



**ESCUELA POLITECNICA NACIONAL  
FACULTAD DE INGENIERIA EN SISTEMAS  
PROGRAMACION I  
PROYECTO FINAL  
GRUPO 5**



## **Manual Técnico: Sistema de Prueba Psicosensométrica GR5**

### **1. Introducción**

Este manual técnico describe la arquitectura, componentes y funcionamiento interno del Sistema de Prueba Psicosensométrica GR5. Está dirigido a desarrolladores, técnicos y personal de soporte que necesiten comprender, mantener o extender el sistema.

### **2. Arquitectura del Sistema**

El sistema GR5\_ExamenPsicosensometrico sigue una arquitectura cliente-servidor para la persistencia de datos y una arquitectura de hardware-software para la interacción física.

- **Capa de Presentación (Frontend Java Swing):**

Interfaz de usuario gráfica desarrollada en Java Swing.

Gestiona la interacción del usuario (ingreso de datos, botones, visualización de resultados).

Las clases GR5\_Inicio.java y GR5ExamenPsicosensometrico.java son los componentes principales de esta capa.

- **Capa de Lógica de Negocio (Backend Java):**

Contiene la lógica para la gestión de usuarios, la comunicación con el Arduino, el procesamiento de datos de la prueba y la interacción con la base de datos.

Clases clave: GR5\_Inicio.java, GR5ExamenPsicosensometrico.java, GR5\_Sesion.java, y GR5\_DBConnexion.java.

- **Capa de Persistencia de Datos (MySQL Database):**

Almacena la información de usuarios y los resultados de las pruebas.

Gestionada por un servidor MySQL. La conexión se realiza a través de JDBC.

- **Capa de Hardware (Arduino):**

Microcontrolador que controla los componentes físicos (LED, Buzzer) y lee la entrada del usuario (Botón).

Contiene la lógica embebida para generar estímulos, detectar respuestas y comunicarse serialmente con la aplicación Java.

El código codigoexamen.ino reside en esta capa.

### **3. Tecnologías y Herramientas Utilizadas**

- **Lenguajes de Programación:**

**Java:** Para la aplicación de escritorio.

**C++ (Arduino IDE):** Para el firmware del microcontrolador Arduino.

- **Base de Datos:**

**MySQL:** Sistema de gestión de bases de datos relacionales.

- **Bibliotecas/Frameworks Java:**

**Java Swing:** Para la construcción de la interfaz gráfica de usuario.

**JDBC (Java Database Connectivity):** Para la conexión y operaciones con la base de datos MySQL.



**ESCUELA POLITECNICA NACIONAL  
FACULTAD DE INGENIERIA EN SISTEMAS  
PROGRAMACION I  
PROYECTO FINAL  
GRUPO 5**



**jSerialComm:** Librería para la comunicación serial entre la aplicación Java y el Arduino.

- **Entornos de Desarrollo Integrado (IDE):**

NetBeans IDE (sugerido por la estructura del proyecto y archivos .form).  
Arduino IDE.

#### **4. Estructura del Proyecto (Directorio Principal app)**

El proyecto está organizado lógicamente en clases Java y un archivo de código Arduino:

- **codigoexamen.ino:**

**Descripción:** Contiene el firmware para el Arduino. Define los pines para LED, Buzzer y Botón. Implementa la lógica para generar estímulos, detectar presiones de botón, calcular tiempos de reacción y gestionar la comunicación serial con la aplicación Java.

**Funcionalidades Clave:**

setup(): Inicializa pines, comunicación serial.

loop(): Bucle principal que gestiona estados (inicio, activo, detenido), lee comandos seriales y genera/detecta estímulos/respuestas.

GR5\_procesarComandoSerial(): Interpreta comandos recibidos de Java (INICIAR, DETENER, CONFIG).

GR5\_generarEstimulo(): Crea estímulos aleatorios (LED/Buzzer) y calcula si la combinación es "correcta" según la configuración.

GR5\_esperarRespuestaUsuario(): Monitorea el botón y calcula el tiempo de reacción.

- **GR5\_DBConnexion.java:**

**Descripción:** Clase Singleton para gestionar la conexión a la base de datos MySQL.

Asegura que solo exista una instancia de conexión a la vez. Contiene las credenciales de conexión (GR5\_DB\_URL, GR5\_DB\_USER, GR5\_DB\_PASS).

**Métodos Clave:**

GR5\_getConnection(): Retorna una conexión activa a la base de datos.

GR5\_closeConnection(): Cierra la conexión a la base de datos.

- **GR5\_Inicio.java / GR5\_Inicio.form:**

**Descripción:** La ventana principal de la aplicación para la gestión de usuarios. Permite agregar, modificar y eliminar usuarios, y seleccionar un usuario para iniciar la prueba.

**Funcionalidades Clave:**

Validación de campos de entrada (nombre, apellido, cédula).

Algoritmo de validación de cédulas ecuatorianas (GR5\_validarCedulaEcuatoriana).

