

Proyecto FINAL PR

Objetivos

- Construir una aplicación cliente-servidor que utilice sockets TCP/UDP para comunicarse y RMI para invocar métodos remotos(*REST*).
- La aplicación debe ser capaz de gestionar y dar servicio a solicitudes de múltiples dispositivos concurrentes.
- El cliente se encargará de la interfaz del usuario y de enviar solicitudes al servidor.
- El servidor manejará las solicitudes, interactuará con la base de datos relacional y enviará respuestas al cliente.
- Se utilizará RMI para conectar diversos dispositivos al sistema central. Estos dispositivos pueden ser de diferentes tipos y estar ubicados en distintas redes. (*Dado que RMI se encuentra deprecated para nuevas versiones de JDK, procuren implementar la lógica de negocio mediante REST a fin de permitir el acceso remoto. Es decir, se utilizará la arquitectura RESTful a fin de conectar diversos dispositivos*).

Requerimientos MÍNIMOS

- **Sockets TCP:**
 - Para establecer una conexión confiable y orientada a la conexión entre el cliente y el servidor.
 - **Sockets UDP:**
 - Para enviar mensajes de forma rápida, por ejemplo, notificaciones o actualizaciones en tiempo real.
 - **RMI:**
 - Para permitir la configuración de dispositivos remotos.
- (RESTful**
- Servicios Web basados en REST en el servidor.
 - Utilizar comandos HTTP: GET, POST, PUT y DELETE.
-)

- **Multithreading:**

- Es necesario utilizar programación concurrente en el lado del servidor, a fin de manejar cada cliente conectado en un hilo separado.

- **Seguridad:**

- Controlar los permisos de acceso, implementando un mecanismo de autenticación robusto. Utilizar protocolos de cifrado para proteger la comunicación entre los dispositivos.
- La aplicación debe funcionar correctamente y permitir la conexión y comunicación de múltiples dispositivos.

- **Funcionalidad:**

- La aplicación debe funcionar correctamente, sin errores críticos.
- Debe cumplir con todos los requisitos establecidos.
- El sistema debe permitir la conexión de al menos dos dispositivos diferentes.
- La interacción con la base de datos debe ser fluida y eficiente, funcionando correctamente a través de la interfaz RMI (RESTful).
- Utilizar los sockets UDP y TCP de manera adecuada.

- **Documentación:**

- Introducción:
 - Clara y concisa.
 - Objetivos y alcance.
- Requisitos:
 - Funcionales.
 - No funcionales.
- Diseño:
 - Arquitectura utilizada.
 - Explicación de la elección de tecnologías.
 - Decisiones de implementación.
 - Diagramas UML.
- Implementación:

- Explicación del código JAVA, librerías, herramientas e instrucciones SQL utilizadas.
- Resultados:
 - Incluir imágenes.
- Manual de usuario.
 - Incluir imágenes.
 - Instrucciones claras y concisas para el uso del sistema.
- Conclusiones:
 - Logros y limitaciones.
- Referencias:
 - Formato APA.

Evaluación Primera PARTE (100% = 20% de la calificación FINAL)

El proyecto será evaluado considerando lo siguiente:

- Funcionalidad (20%):
 - *Debe cumplir con todos los requisitos establecidos.*
 - *Correcta implementación de RMI(RESTful) para comunicarse entre dispositivos.*
 - *La aplicación debe funcionar correctamente, sin errores críticos. Se deben manejar las excepciones relacionadas con las comunicaciones de red y operaciones remotas.*
 - *Gestión eficiente de la BDR, la interacción con la base de datos debe ser fluida y responder de manera adecuada a las acciones del usuario, funcionando correctamente a través de la interfaz RMI(RESTful).*
 - *Manejo correcto de errores y transacciones.*
 - *El sistema debe permitir la conexión de al menos dos dispositivos diferentes.*
 - *Utilizar los sockets UDP y TCP de manera adecuada.*
 - *Implementar mecanismos de seguridad para proteger la comunicación entre los dispositivos.*

- Diseño (40%):
 - *La arquitectura del sistema debe ser clara.*
 - *Las interfaces RMI(RESTful) deben estar bien diseñadas.*
 - *La interfaz de usuario debe ser intuitiva y fácil de usar.*
 - *Se debe definir de manera clara como se almacenarán los datos.*
 - *Implementar medidas de seguridad para proteger los datos (autenticación, autorización)*
- Programación (35%):
 - *El código es limpio, bien estructurado y fácil de entender.*
 - *Se hace uso eficiente de las estructuras de datos y algoritmos utilizados.*
 - *Incluye comentarios.*
 - *Uso de buenas prácticas de programación (KISS, DRY y YAGNI).*
- Documentación (5%):
 - *La documentación debe ser clara, concisa y completa.*
 - *Se debe documentar la configuración para RMI(RESTful).*
 - *Se deben incluir diagramas, código fuente, instrucciones SQL y la presentación de resultados.*
 - *Diseño de la BD.*
 - *Diagrama de la arquitectura cliente-servidor.*
 - *Código para la creación de los sockets, tanto en el cliente como en el servidor.*

Evaluación Segunda PARTE (100% = 15% de la calificación FINAL)

En la segunda parte se evalúa la PRESENTACIÓN ORAL del proyecto:

- Duración: 30 minutos.
- Individual.
- La presentación debe ser clara, el estudiante debe comunicar de forma efectiva los aspectos más relevantes del proyecto.

- Se debe realizar una demostración de la aplicación.
- Explicación de las decisiones de diseño.
- Explicación de los desafíos enfrentados durante el desarrollo.
- Se realizarán modificaciones en tiempo real.

**Nota Bene:*

Este es un conjunto de requerimientos mínimos. Pueden **ampliar y personalizar** el proyecto según sus intereses y conocimientos.