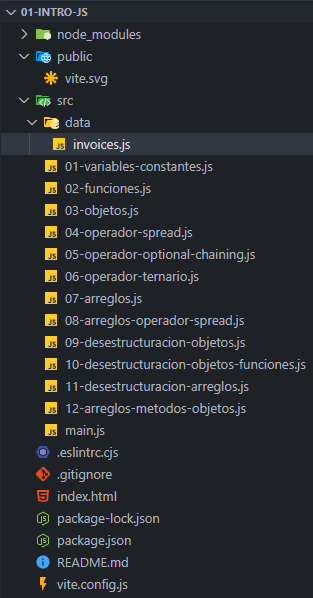
Importar y exportar módulos en JavaScript

En esta sección nos vamos a enfocar en importar y exportar un arreglo y una función que se encuentran en un archivo externo hacia el archivo raíz.

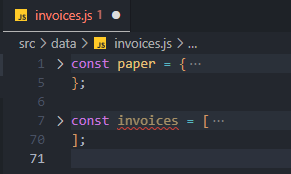
Separar los objetos del archivo main

Los módulos en JavaScript permiten organizar el código en archivos separados y reutilizarlo en diferentes partes de una aplicación. Los módulos pueden exportar funciones, objetos, clases, etc., para que puedan ser importados y utilizados en otros archivos.

Para empezar, crea una nueva carpeta con el nombre **data** dentro de la carpeta **src**. Dentro de esa carpeta crea el archivo **invoices.js**.



Este archivo contendrá el código fuente del objeto **paper** y el arreglo **invoices** (solamente copia y pega el código fuente que se encuentra en la sección anterior).

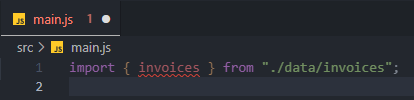


Importar datos al archivo main

Sigue estos pasos:

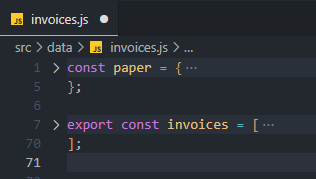
Paso 1: Importar el arreglo invoices al archivo main

Utiliza la sintaxis **import { } from " "** para importar el módulo deseado desde el archivo correspondiente. La ruta se construye con respecto a la carpeta raíz **src**, usando **./**.



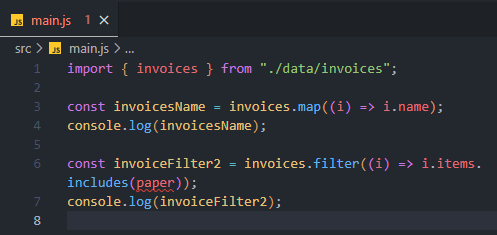
Paso 2: Exportar el arreglo invoices del archivo invoices

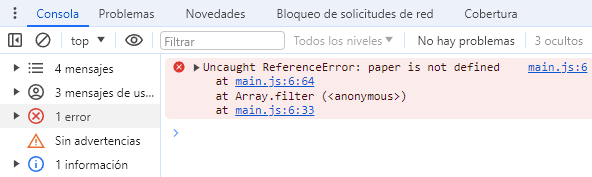
Para que un módulo pueda ser importado en otros archivos, debes marcarlo como exportable utilizando la palabra clave **export** antes de la declaración de la constante **invoices**.



Paso 3: Agregar una función al archivo main

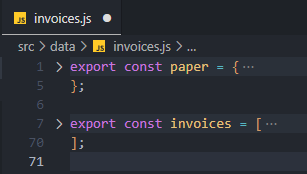
Puedes utilizar los métodos **map** y **filter** para manipular el objeto **invoices** importado (ten en cuenta que las funciones **invoicesName** e **invoiceFilter2** se han definido en la sección anterior). Luego procede a ejecutar la aplicación.



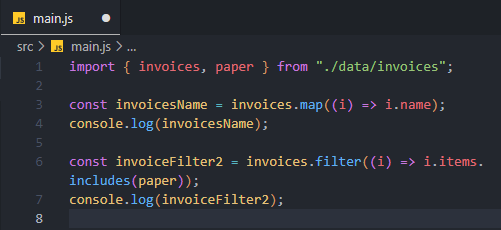


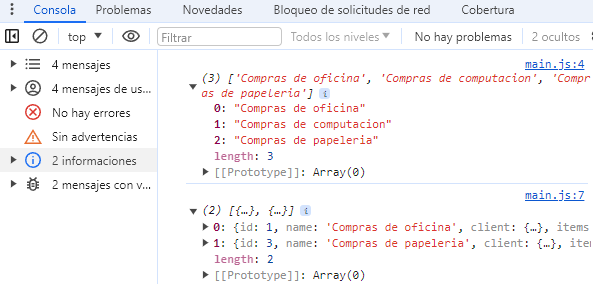
Paso 4: Importar y exportar el objeto paper

Para resolver el error mencionado en la consola (indica que **paper** no está definido), marca el objeto **paper** como exportable en el archivo **invoices.js**.



