

I. INTRODUCCION

Las tecnologías móviles están constituyendo un avance notable en la ultima década, por lo cual se ve afectado en un desarrollo para la ingeniería, un ejemplo notable de esto son los Celulares, PDA's, Netbooks, Tablets, y Smartphones. La mayoría de los dispositivos electrónicos requieren de una interacción con el usuario por lo que para poder operarlos es necesaria una interfaz aplicación de aplicación, que es accesible desde un hardware especial, como lo son Botones, Click Wheels, Pantalla multitáctil Capacitiva y Resistiva ahora utilizadas en la mayoría de los dispositivos móviles , renombrada con el nombre de Pantalla Touch.

Las interfaces de usuario se han ido mejorando gradualmente y cada vez el nivel requerido para su manejo es mínimo, sin perder el nivel de interacción por lo contrario, su funcionalidad esta incrementando. El nivel de calidad y variedad de software móvil cada día esta jugando un rol importante en tareas de campo que demandan un procesamiento de nivel intermedio, lo cual pone en una gran posición importante a los móviles frente a un ordenador de escritorio para actividades de carácter específico.

El desarrollo e implementación de software para las llamadas "Tablets" por su nombre en ingles, serviría para contribuir a la ya existente demanda de dispositivos móviles de la presente década, al implementar software de aplicación de gran funcionalidad y calidad. En este trabajo se presenta el desarrollo de una aplicación que implementa algunos de los protocolos de transporte de datos en red, con una comunicación multi plataforma. El prototipo actual trabaja sobre la plataforma móvil iOS 4 en un Dispositivo Tablet Apple iPad, así como su contraparte de control en un Ordenador con Apple Mac OS X 10.6, y el cual permite una manipulación de la librería multimedia que existe en el iPad, vía un Mac enlazado en red Local. en resumen el iPad Tendría el papel Servidor Multimedia y el Mac el papel cliente Controlador Multimedia. Este proyecto serviría para el estudio de protocolos y vías de comunicación de bajo y alto nivel, como lo son BSD Sockets, TCP, UDP, Bonjour (Zero Conf).

II. ANTECEDENTES

Un tablet PC es una computadora portátil con la que se puede interactuar a través de una pantalla táctil. El usuario puede utilizar una pluma “Stylus” o los dedos para trabajar con el ordenador sin necesidad de teclado físico, o un mouse. Existen modelos de tablet pc que sólo aportan la pantalla táctil a modo de pizarra, siendo así muy ligeros. Utilizan procesadores móviles, que consumen menos energía. El software especial que nos proporciona el sistema operativo nos permite realizar escritura manual, tomar notas a mano alzada y dibujar sobre la pantalla. Así, es útil para hacer trabajos de campo.

Los dispositivos más abundantes son los de 7” y los de 10”. Los primeros son los más fácilmente portables, sin embargo, los segundos son más versátiles dado su mayor tamaño de pantalla, por lo que podrán utilizarse para desarrollar un mayor número de funciones con ellos. A modo de ejemplo, ambos tamaños de pantalla permiten la lectura de un libro electrónico, pero los dispositivos de 10", además permiten visualizar cómics con cierta comodidad. Así pues, dependiendo del uso que se pretenda dar al dispositivo se optará por uno de mayor o menor tamaño. Para utilizar a diario leyendo un libro durante trayectos en transporte público, el modelo óptimo para la mayoría de usuarios sería el del entorno de las 7 pulgadas. Sin embargo, para viajes largos de varios días de duración, y con la pretensión de utilizar el dispositivo para funciones más diversas, el dispositivo preferible sería uno del entorno de las 10 pulgadas.

Algunas empresas ya han empezado a comercializar tablets con pantalla LCD de mas de 14”, que puede rotar 180° sobre si misma y con mayor resolución, este sistema de presentación esta pensado para visualizar revistas, cómics y periódicos. Por otro lado, algunos portátiles incorporan Jack de 3.5 o HDMI para poder ver vídeos en un televisor o monitor. Ejemplos de uso son la lectura de libros electrónicos, lectura de cómics en pantalla, navegación web, juegos, consulta de documentos, visualización de vídeos y películas, reproducción de música, videoconferencia, video juegos.

