

Proyecto Final Diseño de Aplicaciones Web

Plataforma web para una empresa de mensajería

Implementación de una plataforma web para la gestión cliente final y empresa
franquiciada a nivel local



Alberto Garre Muñoz
48693810G

Índice

Resumen	
Justificación del proyecto	
Objetivos	
Análisis del negocio	
Resumen ejecutivo	
Descripción del producto	
Análisis del mercado	
Estructura organizativa de los recursos humanos	
Plan de marketing	
Estados financieros previsionales a 3 años	
Análisis DAFO	
Desarrollo del proyecto	
Revisión bibliográfica, fundamentación teórica.	
Materiales y métodos	
Resultados y análisis	
Conclusiones	
Bibliografía	

1. Resumen

CONTOSOX is a national company dedicated to the transport of light parcel.

It consists of a large logistics chain that can move merchandise throughout the country for further processing in the logistics centers and collection from local franchisees who are in charge of dealing with the client at the end point. Its target audience are Pymes and individual users.

In this project we are going to develop a web platform within the scope of a national transport company. Due to the nature of the sector, this page must be a common access point for all potential CONTOSOX users. For this reason, although it will serve multiple small businesses, we will focus on only one. Its use is intended for the end customer and for small local transport companies franchised by the parent company. It is important to emphasize that there is a logistical dependence with the parent company that we will not be able to resolve in this project due to a lack of data, knowledge and time.

What does the new application have to offer?

- As clients, we will be able to make shipments and check the state of our package.
- As franchisees, we will streamline the company's production processes. Pickups and deliveries will be able to be registered and updated in real time. As employees we will be able to assign the packages that we are going to distribute, we will have instant access to the entire list of packages and we are going to confirm a delivery with just a code and a click.

2. Justificación del proyecto

He escogido este proyecto por una razón básica: durante la duración de estos dos años de curso he estado trabajando en una empresa del sector de reparto. En labores de operario, repartidor y oficinista.

La correcta conjugación de trabajo y estudio ha logrado que consiga finalizar este ciclo formativo.

Es esta empresa he aprendido y desarrollado buenas características para el correcto desempeño laboral.

Y he descubierto posibles puntos a mejorar dentro de los protocolos operacionales de la empresa.

Es por ello, que conocía antes empezar a crear la aplicación del proyecto, que elementos necesitaba, hasta donde iba a llegar y donde podía simplificar sin que la aplicación perdiese el sentido.

He usado React y Javascript porque durante las practicas en empresa he aprendido un poco de estas materias y son las más usadas en el sector.

Esto es así también con la herramienta Postman.

3. Objetivos

Queremos crear un sistema que soporte la operativa de una empresa de paquetería.

Para ello la plataforma debe ofrecer las siguientes características:

- Debe poder ser accesible desde móvil y pc.
- Ver las tarifas que la empresa me ofrece.
- Navegar por la página de forma sencilla.
- El usuario puede solicitar una recogida a domicilio.
- Seguir el estado de un envío.
- Acceso controlado a las secciones de empleado.
- Crear un gestor de usuarios.
- Gestion de empleados con sus roles dentro de la empresa.
- Cada empleado debe poder asignarse sus propias tareas.
- Crear un sistema para asignar paquetes y confirmar entregas y recogidas.
- Ciclo de vida del paquete con seguimiento de su estado, localización y persona responsable.

4. Análisis del negocio

a) Resumen ejecutivo

CONTOSOX es una empresa nacional que se dedica al transporte de paquetería ligera.

Cuenta con una amplia cadena logística que puede mover mercancía por todo el país para su posterior procesamiento en las centrales logísticas y recogida de los franquiciados locales que se encargan del trato con el cliente en punto final.

El papel de los franquiciados es fundamental y representa un pilar en la operativa y la gestión.

Los motivos por los que CONTOSOX ha elegido esta forma organizativa, responden a múltiples factores.

- Delegación de la micro gestión local
- El corporativismo reduce la eficiencia
- Reducción de márgenes
- Dilución de la responsabilidad en la cadena corporativa
- Evita incrementar la masa salarial.

A la luz de estas características no es casual que las principales empresas del sector adopten el mismo formato organizativo no solo en España si no en el resto del mundo.

Contosox orienta su esfuerzos a particulares y PYMES siendo su público objetivo y con tarifas y servicios que se amoldan a sus necesidades.

b) Descripción del servicio

En un mundo cada vez más interconectado, el comercio digital se ha multiplicado por órdenes de magnitud en la última década. El consumidor ha pasado del recelo a comprar por internet, a la aceptación de que es más rápido, más cómodo y ofrece mayor oferta.

¿Pero, que ocurre desde que el cliente compra el producto en una página web hasta que lo recibe?

Ese es el papel de Contosox, participar en la cadena logística del comercio global como un socio más, prestando el mejor servicio.

Pero no solo somos meros transportistas queremos formar parte de la cadena de valor del comercio de proximidad, acercando los productos locales a sus vecinos.

Queremos abrir la oferta de canales de distribución de forma que un pequeño comercio, una farmacia o un productor de naranjas ecológicas por ejemplo puedan enviar sus productos al otro lado de la península de forma fácil, económica y con garantías de satisfacción de todas partes.

Consideramos a las pymes que nos contratan como socios comerciales por compartir un mismo compromiso de mejora y éxito económico.

Y como no, cubrimos el ámbito de la entrega a particulares, que nos tomamos muy en serio puesto que somos la última representación de la experiencia de compra. Para ello contamos con profesionales bien remunerados, formación interna, y protocolos de optimización de recursos que nos permiten no solo garantías de entrega sino que las entregas sean en condiciones óptimas.

c) Análisis del mercado

El mercado está segmentado por los vectores de precio y volumen de transporte de mercancías.

Las empresas que apuestan por grandes volúmenes son Seur, Fedex y DHL

Grandes volúmenes permiten ajustar márgenes a las empresas matrices. Pero esto lo hacen a costa de unas condiciones peores para los franquiciados.

Los franquiciados en estos casos son autónomos o pequeñas pymes donde muchas veces no se conoce aspectos básicos como la repercusión económica por entrega, ya que las grandes matrices firman diferentes contratos con diferentes empresas y cada uno de esos contratos se firma a un precio diferente.

Esto genera incertidumbre, horas extra, desviación en los plazos de entrega, pérdida de material y descontento en el cliente final.

En el otro extremo del mercado está el concepto contrario altos precios y bajo volumen.

La empresa que representa este paradigma es Nacex, perteneciente a la cotizada grupo LOGISTA. Se distinguen por precios caros y garantías de entrega. Tienen unos tamaños y unos pesos máximos para cada elemento.

La limitación de peso y volumen así como un precio más alto ayuda a agilizar la operativa, y en última instancia redundan en un cumplimiento de sus garantías de entrega mejorando así la satisfacción del cliente. Otro aspecto importante es que aunque su estructura organizativa también se basa en franquiciados, estos cubren un área territorial bastante grande y para ello se obliga y se anima a contratar personal propio. Que la recomendación explícita de evitar cubrir puestos con autónomos

CONTOSOX replica el modelo de servicio enfocado en la satisfacción del cliente, con alto precio y escaso volumen, siempre con garantías de entrega.

d) Estructura organizativa de los recursos humanos

Una franquicia suele ser una Pyme dependiendo de la zona con hasta 10 empleados

Para los cometidos que se van a desarrollar, cada franquicia necesita contar con al menos un oficinista y al menos 3 repartidores con carnet de conducir a jornada completa.

La empresa matriz necesita tener a operarios a jornada partida para manipular la maquinaria de filtrado de paquetería en las centrales logísticas. La peculiaridad de este horario es que es de 20:00 a 23:00 y de 05:00 a 08:00

La empresa matriz no solo franquicia a las oficinas de reparto. Resuelve por el mismo modelo empresarial las rutas nocturnas entre plataformas logísticas.

Además la empresa matriz necesitará de un personal con capacitación en el sector IT como ingenieros informáticos, CF DAW o DAM para poder hacerse cargo del mantenimiento de los recursos informáticos.

e) Plan de marketing

A continuación vamos a relatar la gama de productos que ofrece la empresaria

Contosox ofrece solución completa al problema de la cadena logística, desde el punto de partida hasta su destino final en las manos del cliente. Para ello ofrece:

- Recogida en mano
- Transporte logístico por todo el territorio nacional
- Entrega en mano

Los precios se han fijado segmentando por tarifas según el target donde hemos decidido centrar nuestros esfuerzos. Segregando por precio, volumen físico y tamaño

Tras una prospección de mercado donde se han evaluado las preferencias de los potenciales clientes los nichos los precios de cada uno de los competidores, se ha contrastado con los costes operativos y se ha llegado a la conclusión de que para nuestro mercado las tarifas más adecuadas son las siguientes

- Económica

Coste de 5€ por recogida y envío a domicilio
tamaño de 10x25x35cm
peso de hasta 5kilogramo
entrega garantizada antes de 48 horas

- Hogar

Coste de 15€ por recogida y envío a domicilio
tamaño de 50x50x50cm
peso de hasta 7kilogramos
entrega garantizada antes de 48 horas

- Profesional

Coste de 50€ por recogida y envío a domicilio
tamaño de 100x100x100cm
peso de hasta 35kilogramos
entrega garantizada antes de 48 horas

Nos diferenciamos de nuestros competidores por incluir en el precio el servicio de recogida, siendo este un coste marginal en el que podemos incurrir sin mayor problema.

Los márgenes en la tarifa económica están ligeramente por encima del precio de coste con el objetivo de lograr una mayor penetración entre nuestros clientes.

Nuestra mejor publicidad es la satisfacción de nuestros clientes

En el rango más alto de nuestras tarifas estamos muy por encima de otros competidores por volumen y nos quedamos en el rango de los transportistas de calidad.

Esta tarifa esta pensada para pymes y profesionales especializados y transporte de aparatos de diagnóstico, aparatos médicos y veterinarios, aparatos de medición y ensayos no destructivos.

