



TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN ENXEÑARÍA DO SOFTWARE



Desarrollo de una interfaz interactiva de navegación de registros históricos personalizados integrada con un sistema de asistencia en procesos de empresa

Estudiante: Ángel Barral de Jesús
Dirección: Carlos Vázquez Regueiro
Juan Antonio Martín Pernas

A Coruña, julio de 2024.

Dedicatoria

Agradecimientos

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Resumen

Este proyecto consiste en la creación de una interfaz interactiva para generar informes de forma sencilla mediante expresiones de lenguaje natural que va añadiendo el usuario.

Su objetivo principal es que, utilizando los datos recogidos anteriormente mediante SIN-VAD, esta interfaz nos permite manipularlos o consultarlos de tal forma que haciendo llamadas a su API, recupere los datos que el usuario esta buscando.

Esto nos permitirá generar informes de cualquiera de los datos que este en el histórico. Para esto será necesario analizar estas expresiones naturales y procesar las palabras claves que se utilizarán para buscar los datos que necesitemos.

La idea con esto, es que los usuarios que vayan utilizar esta interfaz no pierdan parte importante de su trabajo en configurar el informe, sino que mediante unas simples expresiones y con un diseño intuitivo con ejemplos para cualquiera persona ajena al mundo tecnologico, sea capaz de generar informes de forma sencilla.

Abstract

This project consists of creating an interactive interface to generate reports in a simple way using natural language expressions.

Its main objective is that, using the data previously collected in SIN-VAD, this interface allows us to manipulate or consult them in such a way that by making calls to the SIN-VAD Rest API, the data that the user is looking for is recovered.

This will allow us to generate reports of any of the data that is in the history. For this, it will be necessary to analyze these natural expressions and extract the key words that will be used to search for the data we need.

The idea with this is that the users who are going to use this interface do not lose an important part of their work in configuring the report, but rather that through some simple expressions and with an intuitive design with examples for anyone outside the technological world, they will be able to generate reports in a simple way.

Palabras clave:

- First itemtext
- Second itemtext
- Last itemtext
- First itemtext
- Second itemtext
- Last itemtext
- First itemtext

Keywords:

- First itemtext
- Second itemtext
- Last itemtext
- First itemtext
- Second itemtext
- Last itemtext
- First itemtext

Índice general

1	Introducción	1
1.1	Contexto	1
1.2	Objetivos	1
2	Metología	3
2.1	Metodología elegida	3
3	Desarrollo	5
3.1	1º Iteración: Diseño de la interfaz.	5
3.1.1	1º Opción	5
3.1.2	2º Opción	6
3.1.3	3º Opción	8
3.1.4	Conclusión	8
3.2	2º Iteración: Nueva propuesta de diseño de la interfaz.	9
3.3	3º Iteración: Consolidación de la nueva propuesta.	9
4	Fundamentos tecnológicos	16
4.1	Vue	16
4.2	TailwindCSS	17
4.3	Shadcn-Vue	17
4.4	Visual Studio Code	17
5	Conclusiones	19
A	Material adicional	21
	Bibliografía	23

Índice de figuras

3.1	Primera opción: Tabla con datos para recuperar	5
3.2	Primera opción: Tabla con datos recuperados y/o posibles modificaciones . . .	6
3.3	Segunda opción: Selector con primeras opciones	7
3.4	Segunda opción: Selector las siguientes opciones para acotar nuestro dato . .	7
3.5	Tercera opción: Prompt con datos a añadir	8
3.6	Conclusión: Seleccionamos los datos que nos interesan	10
3.7	Conclusión: Tabla con los datos seleccionados y prompt para buscar	10
3.8	Conclusión: Tabla con los datos seleccionados y las opciones buscadas ante- riormente	11
3.9	2º Iteración: Nueva interfaz	11
3.10	2º Iteración: Introducir columnas y datos para la búsqueda	12
3.11	2º Iteración: Resultados de la búsqueda	12
3.12	3º Iteración: Pantalla principal	14
3.13	3º Iteración: Configurador de informe	14
3.14	3º Iteración: Introducir datos de ejemplo para buscar	15
3.15	3º Iteración: Resultados de la búsqueda	15

Índice de cuadros

Introducción

1.1 Contexto

Cualquier persona de una empresa pierde gran parte de su tiempo para ser capaz de acceder a la información que necesita para llevar su trabajo o en registrar nuevos datos en un sistema de información de una empresa.

