una experiencia de usuario mejorada y aumentando las posibilidades de conversiones. Además, un buen *front-end* es importante porque provee un sitio web rápido, así evitando una mayor tasa de rebote. (2023 Zarttech).

[1] En desarrollo web, la capa de presentación. También conocido como interfaz de usuario.
[2] En desarrollo web, la capa de datos. También conocido como el servidor.

2.1 Objetivos

2.1.1 Objetivo general

Modificar la página de Yo Me Controlo de manera que sea amigable para el usuario sin vulnerar la privacidad del usuario y sus datos. Esto deberá quedar listo a final de año.

2.1.2 Objetivos específicos

Con la mayoría de los usuarios navegando por internet con sus dispositivos móviles, es de suma importancia tener un sitio que se adapte a distintos dispositivos. Así, una buena experiencia de usuario garantiza que la página se ajuste a todos los dispositivos y navegadores, brindando a los usuarios una experiencia uniforme, independientemente de la forma en que accedan al sitio (Zartech 2023). Por ende, uno de los principales objetivos del proyecto es lograr la responsividad, es decir, asegurar que el sitio se ajuste a diferentes tamaños de pantalla sin perder funcionalidad.

Además, el artículo *"The value of UX Design"* de Emily Stevens reúne estadísticas sobre la importancia de una buena interfaz de usuario, estas se encuentran traducidas a continuación:

- *Una interfaz de usuario bien diseñada puede aumentar la tasa de conversión de un sitio web hasta un 200% (Fuente: Forrester Research).*
- *Mejorar el diseño de la experiencia del usuario de un sitio web puede llevar a una tasa de conversión tan alta como el 400% (Fuente: Forrester Research).*

Estas cifras destacan la relevancia de una experiencia de usuario positiva y el valor que se traduce en métricas concretas. Por lo tanto, el próximo objetivo principal se centra en mejorar los tiempos de respuesta para la carga de perfiles y la generación de solicitudes, reconociendo así la importancia de una experiencia de usuario eficiente.

Por otro lado, La Ley de Portabilidad y Responsabilidad del Seguro Médico, conocida por sus siglas en inglés como HIPAA *(Health Insurance Portability and Accountability Act*), se encarga de asegurar la privacidad y la confidencialidad de la información de salud de los pacientes, así como garantizar la portabilidad y la continuidad del seguro médico. Por esto, al desarrollar una página web sobre datos médicos es necesario adherirse a las leyes HIPAA.

2.2 Medidas de desempeño:

Como hay una gran cantidad de diferentes dispositivos y pantallas, Microsoft recomienda seleccionar algunas categorías clave, conocidas como *breakpoints*, y diseñar la página web teniéndolas en cuenta en lugar de intentar optimizar la interfaz para cada dispositivo específico. Microsoft habla de tres

grandes categorías: Pequeño con hasta 640px de ancho, mediano entre 641px y 1007px, y grande con mínimo 1008px. (Microsoft 2023)

Nombre	Ancho en pixeles
Laptop L	1440px
Laptop	1080px
Tablet	768px
Mobile L	420px
Mobile S	320px

Tabla 1 Anchos de pantalla de breakpoints elegidos.

Para medir si el proyecto afecta negativamente a la experiencia de usuario, se medirá el tiempo de carga de cada funcionalidad agregada. Para decidir el tiempo se usaron las estadísticas mencionadas por Sourojit Das en BrowserStack.

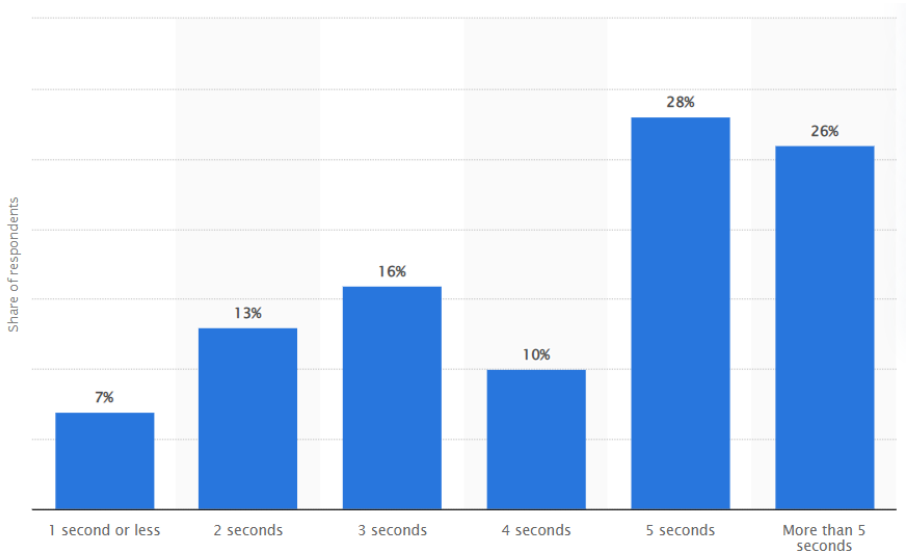


Gráfico 1 Pérdida de usuarios según tiempos de espera para la carga de la página.

Según Das:

- 1 de cada 4 visitantes abandonaría un sitio web que tarda más de 4 segundos en cargar.

- Cada segundo de retraso reduce la satisfacción del usuario en un 16%.

- El 64% de los compradores insatisfechos no vuelven a visitar un sitio web lento.

- El setenta por ciento de los consumidores indican que la velocidad de la página influye en sus decisiones de compra en tiendas en línea.

- Cada segundo adicional de tiempo de carga disminuye las tasas de conversión en un promedio del 4.42% (en los primeros cinco segundos).

(Das 2023)

El Gráfico 1 revela un aumento significativo en la tasa de deserción a partir de los 5 segundos de tiempo de carga. Además, en la misma página, se proporciona una tabla con los tiempos de carga promedio específicos para cada sector de la industria. En particular, el sector de atención médica tiene un tiempo promedio de carga de 5.6 segundos, justo después del mencionado incremento. Con base en estos datos, se concluyó que mantener un tiempo de carga inferior a 5 segundos sería fundamental para preservar una experiencia de usuario óptima.

Sector de la industria	Tiempo de carga promedio
Venta minorista de automóviles	6 segundos
Bienes de consumo empaquetados	6,1 segundos
Financieros	5,1 segundos
Atención médica	5,6 segundos
Medios de comunicación	5,5 segundos
Minoristas	6 segundos
Tecnología	6,8 segundos
Viajes	6,7 segundos

Tabla 2 Tiempo de carga promedio según sector de la industria (Elaborada con información del artículo de Das, 2023)

2.3 Metodologías

La responsividad se alcanza mediante la implementación de diversas funcionalidades proporcionadas por Reactjs y JavaScript. En este proceso, se definen los *breakpoints*, que corresponden a los distintos tamaños de pantalla mencionados, y se utilizan en las *mediaQueries*[3] de CSS. Estas *mediaQueries* permiten ajustar el diseño de la página en función de estos tamaños específicos.