CONTOSOX ha elegido no anunciarse en los medios tradicionales de propaganda principalmente por no incurrir en gastos adicionales. Contosox cuenta con una masa de clientes habituales que crecen de acuerdo al crecimiento orgánico propio del sector. Es por ello que como estrategia defensiva en un entorno de inflación de costes y alta competitividad se ha decidido que esa partida de gasto se destine a mejorar protocolos de eficiencia interna.

f) Estados financieros previsionales a 3 años

En base a los resultado de 2021 preveremos que los pedidos aumenten de acuerdo al crecimiento orgánico del sector de un 5% anual que corresponde con la tasa de inflación prevista también de un 5%

Facturación 2021	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembr	Octubre	Noviembri	Diciembre
x10.000 envios	19	41	21	25	30	29	17	11	27	36	45	53
Economico	23750	51250	26250	31250	37500	36250	21250	13750	33750	45000	56250	66250
Hogar	81429	175714	90000	107143	128571	124286	72857	47143	115714	154286	192857	227143
Profesional	380000	820000	420000	500000	600000	580000	340000	220000	540000	720000	900000	1060000
Suma	485178,571	1046964	536250	638392,9	766071,4	740535,7	434107,1	280892,9	689464,3	919285,7	1149107	1353392,857
									SUMA anual			9.039.642,86 €

Balance 2021		
Facturacion		9.039.642,86 €
Costes operativos		6.327.750,00 €
Provisiones de amortización		451.982,14 €
Pago de la deuda		451.982,14 €
Beneficios después de impuestos		903.964,29 €

A continuación mostraremos la previsión de de estados de ingresos y gastos en una coyuntura estable donde la inflación se mantiene estable como actualmente entre el 2% y el 5%.

Los tipos de interés se mantendrán bajos y estables por lo que la inflación seguirá creciendo pero los costes de la deuda seguirán siendo bajos.

Facturación 2022												
	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembr	Octubre	Noviembri	Diciembre
x10.000 envios	19,95	43,05	22,05	26,25	31,5	30,45	17,85	11,55	28,35	37,8	47,25	55,65
Economico	24937,5	51250	26250	31250	37500	36250	21250	13750	33750	45000	56250	66250
Hogar	85500	175714	90000	107143	128571	124286	72857	47143	115714	154286	192857	227143
Profesional	399000	820000	420000	500000	600000	580000	340000	220000	540000	720000	900000	1060000
Suma	509437,5	1046964	536250	638392,9	766071,4	740535,7	434107,1	280892,9	689464,3	919285,7	1149107	1353392,857
									SUMA anual			9.063.901,79 €

Balance 2022			
Facturacion			9.063.901,79 €
Costes operativos			6.344.731,25 €
Provisiones de amortización			453.195,09 €
Pago de la deuda			453.195,09 €
Beneficios después de impuestos			906.390,18 €

Facturación 2023												
	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembr	Octubre	Noviembri	Diciembre
x10.000 envios	20,9475	45,2025	23,1525	27,5625	33,075	31,9725	18,7425	12,1275	29,7675	39,69	49,6125	58,4325
Economico	26184,375	56503,13	28940,63	34453,13	41343,75	39965,63	23428,13	15159,38	37209,38	49612,5	62015,63	73040,625
Hogar	89775	193725	99225	118125	141750	137025	80325	51975	127575	170100	212625	250425
Profesional	418950	904050	463050	551250	661500	639450	374850	242550	595350	793800	992250	1168650
Suma	534909,375	1154278	591215,6	703828,1	844593,8	816440,6	478603,1	309684,4	760134,4	1013513	1266891	1492115,625
									SUMA anual			9.966.206,25 €

Balance 2023			
Facturacion			9.966.206,25 €
Costes operativos			6.976.344,38 €
Provisiones de amortización			498.310,31 €
Pago de la deuda			498.310,31 €
Beneficios después de impuestos			996.620,63 €

Facturación 2024												
	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembr	Octubre	Noviembri	Diciembre
x10.000 envios	21,994875	47,46263	24,31013	28,94063	34,72875	33,57113	19,67963	12,73388	31,25588	41,6745	52,09313	61,354125
Economico	27493,5938	59328,28	30387,66	36175,78	43410,94	41963,91	24599,53	15917,34	39069,84	52093,13	65116,41	76692,65625
Hogar	94264	203411	104186	124031	148838	143876	84341	54574	133954	178605	223256	262946
Profesional	439897,5	949252,5	486202,5	578812,5	694575	671422,5	393592,5	254677,5	625117,5	833490	1041863	1227082,5
Suma	561654,844	1211992	620776,4	739019,5	886823,4	857262,7	502533,3	325168,6	798141,1	1064188	1330235	1566721,406
									SUMA anual			10.464.516,56 €

Balance 2024			
Facturacion			10.464.516,56 €
Costes operativos			7.325.161,59 €
Provisiones de amortización			523.225,83 €
Pago de la deuda			523.225,83 €
Beneficios después de impuestos			1.046.451,66 €

g) Análisis DAFO

	Origen interno	Origen externo
	Debilidades	Amenazas
Aspectos negativos	- Dependencia de transporte industrial cada vez más caro de mantener en una coyuntura donde la legislación esta volviéndose más restrictiva	- Fuga de la mano de obra a empresas con remuneraciones más altas - Competencia en el sector con dos polos precio y volumen.
	Fortalezas	Oportunidades
Aspectos positivos	- Atención al cliente - Alta retención de clientela - Precios competitivos - Garantías de entrega - Confianza y seguridad - Buenas valoraciones de los usuarios	- Sector en crecimiento - Nuevas necesidades de la población - Entregas en armarios tipo “Amazon” - Subvenciones al vehículo eléctrico industrial - Reducir tiempos de entrega

5. Desarrollo del proyecto

a) Revisión bibliográfica, fundamentación teórica.

Modelo cliente servidor

El modelo cliente-servidor describe cómo un servidor proporciona recursos y servicios a uno o más clientes. Los ejemplos de servidores incluyen servidores web, servidores de correo y servidores de archivos. Cada uno de estos servidores proporciona recursos a los dispositivos del cliente, como computadoras de escritorio, portátiles, tabletas y teléfonos inteligentes.

La mayoría de los servidores tienen una relación de uno a varios con los clientes, lo que significa que un solo servidor puede proporcionar recursos a varios clientes a la vez.

Cuando un cliente solicita una conexión a un servidor, el servidor puede aceptar o rechazar la conexión. Si se acepta la conexión, el servidor establece y mantiene una conexión con el cliente a través de un protocolo específico.

Por ejemplo, un cliente de correo electrónico puede solicitar una conexión SMTP a un servidor de correo para enviar un mensaje. La aplicación SMTP en el servidor de correo solicitará autenticación del cliente, como la dirección de correo electrónico y la contraseña. Si estas credenciales coinciden con una cuenta en el servidor de correo, el servidor enviará el correo electrónico al destinatario previsto.

Los servicios en línea populares distribuyen a los clientes a través de múltiples servidores físicos, utilizando una técnica llamada computación distribuida. En la mayoría de los casos, no importa a qué máquina específica estén conectados los usuarios, ya que todos los servidores brindan el mismo servicio.

Front end Estilos css con Bootstrap

Bootstrap es un framework CSS muy popular en el campo del desarrollo web.

Entre sus cualidades y beneficios están

- Ahorra tiempo: en poco tiempo puede desarrollar cualquier aplicación web. No puede escribir ningún código CSS en la hoja de estilo.

- Personalizable: Un gran aspecto de Bootstrap es que puede personalizarlo. Puede sentarse y hurgar en todo el marco y conservar lo que necesita y deshacerse de lo que no. El objetivo de Bootstrap es que le permite tener en cuenta sus propias necesidades y adaptar su proyecto de desarrollo en consecuencia. Esta es la razón principal por la que muchos desarrolladores sienten que algunas herramientas son completamente

inútiles en Bootstrap, mientras que otros reverencian lo mismo.

- **Coherencia:** una de las principales razones por las que se desarrolló este conjunto de herramientas fue porque Twitter estaba viendo grandes inconsistencias entre los desarrolladores que trabajaban en sus proyectos. Esto causó algunos problemas tanto en el frente del desarrollo como en el frente del usuario final. Dado que funciona con un conjunto central de código de desarrollo, los resultados de Bootstrap son uniformes en todas las plataformas. Verá lo mismo en Internet Explorer, Chrome y Firefox.

- **Responsive:** Bootstrap es totalmente sensible. Si cambia de una computadora portátil a un iPad y de un iPad a una Mac, no tendrá que preocuparse por su trabajo.

- **Soporte:** Bootstrap tiene una gran comunidad de soporte detrás, por lo que generalmente puede obtener ayuda cuando se encuentra con problemas. Además, Bootstrap en sí se actualiza continuamente y los creadores han sido muy buenos para publicar actualizaciones oportunas.

Front-End con React

React de JavaScript es una biblioteca para construir interfaces.

React funciona en el cliente como un SPA (Single Page App), pero puede ser usado para construir aplicaciones Full Stack comunicándose con un servidor o API.

React es frecuentemente referida como un framework de front end porque es capaz y directamente comparable a otros frameworks como Angular o Vue. Fue creada y es mantenida por Facebook.

React no se puede comunicar con una base de datos por ello suele usarse en combinación con otras tecnologías para crear una aplicación Full Stack.

En nuestro caso hemos usado un servidor de JsonServer y PHP Laravel en el BackEnd para crear las APIs. React carece de routes por eso tenemos que instalar un paquete extra que se llama react-router-dom.

¿Porque usar react?

React nos da una estructura para el “view” layer de nuestra aplicación.

MVC es un diseño popular en software.

El modelo trata los datos, el controller trata con las request y el routing, y el view es el UI (User Interface) lo que se muestra al usuario. React es la V del modelo MVC.

Una de las grandes ventajas es el concepto de Componentes reutilizables que pueden tener su propio estado y datos. React usa JSX o JavaScript Syntax Extension que nos permite escribir HTML dinámico. De hecho es JavaScript pero está formateado como HTML.

Las UI creadas son muy interactivas porque React utiliza el Virtual DOM que es el “Document Object Model” que permite actualizar partes de la pagina que necesitan ser actualizadas sin recargar la página.

Una característica de react es el concepto de componente.

¿Que es la reutilización de software y porqué es interesante?

Una definición de reutilización de software podría ser el proceso de creación de sistemas de software a partir de componentes de software predefinidos.

Unas características de la reutilización del software son :

- El desarrollo sistemático de componentes reutilizables.
- La reutilización sistemática de estos componentes como bloques de construcción para crear nuevos sistemas.

Un componente reutilizable puede ser el código, pero los mayores beneficios de la reutilización provienen de una visión más amplia y de mayor nivel de lo que se puede reutilizar. Las especificaciones de software, los diseños, los casos de prueba, los datos, los prototipos, los planos, la documentación, los marcos y las plantillas son todos candidatos para su reutilización. La reutilización de software puede reducir el tiempo y

los costos de desarrollo de software.

Las principales ventajas de la reutilización de software son:

- Aumento la productividad del software.
- Reducción en el tiempo de desarrollo de software.
- Mejora de la interoperabilidad del sistema de software.
- Desarrollo de software con menos gente.
- Muevilidad de personal más fácilmente de un proyecto a otro.
- Reducir los costos de desarrollo y mantenimiento de software.
- Producir software más estandarizado.
- Producir software de mejor calidad y proporcione una poderosa ventaja competitiva.