Para solucionar esto tenemos SINPAD, un sistema de comprensión de lenguaje natural que es desarrollado por la empresa SREC Solutions. El objetivo es asistir al usuario para consultar información, registrar datos y supervisar información desde cualquier dispositivo con sus propias palabras, en múltiples idiomas y adaptado a los procesos de empresa.

Pero esto tiene una gran limitación, y es que el cliente debe construir sus propias aplicaciones para integrar la API REST de SINPAD para así poder generar informes a partir de los registros de los históricos del cliente.

Por tanto, el objetivo de este proyecto es generar una interfaz interactiva de navegación de históricos asociados que facilite la integración de la API de SINPAD y nos permita gestionar la generación, consulta, interacción, edición en tiempo real, descarga y generación de notificaciones en tiempo real de estos informes que se nombraron anteriormente.

Se busca que esta interfaz sea fácil de utilizar y intuitiva, para que cualquiera persona de cualquiera edad sepa utilizarla, y mediante unas pocas y sencillas acciones, sea capaz de generar informes sobre cualquiera de los datos que tienen el histórico.

1.2 Objetivos

En relación con lo anterior, a continuación se detallarán los objetivos que debería cumplir nuestra interfaz.

Esta interfaz debe estar disponible vía web y cualquier usuario que tenga acceso tiene que ser capaz de utilizarla de forma sencilla. Para esto buscaremos que nuestra interfaz sea intuitiva.

tiva y con esto nos referimos que cualquiera persona, con o sin conocimientos tecnologicos, sea capaz mediante una serie sencilla de pasos, sacar los informes que a ellos le interesen.

Dentro de esta interfaz, el usuario podrá hacer los siguientes pasos:

- **Título del informe:** El usuario, antes de empezar a configurar la información que quiere que aparezca en el informe, necesitará indicar el propósito por el cuál se crea, y esto se indicará mediante el título.

- **Periodicidad del informe:** El usuario podrá indicar la periodicidad con la que quiere generar el informe, ya que estos, se generarán automáticamente cada x tiempo.

- **Número de columnas:** El usuario podrá elegir el número de columnas que aparezcan en su informe y el contenido de cada una de ellas. Si alguna de estas, no muestra el dato correcto, se podrá editar antes de generar el informe.

- **Contenido del informe:** Al seleccionar el número de columnas, aparte del título de estas, será necesario el dato a buscar mediante una expresión de lenguaje natural, y esto será lo que el usuario le pasará a nuestro LLM para que lo procese.

Para cumplir todos estos pasos, la interfaz los contendrá de forma sencilla y intuitiva como se mostrará mas adelante en este documento, para facilitar al usuario este proceso lo máximo posible.

Capítulo 2

Metología

En este apartado se detalla la metodología utilizada, así como las razones por las que se seleccionaron esta metodología respecto a otras

2.1 Metodología elegida

Para la realización de este proyecto se seguirá una metodología de desarrollo ágil basada en SCRUM.

Esta se adaptará al proyecto para obtener los mejores resultados de la manera más práctica posible, incluyendo un desarrollo incremental a través de iteraciones en las que definen los requisitos mínimos de funcionalidades, que se desarrollaran, se documentaran y se validarán antes de pasar a la siguiente iteración. Se mantienen reuniones semanales para definir y evaluar el grado de cumplimiento de los objetivos de cada iteración, revisar la planificación y detectar y corregir problemas que dificulten el avance del proyecto.

Gracias al seguir esta metodología nos proporcionan una serie de ventajas respecto a otras como son:

- Resultados anticipados: No es necesario esperar al final para obtener resultados del proyecto. Al cierre de cada semana de trabajo, ya se pueden ver ciertos logros, lo que permite probar el producto sin estar finalizado.
- Flexibilidad y adaptación: Scrum nos permite adaptarnos según las prioridades o los cambios que queremos. Al final de cada iteración, se puede hacer pruebas de concepto sobre el producto para así tomar las mejores decisiones en función del resultado que obtenemos, permitiéndonos hacer cambios para solucionar los problemas que vayan apareciendo en el proyecto.
- Gestión de las expectativas del usuario: Los usuarios pueden participar en cada una de las etapas del proceso y proponer soluciones. Pueden aportar ideas en cualquier parte del proyecto para determinar soluciones a los problemas que vayan apareciendo.

