

## Smart Saw

Rodríguez Mariano	Romano Darío	Sandoval Leandro	Vivas Pablo
39.336.952	33.901.015	41.548.235	38.703.964

Comisión: 02-2900

Grupo: M2

Universidad Nacional de La Matanza,  
Departamento de Ingeniería e Investigaciones Tecnológicas,  
Florencio Varela 1903 - San Justo, Argentina

### Resumen

Smart Saw es una sierra de alta precisión medida en centímetros para realizar cortes para objetos pequeños o usos más “micro” comparado con las sierras automáticas usuales de mayor tamaño. Esta tiene funcionalidades de traslación horizontal para facilitar punto de corte, y activación de corte propia de la sierra; leds indicadores del modo en el que la sierra se encuentra en un momento concreto; ultrasonidos para corroborar la precisión de las distancias y motores de corriente continua para la puesta en marcha de sus modos. Las distancias se introducirán manualmente a través de la interfaz de usuario y además se podrá indicar el sentido de movimiento.

Se pretende que está sea utilizada para cortar objetos en un eje vertical, es decir, únicamente en el eje del largo de la sierra, y estos objetos serán empujados por una prensa para mayor seguridad.

## 1 Introducción

En este trabajo práctico, se presenta el desarrollo de la aplicación Smart Saw, esta aplicación permite a los usuarios controlar la sierra, como prenderla/apagarla e indicar unidades de desplazamiento y sentido de movimiento. Su propósito es simplificar la automatización parcial de pequeños cortes que requieren precisión sobre materiales deseados, proporcionando una interfaz intuitiva y fácil de usar para entornos de uso sobre aplicaciones cotidianas, donde personas comunes puedan beneficiarse de una herramienta accesible, optimizando trabajos que normalmente se harían de forma manual.

Este informe describe la funcionalidad de la aplicación, su estructura, y como se lleva a cabo la interacción con el sistema embebido. Asimismo, se destacan las dificultades durante el desarrollo y las lecciones aprendidas.

## 2 Desarrollo

La aplicación desarrollada constará de 5 actividades posibles que responden o transicionan entre ellos a partir de diversas acciones realizadas por el usuario o por

algún evento dado proveniente el sistema embebido mediante bluetooth. A partir de esto el sistema actuará acorde a lo programado para cumplir sus funciones, entre ellas, encendido y/o apagado de motores, leds, emisión de señales ultrasonidos y cálculos de distancia.

### **Enlaces**

- URL del proyecto : [SOA - Smart Saw - Versión Reentrega](#)
- URL del repositorio: [Smart Saw - GitHub](#)