El diseño de la página se desarrolla de manera relativa en estos diferentes tamaños. Se emplea como vh y vw (con respecto al alto y ancho de la página, respectivamente) y rem (en relación con el tamaño de fuente y el zoom).

En situaciones en las que el diseño original ya no sea adecuado debido a un cambio de tamaño significativo, el diseño se adapta. Esto se logra mediante la implementación de un componente diferente

[3] Función de CSS que ajusta el contenido según el tamaño de la página.

en el archivo JavaScript, diseñado para funcionar de manera óptima con el nuevo tamaño, garantizando así una experiencia de usuario fluida en todas las dimensiones de pantalla.

Para asegurarse cumplir la ley de tener controles de acceso para prevenir que personas no autorizadas accedan a la información de salud del paciente, se utilizará confirmación de dos factores[4] con fin de revisar que el médico realmente tiene permiso. El primer factor sería la contraseña del médico, y el segundo un código de uso único (OTP o One Time Password en inglés).

Para los controles de integridad se tendrá una copia de seguridad para revisar que no hayan cambiado los datos cuando no deberían haber cambiado. Por eso, van a haber registros de quién revisa, edita, o agrega nueva información a la base de datos. Los datos de salud estarán encriptados en la base de datos para que no puedan ser accedidos si logran acceder a la base de datos, así solo se van a poder revisar si es que se pasa por el doble factor.

Para alcanzar el último objetivo, centrado en la experiencia del usuario, se recurrirá a la herramienta PageSpeed de Google. Esta plataforma proporciona cuatro evaluaciones clave: rendimiento, accesibilidad, prácticas recomendadas y Optimización para Motores de Búsqueda *(Search Engine Optimization, SEO)*. El objetivo es obtener una puntuación que se encuentre entre 50 y 100, calificando así como un resultado satisfactorio en todas las áreas evaluadas (Google s.f.)

[4] Método de seguridad que requiere dos llaves diferentes para confirmar identidad.

3 Estado del arte y propuestas de solución

Para encontrar una solución, se exploraron diversas estrategias para abordar el dilema de la privacidad en las encuestas de salud.

En primer lugar, se identificaron formularios externos dedicadas a la captura y entrega de información directamente con el backend de la página, unos ejemplos son Google Forms y Microsoft Forms. También existen plataformas que ofrecen formularios especializados en la recopilación de datos médicos, destacando en la seguridad y privacidad de la información del usuario, como Jotform y Formsite.

La segunda alternativa consiste en mantener las preguntas dentro de la página web, restringiendo su acceso a todos excepto al algoritmo, que puede revisarlas y actuar en consecuencia. Esta medida no solo garantiza la conformidad con HIPAA, sino que también reduce la necesidad de conexiones adicionales.

Por último, se exploraron aplicaciones que se conectan directamente a los datos de salud del usuario, como el caso de "Health" para usuarios de iOS. Estas aplicaciones se emplean para evitar la repetición de preguntas sobre datos personales, requiriendo únicamente la confirmación del usuario para compartir sus datos o no.

Basándose en las diversas opciones exploradas, se han propuesto tres soluciones específicas: El uso de un servicio de formulario en línea, el desarrollo de una aplicación móvil y la incorporación de una encuesta directamente en la propia página.

La primera se concentraría en buscar un formulario que tenga el tratado BAA[5] (Business associate agreement) y que sea conforme con HIPAA, se tomaron en consideración los dos ejemplos de formularios dedicados a la recopilación de datos médicos.

La segunda se conectaría a la aplicación de salud del dispositivo móvil del usuario para registrar, por lado de la empresa, los datos médicos. De esta manera, el usuario no tendría que ingresar por su cuenta sus hábitos y/o condiciones, pues se medirían directo.

La tercera sería una encuesta en la página propia de Yo Me Controlo. La empresa ya tenía una encuesta, pero, como mencionado junto con el problema, contaba con el conocimiento médico del usuario, que puede que este no tenga, así que la solución acá no es hacer una encuesta, sino que

[5] Acuerdo entre entidad médica y socio comercial que establece las reglas del manejo de datos médicos.

extenderla para hacerla más específica. La encuesta propuesta preguntaría sobre hábitos y por detrás establecería qué significa.

4 Solución escogida

Primero se evaluó la seguridad de los Formularios que dicen ser conforme con Hipaa. Se determinó que logran cumplir con HIPAA, pero se descartó ya que las opciones eran demasiados costosas y su enfoque era para empresas de mayor tamaño. Las dos opciones costaban 1548 dólares al año Jotform, o 3000 dólares al año Formsite, como Yo Me Controlo es una empresa pequeña no logra cubrir estos gastos extras.

Segundo, se evaluó la propuesta de la aplicación. Para proponer esta solución a la empresa se hizo un diseño de cómo sería, y cómo se conectaría. Desafortunadamente, no les gustó parte de la propuesta, entendieron que había un problema, pero no consideraban que hacer una aplicación móvil sea una prioridad, así que pidieron resolverlo en la Web App que ya se les estaba desarrollando.

Después de hacer un análisis de privacidad, se encontró un problema. La empresa en su encuesta pide los datos, pero no deberían quedarse con los datos, pues esto son propiedad del usuario, y como son suyos, este decide cuándo entregarlos. Así, se descartó por completo la idea de medir constantemente, pues esto no deja al usuario a cargo de sus datos, sino a la empresa.

Para corregir esto, se pasó a la segunda propuesta, la empresa pedirá los datos mediante la encuesta y solo se usarán para el algoritmo de exámenes, sin mostrárselos a los médicos. Si el usuario nunca se registra, sus datos nunca se guardan, pero si este desea agilizar el proceso para futuras veces, puede dejarlos registrado.

Además, para dejar abierta la opción de telemedicina, donde el médico necesitará revisar los datos, deberá solicitar acceso a estos directo al paciente, esto para mantenerse dentro de los estándares HIPAA, específicamente:

> *HIPAA*-covered entities must ensure there are access controls in place to prevent unauthorized individuals from gaining access to PHI or to the administration control panel. (Alder 2022)

Cuando el médico solicite acceso a los datos del usuario este recibirá una notificación por correo electrónico informándole sobre la solicitud. En este correo se incluirá una OTP, para acceder a los datos el médico deberá ingresar este código. Este proceso garantiza que el acceso a la información del paciente esté debidamente autorizado y protegido.