Luego, importa el objeto **paper** en el archivo **main.js**. Ten en cuenta que el orden no importa al momento de importar.



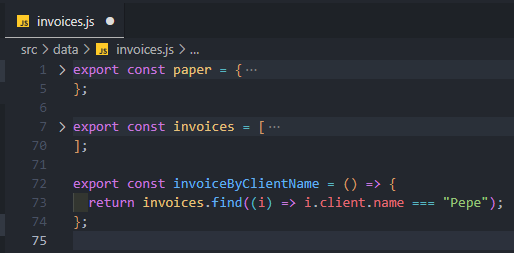


Importar una función al archivo main

Como ya se había mencionado antes, también se puede exportar funciones con la finalidad de reutilizarlo en algún otro archivo. En este mismo contexto, se puede tener un archivo JavaScript que contenga varias funciones y se pueda utilizar para que sea exportables.

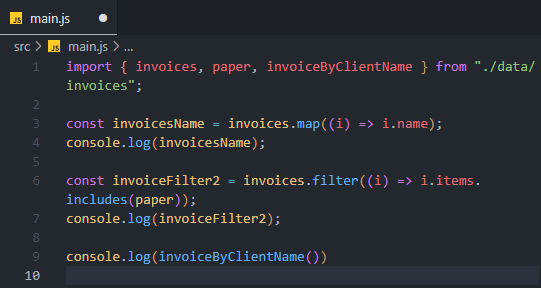
Paso 1: Definir una función en el archivo invoices

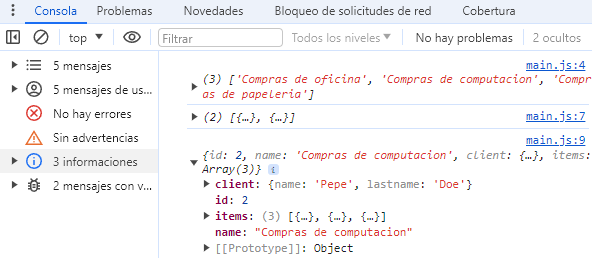
Se define la función **invoiceByClientName** (el mismo de la sección anterior). No olvidar colocar un **export** al inicio de la constante que define la función.



Paso 2: Importar la función en el archivo main

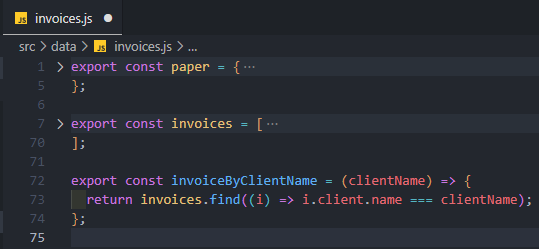
Importa la función dentro de las llaves **{ }** de **import**. Asimismo, también se define la función importada para que lo imprima en la consola:



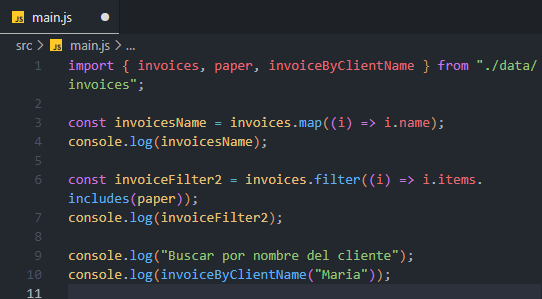


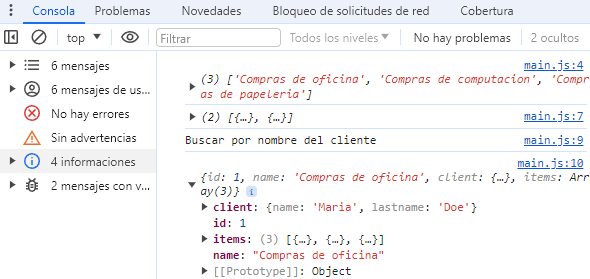
Paso 3: Importar una función que posee un parámetro

Algunas funciones pueden recibir un argumento para que sean más reutilizables y no devuelvan siempre el mismo valor. Para ello, se define un nuevo parámetro en la función **invoiceByClientName** (se encuentra en el archivo **invoices.js**).



Luego, procede a pasar ese parámetro al llamar a la función. En el archivo **main.js**, defina el valor (argumento) que va a tener ese parámetro.





En el ejemplo mostrado, la función **invoiceByClientName**, busca la primera factura que contenga el nombre **"Maria"** en el atributo **name** del objeto anidado **client**, la función está definida en el archivo **invoices.js**.