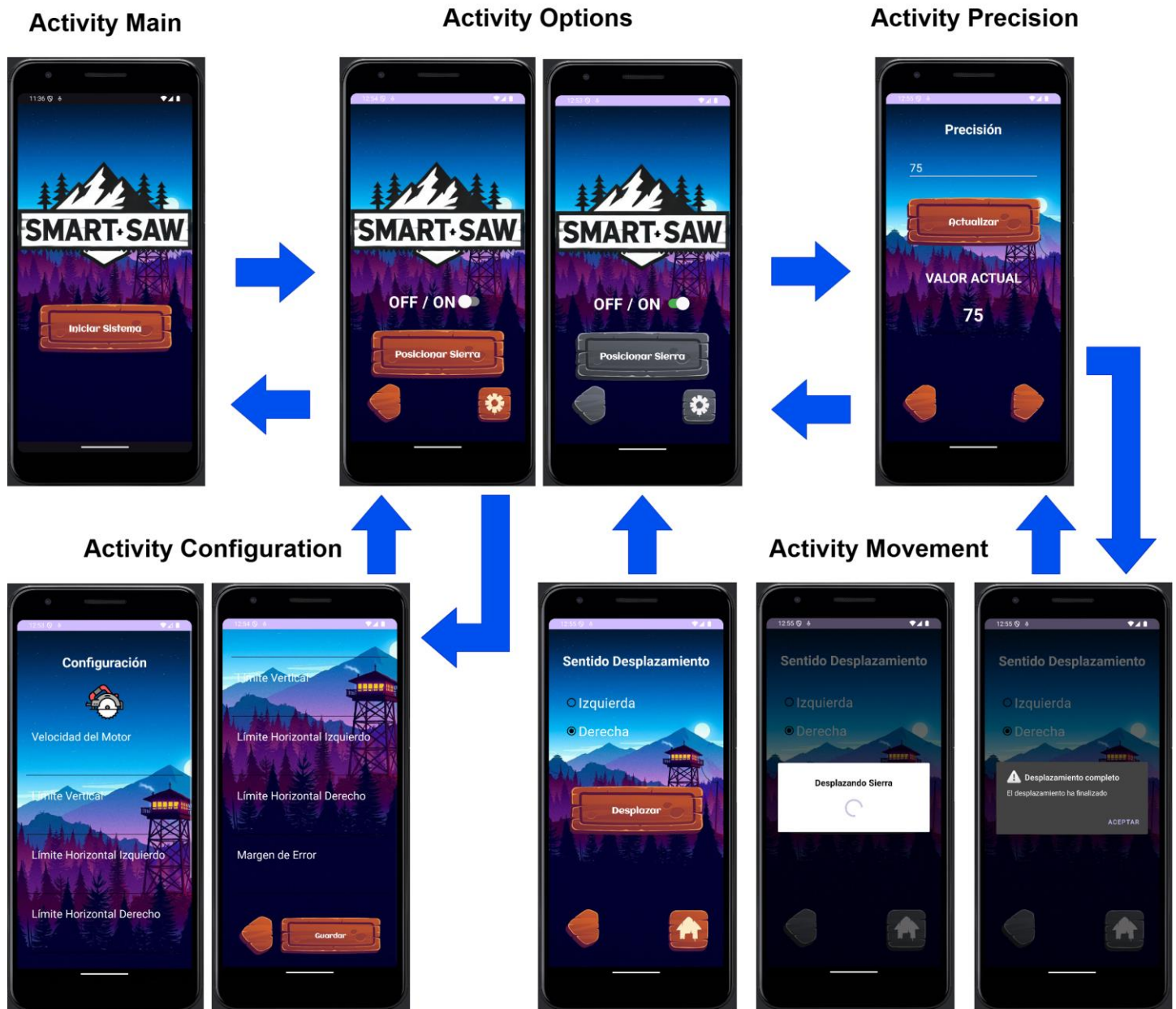
### **Navegacion de Activities**

Se adjunta un archivo PDF correspondiente a los mockups realizados anteriormente al desarrollo de las aplicaciones en Android.



Smart Saw -  
Mockups - Balsamiq

A continuacion, en base al archivo anterior, se adjunta las diferentes pantallas/activies para representar el flujo de navegacion entre ellas.



### Manual de usuario de la aplicación

Se pretende que el usuario de la aplicación que se comunica con el sistema embebido use el sistema para realizar cortes precisos de objetos.

Para esto se debe conectar al dispositivo via bluetooth. En el boton “**Iniciar Sistema**” correspondiente a la **Activity Main** se debera realiza la verificacion exitosa de la conexion para seguir avanzando con la interfaz.

Luego, el usuario se encontrara en la **Activy Options**, la cual podra interactuar con diferentes botones:

- **Switch ON/OFF:** Este switch debera corresponder al encendido y apagado de la sierra. Una vez que el switch se encuentre en ON, los botones de esta actividad se deshabilitaran para evitar interacciones multiples con el sistema embebido.

En caso de que al realizar el corte se pasara del limite vertical permitido, la aplicación lanzara un popup indicando que se paso de dicho limite y a continuacion la sierra se apagara, es decir, el switch pasara a OFF automaticamente junto con las acciones que este implica.

- **Boton de Configuración:** Se podra ir a la **Activity Configuration** para poder realizar ajustes de variables correspondientes al sistema embebido, como modificacion de limites horizontales, limite vertical, margen de error de los sensores ultrasonido y la velocidad del motor. Luego podra guardar esta configuracion o de lo contrario podra volver a **Activity Options**.
- **Boton de Retorno:** El usuario podra regresar a **Activity Main**.
- **Boton Posicionar Sierra:** Como indica el boton, la aplicación se redirijira a **Activity Precision**, en el siguiente parrafo se detallara lo que contendra esta pantalla.