- Gestión sistemática de riesgos: SCRUM nos permite que los problemas que vayan apareciendo, puedan ser gestionados en el momento de su aparición. La intervención será inmediata a la hora de que aparezca un riesgo. Como el trabajo se divide en iteraciones, la cantidad de riesgo a los que nos vamos a enfrentar es menor y esta limitado, lo que ayuda a la complejidad del proyecto.

Desarrollo

3.1 1º Iteración: Diseño de la interfaz.

Lo primero era tener claro cuál sería el diseño más correcto de nuestra interfaz para que así fuera intuitivo y fácil de utilizar para el cliente. Contamos con que este vaya ser la persona más alejada de la tecnología posible y por tanto la interfaz sea intuitiva y fácil de utilizar. Se propusieron 3 opciones para realizar nuestro diseño:

3.1.1 1º Opción

En esta primera opción, se propuso una tabla donde el usuario indica en cada columna el nombre de datos que quiere y mediante un formulario con una ayuda de un LLM recupera los datos del histórico de la empresa.

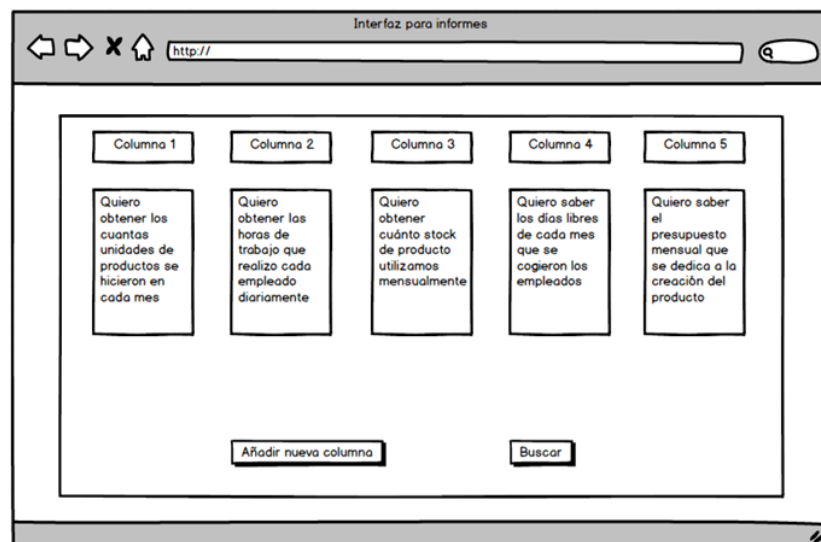


Figura 3.1: Primera opción: Tabla con datos para recuperar

El usuario podría introducir las columnas que quisiera e indicaría el nombre del dato que va a recuperar mediante el nombre de la columna. Con la ayuda de un formulario donde indicamos los datos que queremos recuperar y procesando esta información mediante un LLM, se recuperan estos datos del histórico.

Interfaz para informes

Columna 1	Columna 2	Columna 3	Columna 4	Columna 5
Unidades por mes	Horas de empleado diarias	Stock de producto utilizado mensualmente	Días libres mensuales por empleado	Presupuesto mensual por producto
¿Es correcto o quiere volver a buscar el datos?	¿Es correcto o quiere volver a buscar el datos?	¿Es correcto o quiere volver a buscar el datos?	¿Es correcto o quiere volver a buscar el datos?	¿Es correcto o quiere volver a buscar el datos?
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Buscar	Buscar	Buscar	Buscar	Buscar

Figura 3.2: Primera opción: Tabla con datos recuperados y/o posibles modificaciones

Después de hacer la búsqueda con la ayuda del LLM, se mostrarán los datos que se recuperan de cada columna y tendrá la opción de editar alguno de estos datos sino es el que le interesa al cliente

Es una solución que permite guiar al usuario paso a paso para seleccionar los datos que quiere en su informe, pero tiene un gran problema, y es que, si añadimos muchas columnas, la interfaz se volverá muy compleja, ya que habría demasiada información y podría volverse muy confuso.

3.1.2 2º Opción

Esta segunda solución se basaría en un selector con los datos queremos ver, los seleccionamos e iremos avanzando progresivamente con más selectores hasta tener el dato concreto.

Primero de todo, el usuario selecciona los datos sobre las tablas que le interesa.

Después, se irán mostrando las distintas opciones de cada dato seleccionado e ir avanzando progresivamente mediante selectores.

