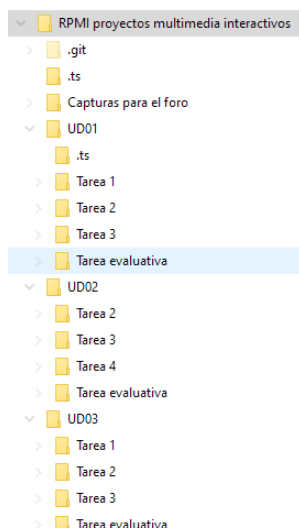


## Administración de recursos digitales

La organización de los archivos de este módulo, proyectos multimedia interactivos, está estructurada de la siguiente manera, cada unidad didáctica tiene su propia carpeta, y dentro de esta hay una subcarpeta para cada tarea a entregar, sea voluntaria o evaluativa. Si fuese necesario por el volumen de archivos se crearía un tercer nivel de subcarpetas donde guardar cosas específicas y si algunos archivos o conjuntos de ellos se utilizasen en tres o más tareas o unidades didácticas se crearían carpetas específicas donde

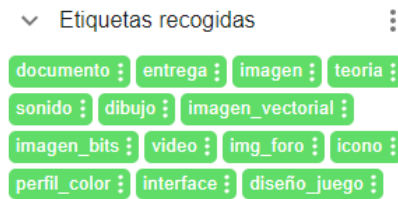


tenerlos.

Es muy importante que todos los nombres de los archivos sean descriptivos de su contenido o finalidad para que se sepa su contenido sin necesidad de abrirlos, salvo en el caso de los documentos de entregas que tienen un nombre obligatorio, pero estos deben ser siempre copia de los originales por lo que no son para usar, solo para entregar.

Para el uso de TagSpaces se han definido los siguientes criterios para la creación de etiquetas

- Extensión del archivo. Da mucha información sobre el tipo de archivo. docx, pdf, png, svg, Pero este tipo de etiquetas no hay que crearlas manualmente, TagSpaces ya reconoce las extensiones y permite hacer búsquedas por extensiones.
- Tipo de archivo. Imagen, documento, sonido, dibujo, etc Aunque sean archivos de diferente extensión son del mismo tipo por lo que es muy útil para agruparlos.
- Tipo específico de archivo. imagen\_bits o imagen\_vectorial ya que son diferentes. Son los subconjuntos más utilizados de las etiquetas del tipo b)
- Finalidad del archivo. entrega, teoría, icono, etc Es más, específica que el tipo de etiqueta anterior y no todos los archivos tienen estas etiquetas. Es útil para el seguimiento de la asignatura
- Módulos para el desarrollo del videojuego. Interface, diseño\_juego. Útil para encontrar los archivos realizados para el videojuego que pueden estar repartidos en diferentes unidades o tareas.



Estas son las etiquetas creadas hasta el momento, es importante resaltar que es algo vivo y se pueden añadir más si se ve necesario, como por ejemplo cuando se creen nuevos archivos para el videojuego.

Además, se podrían crear otras etiquetas como el autor, la prioridad, fecha, resolución, calidad, cliente, etc. pero no son necesarias en este caso. Solo hay un autor, no se utilizan prioridades para las entregas, la fecha es indiferente, las carpetas ya van en orden cronológico, la resolución de las imágenes se escribe en el nombre del archivo, no hay varios clientes, etc.

Pero si surgiese la necesidad de un nuevo tipo de etiqueta más allá de los descritos anteriormente no habría problema en crearlo.

## Repositorio remoto

<https://github.com/Igrtgo/RPMI-proyectos-multimedia-interactivos>

Se han añadido todos los archivos utilizados en el modulo de Realización de proyectos multimedia interactivos.

Se han añadido las siguientes extensiones para ser gestionadas con Git Large File Storage:

- \*.rar filter=lfs diff=lfs merge=lfs -text
- \*.kra filter=lfs diff=lfs merge=lfs -text
- \*.mp4 filter=lfs diff=lfs merge=lfs -text
- \*.docx filter=lfs diff=lfs merge=lfs -text
- \*.aup3 filter=lfs diff=lfs merge=lfs -text
- \*.pptx filter=lfs diff=lfs merge=lfs -text
- \*.jpg filter=lfs diff=lfs merge=lfs -text
- \*.png filter=lfs diff=lfs merge=lfs -text

Son todos los tipos de extensiones utilizados por archivos de más tamaño, más de 1000 KB.

Si se utilizasen nuevos tipos de archivos de gran tamaño se irían añadiendo a lfs, por ejemplo .avi (video), .mkv (video), .xcf (gimp), etc.

---

## Sistemas de almacenamiento y copias de seguridad

Todos los archivos de trabajo del proyecto están guardados en una carpeta local de disco duro c: y es la copia de trabajo.

### Copias locales:

Semanalmente se hace una copia de todos los archivos de trabajo de este y otros proyectos en una memoria USB que se renueva cada 6-9 meses guardándose la vieja, para minimizar los riesgos de que se rompiese. Apenas cuestan unos euros.

En estas copias no hay control de versiones, solo se guarda la última y se reescriben las anteriores. Pero tengo el hábito de crear nuevos archivos con las nuevas versiones y en ese caso se copiarían todas siempre, guardándose las versiones anteriores en una subcarpeta llamada \*\_old

Este habito lo adquirí tras algunos sustos con el sistema operativo, roturas de discos duros y lo tedioso que es recopilar todo antes de formatear.

Ya no utilizo discos duros externos por su tamaño y velocidad, aportan mayor capacidad, pero no me es necesario actualmente, todo lo realizado en Birt no llega a un GB.

También tengo discos duros mecánicos internos donde poder guardar copias de seguridad si el proyecto o partes de el fuesen de gran tamaño.

### Copias en línea:

Esta carpeta de uso diario esta sincronizada con onedrive ya que dispongo de licencia office 365, así que siempre puedo recuperar los archivos desde la nube. Adicionalmente podría sincronizar la carpeta con Dropbox que utilizo para otros proyectos compartidos con terceros, pero es innecesario.

Mientras el proyecto sea de la envergadura actual no habrá problema de capacidad.

Este sistema si tiene control de versiones de tres meses si no recuerdo mal.

Adicionalmente he creado la copia de todos los archivos del módulo RPMI en GitHub por lo que puedo contar con esa copia de respaldo, solo tendría que hacer regularmente la sincronización con GitHub desktop.

### Otras opciones

No tengo copias en otras ubicaciones. Pero si tengo sincronizado Onedrive y Dropbox en un segundo ordenador, de forma que tengo otra copia en local. Si quito internet para que no se sincronice puedo recuperar los archivos en caso de emergencia. Solo perdería entre unas horas y unos días de actividad.