III. IMPLEMENTACION

El presente trabajo implementa un sistema de control remoto sobre una arquitectura cliente-servidor entre un MacBook Pro Bajo Mac OS 10.6 y un iPad bajo iOS 4. El sistema busca establecer la infraestructura base para la exploración de aplicaciones sobre arquitectura de red que hacen el uso eficiente de servicios que son anunciados mediante Bonjour de (Apple), cabe mencionar que , Bonjour no da acceso extra a ningún servicio, meramente los anuncia. El escenario de aplicación para el sistema “iPod Remote” es el siguiente:

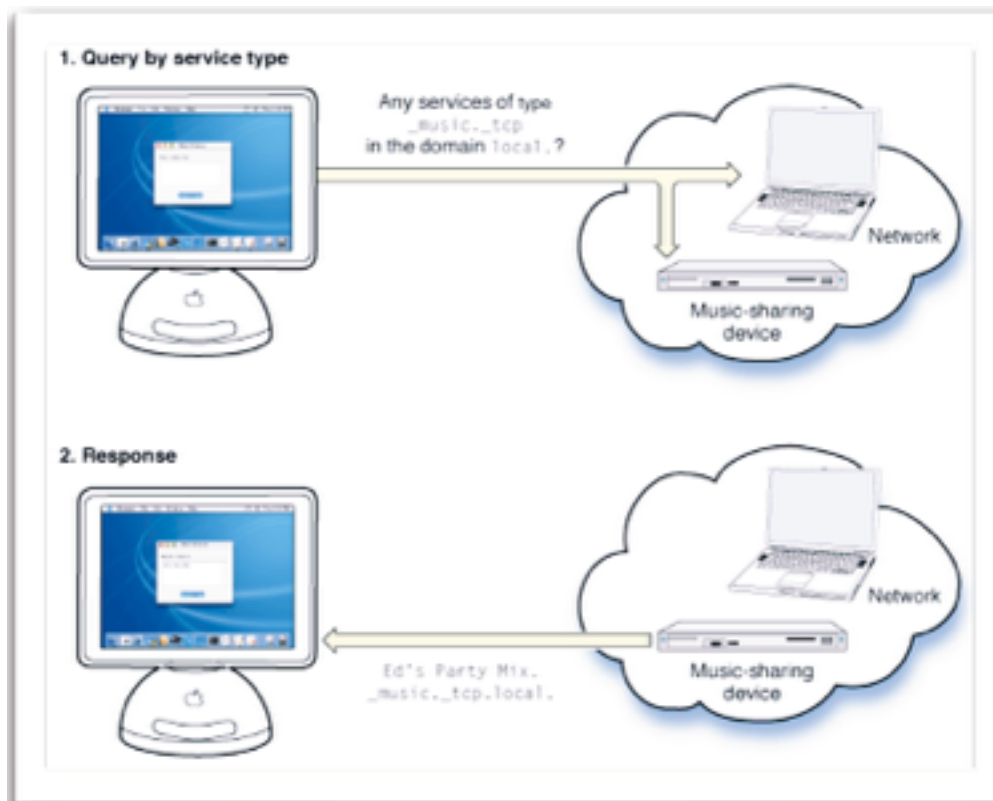
Este software esta diseñado principalmente para lugares en los que se desee tener el control de la librería multimedia de nuestro iPod Remotamente, un ejemplo de ello pudiera ser, situando en un cuarto apartado nuestro iPod a su vez conectado a un auxiliar de audio, mediante la aplicación iPod Remote es posible acceder y controlar el contenido multimedia desde Otro dispositivo Similar bajo iOS (iPod,iPad) o un ordenador bajo el sistema operativo Mac Leopard o Snow Leopard. Fungiendo como Servidor el dispositivo controlado el cual contiene la multimedia, y a su vez como cliente el dispositivo que se utilizara como controlador.

Prestando así al usuario una comodidad, al permitirle controlar su multimedia desde cualquier lugar de su hogar, seleccionar su Artista,Album o Canción preferida, acceder a sus listas de reproducción, control de volumen, avanzar o retroceder un pista reproducida, sin la necesidad de utilizar un dock multimedia que puede tener un rango limitado de control remoto. La arquitectura general de un sistema iPod Remote que ofrece estas características de interacción entre el control de mando móvil y el servidor reproductor de música.

3.1 Implementación del servidor de música

El servidor de música es el componente principal del sistema ya que este se encarga de hacer públicos al cliente los recursos disponibles, atender las peticiones del cliente y aplicar el nivel de control requerido para el servicio. El sistema tiene como base una estructura de programación basada en protocolos y servicios, basada en un servicio llamado BONJOUR con marco de trabajo Zeroconf, Bonjour es un método para descubrir servicios en una red de área local. Para que el servidor pueda localizar los servicios que puede poner a disposición del cliente, este debe interactuar con Bonjour, pero cuando se requiere aplicar control sobre los servicios se requiere tener acceso a algunas capas de bajo nivel del sistema operativo, tal y como se muestra en la figura 2.