En esta solución, los usuarios pueden ir seleccionando de manera progresiva lo que quieren en su informe, lo que puede ser bastante intuitivo para obtener los datos que queremos finalmente, pero tiene el mismo problema que la anterior, al seleccionar muchos datos, la interfaz se puede volver bastante engorrosa. También limita a la hora de crear informes com-

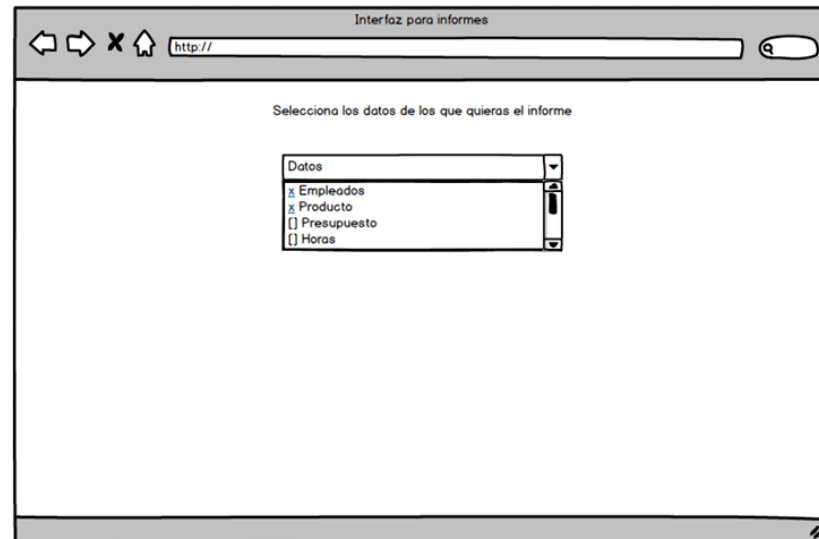


Figura 3.3: Segunda opción: Selector con primeras opciones

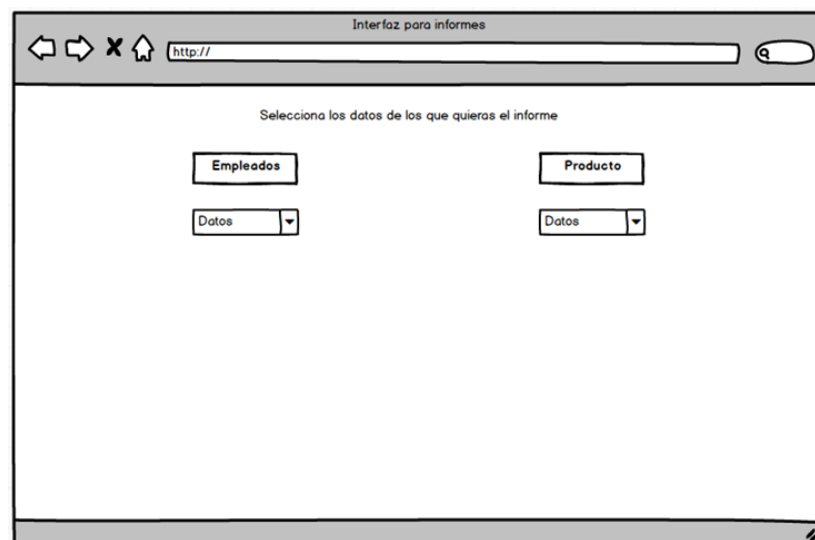


Figura 3.4: Segunda opción: Selector las siguientes opciones para acotar nuestro dato

plejos, ya que se necesitan muchos pasos para crearlos y puede ser una gran pérdida de tiempo para el cliente.

3.1.3 3º Opción

La tercera solución es un prompt donde el usuario interactuara con un LLM viendo en un json o una tabla los datos que van ofreciendo y seleccionando los datos que más nos interesen mediante inputs.

