

ÍNDICE

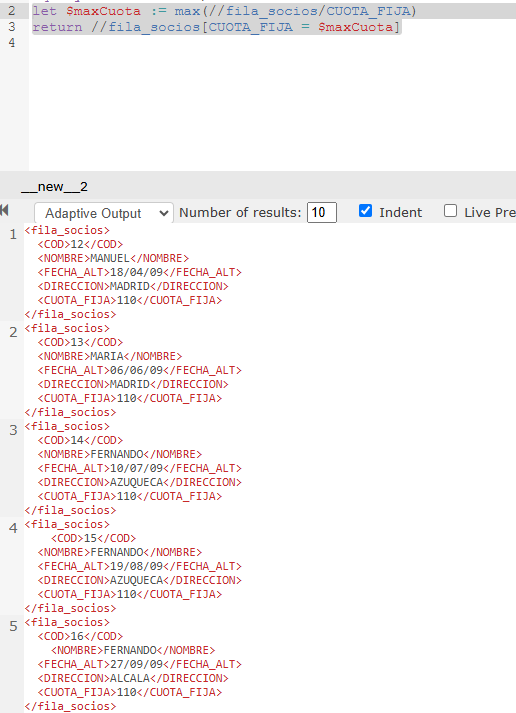
[Ejercicio 1 1](#__RefHeading___Toc10_2857803141)

[Ejercicio 2 8](#__RefHeading___Toc12_2857803141)

# Ejercicio 1

**1. Visualizar el o los socios con la cuota fija más alta. Mostrar todos sus datos.**

let $maxCuota := max(//fila\_socios/CUOTA\_FIJA)  
return //fila\_socios[CUOTA\_FIJA = $maxCuota]



**2. Obtener la información de todos los socios que han realizado actividades de tipo 1 (sólo las de tipo 1).**

for $uso in //fila\_uso

let $actividad := //fila\_actividades[@cod = $uso/CODACTIV]

where $actividad/@tipo = "1"

let $socio := //fila\_socios[COD = $uso/CODSOCIO]

return

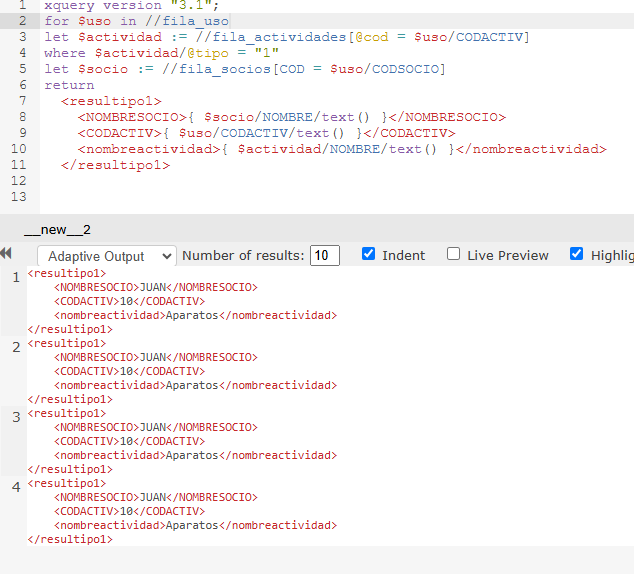
<resultipo1>

<NOMBRESOCIO>{ $socio/NOMBRE/text() }</NOMBRESOCIO>

<CODACTIV>{ $uso/CODACTIV/text() }</CODACTIV>

<nombreactividad>{ $actividad/NOMBRE/text() }</nombreactividad>

</resultipo1>



**3. Obtener por cada uso del gimnasio, el código de socio, su nombre, el código de la actividad, su nombre, las horas de la actividad, que será la hora de fin menos la de inicio, y la cuota adicional que dependerá del tipo (viene indicado arriba).**

for $uso in //fila\_uso

let $socio := //fila\_socios[COD = $uso/CODSOCIO]

let $actividad := //fila\_actividades[@cod = $uso/CODACTIV]

let $horas := number($uso/HORAFINAL) - number($uso/HORAINICIO)

