PROCESO BACK END PAPAS ACME, S.A

HTML:

Lo primero a realizar en el proyecto es una cabecera (header.php) y un index.php .

En el **header.php** simplemente ponemos ciertos metadatos, y en el **index.php** creamos un formulario para poder introducir los ficheros que nos proporciona la empresa, estos ficheros los enviamos a **upload.php**

UPLOAD.PHP:

-Leer ficheros

Aquí tuve que hacer dos búsquedas, una para saber como coger los datos del fichero que el usuario ha introducido : https://code.tutsplus.com/es/tutorials/how-to-upload-a-file-in-php-with-example--cms-31763 (aquí también aprendí a como pasar un fichero por post)

Por otra parte encontré la manera de comprobar si la extensión introducida era la correcta: https://parzibyte.me/blog/2018/11/06/extraer-extension-de-archivo-en-php/#La_respuesta_es_pathinfo

Más tarde si no hay errores comienzo a leer los ficheros : https://www.jose-aguilar.com/blog/leer-archivo-csv-con-php/

Comprendiendo como funciona fgetcsv:
https://www.php.net/manual/es/function.fgetcsv.php

-Crear objetos

En estos whiles recorro el Excel e introduzco la información en 3 tipos de clases distintas que he creado: Costumer,Product y Order, introduciendo estos objetos en arrays

-Reporte 1 (líneas 99-149)

Iremos introduciendo el id y el total del pedido en un array llamado $arrayReporte1, por lo que recorremos todos los pedidos(guardando sus Id de producto), y recorremos estos id de producto(Recorremos la posición par del String) que se encuentran en el pedido, por último recorremos los productos, y si coincide el id del producto del servicio con el producto que estamos recorriendo entonces añadimos su coste a una variable $sum.

Una vez con los gastos de cada pedido añadimos al array $arrayReporte1 el id del pedido, junto con su coste total

Solo nos queda generar el CSV gracias a la explicación de este artículo: https://denisseestrada.com/crear-un-archivo-csv-con-php/

-Reporte 2 (151-183)

Creamos un for para recorrer los productos, un for para recorrer los pedidos, y un for(igual que antes) para recorrer los Id de producto del pedido en el que nos encontramos, por último, si coincide el id de producto del pedido con el producto en el que nos encontramos entonces guardamos en un array $idClientesConProducto el id del Cliente en la posición del producto en el que estemos.

Luego he utilizado la siguiente instrucción:

$idClientesConProducto = array_map('array_unique', $idClientesConProducto);

Que a partir de un arreglo crea otro sin que se repita ningún valor dentro de cada sub array(gracias a la función array_unique de array_map): https://stackoverflow.com/questions/28736091/get-unique-array-for-key

Por último quiero crear un array con el id de producto y los ids de cliente, los cuales los quiero separados por espacios, esto lo consigo gracias a la función implode:

array_push($array_final,implode(" ", $sub_array));

Gracias a este artículo:

https://toninieto.com/knowledge-base/php/arrays/convertir-un-array-en-un-string-separado-por-comas/#:~:text=PHP%3A%20Convertir%20un%20array%20en%20un%20string%20separado%20por%20comas&text=Si%20tenemos%20un%20array%20y,(%27%2C%20%27%2C%20%24array_var)%3B

Luego convierto el array en un arreglo en el que cada sub array dos valores para que cada id de producto este con el string de ids de clientes: (Con esto estuve dando muchas vueltas por internet, hasta que al final encontré la función array_chunk en el manual de php, el problema fue que no sabia como expresarlo en el buscador de Google, pero al final lo conseguí)

-Reporte 3

Este lo realicé en los mismos bucles que el reporte 1 ya que pude utilizar el valor $sum pero guardándolo con el id del cliente en $arrayReporte3.

Esto fue sin duda alguna lo que más me costó del proyecto, conseguir pasar de este array:

```
array (size=50)
  1 =>
    array (size=2)
      0 => int 0
      1 => float 18.943120182
  2 =>
    array (size=2)
      0 => int 22
      1 => float 61.425421361
  3 =>
    array (size=2)
      0 => int 57
      1 => float 23.145479225
```

A este:

```
0 => float 18.943120182
  22 => float 124.710690306
  57 => float 23.145479225
  20 => float 34.399455367
```

Es decir, quiero sumar el segundo valor del array cuando el primero coincida, esto esta explicado en el código a partir de la línea 125

Acomodamos el array en la línea 137, y a partir de la línea 188 creamos dos fors para ir introduciendo los nombres y los apellidos cuando coincide el id del cliente del for con el id de cliente del array que traemos, finalmente generamos el csv

-Descargar csv:

Aquí creo hrefs que apuntan a un fichero download.php al que le envía el fichero correspondiente, y este lo descarga en el navegador del cliente:

https://programacion.net/articulo/como_forzar_la_descarga_de_un_fichero_en_php_1935

Aclaraciones:

Tengo que ser sincero, y no salió a la primera, en primer lugar hice el proyecto solo con arrays, pero duré poco tiempo hasta que me di cuenta de que podía hacerlo con objetos. Al intentarlo con objetos, introduje la información de las variables de los objetos en arrays distintos, generando arrays excesivos. Hasta que al final me percaté de otra manera mucha más sencilla, que era almacenar todos los objetos enteros en arrays.