Mi backened es PHP Laravel y tiene la lógica de negocio ,el “Business Intelligence“.

Hay diferentes frameworks para construir robustas aplicaciones web pero Laravel retiene la posición numero uno por su flexibilidad que proporciona a los desarrolladores.

Laravel es un framework de desarrollo de aplicaciones web con una sintaxis que hace que todo el proceso de desarrollo web sea rápido, fácil y agradable para los desarrolladores al eliminar todos los puntos débiles asociados con el manejo de código PHP complejo.

Facilita algunas de las tareas comúnmente ejecutadas como routing , autenticación, sesiones, almacenamiento en caché y entre otras, para que los desarrolladores puedan concentrarse en crear funciones de la aplicación relacionadas con el negocio.

El framework Laravel fue creado por Taylor Otwell con la única intención de ayudar a los desarrolladores de sitios web a hacer que la codificación compleja sea simple, más rápida y mejor.

Entre los beneficios de usar Laravel tenemos:

- Eloquent ORM (object-relational-mapping)

Eloquent ORM en Laravel proporciona a los desarrolladores web una implementación simple de ActiveRecord que hace que su interacción con las bases de datos sea fácil y requiera menos tiempo. Les permite escribir consultas de bases de datos utilizando la sintaxis PHP, y no necesitan escribir ni conocer SQL para actualizar o modificar bases de datos.

- Arquitectura Laravel MVC

Laravel es un marco PHP basado en MVC que garantiza una estrecha separación entre las capas de presentación y la lógica empresarial. Al estar basado en MVC, el marco proporciona muchas características como alto rendimiento, mayor seguridad y escalabilidad.

- Código abierto y una gran comunidad

Aunque muchos frameworks PHP vienen con una etiqueta de precio, Laravel es un marco de código abierto para desarrolladores web. Además, tiene una comunidad poderosa que respalda el marco para hacerlo más avanzado y flexible.

- Bibliotecas orientadas a objetos

El framework PHP de Laravel está repleto de muchas bibliotecas integradas orientadas a objetos que están llenas de características sorprendentes para los desarrolladores.

Junto con estas fantásticas características del marco de Laravel, varias razones hacen que las organizaciones estén listas para seguir adelante con los servicios de desarrollo de Laravel. Estas mismas razones, o más bien los beneficios de Laravel Framework, se enumeran a continuación.

Para comunicar el Back-end con el Front-End he usado una API Rest tambien conocida como RESTful API es una interfaz de programación de aplicaciones (API o API web) que se ajusta a las limitaciones del estilo REST y permite la interacción con los servicios web RESTful. REST significa transferencia de estado representacional y fue creado por el científico informático Roy Fielding.

¿Qué es una API?

Una API es un conjunto de definiciones y protocolos para crear e integrar una aplicación de software. A veces tiene la apariencia de una relación entre un proveedor de información y un usuario de información, que establece el contenido requerido por el consumidor (la llamada) y el contenido requerido por el productor (la respuesta).

Para interactuar con una computadora o sistema para recuperar información o realizar una función, una API ayuda a comunicar lo necesario a ese sistema para que pueda comprender y cumplir la solicitud.

Una API puede verse como un mediador entre los usuarios o clientes y los recursos o servicios web que desean obtener. También es una forma de que una organización comparta recursos e información mientras mantiene la seguridad, el control y la autenticación, lo que determina quién tiene acceso a qué. Otra ventaja de una API es que no es necesario que conozca los detalles del almacenamiento en caché: cómo se recupera su recurso o de dónde proviene.

REST es un conjunto de restricciones arquitectónicas, no un protocolo ni un estándar. Los desarrolladores de API pueden implementar REST de diversas formas. Cuando una solicitud de cliente se realiza a través de una API RESTful, transfiere una representación del estado del recurso al solicitante o al punto final. Esta información, o representación, se entrega en uno de varios formatos a través de HTTP: JSON (Notación de Objetos Javascript), HTML, XML, Python, PHP o texto sin formato.

JSON es el formato de archivo más popular en general porque, a pesar de su nombre, es independiente del idioma y es legible tanto por humanos como por máquinas. Algo más a tener en cuenta: los encabezados y los parámetros también son importantes en los métodos HTTP de una solicitud HTTP de la API RESTful, ya que contienen información de identificación importante en cuanto a los metadatos de la solicitud, la autorización, el identificador uniforme de recursos (URI), el almacenamiento en caché, las cookies y más. Hay encabezados de solicitud y encabezados de respuesta, cada uno con su propia información de conexión HTTP y códigos de estado.

Para que una API se considere RESTful, debe cumplir con estos criterios:

- Una arquitectura cliente-servidor compuesta por clientes, servidores y recursos, con solicitudes gestionadas a través de HTTP.
- Comunicación cliente-servidor sin estado, lo que significa que no se almacena información del cliente entre las solicitudes de obtención y cada solicitud es independiente y no está conectada.
- Datos almacenables en caché que agilizan las interacciones cliente-servidor. Una interfaz uniforme entre componentes para que la información se transfiera de forma estándar.

Como bases de datos hemos usado MySQL desde PHPmyadmin

MySQL es un sistema de administración de bases de datos relacionales, lo que significa que almacena registros en tablas múltiples, separadas y altamente codificadas en lugar de un solo repositorio. Fue desarrollado por primera vez por una empresa sueca, MySQL AB, a mediados de la década de 1990 y su popularidad creció rápidamente después de que se convirtió en código abierto en 2000. Más tarde, Sun Microsystems compró MySQL AB, que a su vez fue adquirida por el gigante tecnológico estadounidense Oracle en 2010.

Hoy, MySQL es la segunda solución de bases de datos relacionales del mundo, según DB Engines. Sus usuarios incluyen una amplia gama de sitios web y aplicaciones, incluidas marcas domésticas como Spotify, Netflix, Facebook y Booking.com.

Beneficios de MySQL

- MySQL se desarrolló para brindar velocidad y mantiene la reputación de ser rápido, incluso si esto ocurre a expensas de algunas características adicionales. También es conocido por su confiabilidad, respaldado por una gran comunidad de programadores que han sometido el código a duras pruebas a lo largo de los años. Otro beneficio es que es relativamente simple de aprender y usar, y dado que ha existido durante casi tres décadas, no es difícil encontrar desarrolladores de MySQL experimentados cuando los necesita.
- Seguridad: esta es siempre una consideración importante para las empresas, ya que necesitan proteger los

datos confidenciales y defenderse de los ciberataques. MySQL ofrece cifrado mediante el protocolo Secure Sockets Layer (SSL), enmascaramiento de datos, complementos de autenticación y otras capas de seguridad para proteger la integridad de los datos. El paquete Enterprise también incluye protección por firewall contra ciberataques.

- Escalabilidad: a medida que aumentan los volúmenes de datos y las cargas de usuarios, la base de datos debe ampliarse para hacer frente a la carga de trabajo adicional sin una caída en el rendimiento. MySQL se puede escalar de diferentes formas, normalmente mediante replicación, agrupamiento o fragmentación (o una combinación de ellos). Es capaz de admitir y procesar bases de datos muy grandes, aunque es probable que esto tenga un impacto en la velocidad.

b) Materiales y métodos

Tecnologías previstas para su implementación

Elegiremos una solución centrada en tecnología web responsive que permita adaptarme al máximo número de dispositivos limitando el tiempo de desarrollo. Nota bibliográfica

Escogeré una arquitectura Cliente-Servidor usando el patrón Modelo-Vista-Controlador. Desarrollaré el frontend empleando tecnologías web habituales (HTML5, CSS3, JavaScript), aprovechando los frameworks de React y Bootstrap 4. Usaré React por ser el framework más utilizado en el sector empresarial, así como para poder aprovechar la experiencia adquirida en su desarrollo en mis prácticas actuales.

Emplearé la penúltima versión de Bootstrap, la 4, para los estilos CSS, por ser la versión con mayor soporte en cuanto a la comunidad y los manuales. En cuanto al backend, construiré una aplicación PHP con el framework Laravel, para aprovechar la experiencia adquirida durante mi formación.

Como servidor web emplearé Apache, el servidor más usado y con el que ya he trabajado durante el grado.

Así mismo, como almacén de datos aprovecharé el servidor SQL de MySQL.

Planeo ejecutar el backend PHP y el servidor MySQL sobre un sistema Linux, Ubuntu 20.04.

En cuanto a las tecnologías para el desarrollo:

- Emplearé Visual Studio Code y su gran repertorio de plugins para escribir el código en los diferentes lenguajes, interactuar con los repositorios git y acceder a los diferentes terminales.
- Usaré varios repositorios privados en GitHub para realizar el control de versiones con Git de las diferentes partes del proyecto.
- Emplearé VirtualBox para ejecutar una máquina virtual de Ubuntu 20.04 con la que realizar el desarrollo del backend.

c) Resultados y análisis

El proceso a lo largo del trimestre se ha dividido en tres partes

- El diseño
- La implementación del front end
- La implementación del back end con son base de datos.

El diseño inicial es una simplificación esquemática de un compendio de empresas del sector.

Se sacaron una lista de casos de uso. Estos casos de uso se comparó como se resolvían por parte de la competencia. ANEXO Casos de Uso

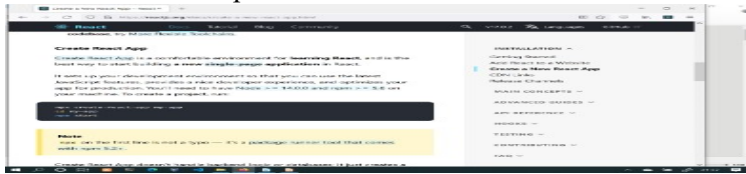
Para encontrar puntos comunes inicialmente se usó el dibujo a mano tras descartar el uso de plataformas como Figma por añadir complejidad. Anexo diseño a mano

Tras la representación manual se hizo una representación en html con estilos css en bootstrap quedando las páginas y el alcance de la aplicación definido.

Este fue el momento en el que se decidió empezar a hacer la aplicación React.