Este enfoque implica ajustes en el proyecto, como agregar perfiles de médico, modificar el perfil de usuario actual y encuestas más detalladas, e integrar funciones como OTP y solicitud de datos. Estas medidas aseguran seguridad y conformidad con HIPAA, estableciendo una base sólida para la telemedicina.

5 Evaluación Económica

También, se hizo un análisis económico[6], con un caso optimista y un caso pesimista. No importar el caso, se asume que cada usuario compra una la orden médica, con un valor de $3000, y los exámenes recomendados por su edad, con un valor de $3000 también, teniendo un total de compra de $6000 pesos.

En el caso optimista, se asume que cada persona que usó la página, se la recomienda a dos amigos o conocidos, y estos empiezan a usarla el siguiente año, y así. Con estos datos, el VAN queda cerca de 7 millones, con una TIR de 305%.

En el caso pesimista, se asume que solo una persona nueva usa el servicio cada mes, sumando un total de 12 extra cada año únicamente. Con estos datos, el VAN queda negativa, con un valor cerca de nueve millones, y con una TIR negativa también, de -11.9%.

Al modificar la cantidad de usuarios que se agregan al mes se encontró que será necesario que al menos 30 usuarios nuevos deben usar la página para tener un VAN positivo por lo haría el proyecto viable.

[6] Revisar en Evaluación económica.xlsx

6 Matriz de riesgos

Al implementar este proyecto, podrían ocurrir accidentes o riesgos de seguridad. Por esto se hizo una matriz de riesgo. Los cuatros principales son:

- Inyección de código malicioso.
- Componentes vulnerables y desactualizados.
- Pagina deja de funcionar.
- Retraso en el desarrollo.

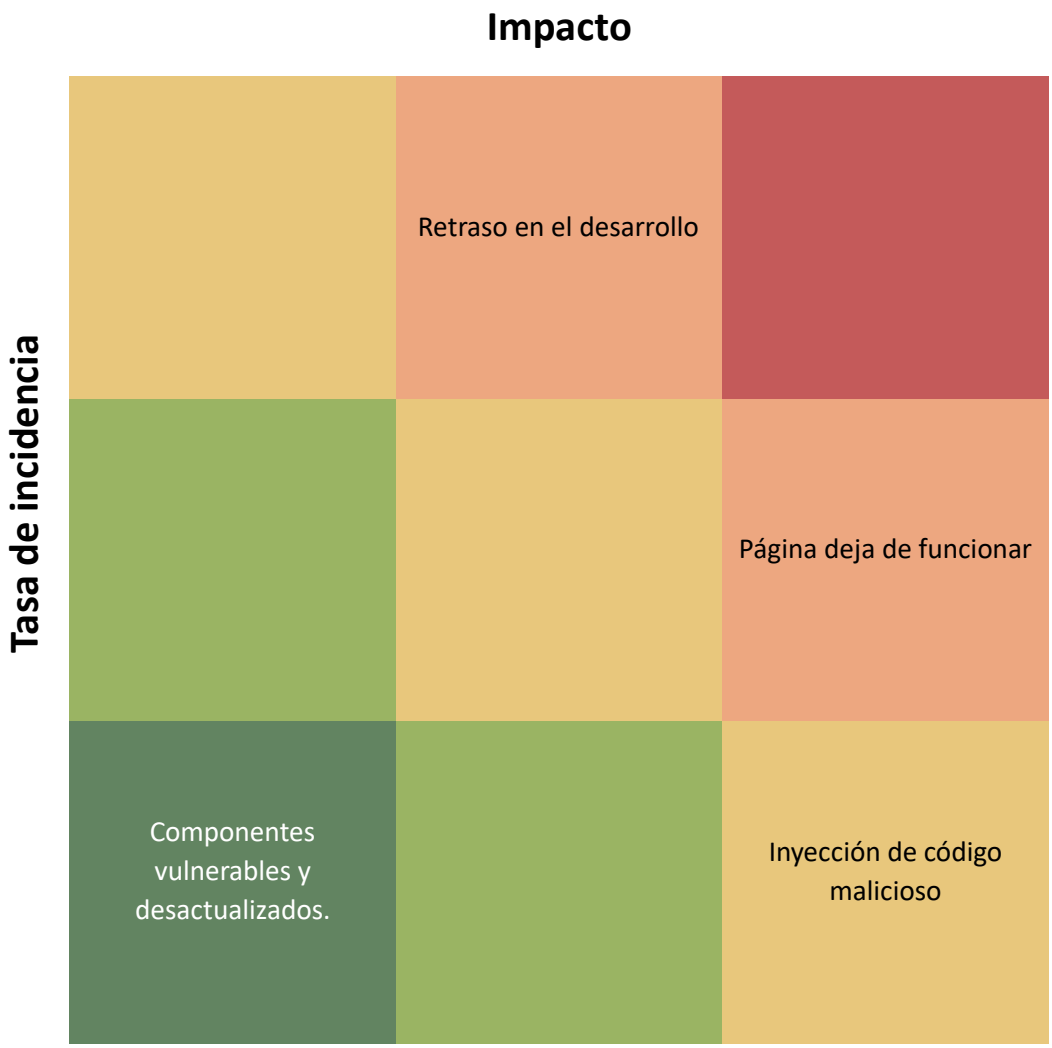


Tabla 3 Matriz de riesgo

Para temas de ciberseguridad la información de tasa de incidencia e impacto se obtuvo de la lista de top 10 vulnerabilidades 2021 del Proyecto Abierto de Seguridad en Aplicaciones Web (OWASP por sus

siglas en inglés). La fundación ofrece una lista de factores, entre ellos, la tasa de incidencia, y el impacto ponderado promedio. (OWASP 2023)

El primer riesgo, inyección, se refiere a la inserción de código en la página de manera que se manden consultas, en este caso de SQL, con intenciones maliciosas. Como SQL corre los comandos que se le entreguen, los filtros se tienen que hacer de antes, y si estos no están bien hechos, pueden sufrir de inyección SQL. (Microsoft, 2023) Esta se mitiga poniendo filtros en los campos de escritura abierta, para ver ejemplo de filtros usados ver anexo 11.3, el filtro de correo se encuentra en 11.1.

Para ver fuente de impacto e incidencia ver Tabla 3, provista por OWASP.

CWEs Mapped	Max Incidence Rate	Avg Incidence Rate	Avg Weighted Exploit	Avg Weighted Impact	Max Coverage	Avg Coverage	Total Occurrences	Total CVEs
33	19.09%	3.37%	7.25	7.15	94.04%	47.90%	274,228	32,078

Tabla 4 Impacto e índice de ocurrencia Inyección

El segundo, componentes desactualizados, se refiere a cuando un código, abierto o de propiedad, contiene vulnerabilidades o ya no se mantiene. Este tipo de vulnerabilidades generalmente se encuentra en formato de bibliotecas. Este tipo de código se suele utilizar sin consideraciones por la seguridad, y eso lleva a brechas de seguridad. Este se mitiga eliminando componentes que no son necesarios, para no depender de sus actualizaciones de seguridad, y siempre manteniendo en la última versión los que son indispensables.

