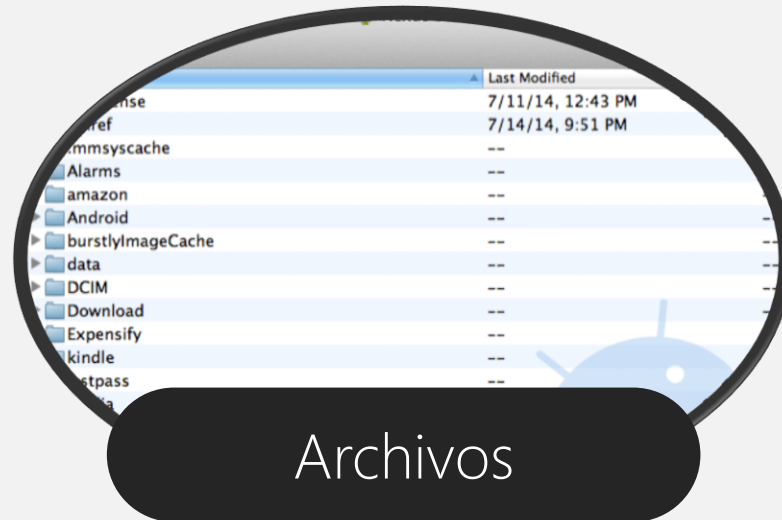


# 79. Opciones para almacenar datos locales

# Opciones para almacenar datos

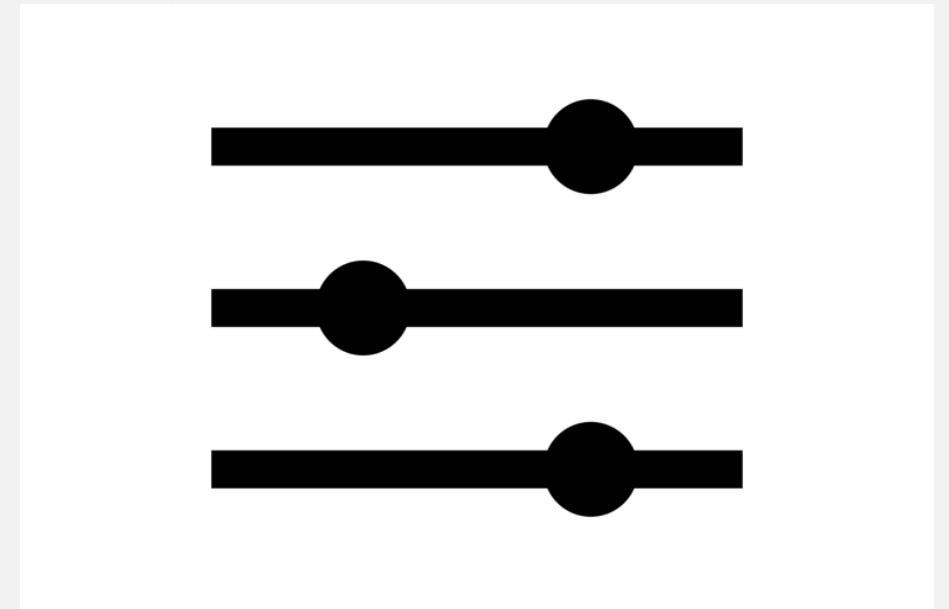
Al almacenar información local, su aplicación tiene varias opciones para elegir



¿Qué enfoque tiene más sentido para el tipo de datos con los que está trabajando?

# Preferencias

- ❖ iOS, Android y Windows admiten el almacenamiento de preferencias específicas de la aplicación como simples pares clave-valor
- ❖ Útil para almacenar la configuración de la aplicación, las preferencias del usuario y otros ajustes de personalización que el usuario puede controlar