Interacción con la base de datos para operaciones CRUD de usuarios.

Carga y visualización de usuarios en una JTable.

Transfiere los datos del usuario seleccionado a GR5\_Sesion.java.

- **GR5\_Sesion.java:**

**Descripción:** Clase estática que actúa como un contenedor de datos de sesión.

Almacena temporalmente la usuarioID, nombre, apellido y cedula del usuario actualmente seleccionado.



**ESCUELA POLITECNICA NACIONAL  
FACULTAD DE INGENIERIA EN SISTEMAS  
PROGRAMACION I  
PROYECTO FINAL  
GRUPO 5**



- **GR5ExamenPsicosensometrico.java / GR5ExamenPsicosensometrico.form:**

**Descripción:** La ventana principal donde se realiza el examen. Gestiona la comunicación serial, la configuración del examen, la visualización de resultados en tiempo real y la persistencia del historial de pruebas.

**Funcionalidades Clave:**

**Comunicación Serial:** Utiliza jSerialComm para conectar y escuchar datos del Arduino en un hilo separado (hiloEscuchaGR5).

**Configuración Dinámica:** Envía comandos de configuración al Arduino (intensidad, tipo de estímulo, velocidad).

**Control de Prueba:** Botones para iniciar, detener y reiniciar la prueba.

**Actualización de UI:** Actualiza los JLabel con el estado del estímulo, resultado de reacción, tiempo de reacción, aciertos y errores.

**Persistencia de Resultados:** Guarda los resultados de la prueba (aciertos, errores, tiempo promedio, fecha) en la base de datos.

**Historial:** Carga y muestra el historial de pruebas del usuario desde la base de datos.

**Exportación CSV:** Permite exportar el historial a un archivo .csv.

**GR5\_detectarYConectarArduino():** Maneja la auto-detección y conexión con el puerto serial del Arduino.

**GR5\_enviarComandoArduino():** Envía cadenas de comando al Arduino.

**GR5\_procesarMensajeArduino():** Parsea los mensajes recibidos del Arduino (GR5\_ESTIMULO, GR5\_RESPUESTA).

## **5. Base de Datos (MySQL)**

El sistema interactúa con una base de datos MySQL llamada PsicoAppDB donde existen al menos dos tablas principales:

**usuarios:**

id (PRIMARY KEY, AUTO\_INCREMENT)

nombre

apellido

cedula (UNIQUE)

**resultados:**

id (PRIMARY KEY, AUTO\_INCREMENT)

usuario\_id (FOREIGN KEY a usuarios.id)

fecha (timestamp)

aciertos

errores

tiempo\_promedio\_reaccion

## **6. Flujo de Datos y Comunicación**

**Inicio:** La aplicación Java se ejecuta, intenta conectar a la BD y al Arduino.

**Gestión de Usuario:** Datos ingresados por el usuario se validan y se envían a la BD (GR5\_DBConnexion). Datos de usuario seleccionado se cargan en GR5\_Sesion.

**Configuración de Prueba:** Parámetros seleccionados en



**ESCUELA POLITECNICA NACIONAL  
FACULTAD DE INGENIERIA EN SISTEMAS  
PROGRAMACION I  
PROYECTO FINAL  
GRUPO 5**



GR5ExamenPsicosensometrico.java se serializan y se envían vía comunicación serial al Arduino (codigoexamen.ino).

**Ejecución de Prueba:**

El Arduino genera un estímulo.

El Arduino envía el estado del estímulo y si es "correcto" a Java.

El usuario presiona el botón.

El Arduino detecta la presión, calcula el tiempo, y envía el resultado (acierto/fallo y tiempo de reacción) a Java.

Java actualiza la UI y acumula contadores.

**Fin de Prueba / Guardado:**

Resultados finales (aciertos, errores, tiempo promedio) se envían a la BD vía

GR5\_DBConnexion.

El historial se consulta de la BD y se muestra/exporta.

## **7. Consideraciones de Desarrollo y Mantenimiento**

**Singleton para DB Connection:** El patrón Singleton en GR5\_DBConnexion es una buena práctica para gestionar recursos de conexión.

**Hilo Separado para Serial:** La comunicación serial se maneja en un Thread dedicado en GR5ExamenPsicosensometrico.java para evitar el bloqueo de la interfaz de usuario.

**Manejo de Errores:** Se utilizan try-catch para manejar excepciones de SQL y serialización. Considerar un manejo más robusto y mensajes de error más informativos.

**Seguridad:** Las credenciales de la base de datos están hardcodeadas en GR5\_DBConnexion.java. Para entornos de producción, se recomienda externalizarlas o usar prácticas más seguras (variables de entorno, archivos de configuración protegidos).

**Escalabilidad:** Para un entorno multiusuario concurrente más allá de una aplicación de escritorio, la clase GR5\_Sesion podría necesitar un rediseño.

**Extensibilidad:** La estructura modular permite añadir nuevos tipos de estímulos o pruebas con cambios limitados a las clases existentes.