Para ello tenemos que instalar nodeJs, npm

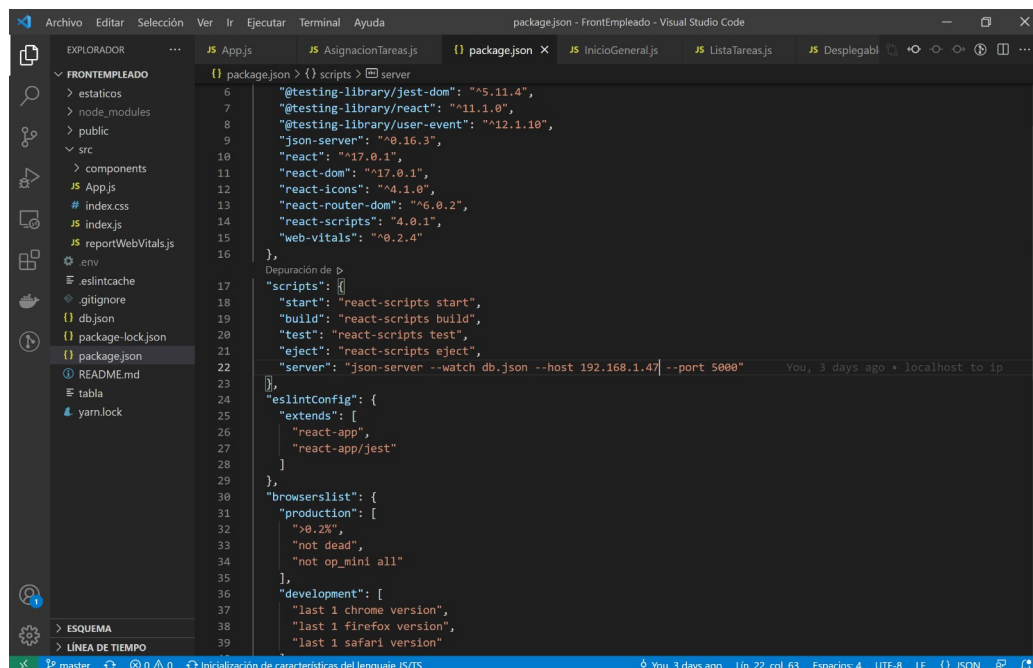
Después crearemos una aplicación React con:



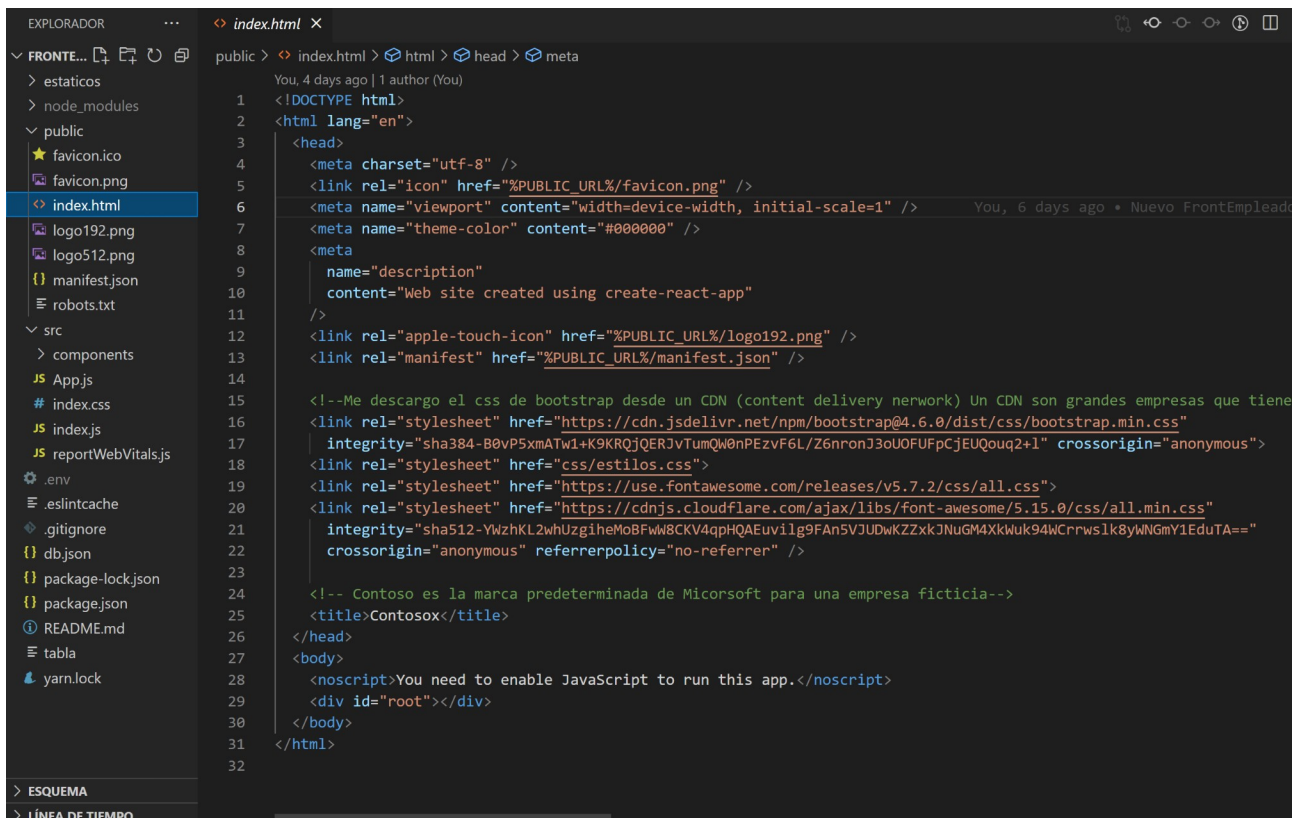
Una vez instalada podemos ejecutar la aplicación con

\$ npm start

Tenemos el archivo package.json que nos muestra versiones y scripts



El archivo index.html es un html normal con la peculiaridad de que sirve de punto de entrada a la aplicación react a través de id="root" en el <body>



```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <link rel="icon" href="%PUBLIC_URL%/favicon.png" />
6     <meta name="viewport" content="width=device-width, initial-scale=1" />
7     <meta name="theme-color" content="#000000" />
8     <meta
9       name="description"
10      content="Web site created using create-react-app"
11    />
12    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
13    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
14
15    <!--Me descargo el css de bootstrap desde un CDN (content delivery network) Un CDN son grandes empresas que tiene
16    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css"
17      integrity="sha384-B0vP5xmATw1+K9KRQjQERjvTumQW0nPEzvF6L/Z6nronJ3oU0FpCjEQUouq2+1" crossorigin="anonymous">
18    <link rel="stylesheet" href="css/estilos.css">
19    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css">
20    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@5.15.0/css/all.min.css"
21      integrity="sha512-YwzhK2whUzgiheMoBFw8CKV4qpHQAeuviIg9FAn5VJUDwKZZxkJNuGM4XKwuk94WCrrwslk8yWNGmY1EduTA=="
22      crossorigin="anonymous" referrerpolicy="no-referrer" />
23
24    <!-- Contoso es la marca predeterminada de Micorsoft para una empresa ficticia-->
25    <title>Contosox</title>
26  </head>
27  <body>
28    <noscript>You need to enable JavaScript to run this app.</noscript>
29    <div id="root"></div>
30  </body>
31 </html>
```

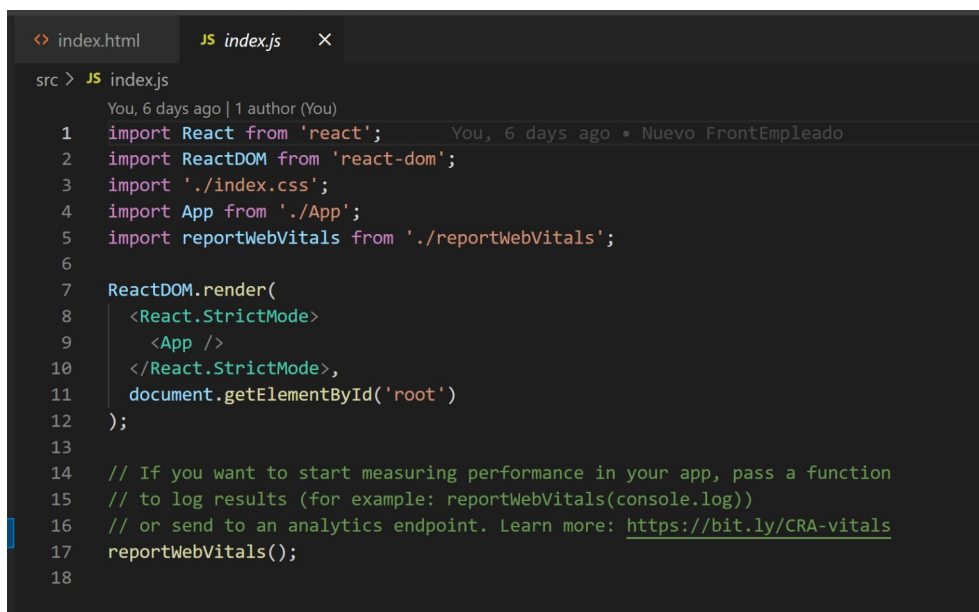
Me descargo el css de Bootstrap desde un CDN (content delivery network)

Un CDN son grandes empresas que tienen redes grandes de servidores distribuidos por todo el mundo que pueden ofrecer archivos muy usados con muy poca latencia, como en este caso bootstrap.

Con el fin por ejemplo de mejorar la experiencia de usuario a la hora de comprar en una tienda.

En la carpeta src es donde ponemos todos nuestros componentes

Dentro de esta carpeta tenemos el archivo index.js que es un poco como el punto de entrada para nuestra aplicación react



```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5 import reportWebVitals from './reportWebVitals';
6
7 ReactDOM.render(
8   <React.StrictMode>
9     <App />
10   </React.StrictMode>,
11   document.getElementById('root')
12 );
13
14 // If you want to start measuring performance in your app, pass a function
15 // to log results (for example: reportWebVitals(console.log))
16 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
17 reportWebVitals();
18
```

Dentro de ReactDOM tenemos el método render y estamos insertando nuestra aplicación en el div con id "root".