let $cuotaAdicional :=

if ($actividad/@tipo = "1") then 0

else if ($actividad/@tipo = "2") then 2 \* $horas

else if ($actividad/@tipo = "3") then 5 \* $horas

else 0

return

<resul>

<CODSOCIO>{ $socio/COD/text() }</CODSOCIO>

<NOMBRE>{ $socio/NOMBRE/text() }</NOMBRE>

<CODACTIV>{ $uso/CODACTIV/text() }</CODACTIV>

<NOMBRE>{ $actividad/NOMBRE/text() }</NOMBRE>

<horasactividad>{ $horas }</horasactividad>

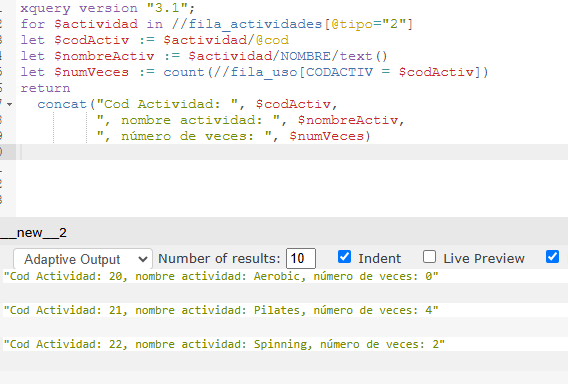
<cuotaadicional>{ $cuotaAdicional }</cuotaadicional>

</resul>



**4. Obtener el número de veces que se practica cada actividad del tipo 2. Obtener una salida de datos concatenados:**

for $actividad in //fila\_actividades[@tipo="2"]  
let $codActiv := $actividad/@cod  
let $nombreActiv := $actividad/NOMBRE/text()  
let $numVeces := count(//fila\_uso[CODACTIV = $codActiv])  
return   
 concat("Cod Actividad: ", $codActiv,   
 ", nombre actividad: ", $nombreActiv,   
 ", número de veces: ", $numVeces)



**5. Añadir el nodo en todas las actividades indicando el número de veces que los socios han realizado esa actividad.**

for $actividad in //fila\_actividades

let $codActiv := $actividad/@cod

let $numVeces := count(//fila\_uso[CODACTIV = $codActiv])

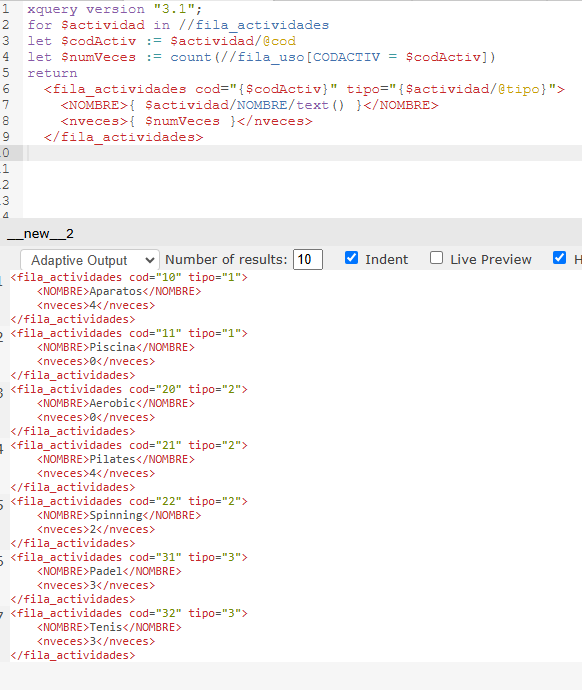
return

<fila\_actividades cod="{$codActiv}" tipo="{$actividad/@tipo}">

<NOMBRE>{ $actividad/NOMBRE/text() }</NOMBRE>

<nveces>{ $numVeces }</nveces>

</fila\_actividades>



**6. Calcular la cuota adicional por socio. Añadir el nodo cuotaadicional a los socios del gimnasio. Que es la cantidad adicional a pagar por cada socio dependiendo de las actividades realizadas (será la suma de las cuotas adicionales que paga el socio por cada actividad que realice). Crear lo necesario para poder añadir este nodo.**

for $socio in //fila\_socios

let $codSocio := $socio/COD/text()

let $cuotaAdicional :=

sum(

for $uso in //fila\_uso[CODSOCIO = $codSocio]

let $codActiv := $uso/CODACTIV/text()

let $tipoActiv := //fila\_actividades[@cod = $codActiv]/@tipo

return

if ($tipoActiv = "1") then 0

else if ($tipoActiv = "2") then 5

else if ($tipoActiv = "3") then 10

else 0

)