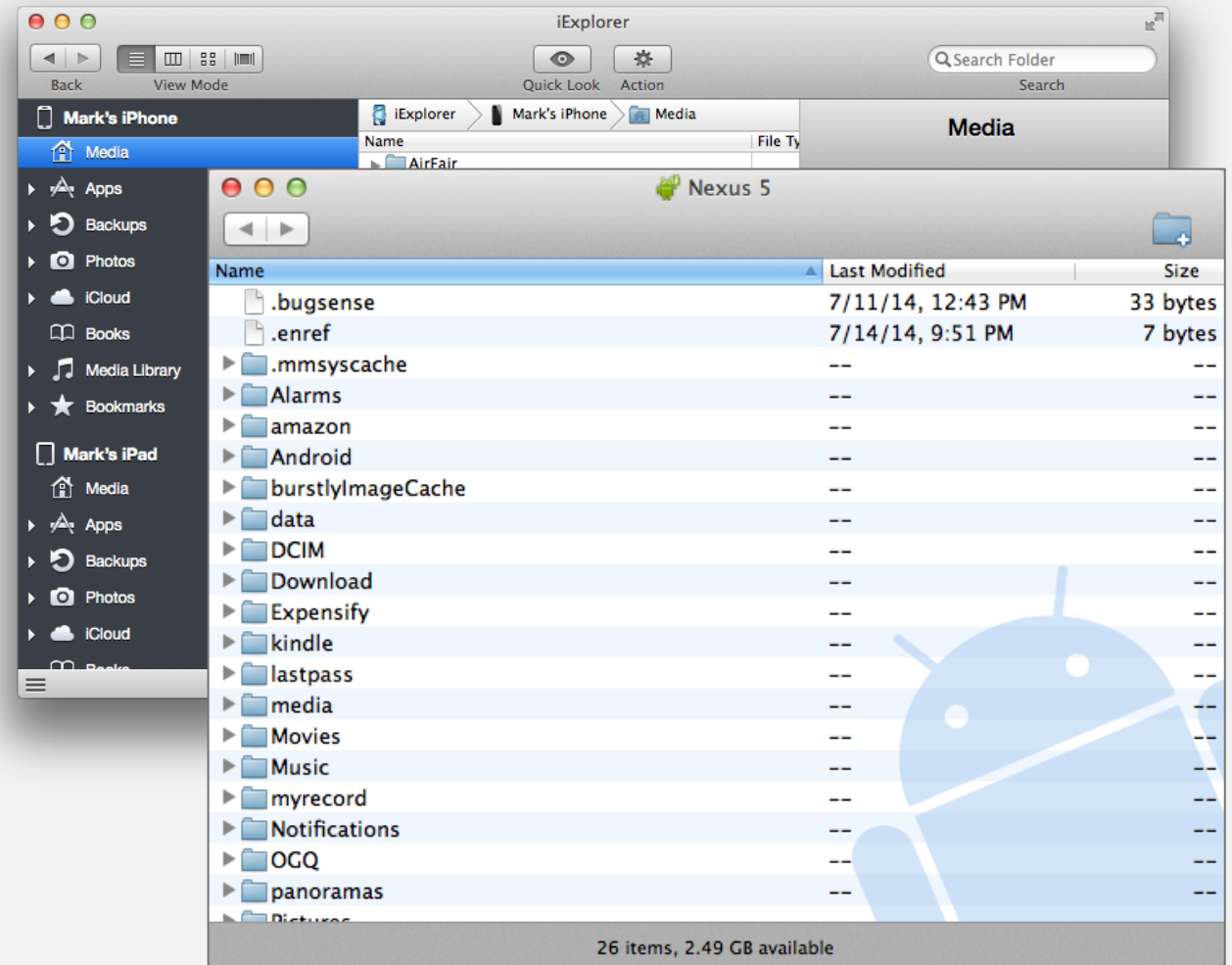
# Preferences de Essentials

La clase **Preferences** ayuda a almacenar las preferencias de la aplicación en un almacén de clave y valor.

```
Preferences.Set("my_key", "my_value");
```

# El sistema de archivos


- Los dispositivos tienen sistemas de archivos persistentes para almacenar configuraciones, aplicaciones, datos, etc.
- La estructura y el contenido del sistema de archivos varían según el sistema operativo.