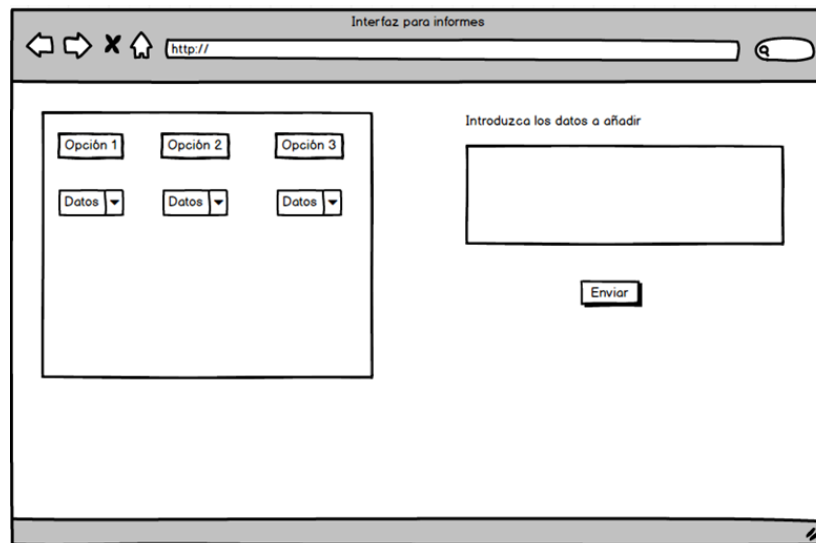


Figura 3.5: Tercera opción: Prompt con datos a añadir

El usuario introducirá los datos que le interese en el input y se irán mostrando las distintas opciones de datos para los elementos del informe generar.

Esta es una opción bastante buena para la generación de informes complejos, ya que con la ayuda del LLM puede ayudarnos a buscar lo que encontramos de forma correcto, pero esto se puede convertir en una contra también, ya que en el caso contrario podría ser que el LLM no entienda claramente lo que se quiere buscar y esto puede ser un problema para un usuario no familiarizado con la interfaz.

3.1.4 Conclusión

De estas 3 opciones, después de estudiar sus pros y contras, se concluyo que la mejor opción sería una mezcla de las 3 de esta forma.

Primero, el usuario seleccionará los datos de los que le interese generar el informe sin sus opciones.

Con los datos seleccionados, se nos mostrará una tablita donde con esos datos y un prompt,

el usuario indicará las opciones que quiere de cada dato recogiendo esa información con un LLM para posteriormente mostrarlo en la tabla.

Cuando el LLM procesé la información, mostrará un selector desplegable donde se puede seleccionar la opción o las opciones que el usuario considere opinión según lo introducido en el prompt y su criterio.

Esta es la mejor opción mezclando las 3 anteriores, ya que el usuario puede comenzar con una serie de columnas claras e ir modificándolas o no según su criterio. Con la ayuda del LLM, este nos muestra las opciones más acordes a nuestra búsqueda y ayudando así a no tener que navegar mediante selectores, lo cuál es mucho más engorroso para un usuario sin experiencia.

3.2 2º Iteración: Nueva propuesta de diseño de la interfaz.

Una vez comentadas las ideas anteriores junta UX, y tras refinarlas, se llegó a una conclusión conjunta para como se quiere que sea esta interfaz.

En esta nueva pantalla, se nos mostrará diferentes opciones para configurar nuestro informe. Se podrá seleccionar la periodicidad con la que se generé (la idea de esta interfaz es que se generé cada x tiempo el informe una vez configurado), pudiendo seleccionar las unidades de tiempo y el período en el que se van a generar. Podemos ir añadiendo las columnas que creamos oportunas que aparezcan en él y en ellas indicar tanto el nombre como el dato que queremos buscar.

Una vez que introduzcamos la información que queremos, con la ayuda de un LLM, buscará está en nuestro histórico, para así mostrarle posteriormente los datos que busca el cliente.

Si alguno de los datos que aparecen no nos encaja con lo que buscamos, podremos seleccionar mediante un desplegable la columna del informe que queremos cambiar y con la ayuda de un prompt, introduciremos los cambios que consideraremos oportunos.

Tendremos también una serie de palabras clave sugeridas para que así esta sea más fácil y nos facilite a la hora de generar el informe.

3.3 3º Iteración: Consolidación de la nueva propuesta.

Se volvió a comentar la nueva propuesta con UX y se tomaron algunas decisiones de diseño para la interfaz. Primero de todo y más importante, habría que definir la finalidad de ese informe y esto se haría a través del título.

Se añadirá un título que describirá el propósito del informe. Para que sea más fácil para el usuario la configuración de este, se mostrará una serie de ejemplos donde el usuario podrá ver los diferentes pasos que debe seguir para generar el informe.