La aplicación está importada de App.js

App.js es el componente principal dentro hay una función App

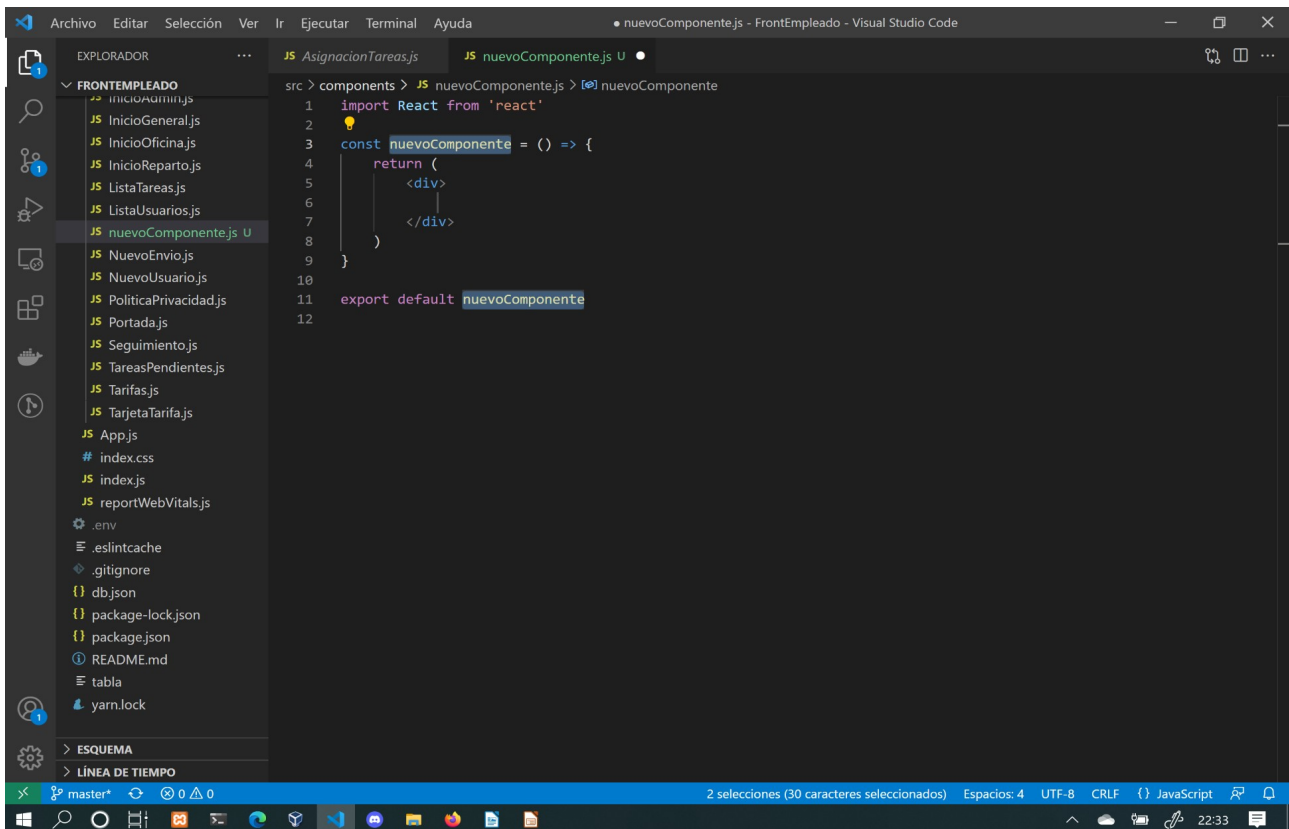
React utiliza JSX que es un javascript formateado como html.

Tiene unas diferencias como el atributo class que debe ser className o for que es htmlFor.

¿Cómo creamos un componente?

Creamos una carpeta components dentro de la cual creamos un componente.js

En Visual Studio Code con el atajo *rafce* nos crea un esquema de lo que lleva un componente

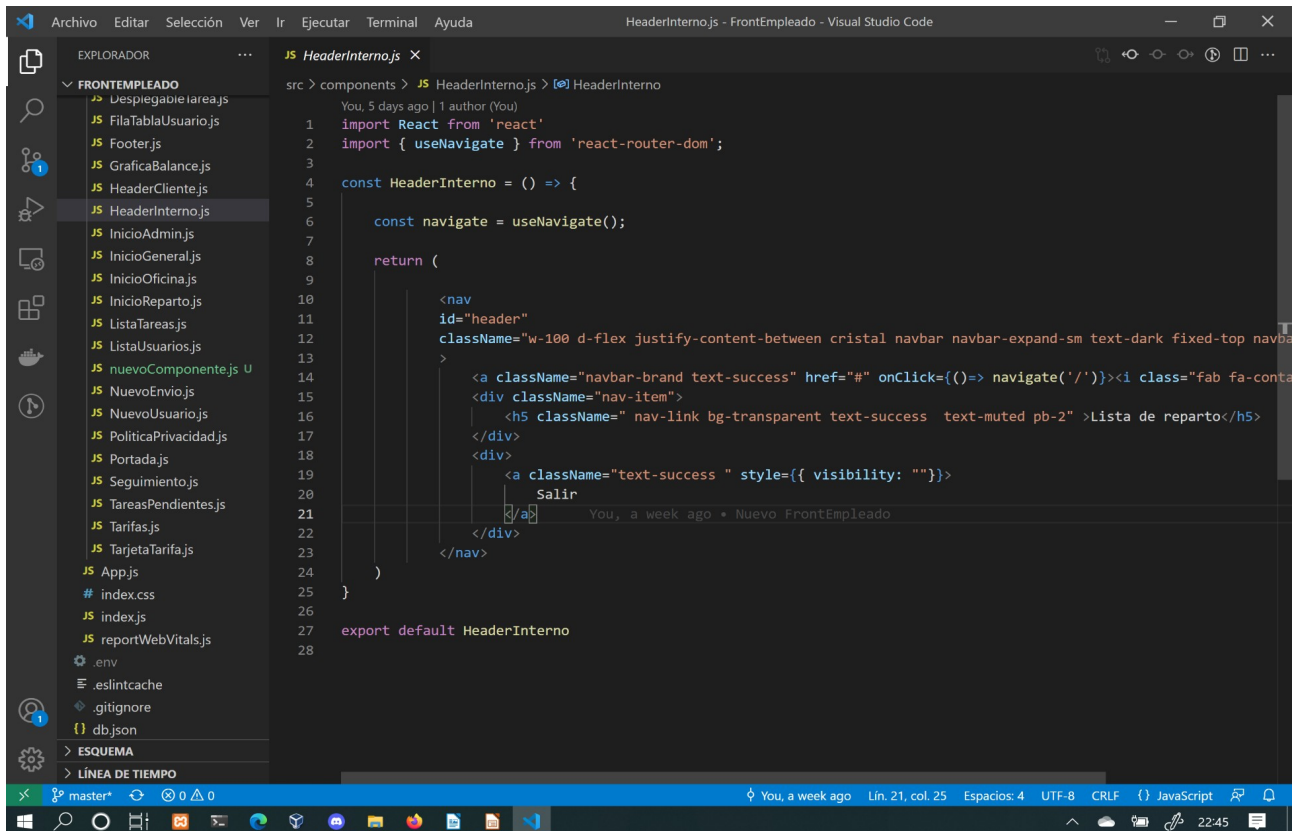


```
1 import React from 'react'
2
3 const nuevoComponente = () => {
4   return (
5     <div>
6
7     </div>
8   )
9 }
10
11 export default nuevoComponente
12
```

Todo lo que debe contener el return del componente debe estar en un único elemento JSX

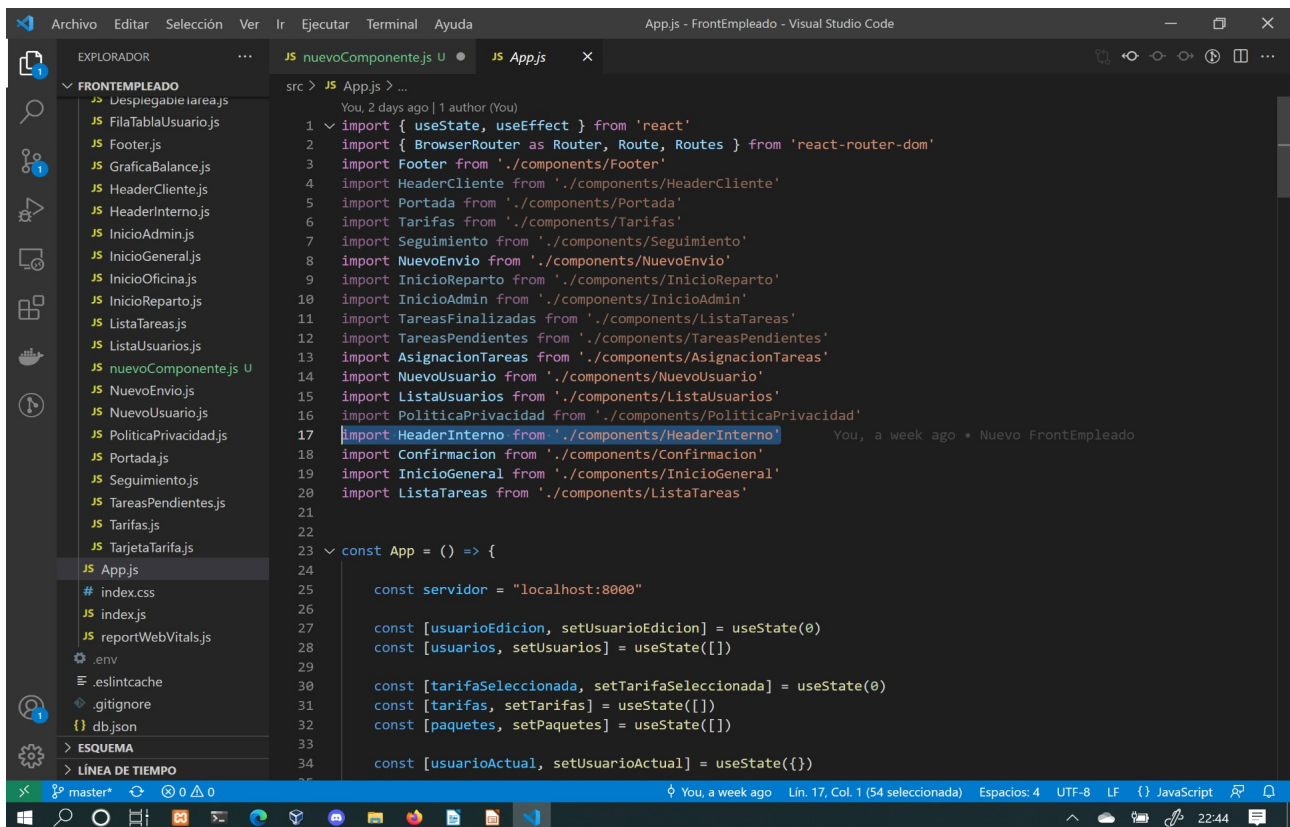
Import React from 'react' no es necesario