Para ver fuente de impacto e incidencia ver Tabla 4, provista por OWASP.

CWEs Mapped	Max Incidence Rate	Avg Incidence Rate	Max Coverage	Avg Coverage	Avg Weighted Exploit	Avg Weighted Impact	Total Occurrences	Total CVEs
3	27.96%	8.77%	51.78%	22.47%	5.00	5.00	30,457	0

Tabla 5 Impacto e índice de ocurrencia

El tercero, página deja de funcionar, se refiere a cuando ocurre un error que deja a los usuarios inhabilitados para seguir con el proceso de la página, si esto ocurre se tiene que llamar al equipo de TI que se encargaría en solucionar el problema. Para mitigarlo se generará *un SLA (Service Level Agreement*), esto

es un acuerdo que se hace con el cliente en donde se define los tiempos que la página puede estar abajo y establece un tiempo dedicado para mantención de la página.

El último, atraso en desarrollo, tiene una tasa de incidencia alta, pues es bien fácil que una tarea no se termine cuando se esperaba, puede que el equipo subestime el tiempo que tomará una tarea, o que salgan problemas en el camino. El impacto ed bajo-medio, pues si ocurre unavez no es tanto problema, pero si ocurre repetidamente, puede hacer que toda la producción se atrase.

Para mitigar esto se usa el método ágil SCRUM, donde se planea lo que se hará por las siguientes dos semanas, ajustandolo al inicio de la segunda si es necesario, y diariamente se revisa como va el equipo y si se necesita algo para que no se atrasen.

7 Implementación del proyecto

Al inicio querían un demo de solo *front-end* en un corto tiempo así que se tenía que hacer rápido, por este motivo se usó la biblioteca[7] de React Bootstrap, para agilizar el proceso.

Bootstrap tenía un solo problema, este es muy poco personalizable. Estos no son modificables. A algunos se les pueden cambiar cosas mínimas como el color del texto o el fondo, pero para eventos como *hover* o active se mantenía el original, haciendo que los colores no coincidan.

Como el primer diseño de *front-end* era azul con naranjo[8], era suficiente en un inicio. Pero cuando pidieron modificarlo, ya fue difícil mantener Bootstrap, así que, como el nuevo diseño se hizo desde cero, se dejó Bootstrap a un lado y se hicieron los componentes reutilizables que se usaban con pure Reactjs[9].

Además, algo principal en el cambio de Bootstrap fue cambiar los forms que venían prehechos, a un Reactjs puro. Hay 3 partes que incluir en un Formulario. La primera son los campos de input, la segunda son los filtros para validar que lo ingresado está bien, y la tercera es el mensaje de error para cuando no está validado.

La primera es simple, pues el input es un componente básico de cualquier HTML[10], Esto se guarda en una variable con *useState*[11] y se va actualizando con cada cambio que hace el usuario. La función que actualiza la variable hace el filtro para evitar inyección de código y al mismo tiempo confirmar de que está correcto.

Para la validación se usa un diccionario[12] con cada input como llave, y como valor se pone la palabra “valid” en un inicio, y si la función encuentra que no es válido se cambia por la palabra "invalid”. Esto se conecta con la tercera parte, se evalúa la llave respectiva, si es invalid se muestra el mensaje de error, si no, no se muestra. Al mismo tiempo, el borde del input se pone rojo si es inválido o se deja color original si es válido.

---

[7] Conjunto de funciones para un lenguaje específico.
[8] Para revisar diseño anterior y diseño actual ir a https://alumnosuaicl-my.sharepoint.com/:f:/g/personal/cfermandois_alumnos_uai_cl/EgIRrsaSJVNDnVmJJtW36K0BnEBfhfqe9F3lgP1E191jwg?e=vgIJ1R.
[9] Extracto de componentes reutilizables en anexo 11.4
[10] De sus siglas en inglés HyperText Markup Language.
[11] Hook de Reactjs utilizada para cambiar el estado de una variable dentro de una función.
[12] Tipo de variable, es una colección de valores con una llave asociada para acceder a este.

Usando archivos context[13] se conecta la información entregada por el usuario. La empresa nos entregó un archivo Excel con una tabla que tenía el nombre del examen, los antecedentes familiares que tenía que cumplir el género, la edad requerida, y los antecedentes personales que tenía que cumplir. De esta manera:

Examen	Edad	Personal	Familiar	Género
Nombre	15	[1, 2, 3]	[14, 15]	2

Tabla 6 Ejemplo de requisitos de un examen

Con esto, dentro del archivo *context*, se hizo un diccionario con los códigos, uno para los antecedentes familiares y uno para los personales, que seguían la siguiente lógica:

Códigos = {“antecedente”: 1 … “último”: n}[14]

La forma de revisar si el usuario cumplía con los antecedentes era: la lista de antecedentes se cambiaba de palabras a código, usando el diccionario mencionado, y luego se juntaban ambos, personal y familiar. De esta manera la lista se vería igual a la del examen y se podía usar la función *.include*[15] para para filtrar los que le correspondían al usuario.

Antes de pasar la lista de exámenes por el filtro de antecedentes, se pasaba por el filtro de género y luego por el de edad. Con el de edad se hacía una lista aparte ya que eran dos paquetes diferentes.

Para hacer la página de compra se hicieron dos archivos, uno donde salen los paquetes disponibles, y otro donde salen los que ya se seleccionaron.

Los paquetes parten colapsados, mostrando solo el título y el número de exámenes, Y si el usuario hace clic en la flecha a la derecha se expande y ve lo que les corresponde.

Esto se logró usando un diccionario. Como ya ve está definido el nombre de la sección, y en el valor está la palabra “visible” o “colapse”. Si es colapse la flecha está rotada 90° hacia adentro, si no, apunta hacia abajo.

Para logra que el botón abra o cierre según corresponda, se pusieron dos botones separados uno solo se muestra si está visible y el otro si no está visible, de esta manera cada uno tiene una función diferente que hace que cambie el valor de “visible” a “colapse” y viceversa.

[13] Archivo para guardar variables que son globales sin necesidad de pasarlos por jerarquía.
[14] Revisar anexo 11.2 para ver los diccionarios, y cómo se crearon los requerimientos de los exámenes.
[15] Función para revisar si una lista incluye lo mencionado, retorna verdadero o falso.