Una vez añadido el título, deberíamos configurar el informe. Para esto seleccionaremos la

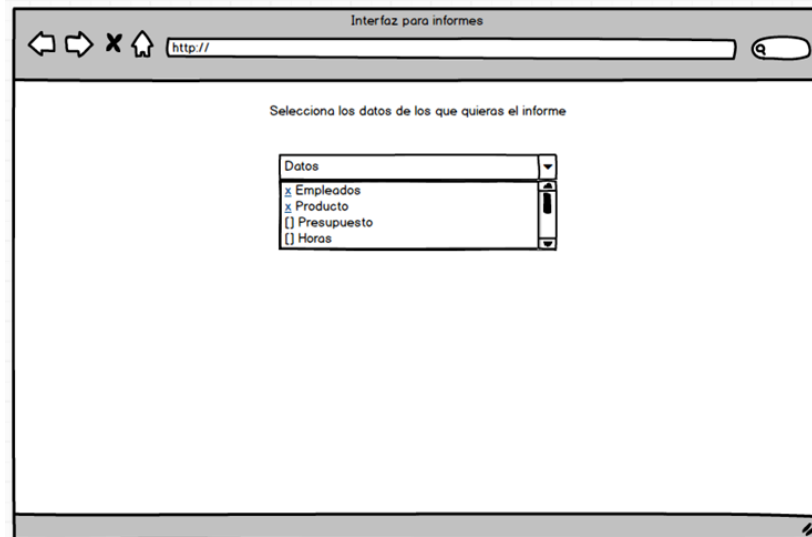


Figura 3.6: Conclusión: Seleccionamos los datos que nos interesan

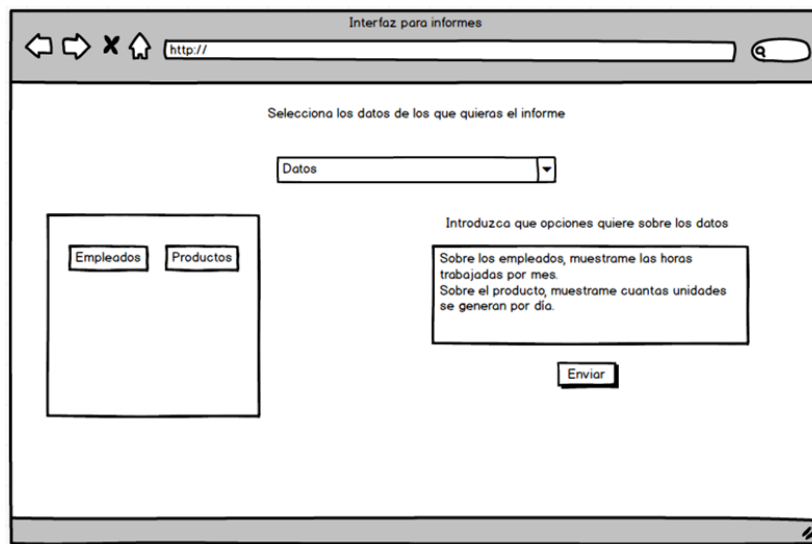


Figura 3.7: Conclusión: Tabla con los datos seleccionados y prompt para buscar

Interfaz para informes

Selecciona los datos de los que quieras el informe

Datos

Empleados

Productos

Horas por mes

Unidades por día

Introduzca que opciones quiere sobre los datos

Sobre los empleados, muéstrame las horas trabajadas por mes.

Sobre el producto, muéstrame cuantas unidades se generan por día.

Enviar

Figura 3.8: Conclusión: Tabla con los datos seleccionados y las opciones buscadas anteriormente

Interfaz para informes

Periodicidad del informe

3

Horas

Días

Meses

Años

Añadir nueva columna

Introduzca el nombre de la columna que quiera generar y el tipo de dato a buscar

Columna

Dato a buscar

Buscar

Figura 3.9: 2º Iteración: Nueva interfaz

Figura 3.10: 2º Iteración: Introducir columnas y datos para la búsqueda

Figura 3.11: 2º Iteración: Resultados de la búsqueda

periodicidad con la que se genera y el nombre de la columna más el dato que queremos añadir a nuestro informe, como se comentara en un principio. Si necesitamos añadir más datos a nuestro informe, añadiríamos una nueva columna a nuestra tabla.

En este caso, añadimos 2 columnas con los datos oportunos para buscarlos.

A la hora de buscar estos datos, nos mostraría las opciones que al LLM le pareció más oportuno según nuestros datos registrados en el histórico. Podemos seleccionar las distintas columnas de nuestro informe y mediante un prompt con algunas opciones sugeridas, poder refinar nuestra búsqueda para que nos muestre el dato de forma correcta

Si tenemos claro que los datos son correctos, podemos generar nuestro informe, que quedara configurado para generarse periódicamente según el tiempo que le habremos indicado.