El componente debe ser exportable

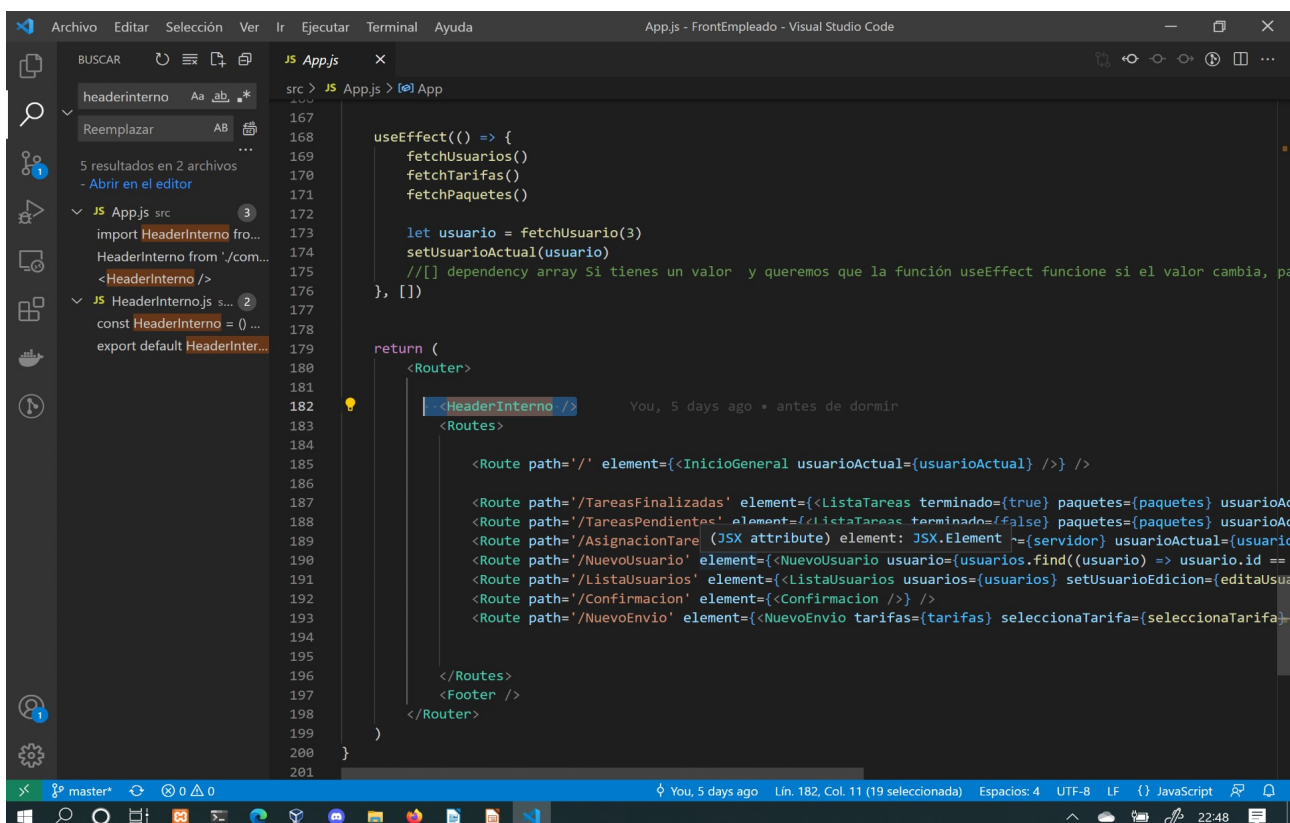


```
src > components > JS HeaderInterno.js > HeaderInterno
You, 5 days ago | 1 author (You)
1 import React from 'react'
2 import { useNavigate } from 'react-router-dom';
3
4 const HeaderInterno = () => {
5
6     const navigate = useNavigate();
7
8     return (
9
10         <nav
11             id="header"
12             className="w-100 d-flex justify-content-between cristal navbar navbar-expand-sm text-dark fixed-top navbs
13         >
14             <a className="navbar-brand text-success" href="#" onClick={()=> navigate('/')}><i class="fab fa-conta
15             <div className="nav-item">
16                 <h5 className=" nav-link bg-transparent text-success text-muted pb-2">Lista de reparto</h5>
17             </div>
18             <div>
19                 <a className="text-success " style={{ visibility: ""}}>
20                     Salir
21                 </a>
22             </div>
23         </nav>
24     )
25 }
26
27 export default HeaderInterno
28
```

El componente luego tiene que ser importado en App.js y entonces puede ser incluido en lo que devuelve en el return



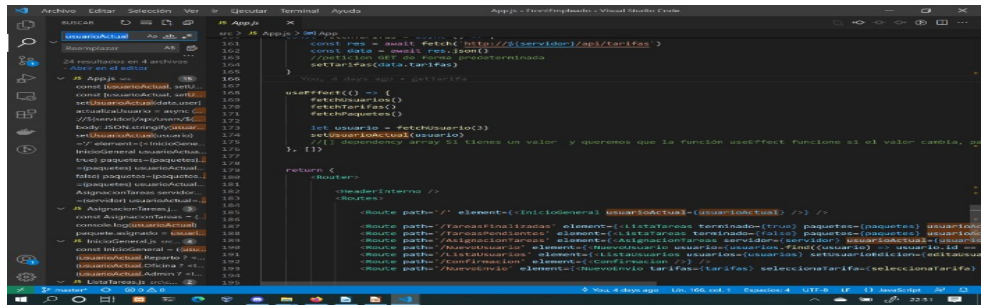
```
src > JS App.js > ...
You, 2 days ago | 1 author (You)
1  import { useState, useEffect } from 'react'
2  import { BrowserRouter as Router, Route, Routes } from 'react-router-dom'
3  import Footer from './components/Footer'
4  import HeaderCliente from './components/HeaderCliente'
5  import Portada from './components/Portada'
6  import Tarifas from './components/Tarifas'
7  import Seguimiento from './components/Seguimiento'
8  import NuevoEnvio from './components/NuevoEnvio'
9  import InicioReparto from './components/InicioReparto'
10 import InicioAdmin from './components/InicioAdmin'
11 import TareasFinalizadas from './components/ListaTareas'
12 import TareasPendientes from './components/TareasPendientes'
13 import AsignacionTareas from './components/AsignacionTareas'
14 import NuevoUsuario from './components/NuevoUsuario'
15 import ListaUsuarios from './components/ListaUsuarios'
16 import PoliticaPrivacidad from './components/PoliticaPrivacidad'
17 import HeaderInterno from './components/HeaderInterno'
18 import Confirmacion from './components/Confirmacion'
19 import InicioGeneral from './components/InicioGeneral'
20 import ListaTareas from './components/ListaTareas'
21
22
23 const App = () => {
24
25     const servidor = "localhost:8000"
26
27     const [usuarioEdicion, setUsuarioEdicion] = useState(0)
28     const [usuarios, setUsuarios] = useState([])
29
30     const [tarifaSeleccionada, setTarifaSeleccionada] = useState(0)
31     const [tarifas, setTarifas] = useState([])
32     const [paquetes, setPaquetes] = useState([])
33
34     const [usuarioActual, setUsuarioActual] = useState({})
35 }
```



```
src > JS App.js > App
167
168
169 useEffect(() => {
170     fetchUsuarios()
171     fetchTarifas()
172     fetchPaquetes()
173
174     let usuario = fetchUsuario(3)
175     setUsuarioActual(usuario)
176     //[] dependency array Si tienes un valor y queremos que la función useEffect funcione si el valor cambia, pa
177 }, [])
178
179
180 return (
181     <Router>
182     <HeaderInterno />
183     <Routes>
184
185         <Route path="/" element={<InicioGeneral usuarioActual={usuarioActual} />} />
186
187         <Route path="/TareasFinalizadas" element={<ListaTareas terminado={true} paquetes={paquetes} usuarioAc
188         <Route path="/TareasPendientes" element={<ListaTareas terminado={false} paquetes={paquetes} usuarioAc
189         <Route path="/AsignacionTare (JSX attribute) element: JSX.Element ~=(servidor) usuarioActual={usuario
190         <Route path="/NuevoUsuario" element={<NuevoUsuario usuario={usuarios.find((usuario) => usuario.id ==
191         <Route path="/ListaUsuarios" element={<ListaUsuarios usuarios={usuarios} setUsuarioEdicion={editaUsua
192         <Route path="/Confirmacion" element={<Confirmacion />} />
193         <Route path="/NuevoEnvio" element={<NuevoEnvio tarifas={tarifas} seleccionaTarifa={seleccionaTarifa}
194
195     </Routes>
196     <Footer />
197 </Router>
198
199 )
200
201 }
```

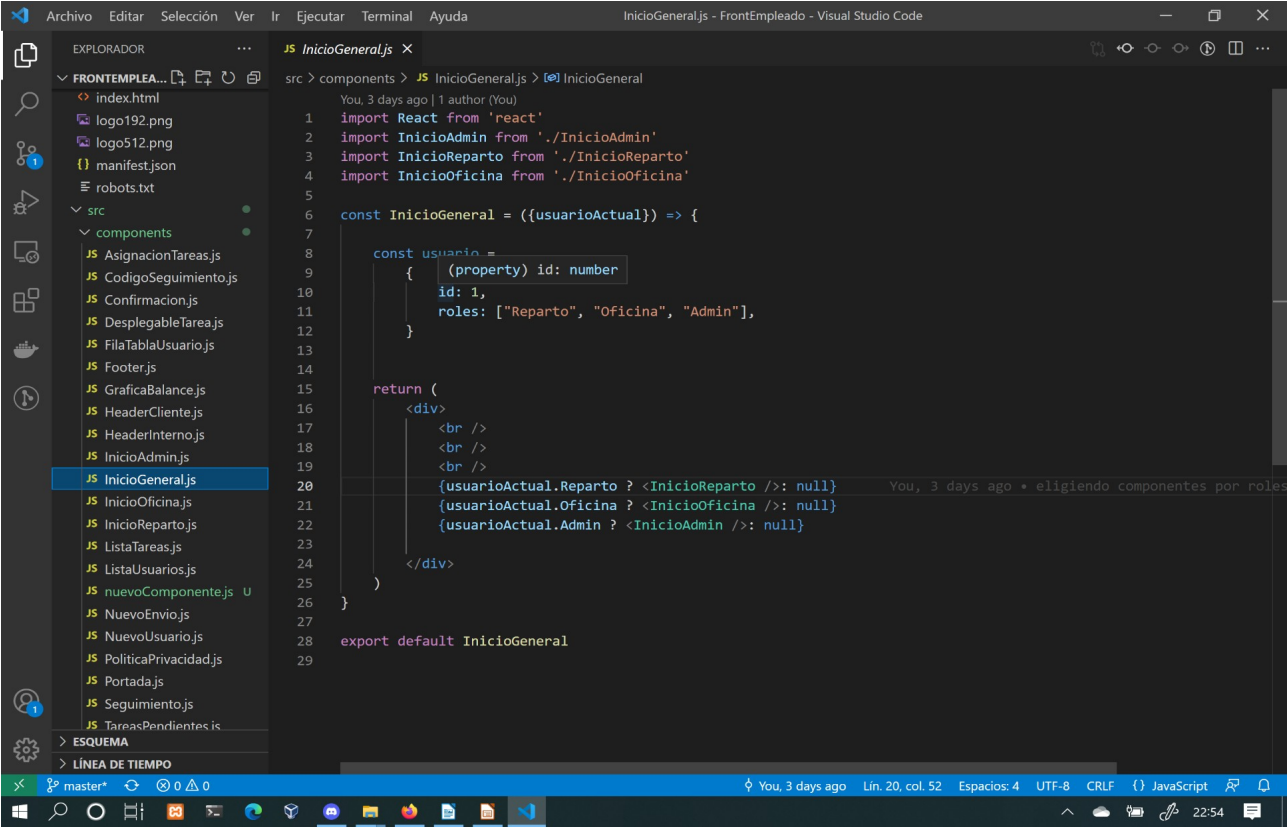
Podemos pasar props dentro del componente