Para agregar paquetes a la cesta, el usuario debe apretar en el botón morado de la derecha punto del mismo modo que para colapsar hay un diccionario con las llaves según las secciones, Pero esta vez los valores son "*InBasket*" o "*OutBasket*" según corresponda, todos los valores parten en "*OutBasket*" y cuando hacen clic se cambia a "*InBasket*", y el botón se cambia a una ganaste con un signo menos, para indicar que ahora es botón para eliminar, que lo cambiaría de vuelta a "*OutBasket*".

En el segundo archivo también están con la funcionalidad de colapsar / mostrar, pero solo se muestra los "*InBasket*" y, por eso, solo el botón de eliminar. En este se calcula el precio tomando las llaves y multiplicándolas por el número correspondiente, así se actualiza a tiempo real. Una vez se hace la compra, se genera la orden con los exámenes que estaban en la lista, y la compra está terminada. Para ver código de estas dos funcionalidades, colapsar o mostrar paquetes y agregar o quitar elementos, ver anexo 11.1.

Lamentablemente, eso no fue todo, la empresa solicitó cambiar la forma de hacer el filtro y ya no era que tengan todo lo de la lista como era que tuvieran al menos uno, así que junto con *include* se usó un *Some*[16].

Después, solicitaron cambios nuevamente, así que se tuvieron que hacer unos filtros más específicos. Estos filtros ya iban siendo según examen.

Para implementar la seguridad de los datos, se aplicó la autentificación de dos factores, específicamente en los perfiles y en la solicitud de datos.

Cuando el usuario que inicia sesión tiene el rol "*medic*", la barra superior cambia para darle acceso su perfil de médico, donde puede cambiar sus datos personales. Esta página parte en modo solo lectura, pero si aprieta editar datos personales, los campos de nombre, apellido, y correo, se vuelven editables. En un futuro, si la página la amplían a más países e idiomas, se volverán editables también.

Cuando el usuario quiere cambiar los datos, le aparece un modal para que ingrese su contraseña. Cuando esta es confirmada, se envía un código a su correo de único uso. Este dura 5 minutos, y una vez pasan, se tiene que enviar otro código. Una vez el código es aprobado, los datos se modifican, y aparece un modal confirmando el cambio de datos. Para el código de este modal revisar anexo 11.6.

Lo mismo ocurre para el buscador de usuario, pero en este caso el correo se envía al paciente dueño de los datos, esto para que el paciente decida cuando entregar su información.

[16] Función para revisar que al menos uno se cumpla.

Para desarrollar este buscador de usuario, se hicieron dos pestañas, una para buscar por rut, y una para buscar por apellido. Por una API[17] se obtienen los usuarios y se despliegan en una lista. Al seleccionar un nombre aparece la información bloqueada a la derecha hasta que se tenga acceso.

Para hacer este buscador responsivo se necesitó más que un CSS con mediaQueries, pues como tenía doble columna, si se mostraba en dispositivos móviles no muy anchos, o ventanas de ancho menor los componentes no se alcanzan a ver bien. Para esto se usó lo que se mencionó en la metodología, en el archivo JavaScript se usaron operadores ternarios para revisar el ancho de la ventana y poner el componente necesario para ese tamaño. Se usó un event listener[18] para poder estar al tanto de cambios de ancho en la ventana. Para ver el código de responsividad ver anexo 11.5.

[17] Application Programming Interface, sigla en inglés.

[18] es una función que espera y responde a eventos específicos, en este caso, el cambio de tamaño de la ventana.

8 Resultados:

Se uso la página de PageSpeed para revisar el tiempo de carga necesario, obteniendo una calificación del 88% en rendimiento, dentro del rango esperado. Además, proporcionó información sobre el tiempo que tarda el usuario en visualizar la información, el cual fue de 0,9 segundos, calificado como rápido.

Solicitud	Tiempo
Carga de datos perfil médico	1-1.4s
Verificación de contraseña	0.5-1s
Envío de OTP	1.9s
Búsqueda de usuario	915ms

Tabla 7 Resultado de medición de tiempo de carga para cada solicitud

9 Conclusión

En este proyecto, la principal preocupación era cumplir con HIPAA, especialmente implementar el control de acceso pues el resto de las reglas se cumplían por el equipo de base de datos, o el equipo de backend. Así que el proyecto se considera logrado con la implementación de los dos factores.

De igual manera, los proyectos siempre se pueden continuar, sobre todo mantener. La refactorización del código ayuda a que este esté mejor hecho, pero no arregla errores o agrega funcionalidades. Esto no se hizo durante el proyecto ya que se perdería tiempo necesario para desarrollar este, además, una persona está en constante aprendizaje, así que siempre van a aparecer nuevas y mejores formar de hacer algo, pero si no afectan funcionalidad, es mejor dejar esto para después y no quedarse atascado.

10 Bibliografía

Alder, S. (2022, 3 de noviembre). Can You Make WordPress HIPAA Compliant? HIPAA Journal. Recuperado 24 noviembre 2023 de https://www.HIPAAjournal.com/wordpress-HIPAA-compliant/

Alder, S. (2023, 26 de julio). Majority of Americans Mistakenly Believe Health App Data is Covered by HIPAA. HIPAA Journal. Recuperado 3 diciembre 2023 de https://www.HIPAAjournal.com/americans-mistakenly-believe-health-app-HIPAA/

Das, S. (2023, 6 de abril). How fast should a website load in 2023? BrowserStack. Recuperado 24 noviembre 2023 de https://www.browserstack.com/guide/how-fast-should-a-website-load

Google. (s.f.). PageSpeed Tools. Recuperado 24 noviembre 2023 de https://pagespeed.web.dev

Instituto Nacional del Cáncer. (2022, 17 de agosto). Genética del cáncer. Recuperado 2 diciembre 2023 de https://www.cancer.gov/espanol/cancer/causas-prevencion/genetica

Microsoft. (2023, 2 de abril). Inyección de código SQL. Recuperado 2 diciembre 2023 de https://learn.microsoft.com/es-es/sql/relational-databases/security/sql-injection

Microsoft. (2023, 3 de diciembre). Screen Sizes and Breakpoints. Recuperado 2 diciembre 2023 de https://learn.microsoft.com/en-us/windows/apps/design/layout/screen-sizes-and-breakpoints-for-responsive-design

OWASP. (2023). OWASP Top 10. Recuperado 24 noviembre 2023 de https://owasp.org/Top10/

React Bootstrap. (s.f.). Buttons - React Bootstrap Documentation. Recuperado 24 noviembre 2023 de https://react-bootstrap.netlify.app/docs/components/buttons/