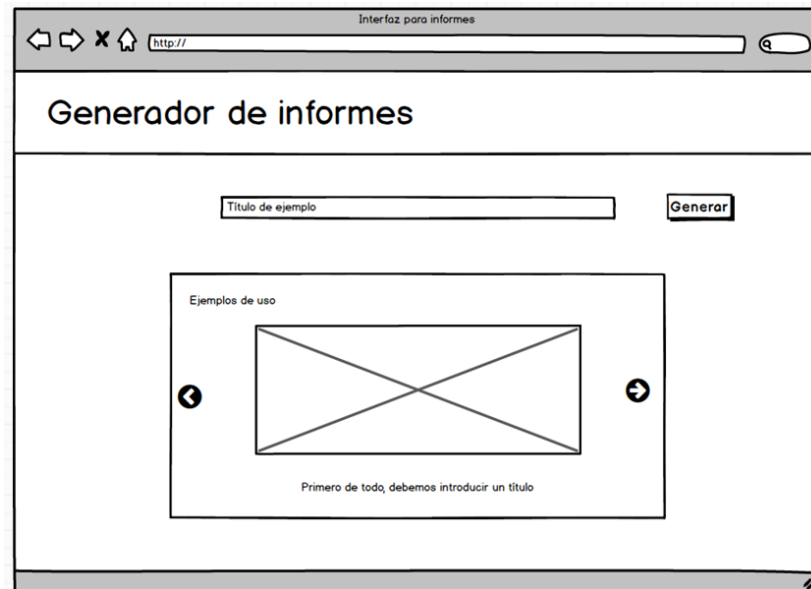


Figura 3.12: 3ª Iteración: Pantalla principal

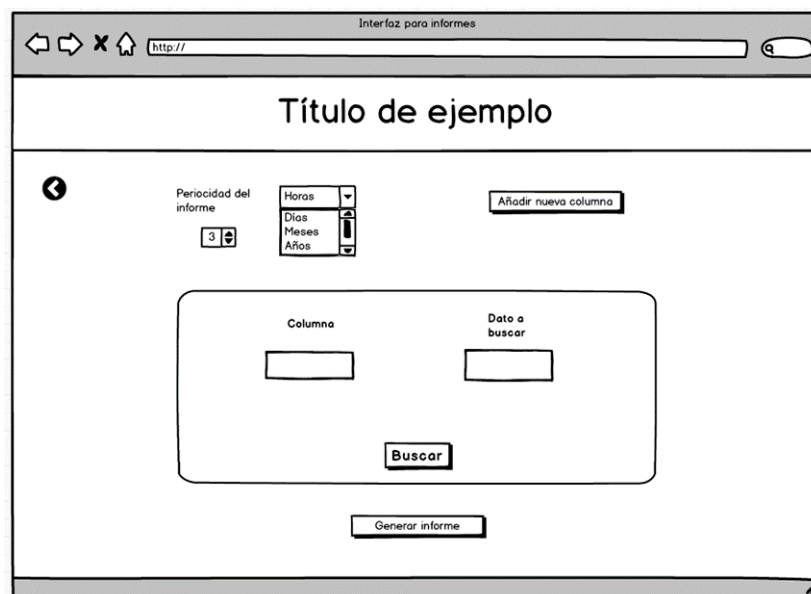


Figura 3.13: 3ª Iteración: Configurador de informe

Interfaz para informes

http://

Título de ejemplo

Período del informe

Horas
Días
Meses
Años

3

Añadir nueva columna

Columna	Dato a buscar
Unidades de producto por mes	Quiero obtener las cuantas unidades de productos se hicieron en cada mes
Stock/mes	Quiero obtener cuánto stock de producto utilizamos mensualmente

Buscar

Generar informe

Figura 3.14: 3º Iteración: Introducir datos de ejemplo para buscar

Interfaz para informes

http://

Título de ejemplo

Período del informe

Horas
Días
Meses
Años

3

Añadir nueva columna

Columna	Dato a buscar
Unidades de producto por mes	Unidades por mes
Stock/mes	Stock utilizado mensualmente

Unidades de producto por mes
Unidades de producto por mes
Horas trabajados por mes
Días libres por mes de empleado
Stock/mes

Unidades por año Unidades por día Unidades por semana

Generar informe

Figura 3.15: 3º Iteración: Resultados de la búsqueda

Fundamentos tecnológicos

4.1 Vue

Vue es un framework de JavaScript que permite crear potentes y versátiles interfaces de usuario. Una de las grandes ventajas que tiene vue es que combina HTML, CSS y JavaScript en un mismo fichero, permitiéndonos tener más ordenado y localizable nuestro código.