En este ejemplo pasamos el resultado de la función setUsuarioActual que como veremos más adelante establece el resultado de usuarioActual dentro del componente.



```
101 // ...
102 const res = await Fetch(`${url}/${usuario}/api/tarifas`);
103 const data = await res.json();
104 // ...
105 // ...
106 // ...
107 // ...
108 // ...
109 // ...
110 // ...
111 // ...
112 // ...
113 // ...
114 // ...
115 // ...
116 // ...
117 // ...
118 // ...
119 // ...
120 // ...
121 // ...
122 // ...
123 // ...
124 // ...
125 // ...
126 // ...
127 // ...
128 // ...
129 // ...
130 // ...
131 // ...
132 // ...
133 // ...
134 // ...
135 // ...
136 // ...
137 // ...
138 // ...
139 // ...
140 // ...
141 // ...
142 // ...
143 // ...
144 // ...
145 // ...
146 // ...
147 // ...
148 // ...
149 // ...
150 // ...
151 // ...
152 // ...
153 // ...
154 // ...
155 // ...
156 // ...
157 // ...
158 // ...
159 // ...
160 // ...
161 // ...
162 // ...
163 // ...
164 // ...
165 // ...
166 // ...
167 // ...
168 // ...
169 // ...
170 // ...
171 // ...
172 // ...
173 // ...
174 // ...
175 // ...
176 // ...
177 // ...
178 // ...
179 // ...
180 // ...
181 // ...
182 // ...
183 // ...
184 // ...
185 // ...
186 // ...
187 // ...
188 // ...
189 // ...
190 // ...
191 // ...
192 // ...
193 // ...
194 // ...
195 // ...
196 // ...
197 // ...
198 // ...
199 // ...
200 // ...
```

Dentro del componente se pasa el prop entre llaves como argumento de la arrow function



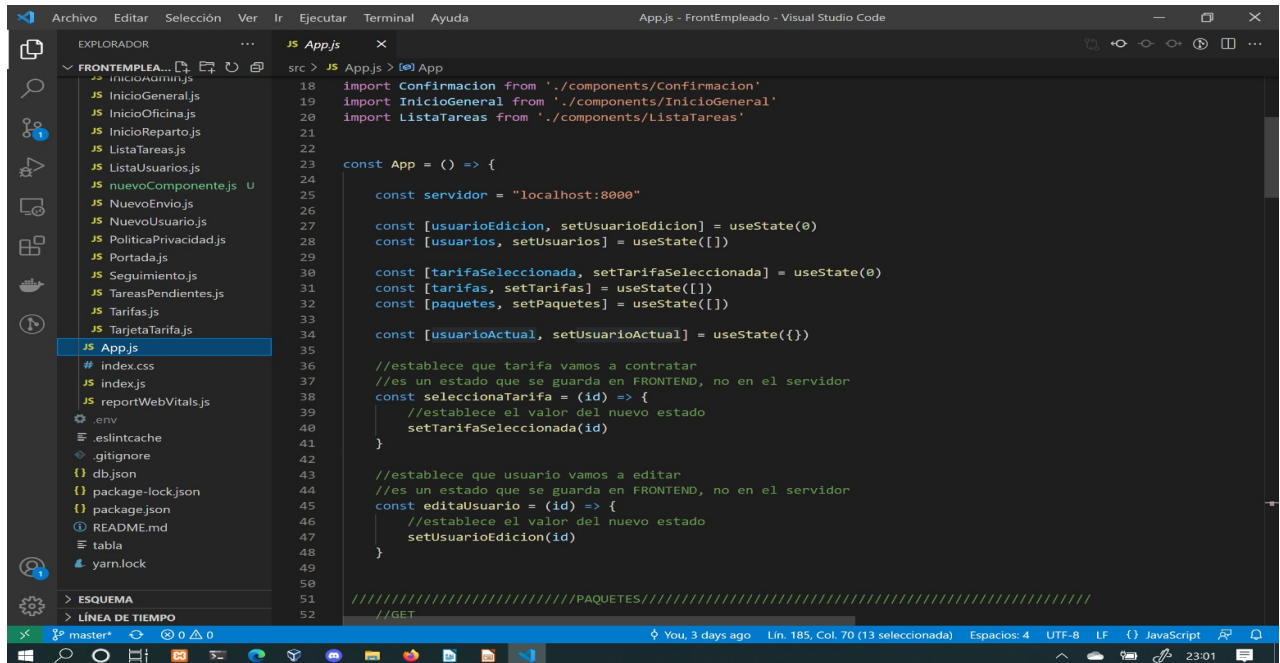
```
1 import React from 'react'
2 import InicioAdmin from './InicioAdmin'
3 import InicioReparto from './InicioReparto'
4 import InicioOficina from './InicioOficina'
5
6 const InicioGeneral = ({usuarioActual}) => {
7
8   const usuario =
9     {
10       (property) id: number
11       id: 1,
12       roles: ["Reparto", "Oficina", "Admin"],
13     }
14
15   return (
16     <div>
17       <br />
18       <br />
19       <br />
20       {usuarioActual.Reparto ? <InicioReparto />: null}
21       {usuarioActual.Oficina ? <InicioOficina />: null}
22       {usuarioActual.Admin ? <InicioAdmin />: null}
23     </div>
24   )
25 }
26
27 export default InicioGeneral
```

Los props sirven para pasar variables o funciones entre componentes (de padres a hijos).

- Las variables sirven para guardar estados de la página.

- Las funciones sirven para realizar acciones en la pagina por ejemplo:

1. Cambiar estados con las funciones setEstado

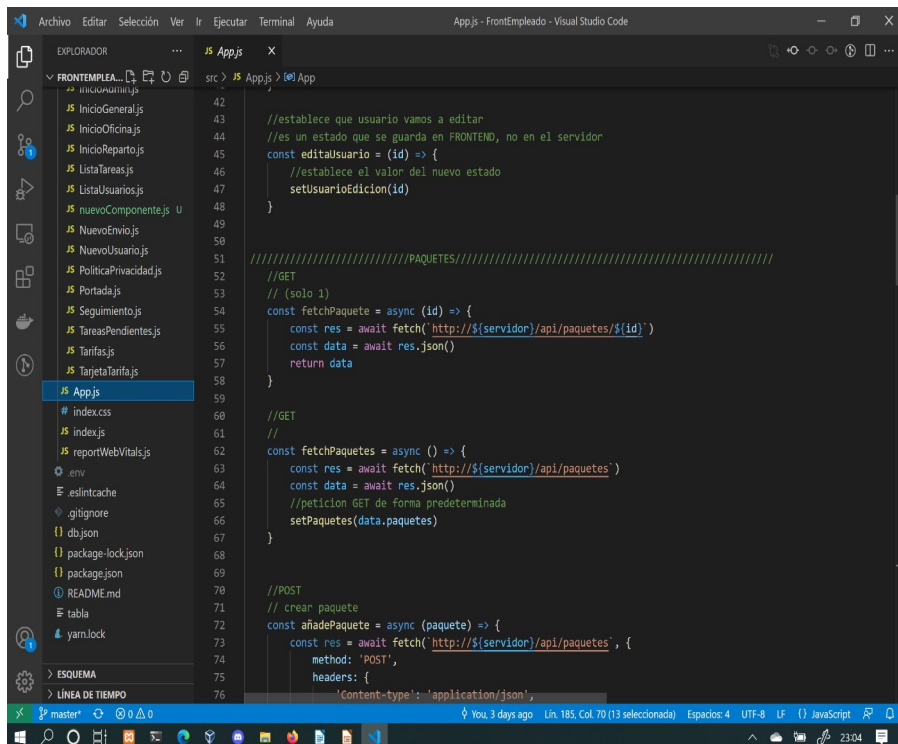


```

18 import Confirmacion from './components/Confirmacion'
19 import InicioGeneral from './components/InicioGeneral'
20 import ListaTareas from './components/ListaTareas'
21
22
23 const App = () => {
24
25   const servidor = "localhost:8000"
26
27   const [usuarioEdicion, setUsuarioEdicion] = useState(0)
28   const [usuarios, setUsuarios] = useState([])
29
30   const [tarifaSeleccionada, setTarifaSeleccionada] = useState(0)
31   const [tarifas, setTarifas] = useState([])
32   const [paquetes, setPaquetes] = useState([])
33
34   const [usuarioActual, setUsuarioActual] = useState({})
35
36   //establece que tarifa vamos a contratar
37   //es un estado que se guarda en FRONTEND, no en el servidor
38   const seleccionaTarifa = (id) => {
39     //establece el valor del nuevo estado
40     setTarifaSeleccionada(id)
41   }
42
43   //establece que usuario vamos a editar
44   //es un estado que se guarda en FRONTEND, no en el servidor
45   const editaUsuario = (id) => {
46     //establece el valor del nuevo estado
47     setUsuarioEdicion(id)
48   }
49
50
51   //////////////////////////////////PAQUETES////////////////////////////////////
52   //GET