Valezquez, A. OWASP Top 10: #6 Vulnerable and Outdated Components. Foresite. Recuperado 24 noviembre 2023 de https://foresite.com/blog/owasp-top-10-vulnerable-and-outdated-components/

Xercavins Comas, X. (2014, 15 de enero). Medicina preventiva. Top Doctors. Recuperado 2 diciembre 2023 de https://www.topdoctors.es/diccionario-medico/medicina-preventiva

Zarttech. (2023, 28 de marzo). The Importance of a Skilled Frontend Developer in Building User-Friendly Interfaces. Recuperado 24 noviembre 2023 de https://www.linkedin.com/pulse/importance-skilled-frontend-developer-building-user-friendly

11 Anexo:

11.1 Funciones para colapsar/mostrar y agregar/quitar ítems al carro de compras.

```
const handleCollapse = (id) => {
  setVisibleExams((Visible) => ({
    ...Visible,
    [id]: "Collapse",
  }));
};

const handleVisible = (id) => {
  setVisibleExams((Visible) => ({
    ...Visible,
    [id]: "Visible",
  }));
};

const removeBasket = (id) => {
  setBasket((Visible) => ({
    ...Visible,
    [id]: "OutBasket",
  }));
};
const pay = () => {…
};


function ValidateEmail(input) {

  var validRegex = /^[a-zA-Z0-9.!#$%&'*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/;

  if (input.match(validRegex)) {
    setEmailValidation(true)
    pay()
  } else {
    setEmailValidation(false)
  }

}
```

Código 1 Funciones segundo archivo carro de compras

En este código se ven cinco funciones: handleColapse, handleVisible, removeBasket, pay, ValidateEmail. Las primeras dos se encargas de colapsar o mostrar los exámenes que hay dentro de cada paquete.

La función removeBasket se encarga de eliminar elementos que hay en el carro de compras. En este código no se encuentra su contraparte addBasket, ya que este es el archivo de los paquetes ya elegidos por el usuario, por lo que ya están añadidos.

La función pay se activa cuando el usuario aprieta el botón “pasar a pagar”, esto le genera la orden de pago y lo envía a la pasarela de pago para que pueda realizar su compra.

Por último, la función validateEmail se usa para filtrar lo que se ingrese en el campo abierto, esto valida si lo ingresado tiene formato de correo o no, y también ayuda a mitigar la posibilidad de inyección de código.

```
const {
  Basket, setBasket
} = useContext(BasketContext);

const handleCollapse = (id) => {
  setVisibleExams((Visible) => ({
    ...Visible,
    [id]: "Collapse",
  }));
};

const handleVisible = (id) => {
  setVisibleExams((Visible) => ({
    ...Visible,
    [id]: "Visible",
  }));
};
const addBasket = (id,divId) => {
  document.getElementById(divId).style.backgroundColor = "#cff9ff";
  setBasket((Visible) => ({
    ...Visible,
    [id]: "InBasket",
  }));
};

const removeBasket = (id,divId) => {
  document.getElementById(divId).style.backgroundColor = "white";
  setBasket((Visible) => ({
    ...Visible,
    [id]: "OutBasket",
  }));
};
```

Código 2 Funciones primer archivo carro de compras

En este código se encuentran las funciones par colapsar y mostrar los exámenes de cada paquete, y las funciones para añadir o eliminar elementos del carro de compras.

11.2 Archivo MedicalContext donde se guardan los requisitos de exámenes

```
const CreatePacks = (
  name,
  exams,
  code,
  id,
) => {
  const Pack = {
    Name: name,
    Exams: exams,
    Code: code,
    ID: id,
  };
  return Pack;
};


const EstadoNutri = CreatePacks(
  "Estado Nutricional",
  [CFierro, Magnesio, ÁcidoFólico, Calcio, VitaminaB12, VitaminaD, FosfaZinc],
  [301030, 302056, 301002, 302015, 302077, 302078, 111112],
  0
);
const VegaVegeta = CreatePacks(
  "Vegano / Vegetariano",
  [Orina, Urocultivo, Creatinina, NitrogeoUrico, ÁcidoUrico, VitaminaB12, VitaminaD],
  [309022, 306011, 302023, 302057, 302005, 302077, 302078],
  1
);
const transmisiónsexual = CreatePacks(
  "Detección enfermedades de transmisión sexual",
  [ElisaVIH, Sífilis, ElisaVHC, HBsAG, Gonotthoeae, Trachomatis, Cultivo],
  [306169, 306038, 306081, 306079, 306016, 304500, 306023],
  2
);
```

Código 3 Función para crear paquetes de exámenes y ejemplos

En este código se ve una función para crear paquetes de exámenes. La función requiere cuatro elementos como parámetros: El nombre del pack, una lista con los exámenes, los códigos de los exámenes, y el id del paquete. Luego de la declaración se puede ver a esta en uso para la creación de tres paquetes.

```
const CProtein = CreateExamRequirements(
  'Proteína C Reactiva',
  1,
  "Laboratorio",
  1,
  'No necesita preparación',
  [15],
  [],
  [],
  [],
  "Recomendaciones por rango etario",
  305031,
  1
);
const PerfilLipidico = CreateExamRequirements(
  'Perfil Lipídico',
  1,
  "Laboratorio",
  2,
  'Ayuno 8-12hrs',
  [15],
  [],
  [],
  [],
  "Recomendaciones por rango etario",
  302034,
  2
);
```

Código 4 Ejemplo función para crear exámenes

En este código se puede ver el uso de una función llamada CreateExamRequirements. Esta función se usa para generar los exámenes como objetos con sus atributos. La función pide como parámetros el nombre del examen, código del tipo de examen, texto del tipo de examen, código de preparación, texto de preparación, edad mínima, antecedentes médicos necesarios (es decir, debe tener todos), antecedentes médicos opcionales (debe tener mínimo uno de ellos), género, subtitulo, código, e ID del examen.

```
const MedicalContextProvider = (props) => {
  const [OutputListOfExams, SetOutputListOfExams] = useState(() => {
    const localData = localStorage.getItem("OutputListOfExams");
    return localData ? JSON.parse(localData) : [];
  });

  const [PerRecordsCodes, SetPerRecordsCodes] = useState(() => {
    const localData = localStorage.getItem("PerRecordsCodes");
    return localData ? JSON.parse(localData) :
    {
      "Sobrepeso": 6,
      "Obesidad": 7,
      "Hipotiroidismo": 11,
      "Resistencia a la insulina": 8,
      "Diabetes": 9,
      "Hipertensión": 10,
      "Rinitis alérgica": 12,
      "Asma": 13,
      "Tabaquismo":14,
      "Sedentarismo":15,
      "Colesterol alto":20
    };
  });

  const [FamRecordsCodes, SetFamRecordsCodes] = useState(() => {
    const localData = localStorage.getItem("FamRecordsCodes");
    return localData ? JSON.parse(localData) :
    {
      "Resistencia a la insulina": 16,
      "Diabetes": 1,
      "Hipertensión": 2,
      "Cáncer de próstata": 3,
      "Cáncer de mama": 4,
      "Hipotiroidismo": 5,
      "Cáncer de colon": 17,
      "Infarto": 18,
      "Accidente cerebro vascular": 19
    };
  });
```

Código 5 Diccionarios código-antecedente

En este código se ven dos diccionarios, uno para los antecedentes personales, y otro para los antecedentes familiares. Se usa para asociar lo que ingresó el paciente, que viene en texto, con los requerimientos del examen, que vienen en código.

