###### Licenciatura en Ingeniería de Software

###### Taller de Programación Visual III

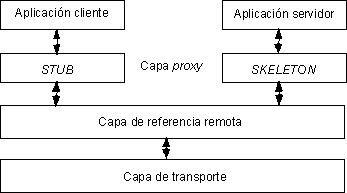
|  |  |
| --- | --- |
| UNIDAD 3 | Tarea # 9 |
| Nombre: Miguel Alejandro Guevara Mendoza | Fecha: 09/12/2018 |

Java RMI, C# WCF: 3er parcial

**JAVA RMI**

RMI (Remote Method Invocation) es un mecanismo que permite realizar llamadas a métodos de objetos remotos situados en distintas (o la misma) máquinas virtuales de Java, compartiendo así recursos y carga de procesamiento a través de varios sistemas.

La arquitectura RMI puede verse como un modelo de cuatro capas:



Capa 1

La primera capa es la de aplicación y se corresponde con la implementación real de las aplicaciones cliente y servidor. Aquí tienen lugar las llamadas a alto nivel para acceder y exportar objetos remotos.  
Cualquier aplicación que quiera que sus métodos estén disponibles para su acceso por clientes remotos debe declarar dichos métodos en una interfaz que extienda java.rmi.Remote.

Capa 2

Es la capa proxy, o capa stub-skeleton. Esta capa es la que interactúa directamente con la capa de aplicación. Todas las llamadas a objetos remotos y acciones junto con sus parámetros y  
retorno de objetos tienen lugar en esta capa.

Capa 3

Es la de referencia remota, y es responsable del manejo de la parte semántica de las invocaciones remotas. También es responsable de la gestión de la replicación de objetos y realización  
de tareas específicas de la implementación con los objetos remotos, como el establecimiento de las persistencias semánticas y estrategias adecuadas para la recuperación de conexiones perdidas. En esta capa se espera una conexión de tipo stream (stream-oriented connection)  
desde la capa de transporte.

Capa 4

Es la capa de transporte. Es la responsable de realizar las conexiones necesarias y manejo del transporte de los datos de una máquina a otra. El protocolo de transporte subyacente para RMI es JRMP (Java Remote Method Protocol), que solamente es “comprendido” por programas Java.

**Spring MVC**

Spring Mvc es una alternativa de framework basado en el patrón modelo-vista-controlador, después de haber aprendido de errores de frameworks anteriores a estos como Jakarta Struts y otras alternativas.

Particularmente, el equipo Spring siente que no hay suficiente separación entre las capas de presentación y la capa de manejo de request, y entre la capa de manejo de request y el modelo.

El framework tiene un conjunto de interfaces que después se implementan para proporcionar la funcionalidad correspondiente. Las interfaces están acopladas claramente al Servlet Api. La clase DispatcherServlet está en el front controller y es responsable de delegar y coordinar el control entre varias interfaces en la fase de ejecución durante una petición Http. Las interfaces más importantes definidas en Spring Mvc, y sus responsabilidades, son las siguientes:

1. HandlerMapping: permite manejar peticiones de entrada.
2. HandlerAdapter: ejecución de objetos que permiten manejar las peticiones entrantes.
3. Controller: está entre el modelo y la vista, y permite manejar peticiones entrantes y redirigirlas a la respuesta adecuada.
4. Vista: responsable de retornar una respuesta al cliente.
5. ViewResolver: selecciona una vista basada en un nombre lógico de la vista.
6. HandlerInterceptor: intercepta las peticiones entrantes , es comparable pero no igual a los filtros de Servlet.
7. LocaleResolver: resuelve y opcionalmente salva el local de un usuario individual.
8. MultipartResolver: facilita trabajar con ficheros de subida wrapping peticiones de entrada.

**WFC**

Windows Communication Foundation o WCF, (también conocido como Indigo), es la nueva plataforma de mensajería que forma parte de la API de la Plataforma .NET 3.0 (antes conocida como WinFX, y que no son más que extensiones para la versión 2.0). Se encuentra basada en la Plataforma .NET 2.0 y de forma predeterminada se incluye en el Sistema Operativo Microsoft Windows Vista. Consiste en un número de bibliotecas de clase para desarrollar sistemas que envían mensajes entre servicios y clientes sobre diversos mecanismos de transporte.

Una aplicación WCF está compuesta por:

1. Clientes: Son aplicaciones que inician la comunicación.
2. Servicios: Son aplicaciones que esperan los mensajes de los clientes y responden a los mismos.

Los mensajes son enviados entre Endpoints. Un Endpoint es un lugar donde un mensaje es enviado, o recibido, o ambos.  
  
Con WCF, nos independizamos de los mecanismos de transporte y de los protocolos, ya que podemos tener un único servicio programado de una manera particular y exponerlo de diferentes formas, cada punto de contacto con el exterior de un servicio en WCF se conoce como Endpoint. Por lo tanto decimos que un servicio WCF tiene un número de Endpoints, cada cual diferenciado por dirección, puerto, protocolo de transporte, etc.

**ASP.NET**

Es una tecnología gratuita que permite a los programadores crear páginas web dinámicas, desde websites personales hasta aplicaciones web empresariales. Todo lo que necesitas para empezar es descargar .NET Framework y una herramienta de desarrollo.  
ASP.NET es un Framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML.

Características

Las páginas de ASP.Net, conocidas oficialmente como "web forms" (formularios web), son el principal medio de construcción para el desarrollo de aplicaciones web. Los formularios están contenidos en archivos con esta extensión ASPX; en jerga de programación, estos archivos típicamente contienen etiquetas HTML o XHTML estático, y también etiquetas definiendo controles Web que se procesan del lado del servidor y Controles de Usuario donde los desarrolladores colocan todo el código estático y dinámico requerido por la página web.