return

<fila\_socios>

<COD>{ $codSocio }</COD>

<NOMBRE>{ $socio/NOMBRE/text() }</NOMBRE>

<FECHA\_ALT>{ $socio/FECHA\_ALT/text() }</FECHA\_ALT>

<DIRECCION>{ $socio/DIRECCION/text() }</DIRECCION>

<CUOTA\_FIJA>{ $socio/CUOTA\_FIJA/text() }</CUOTA\_FIJA>

<cuotaadicional>{ $cuotaAdicional }</cuotaadicional>

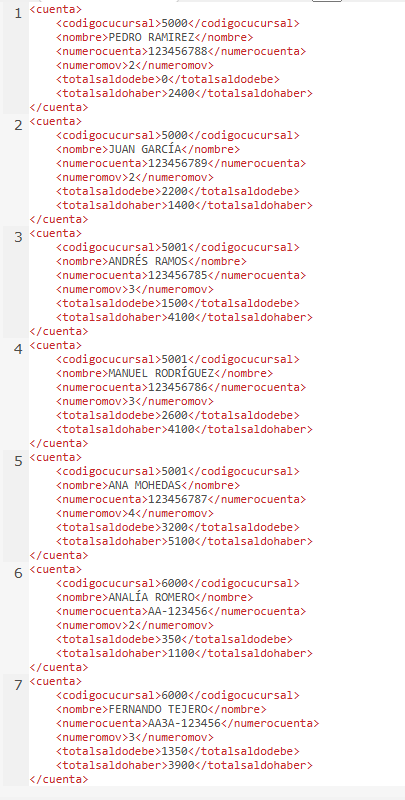
</fila\_socios>



# Ejercicio 2

**1. Obtener una salida con los datos de las cuentas. Con la sucursal a la que pertenece, el nombre y número de cuenta, el total saldo debe, el total saldo haber, y el número de movimientos de la cuenta.**

for $banco in //banco  
for $sucursal in $banco/sucursales/sucursal  
for $cuenta in $sucursal/cuentas/cuenta  
let $codSucursal := $sucursal/codigocucursal/text()  
let $nombreCuenta := $cuenta/nombre/text()  
let $numCuenta := $cuenta/numerocuenta/text()  
let $saldoDebe := xs:decimal($cuenta/saldodebe/text())  
let $saldoHaber := xs:decimal($cuenta/saldohaber/text())  
let $movimientos := $cuenta/movimientos/movim  
let $totalDebe := $saldoDebe + sum($movimientos[tipomov = "R"]/importe/xs:decimal(.))  
let $totalHaber := $saldoHaber + sum($movimientos[tipomov = "I"]/importe/xs:decimal(.))  
let $numMovimientos := count($movimientos)  
return  
 <cuenta>  
 <codigocucursal>{ $codSucursal }</codigocucursal>  
 <nombre>{ $nombreCuenta }</nombre>  
 <numerocuenta>{ $numCuenta }</numerocuenta>  
 <numeromov>{ $numMovimientos }</numeromov>  
 <totalsaldodebe>{ $totalDebe }</totalsaldodebe>  
 <totalsaldohaber>{ $totalHaber }</totalsaldohaber>  
 </cuenta>



**2. Añadir el nodo a cada sucursal que incluya el número de cuentas, total haber, y total debe a cada sucursal. El total haber es la suma de los saldos haber de las cuentas + la suma de todos los importes de tipo Ingreso de las cuentas de la sucursal.**

for $banco in //banco

for $sucursal in $banco/sucursales/sucursal

let $codSucursal := $sucursal/codigocucursal/text()

let $cuentas := $sucursal/cuentas/cuenta

let $numCuentas := count($cuentas)

let $totalDebeSucursal := sum($cuentas/saldodebe/xs:decimal(.)) + sum($cuentas/movimientos/movim[tipomov = "R"]/importe/xs:decimal(.))

let $totalHaberSucursal := sum($cuentas/saldohaber/xs:decimal(.)) + sum($cuentas/movimientos/movim[tipomov = "I"]/importe/xs:decimal(.))

return

<sucursal>

<codigocucursal>{ $codSucursal }</codigocucursal>

{

for $cuenta in $cuentas

let $nombreCuenta := $cuenta/nombre/text()

let $numCuenta := $cuenta/numerocuenta/text()

let $saldoDebe := xs:decimal($cuenta/saldodebe/text())

let $saldoHaber := xs:decimal($cuenta/saldohaber/text())

let $movimientos := $cuenta/movimientos/movim

let $totalDebe := $saldoDebe + sum($movimientos[tipomov = "R"]/importe/xs:decimal(.))

let $totalHaber := $saldoHaber + sum($movimientos[tipomov = "I"]/importe/xs:decimal(.))

let $numMovimientos := count($movimientos)

return

<cuenta>

<nombre>{ $nombreCuenta }</nombre>

<numerocuenta>{ $numCuenta }</numerocuenta>

<numeromov>{ $numMovimientos }</numeromov>

<totalsaldodebe>{ $totalDebe }</totalsaldodebe>

<totalsaldohaber>{ $totalHaber }</totalsaldohaber>

</cuenta>

}

<Totales>

<numerocuentas>{ $numCuentas }</numerocuentas>

<totalsaldodebe>{ $totalDebeSucursal }</totalsaldodebe>

<totalsaldohaber>{ $totalHaberSucursal }</totalsaldohaber>

</Totales>

</sucursal>