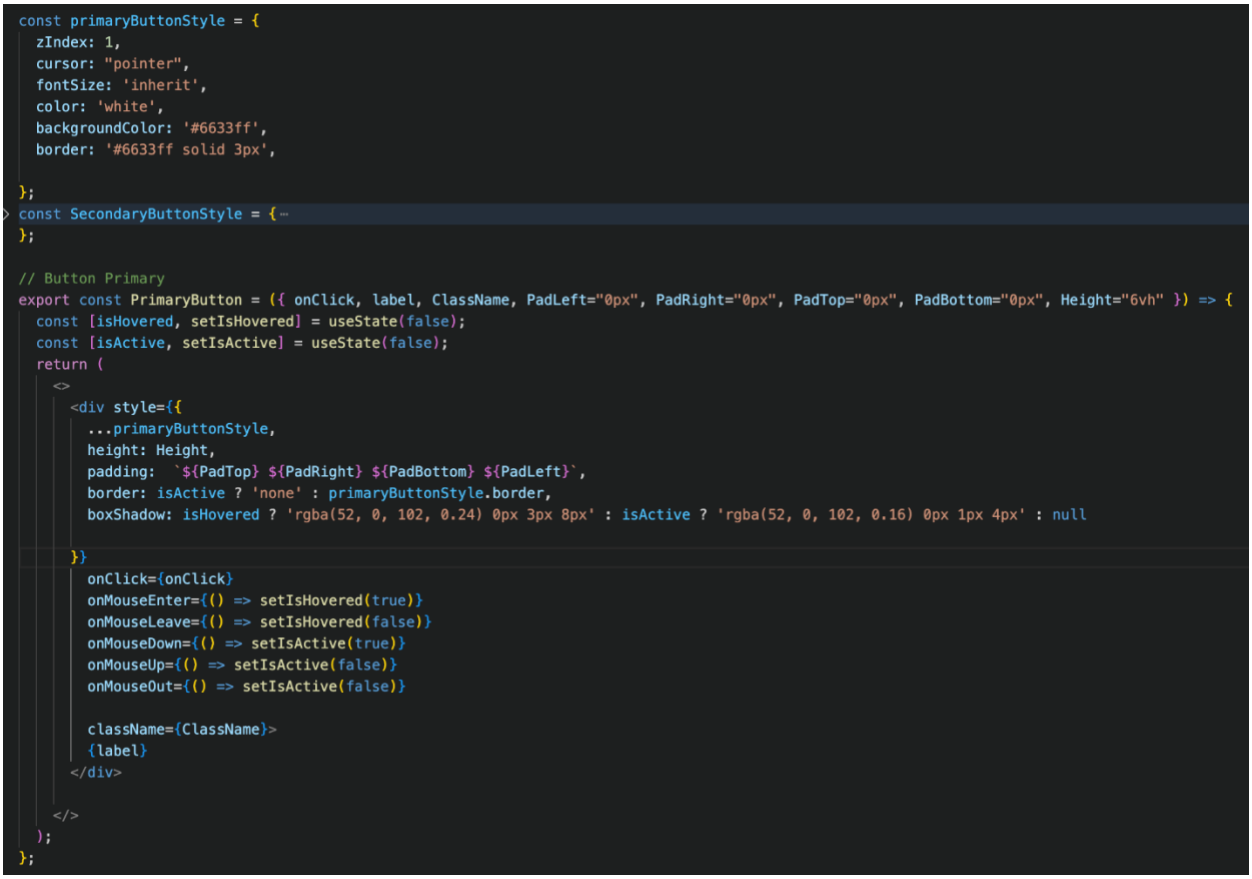
11.3 Filtro texto campo abierto para mitigar inyección

```
const handleName = (value) => {
  const result = value.replace(/\d/g, '');
  setName(result);
};
const handleChange = (value, setValue) => {
  const result = value.replace(/\D/g, '');

  setValue(result.slice(0, 3));
};
```

Código 6 Filtros para cuando se ingresa nombre y números

En este código se ven dos funciones: handleName y handleChange. Estas funciones se usan para mitigar inyección de código malicioso en los campos de nombre, peso, y altura.

11.4 Botón reutilizable primario con paleta de colores de la empresa.

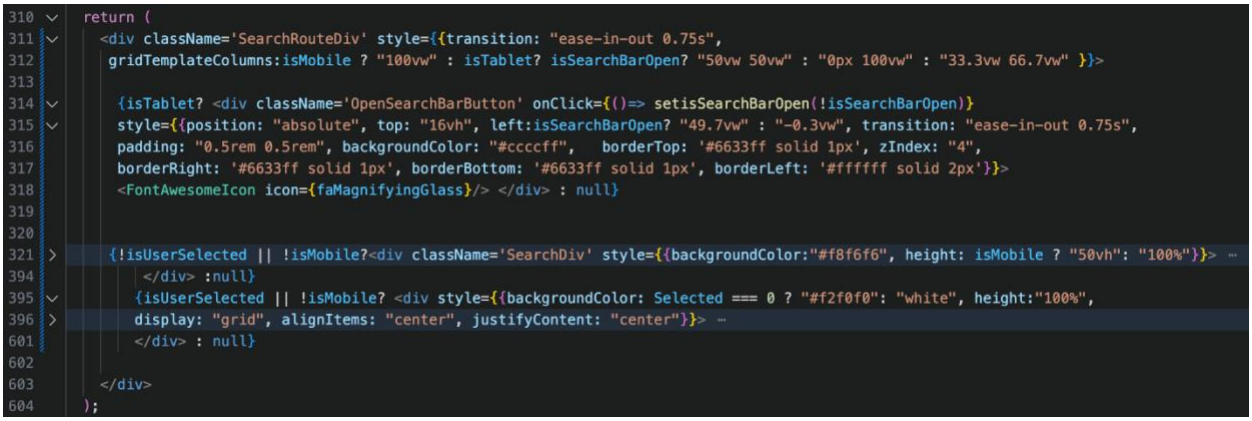


```
const primaryButtonStyle = {
  zIndex: 1,
  cursor: "pointer",
  fontSize: 'inherit',
  color: 'white',
  backgroundColor: '#6633ff',
  border: '#6633ff solid 3px',

};
const SecondaryButtonStyle = {…
};

// Button Primary
export const PrimaryButton = ({ onClick, label, ClassName, PadLeft="0px", PadRight="0px", PadTop="0px", PadBottom="0px", Height="6vh" }) => {
  const [isHovered, setIsHovered] = useState(false);
  const [isActive, setIsActive] = useState(false);
  return (
    <>
      <div style={{
        ...primaryButtonStyle,
        height: Height,
        padding:  `${PadTop} ${PadRight} ${PadBottom} ${PadLeft}`,
        border: isActive ? 'none' : primaryButtonStyle.border,
        boxShadow: isHovered ? 'rgba(52, 0, 102, 0.24) 0px 3px 8px' : isActive ? 'rgba(52, 0, 102, 0.16) 0px 1px 4px' : null

      }}
        onClick={onClick}
        onMouseEnter={() => setIsHovered(true)}
        onMouseLeave={() => setIsHovered(false)}
        onMouseDown={() => setIsActive(true)}
        onMouseUp={() => setIsActive(false)}
        onMouseOut={() => setIsActive(false)}

        className={ClassName}>
        {label}
      </div>

    </>
  );
};
```