# Trabajando con carpetas y archivos

Puede trabajar directamente con archivos y carpetas usando clases en el espacio de nombres **System.IO**

```
using System.IO;
...
public IEnumerable<Todo> LoadTodoTasks(string filename)
{
    StreamReader reader = File.OpenText(filename);
    ...
}
```



Clases familiares como **File**, **Directory** y **StreamReader** están disponibles en sus proyectos específicos de plataforma

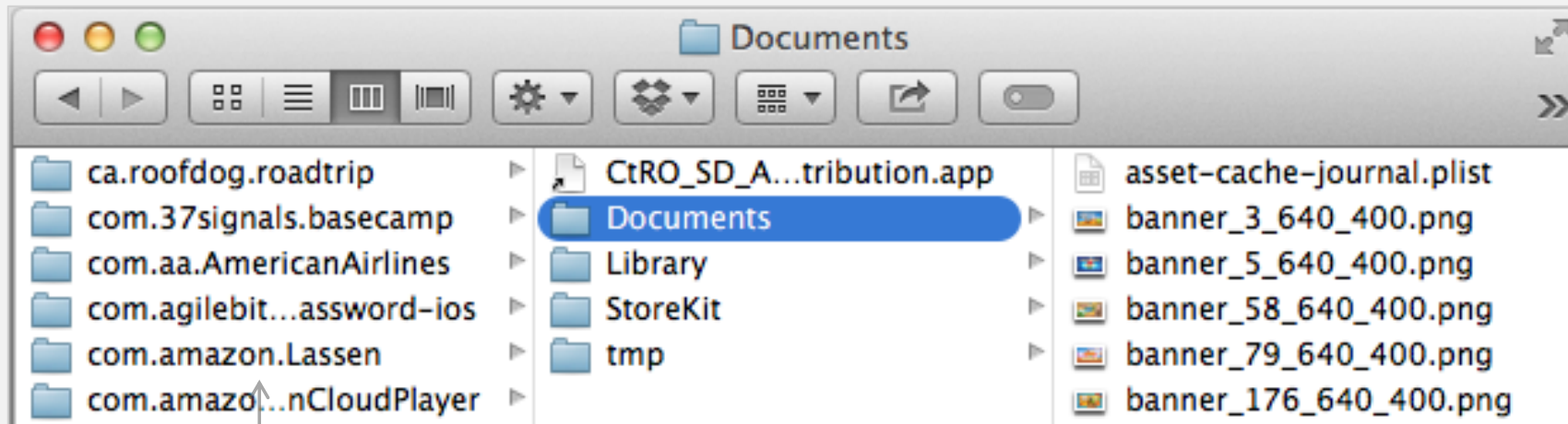
# Formato de archivos

Cada plataforma admite formatos de texto, binarios, XML y JSON: use el que tenga sentido para su estilo de datos

```
using System.Xml.Linq;
...
public IEnumerable<string> LoadTasks(string filename)
{
    XDocument doc = XDocument.Load(filename);
    return (from item in doc.Root.Descendants("todo")
            select (string) item.Attribute("PartNumber"));
}
```

# Sandbox

Su aplicación tiene una carpeta dedicada, llamada carpeta de la aplicación o sandbox, en el sistema de archivos que contiene contenido específico de la aplicación.






Cada aplicación de iOS tiene una carpeta, que contiene subcarpetas que, a su vez, contienen los datos y los activos que crea



# Localización de archivos

La ubicación recomendada para su archivo de datos difiere entre plataformas

 Android	<code>&lt;AppHome&gt;/files</code>
 iOS	<code>&lt;AppHome&gt;/Library/[subdirectory]</code>
 UWP	<code>&lt;AppHome&gt;\LocalState</code>

Estas ubicaciones son comunes, pero otras opciones están disponibles (por ejemplo, Android tiene una carpeta de base de datos)

# Folder path [.NET]

Puedes usar las API de .NET para obtener la ruta completa a la carpeta de la aplicación



```
// <AppHome>/files  
string path = Environment.GetFolderPath(Environment.SpecialFolder.Personal);
```



```
// <AppHome>/Documents  
string docFolder = Environment.GetFolderPath(Environment.SpecialFolder.Personal);  
// to meet Apple's iCloud terms, content that is not generated by the user  
// should be placed in the /Library folder or a subdirectory inside it  
string libFolder = System.IO.Path.Combine(docFolder, "..", "Library");
```



```
// <AppHome>\LocalPath  
string path = Windows.Storage.ApplicationData.Current.LocalFolder.Path;
```

# SQLite.NET

El motor de base de datos de SQLite permite que las aplicaciones .NET MAUI carguen y guarden datos en una base de datos local usando código compartido.