En **Activity Precision**, el usuario debera ingresar un valor numerico, el cual indicara la unidad de desplazamiento (CM) que deseara desplazar la sierra. Luego, con el boton “**Actualizar**”, establece el nuevo valor de desplazamiento en el Sistema Embebido, esto es por si es que anteriormente se hizo alguna accion de desplazamiento, queda guardado el valor del ultimo desplazamiento realizado. Una vez indicado el **VALOR ACTUAL**, se podra avanzar a **Activity Movement**.

En esta ultima actividad, se debera seleccionar la direccion de desplazamiento que el usuario desee para correr la sierra horizontalmente. Las interacciones con esta pantalla son las siguientes:

- **Boton Desplazar:** Al apretar este boton, la sierra comenzara a desplazarse horizontalmente, encendiendo el motor de desplazamiento hasta la cantidad de unidades que se le indico anteriormente. Esto desde la aplicación muestra un popup con un circulo de carga (componente ProgressBar). Una vez finalizado el desplazamiento, el popup se actualiza indicando que el desplazamiento finalizo. Anteriormente a realizar la accion de desplazamiento, se realiza una verificacion de los limites horizontales para conocer si es posible realizar el desplazamiento con la direccion y la cantidad solicitada, en caso afirmativo, se realiza la accion de desplazar; caso negativo, se mostrara un popup indicando que el desplazamiento no es posible de realizar.
- **Boton de Retorno:** El usuario podra ir a la actividad anterior (Activity Precision) por si quiere modificar el valor de desplazamiento.
- **Boton de Retorno al Menu de Opciones:** Este boton retorna al menu de opciones, facilitando al usuario por si quiere desplazar la sierra y enseguida encender la sierra, sin tener que pasar por la activy intermedia, mejorando la experiencia de usuario.

### 3 Conclusión

Durante el desarrollo de la aplicación nos encontramos con distintos problemas, a continuación, se mencionan los más importantes:

- **Problemas de versionado:** Las dependencias de Android variaban entre cada integrante del grupo, y es por esto que al integrar algunas funcionalidades nos encontrábamos con inconvenientes. Para solucionar esto, definimos una versión entre el equipo para poder trabajar sobre esta misma. Destacando Gradle y .Idea como el mayor problema a la hora de hacer un pull y push en el repositorio.
- **Uso de componentes para el front-end:** Tanto para diseño de botones, ajustes responsive o componentes no vistos en clase se debió investigar para poder sumarlos al proyecto.
- **Inconvenientes entre el sistema embebido/bluetooth:** La lectura que se enviaba vía bluetooth en el sistema embebido no se leía bien. Además, se tuvo problemas al probar la implementación de bluetooth, ya sea, en el Arduino como en el Android dado que al no tener el componente físico dificultaba realizar las pruebas de funcionamiento correcto, en donde se terminó delegando dicha prueba a la persona que tenía el Arduino. Todo esto tuvo como consecuencias demoras en avances del proyecto y dificultades al paralelizar tareas.
- **Dificultad con el Android Studio:** Dado que dicha herramienta requiere grandes recursos para funcionar, la misma nos ralentizó en determinados momentos perjudicando y prolongando los tiempos en testeo de nuevas funcionalidades.

#### 4 Bibliografía / Referencias

Ing. Esteban A. Carnuccio, Ing. Mariano Volker, Ing. Raul Villca & Ing. Matias Adagio: [Apunte Teorico sobre el Sistema Operativo Android](#).

Ing. Matias Adagio: [Sistemas Embebidos Android](#). Universidad Nacional de la Matanza.

Ing. Esteban A. Carnuccio: [Sensores en Android](#). Universidad Nacional de la Matanza.

Ing. Esteban A. Carnuccio: [Repositorio de Ejemplos Android](#). Material SOA. GitLab