```

2. Las funciones para hacer un fetch para comunicarme con el back end a traves de la api rest



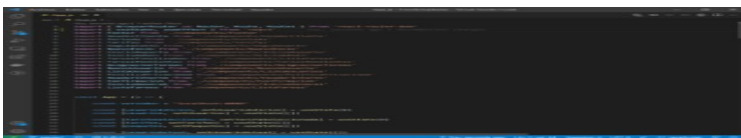
```

42 //establece que usuario vamos a editar
43 //es un estado que se guarda en FRONTEND, no en el servidor
44 const editaUsuario = (id) => {
45   //establece el valor del nuevo estado
46   setUsuarioEdicion(id)
47 }
48
49
50 //////////////////////////////////PAQUETES////////////////////////////////////
51 //GET
52 // (solo 1)
53 const fetchPaquete = async (id) => {
54   const res = await fetch(`http://${servidor}/api/paquetes/${id}`)
55   const data = await res.json()
56   return data
57 }
58
59 //GET
60 //
61 const fetchPaquetes = async () => {
62   const res = await fetch(`http://${servidor}/api/paquetes`)
63   const data = await res.json()
64   //petición GET de forma predeterminada
65   setPaquetes(data.paquetes)
66 }
67
68 //POST
69 // crear paquete
70 const añadePaquete = async (paquete) => {
71   const res = await fetch(`http://${servidor}/api/paquetes`, {
72     method: 'POST',
73     headers: {
74       'Content-Type': 'application/json',

```

Para usar rutas

tenemos que importar BrowserRouter de react router dom

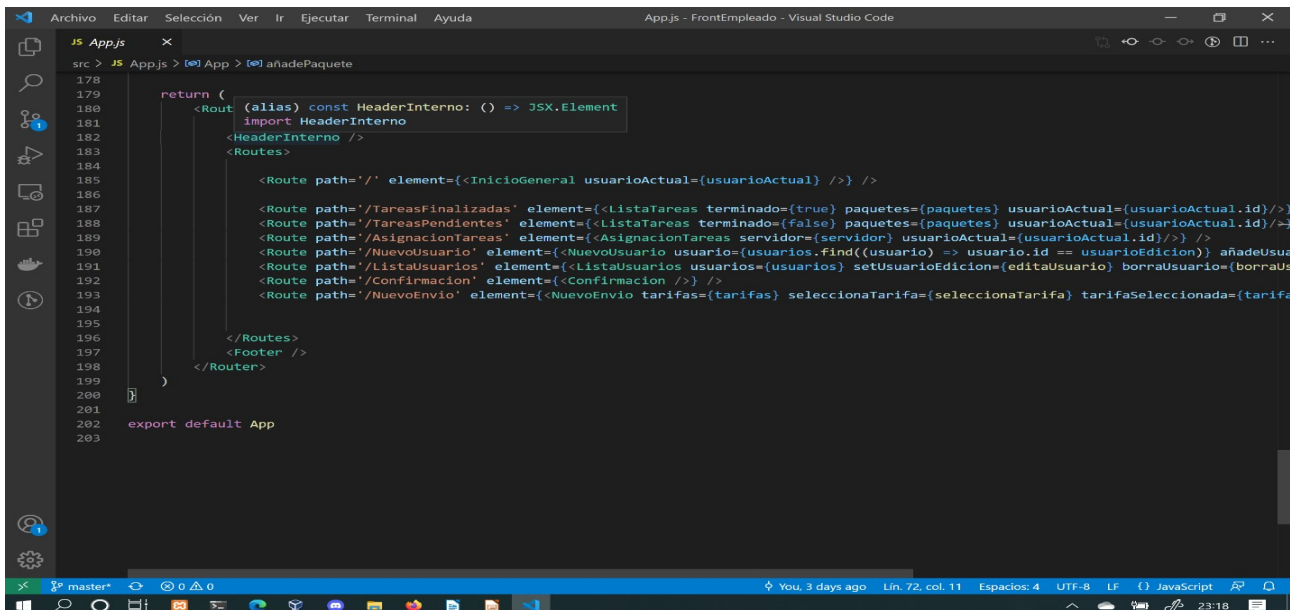


Routes es un componente de react que se coloca en JSX que sirven para indicar que componentes hay que

renderizar según la ruta

<Route> se puede agrupar dentro de <Routes>

<Route tiene dos atributos path indica la url element indica los componentes que se le van a pasar como props



```
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203

return (
  <Route (alias) const HeaderInterno: () => JSX.Element
    import HeaderInterno
  <HeaderInterno />
  <Routes>

    <Route path="/" element={(<InicioGeneral usuarioActual={usuarioActual} />)} />

    <Route path="/TareasFinalizadas" element={(<ListaTareas terminado={true} paquetes={paquetes} usuarioActual={usuarioActual.id}/>)} />
    <Route path="/TareasPendientes" element={(<ListaTareas terminado={false} paquetes={paquetes} usuarioActual={usuarioActual.id}/>)} />
    <Route path="/AsignacionTareas" element={(<AsignacionTareas servidor={servidor} usuarioActual={usuarioActual.id}/>)} />
    <Route path="/NuevoUsuario" element={(<NuevoUsuario usuario={usuarios.find(usuario => usuario.id === usuarioEdicion)} añadeUsua
    <Route path="/ListaUsuarios" element={(<ListaUsuarios usuarios={usuarios} setUsuarioEdicion={editaUsuario} borraUsuario={borraUs
    <Route path="/Confirmacion" element={(<Confirmacion />)} />
    <Route path="/NuevoEnvio" element={(<NuevoEnvio tarifas={tarifas} seleccionaTarifa={seleccionaTarifa} tarifaSeleccionada={tarifa

  </Routes>
  <Footer />
</Router>
)

export default App
```

El back end consiste en MySQL con una base de datos de unas pocas tablas que son

- users
- paquetes
- tarifas
- localizaciones

Como intermediario empleo una instalación estándar de Laravel a la que le he añadido la extensión oficial para APIs REST por lo que no he usado vistas, solo directamente controladores

d) Conclusiones

He conseguido las siguientes características

- Debe poder ser accesible desde móvil y pc.
- Ver las tarifas para los que he necesitado el uso de estado.
- Navegar por la página de forma sencilla para el usuario
- El usuario puede solicitar una recogida a domicilio.
- Seguir el estado de un envío.
- Crear un gestor de usuarios
- Gestion de empleados con sus roles dentro de la empresa.
- Cada empleado debe poder asignarse sus propias tareas.
- He creado un sistema para asignar paquetes y confirmar entregas y recogidas.

- Ciclo de vida del paquete con seguimiento de su estado, localización y persona responsable.

Cuales no he podido

He intentado asumir un proyecto muy grande en solitario que al principio me parecía asumible pero me he ido encontrando con escollos que no he podido solucionar por la falta de tiempo y conocimiento, y falta de experiencia con tecnologías.

React y las API REST por ejemplo no las conocía antes de comenzar el proyecto

Y lo he comenzado ha aprender

No he podido implementar ni usuarios ni seguimiento del paquete.

Los botones colapsables eran una propiedad de bootstrap que no tenía una apropiada transición a react

Cuales he decidido dejar fuera por complejidad

Tampoco he podido implementar pagos en su momento encontré una API para paypal con opciones de prueba para desarrolladores pero he tenido tantas otras cosas que hacer en el proyecto que no he podido volver a verlo.

e) Líneas de investigación futuras

- Pago por PayPal usando su API
- Generación y lectura de códigos de barras
- Generación y lectura de códigos Qr
- Seguimiento gps de la ruta
- Organización de rutas óptimas de reparto

6. Bibliografía

React JS Crash Course 2021

<https://www.youtube.com/watch?v=w7ejDZ8SWv8sadf>

Cliente

• **Contrata un servicio**

1. Rellena datos necesarios

Se muestra un formulario con todos los campos donde el usuario rellenará los datos necesarios

1. E-mail
2. Solicitante(remitente)
3. Teléfono
4. NIF

Datos de recogida

5. Fecha de recogida(date)
6. Horario de recogida(10:00, 15:00, 17:00, 19:00)
7. España/Portugal/Andorra
8. CP
9. Población
10. Dirección (Nombre, Via, número, piso)
11. Empresa /persona de recogida
12. Telefono
13. Persona de contacto
14. Observaciones de la recogida

Datos de entrega

1. España/Portugal/Andorra
2. CP
3. Poblacion
4. Dirección(Nombre, via, número, piso)
5. Destinatario
6. Telefono
7. Observaciones de la entrega
- 8.

Información del servicio

1. tipo de servicio(8:30, 10:00, 12:00, 19:00)
2. Envase(paquete/bolsa)
3. N.º de bultos
4. Peso aprox

checkbox* **campos obligatorios**

He leído y acepto [la política de privacidad y condiciones de uso de la web](#)

Debe escoger tipo de tarifa y describir el paquete

si no se marcan los datos OBLIGATORIOS devolverá un error.

- Recogida en casa
- Dejar en la oficina

2. Paga

Dinero que cuesta(mostrar una factura)
Escoger métodos de pago
En una única página

3.página de confirmación pago exitoso/datos de identificación/

- NECESITA GENERAR(PDF) E IMPRIMIR ETIQUETA

4. Recibe correo de confirmación

• Muestra tabla de precios

dos desplegables donde el cliente selecciona el servicio que quiere
y al confirmar le sale el precio.

Desplegable

Inferior a 1kg

Superior a 20kg

Antes de las 10h

Entrega en el día

• Consulta seguimiento de envío

Usuario introduce el número de identificación envío.

Señala estado actual

Obtiene lista de datos por los que el paquete ha pasado

recogida	hora
plataforma logística	hora
en proceso de entrega	hora
...	

Repartidor

1. Pagina de registro (usuario/contraseña)
 2. Añadir paquetes
 3. Mostrar lista de tareas
 - hora de entrega/población/dirección abreviada(calle número)
 - Las tareas realizadas están marcados de otra forma
 4. Ver detalles tarea
nombre / dirección / cp / numero de teléfono / horario de entrega/numero de identificación del paquete
- Entrega paquete

formulario rellenar NOMBRE COMPLETO // DNI
 - selecciono paquete/reconocer paquete
 - procedimiento entrega
 - entrega válida
 - entrega fallida
 - Recoger envío
 - Lectura del código qr
 - Confirmar entrega con formulario dni/nombre

Oficina

- Loguearse
- Entrega paquete
Igual que en el repartidor
- Recoger envío
 - Formulario datos
 - Pago

si es válido el proceso => pagina de aterrizaje exitoso con opcion de descarga de documentos
Generamos etiqueta
Generar factura
Escanear etiqueta qr

- Lista de tareas pendientes empresa
 - Panel de tareas completadas y por hacer.
 - Filtrar por repartidor
- Localización de repartidor
 - mapa en tiempo real

Administración

- Gestión de usuarios
- Gestión de cuentas y pasarelas de pago(Paypal)