Las imágenes que se muestran a continuación, preferiría fueran parte de la información general en los apéndices... lo que buscamos aquí es que tengas una idea firme de cómo plasmar este proceso de acceder a los recursos disponibles en un diagrama propio...



3.1.1 Descubriendo recursos a través de Bonjour

El trabajo de configuración o programación de servicios bajo Bonjour requiere desarrollar la codificación adecuada para las librerías (interface.h) así como con la implementación de la heurística de control deseada (implementation.m)

Basado en 3 etapas: Publicación, Descubrimiento, Resolución.

Publicación:

El primer paso para anunciar un servicio Bonjour es crear una application Cocoa, con ambos archivos, interface e implementación (Cabecera y Programa respectivamente). Esto se hace mediante la creación de una instancia de NSNetService para así conseguir el objeto necesario para la publicación. En este caso se le dio el nombre de **netService**.

```
#import <UIKit/UIKit.h>
@interface NayberzAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    NSNetService *netService;
}
@property (nonatomic, retain) IBOutlet UIWindow *window;
@end
```

Toda vez que se obtiene el apuntador al objeto NSNetService, es trabajar sobre el archivo de implementación. Es aquí en donde será posible la creación y publicación del servicio.

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Crear una instancia de NSNetService
    netService = [[NSNetService alloc]
    initWithDomain:@""
    type:@"_nayberz._tcp."
    name:[[[UIDevice currentDevice] name]
    port:9090];
    //Se le asigna un delegado al servicio para que estos metodos delegados esten a la
    espera de un cambio en el flujo del programa.
    [netService setDelegate:self];
    // Se ordena el metodo de publicacion
    [netService publish];
    [window makeKeyAndVisible];
    return
    YES;
}
```

Adicionalmente a la especificación de los métodos delegados establecidos por la clase NSnetService, para hacer saber al sistema si se publico el servicio u ocurrió algún error.

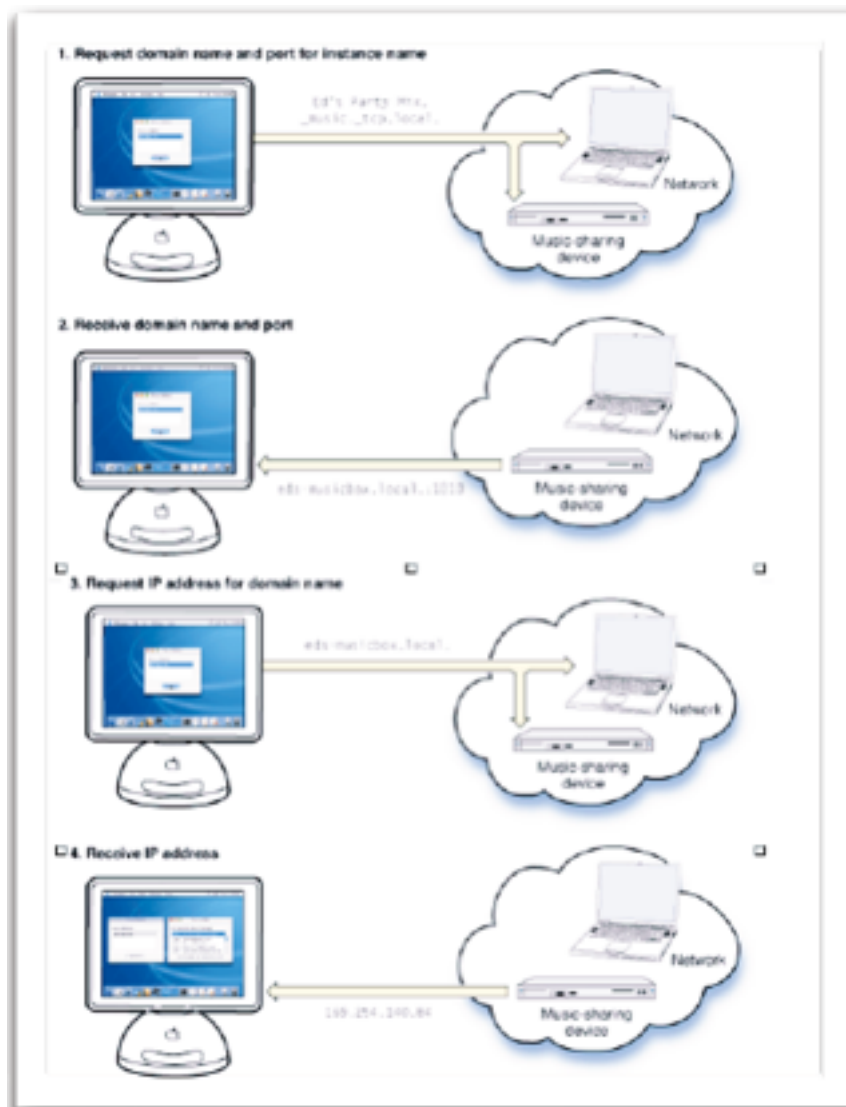
```
- (void)netServiceDidPublish:(NSNetService *)sender
{
    NSLog(@"published: %@", sender);
}
- (void)netService:(NSNetService *)sender didNotPublish:(NSDictionary *)
errorDict
{
    NSLog(@"not published: %@ -> %@", sender, errorDict);
}
```

Para indicar al programa el paro de publicación de servicio se utiliza este método.