Este framework nos permite crear componente que después van poder ser reutilizados por otros componentes, lo que nos va dar una gran ventaja a la hora de ahorrar tiempo y costes en el proyecto. Cada vez que hagamos un cambio en alguno de nuestro componente, este se actualiza automáticamente en la interfaz gracias a su propiedad reactiva.

Incluye algunas herramientas, como Vue Router, para gestionar el enrutamiento de las páginas que lo utilizaremos en el proyecto para nuestra interfaz.

Respecto a otros frameworks como Angular o React, vue nos proporcionan una serie de ventajas:

- Curva de aprendizaje suave: Es simple y fácil de aprender. Contiene bastante documentación clara, lo que permite desarrollar código de forma rápida.
- Flexibilidad respecto a frameworks como Angular, Vue permite al desarrollador elegir y configurar las partes que este considere oportunas únicamente.
- Tamaño ligero: Comparado con los otros frameworks, Vue.js es relativamente pequeño, lo que puede resultar de gran ayuda a la hora de tiempos de carga y rendimiento.
- Es muy parecido a HTML, por lo que la sintaxis es muy fácil de entender si ya trabajaste anteriormente con este.

Utilizaremos este framework junta a Tailwindcss para el desarrollo de nuestra interfaz.

4.2 TailwindCSS

Tailwindcss es un framework de CSS que nos permite componer estilos de manera mucho más sencilla. Nos permite relacionar y crear clases css para utilizar en nuestros elementos HTML fácilmente.

A diferencia de Bootstrap o foundation, tailwind lo que nos proporcionan son conjunto de clases css de bajo nivel que son fácilmente combinables para así crear diseños sin la necesidad de escribir CSS personalizado.

Gracias a esta característica, nos proporciona un desarrollo de código más rápido al eliminar la necesidad de escribir todo el CSS personalizado para cada componente.

Gracias al tener una clases de utilidad comunes, esto nos permite que el diseño en nuestra aplicación sea fácilmente mantenible y no preocuparnos de tener conflictos con los css.

Una clave en todo esto es que Tailwind no impone estilos predeterminados ni componentes específicos, lo que nos da una gran libertad a la hora de crear los diseños.

Por eso se eligió Tailwind, por su gran facilidad, flexibilidad y personalización a la hora de crear nuestro proyecto.

4.3 Shadcn-Vue

Shadcn es una biblioteca de componentes que permite una fácil personalización de estos y de su estilo. Esto nos permite construir nuestra aplicación de forma eficiente y manejable, al utilizar componentes reutilizables.

La versión que utilizaremos en el proyecto es shadcn junto a vue o shadcn-vue, que permite una mayor facilidad a la hora de incluir los componentes, ya que estos están creados en vue.js.

Esto también nos permite que a la hora de añadir un componente, este se actualice automáticamente en la interfaz y si hacemos algún cambio en él, también se añada de forma correcta.

4.4 Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft. Es conocido por ser ligero, rápido y extensible. Es gratuito y de código abierto, lo que nos permite tener un gran variedad de extensiones para añadir a nuestro código.

Gracias a su gran variedad de lenguajes soportados, se eligió este ide ya que soporta Vue.js, por lo que es perfecto para desarrollar nuestra interfaz. También se eligió porque nos permite una serie de característica como son:

- Es ligero y rápido, ya que como vue.js, este tiene un rendimiento eficiente y tiempos de carga rápidos, lo que nos permite un desarrollo más eficiente.
- Tiene miles de extensiones disponibles y muchas muy útiles para nuestro proyecto.
- Ofrece IntelliSense, que nos autocompleta el código para así poder avanzar rápidamente con nuestras funciones y métodos.
- Contiene una terminal integrada, que permite ejecutar comandos en línea para añadir librerías o componentes de las bibliotecas que se nombraron anteriormente.

Conclusións

DERRADEIRO capítulo da memoria, onde se presentará a situación final do traballo, as leccións aprendidas, a relación coas competencias da titulación en xeral e a mención en particular, posibles liñas futuras,...

Apéndices

Apéndice A

Material adicional

EXEMPLO de capítulo con formato de apéndice, onde se pode incluír material adicional que non teña cabida no corpo principal do documento, suxeito á limitación de 80 páxinas establecida no regulamento de TFGs.

Bibliografía
