

<b>Práctica Build 1</b>	Entregable: Este documento y el histórico bash
	Fecha Entrega: 19/2/20
	Equipo: Individual
	Herramientas: IDE Android y Arduino IDE

## Introducción

La construcción de un entregable puede hacerse empleando un IDE pero en entornos con múltiples plataformas y diferentes equipos especializados esto no es recomendable y debe automatizarse con procesos independientes del IDE.

Acceso a máquina virtual: <https://eim-laboratoriovirtual.unavarra.es/laboratorio/#/>

- 1) Acceso con alumno@ldap
- 2) En cada contenedor: alumno/P6dJbyA2020

Para obtener histórico desde CLI: `>$ history`

## Desarrollo de la práctica

1.-Instale en su contenedor dos IDE específicos:

Android Studio  
Arduino IDE

2.-Prepare la automatización de dos *entregables*:

2.1.-Android

Crear proyecto con una actividad ActivityAxxxxx (donde xxxxx es el número de alumno) que compile. No es necesario añadir funcionalidad alguna.  
Como *Android Studio* instala *Gradle*, buscar un *target* adecuado y ejecútelo desde CLI.

Ver “compilarActividadAndroid” adjunto a este documento

2.2.-Sensor basado en Arduino.

Proyecto referencia: <https://github.com/ladislav/Bare-Arduino-Project>

Según lo familiarizado que esté el alumno con el entorno, hay dos opciones:

- a) Crear un sketch propio y compilarlo desde CLI con un *makefile* (recomendado ***Bare-Arduino-Project***).
- b) Emplear el proyecto ***FooProject*** incluido en ***Bare-Arduino-Project*** siguiendo las instrucciones para compilar desde CLI.

Ver “compilarProyectoArduino” adjunto a este documento

3.- Comentar cómo afecta tener diferentes plataformas de compilación en un mismo proyecto y qué características debería tener un servicio SaaS que permitiera compilar aplicaciones para diferente hardware.

Tener diferentes plataformas de compilación en un mismo proyecto puede ser una molestia contraproducente para los desarrolladores del proyecto. Lo ideal sería tener una sola plataforma de compilación que compilase todo tipo de software sin importar el lenguaje, hardware, etc.

No obstante, el tener varias plataformas de compilación puede ofrecer mayor libertad a la hora de configurar las distintas plataformas para que se adapten mejor a tu proyecto. Otra ventaja puede ser la posibilidad de sustituir un compilador que no se adapte a nosotros o no sea de nuestro gusto por otro más a nuestro estilo, como puede pasar con el proyecto “Bare-Arduino-Project”, el cual huía del “terrible” IDE de Arduino.

Un servicio SaaS que permita compilar aplicaciones para diferente hardware debería tener entre sus características más evidentes la elección del hardware que se va a usar y si se pudiera hacer consultando la BIOS para una autoselección, sería lo ideal. Guardar perfiles sería lo siguiente que tendría que tener, así nos evitaríamos tener que configurar de nuevo cada vez que queramos volver a compilar.