```
- (void)applicationWillTerminate:(UIApplication *)application
{
    [netService stop];
}
```

Cuando el servicio esta listo y publicado el cliente puede ver que servicio existe y esta disponible en el servidor. Sin embargo, el cliente no puede interactuar con el servicio aun ya que para poder tener acceso a los recursos puestos a disposición por Bonjour.

3.2 Búsqueda de servicios



Descubrimiento:

Es necesario crear una instancia de la clase `NSNetServiceBrowser`.

```
#import <UIKit/UIKit.h>
```

```
@interface TableController : UITableViewController
```

```
{  
    NSMutableArray *netServices;  
    NSNetServiceBrowser *serviceBrowser;  
}  
@end
```

Cuando existen servicios disponibles, el cliente ya puede seleccionar el servicio deseado. La selección de servicios se realiza a través de una búsqueda en una tabla. En el código siguiente se desarrollan las actividades siguientes: El siguiente paso es iniciar la búsqueda de servicios indicando el nombre del servicio que se esta buscando:

```
- (id)init
{
    serviceBrowser = [[NSNetServiceBrowser alloc] init];
    // Se asigna un delegado que estara a la espera cuando se encuentren servicios
    [serviceBrowser setDelegate:self];
    // Se inicia la busqueda
    [serviceBrowser searchForServicesOfType:@"_nayberz._tcp."
    inDomain:@""];
    return self;
}
```

Luego es necesario implementar los métodos delegados que serán llamados cuando halla servicios disponibles en la red.

```
// Es llamado cuando se encuentran servicios
- (void)netServiceBrowser:(NSNetServiceBrowser *)browser
didFindService:(NSNetService *)aNetService
moreComing:(BOOL)moreComing
{
    NSLog(@"%@@", aNetService);
}
```

3.3.3 Resolución:

En esta etapa es necesario implementar los métodos delegados de NSNetService que serán llamados para resolver el servicio elegido.