Código 7 Botón temático reutilizable

Este código muestra un componente reutilizable con la temática de la empresa. Tiene parámetros para poder ajustarlo a diferentes situaciones.

11.5 Cambio de componentes y disposición para responsividad



```
return (
  <div className='SearchRouteDiv' style={{transition: "ease-in-out 0.75s",
   gridTemplateColumns:isMobile ? "100vw" : isTablet? isSearchBarOpen? "50vw 50vw" : "0px 100vw" : "33.3vw 66.7vw" }}>

    {isTablet? <div className='OpenSearchBarButton' onClick={()=> setisSearchBarOpen(!isSearchBarOpen)}
    style={{position: "absolute", top: "16vh", left:isSearchBarOpen? "49.7vw" : "-0.3vw", transition: "ease-in-out 0.75s",
    padding: "0.5rem 0.5rem", backgroundColor: "#ccccff",   borderTop: '#6633ff solid 1px', zIndex: "4",
    borderRight: '#6633ff solid 1px', borderBottom: '#6633ff solid 1px', borderLeft: '#ffffff solid 2px'}}>
    <FontAwesomeIcon icon={faMagnifyingGlass}/> </div> : null}


   {!isUserSelected || !isMobile?<div className='SearchDiv' style={{backgroundColor:"#f8f6f6", height: isMobile ? "50vh": "100%"}}> …
      </div> :null}
     {isUserSelected || !isMobile? <div style={{backgroundColor: Selected === 0 ? "#f2f0f0": "white", height:"100%",
     display: "grid", alignItems: "center", justifyContent: "center"}}> …
     </div> : null}

  </div>
);
```

Código 8 Búsqueda de usuario, funciones colapsadas para mostrar responsividad

En el código se puede ver primero un con atributo style. Dentro de este se establece gridTemplateColumns para dividir la pantalla en columnas. El ancho de estas columnas se define por unos cuantos operadores ternarios, primero se evalúa si el ancho de la ventana es menor o igual a 425px y si se cumple, se establece que es mobile, así que hay solo una columna. Si no se cumple, se evalúa si es menor o igual a 768px, si se cumple se establece que es tablet y se revisa si la barra de búsqueda está abierta, si lo está se pone mitad y mitad para cada columna, si no se pone la primera columna como 0px, es decir, no se muestra, y la segunda como todo el ancho de la pantalla. Si no es mobile ni tablet se establece que es laptop o mayor y se deja la barra de búsqueda con un tercio del ancho de la pantalla, y la segunda columna con dos tercios del ancho.

Luego, hay un segundo componente, ese solo se muestra si se cumple la condición de tablet. Este componente es un marcador con un símbolo de una lupa para hacer referencia a que es una barra de búsqueda y el usuario entienda para qué es esta pestaña. Al este ícono se cierra o abre la barra de búsqueda, es decir, se cambia la variable isSearchBarOpen a verdadero o falso según corresponda.

Finalmente se pueden ver dos componentes colapsados, estos son las columnas. El primero es la barra de búsqueda, y el segundo es donde se muestra la información del paciente. La primera tiene un operador ternario con condición doble, si es que no hay usuario seleccionado o si es que no es mobile. Si no es mobile, como se estableció con las columnas, se muestran ambas, pero si lo es, se muestra solo una. Para elegir qué columna se muestra se usa la variable booleana isUserSelected, si no hay usuario seleccionado se muestra la barra de búsqueda para que pueda seleccionar uno.

La segunda columna también tiene condición doble, si es que hay un usuario seleccionado, o si es que no es mobile. Del mismo modo que la anterior columna, solo si es que es mobile habría necesidad de

ocultarla, por lo que, si no lo es, se muestra. Si es mobile, se revisa si hay un usuario seleccionado y si lo hay, se muestra el usuario.

11.6 Modal perfil médico confirmación

```
  <Modal
  open={ModalOpen}
  ModalHeight='fit-content'
  closeIcon={false}
  BackdropClick = {ConfirmationModal ? ()=>HandleConfirmation() : ()=>console.log("")}
  onClose={()=>setModalOpen(false)}
  >
{(!EmailModal)?
<div style={{ display: "grid", justifyItems: "center", gridTemplateRows: "6rem 6rem 6rem", height: "100%" }}>…
  </div>
  :  (!ConfirmationModal)?
  <div style={{ display: "grid", justifyItems: "center", gridTemplateRows: "9rem 6rem 6rem 6rem"}}>…
</div>
:
  <div style={{ display: "grid", justifyItems: "center", alignContent: "center"}}>…
  </div>
}


  </Modal>
```

Código 9 Extracto de código perfil médico: Modal confirmación de cambios

Este código es parte del perfil de médico. Cuando este desea cambiar sus datos, se le pide una confirmación de dos factores, primero su contraseña, y luego un código enviado a su correo. Por eso, este modal tiene tres partes: Contraseña, Correo, confirmación.

Para saber qué mostrar en el modal se usaron dos variables y dos operadores ternarios. Primero, la variable EmailModal no es verdadera, significa que aún no se llega a esa etapa, por lo que se muestra el modal de contraseña, si es verdadera, se revisa si ConfirmationModal es verdadera, si no es que aún no se llega a la confirmación, así que se muestra la etapa de ingresar el código. Una vez se ingresa y se valida, se muestra la confirmación de que los datos fueron cambiados correctamente.