```
- (void)netServiceDidResolveAddress:(NSNetService *)sender
{
    int row = [netServices indexOfObjectIdenticalTo:sender];
    NSIndexPath *ip = [NSIndexPath indexPathForRow:row inSection:0];
    NSArray *ips = [NSArray arrayWithObject:ip];
    [[self tableView] reloadRowsAtIndexPaths:ips
    withRowAnimation:UITableViewRowAnimationRight];
}
```


3.3 Interface de usuario

El sistema iPod Remote es un software que cuenta con una interface visual, la cual esta compuesta con los elementos necesarios para publicar y establecer un servicio entre el dispositivo móvil y la computadora.

Interacción Servidor:

Se presenta una pantalla donde se encuentran solamente 2 elementos que representan lo necesario para que el sistema publique su servicio, el primero es un botón al cual se le pulsara para iniciar el servicio, este al ser pulsado se tornara transparente, el segundo elemento sera un indicador de texto que servira para avisar al usuario que el servicio se ha publicado, una vez cerrada la aplicación la conexión y el flujo de datos es perdido.

Interacción Cliente:

Una vez publicado el servicio del lado del servidor, en la interface cliente se presentara una ventana, que mostrara un listado de los servicios disponibles en la red mediante bonjour, para así al dar click en el nombre del servicio deseado, establecer el enlace, después de que la conexión y es establecida y exitosa, se le presenta automáticamente al usuario, un listado del contenido multimedia del dispositivo en este caso la librería musical, una vez cerrada la aplicación la conexión y el flujo de datos es perdido.

IV. Conclusiones Y TRABAJO A FUTURO

En este trabajo se ha presentado la implementación de un software de aplicación que tiene la funcionalidad de permitir a equipos de diferentes plataformas intercambiar información multimedia, estas aplicaciones permiten al usuario tener el control básico de la librería multimedia en el dispositivo móvil desde la comodidad de su ordenador. Un ejemplo, de aplicación se puede presentar cuando el usuario desea enlazar su dispositivo móvil con un componente de audio superior, mediante el conector auxiliar de su dispositivo. Este prototipo logro contar con el enlace y establecimiento de conexión para la transferencia de datos entre un sistema operativo Mac OS 10.6 y un sistema operativo iOS 3.0, lo cual fue utilizando protocolos de red como es el caso de Bonjour, básicos como contar con fundamentos básicos sobre programación de Sockets en el lenguaje Objective-C, así como también el uso de librerías para el manejo de la multimedia, la interacción humano-maquina fue posible mediante una interfaces programada utilizando el API de programación "Cocoa Touch.

En un futuro esta proyectado realizar cambios en la programación para lograr una comunicación multi punto, para así añadir en enlace de mas de un dispositivo, este proyecto tiene fundamentos esenciales para un comunicación de red, por lo que su logica y codificación puede ser de gran ayuda para proyectos en los cuales implique comunicación y transferencia de datos entre dispositivos mediante una red de area local, ayudando así a la busqueda, reconocimiento y enlace, mediante el protocolo de "Zero Configuration" llamado Bonjour.

REFERENCIAS Y BIBLIOGRAFIA

[http://es.wikipedia.org/wiki/Bonjour_\(software\)](http://es.wikipedia.org/wiki/Bonjour_(software))

<https://developer.apple.com/library/mac/navigation/>

<https://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/NetServices/Introduction.html>

https://developer.apple.com/library/mac/#documentation/MacOSX/Conceptual/OSX_Technology_Overview/About/About.html

<https://developer.apple.com/library/mac/#documentation/Networking/Conceptual/NSNetServiceProgGuide/Introduction.html>

https://developer.apple.com/library/mac/#documentation/Networking/Conceptual/dns_discovery_api/Introduction.html

<https://developer.apple.com/technologies/mac/networking.html>

<http://www.apple.com/support/bonjour/>

APENDICES

Apéndice A

Comprendiendo la arquitectura de servicios de red Apple? Bonjour?

El Sistema de control remoto de librería multimedia iPod Remote es un ejemplo de aplicación de protocolos y tecnología de redes de ordenadores que son usados en una amplia variedad de sistemas operativos, para el enlace y la transferencia de datos a través de una red. Esta tecnología es muy utilizada en Mac OS X y permite a los usuarios establecer una red sin ningún tipo de configuración. Actualmente se usa en Mac OS X y otros sistemas operativos para encontrar impresoras y servidores de ficheros. También es usada por iTunes para encontrar música compartida, en iPhoto para encontrar fotos compartidas, iChat, Skype y el Proyecto Gizmo para encontrar otros usuarios de la red local, TiVo Desktop para encontrar video grabadores digitales, y SubEthaEdit para encontrar colaboradores de documentos. También es usado por Safari para encontrar servidores web locales y páginas de configuración para dispositivos locales, y por Asterisk para notificar de servicios y parámetros de configuración para teléfonos VoIP y llamadores automáticos. Sin una configuración especial de DNS, Bonjour sólo funciona en una única subred, lo cual suele ser una pequeña área. A veces se confunde Bonjour con un servicio de ordenador (por ejemplo, compartir ficheros) disponible a la Internet pública, lo cual puede ser considerado como un riesgo de seguridad. De hecho, Bonjour no da acceso extra a ningún servicio; meramente los anuncia. Por ejemplo, un usuario puede explorar por una lista de ordenadores cercanos con los que puede compartir ficheros -Bonjour en esos ordenadores ha dicho al usuario que ese servicio está disponible- pero aun así tiene que proveer una contraseña para acceder al servicio en esas máquinas.

Este sistema tiene como base protocolos, servicios y estándares de programación cliente/servidor como primera fase de desarrollo en la cual reside la estructura principal de todo sistema orientado a la red, cuando un sistema cliente/servidor es implementado es necesario tomar en cuenta un procedimiento para llevar a cabo el enlace y por consiguiente una transferencia de datos.

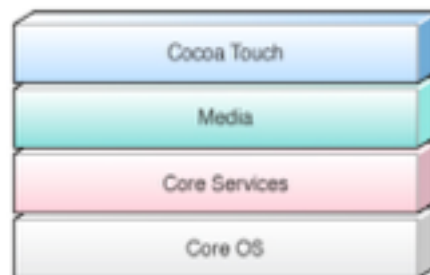
Es de suma importancia tener un conocimiento de las tecnologías disponibles que trabajan en las redes actuales, como se daría el enlace físico y lógico de la aplicación, si el sistema es alambico o inalámbrico. una vez conocido el material necesario, es posible continuar a formar la estructura base de la aplicación, Esto consiste en seleccionar la o las plataformas en las cual operara el software, el lenguaje de programación en el cual se codificara, que servicios y protocolos existen, para así seleccionar la complejidad de estos según se necesite o se demande en base a que tipo de funcionalidad prestara la aplicación, no es necesario seleccionar un protocolo muy complejo si la aplicación no lo requiere, un ejemplo de ellos son TCP y UDP, en la cual TCP es un protocolo orientado a la seguridad de el envío y acuse de recibo de los paquetes, por lo contrario UDP es un protocolo por el cual viajan datos en los cuales no es de suma importancia si es perdido alguno, lo cual para fines de esta aplicación no es necesario que se mantenga un enlace permanente de transferencia. Una siguiente fase es conocer el funcionamiento multimedia del dispositivo, cual es el alcance multimedia que ofrece, así como los componentes y librerías disponibles para su manejo.

Arquitectura General.

El sistema tiene como base una estructura de programación basada en protocolos y servicios, su funcionamiento está basado en servicios, el más utilizado tiene como nombre BONJOUR, con marco de trabajo Zeroconf, Bonjour es un método para descubrir servicios en una red de área local. Esta tecnología es muy utilizada en Mac OS X y permite a los usuarios establecer una red sin ningún tipo de configuración. Actualmente se usa en Mac OS X y otros sistemas operativos para encontrar impresoras y servidores de ficheros. También es usada por iTunes para encontrar música compartida, en iPhoto para encontrar fotos compartidas, iChat, Skype y el Proyecto Gizmo para encontrar otros usuarios de la red local, TiVo Desktop para encontrar video grabadores digitales, y SubEthaEdit para encontrar colaboradores de documentos. También es usado por Safari para encontrar servidores web locales y páginas de configuración para dispositivos locales, y por Asterisk para notificar de servicios y parámetros de configuración para teléfonos VoIP y llamadores automáticos. Sin una configuración especial de DNS, Bonjour sólo funciona en una única subred, lo cual suele ser una pequeña área. A veces se confunde Bonjour con un servicio de ordenador (por ejemplo, compartir ficheros) disponible a la Internet pública, lo cual puede ser considerado como un riesgo de seguridad. Bonjour no da acceso extra a ningún servicio; meramente los anuncia. Por ejemplo, un usuario puede explorar por una lista de ordenadores cercanos con los que puede compartir ficheros -Bonjour en esos ordenadores ha dicho al usuario que ese servicio está disponible- pero aun así tiene que proveer una contraseña para acceder al servicio en esas máquinas.

Apéndice B

Es importante mencionar y hacer una descripción básica de como es que esta compuesto el sistema operativo, observando como se organizan las capas del sistema, a continuación se muestra el sistema operativo, compuesto por las siguientes capas mencionandolas desde su nivel mas alto al mas bajo: CocoaTouch, Media, Core Services y Core OS.



Este proyecto tiene mayor énfasis en las capas Core OS y Media.
Core OS:



La capa nucleo del sistema operativo iOS permite tener un acceso y manejo de recursos necesarios para establece una conexión de red y hacer posible la transferencia de datos, es necesario tomar en cuenta recursos como los son Sockets para establecer el puerto de transferencia, Bonjour que facilita la busqueda y enlace en la red sin necesidad de configuración, y File System para hacer posible la transferencia y empaquetado de datos.

Media:



La capa Media consta de amplias librerías que permiten al programador tener un manejo de recursos multimedia, como lo es el Video, Audio e imagen, así como otros recursos como el manejo de animación en 2 y 3 Dimensiones.

En general es importante tener un control sobre la secuencia necesaria para establecer una publicación, un enlace y una transferencia de datos. esto se consigue accediendo debidamente a cada una de las capas del sistema, y utilizando las librerías y clases necesarias, siguiendo la metodología establecida por el lenguaje.