TECNOLÓGICO NACIONAL DE MÉXICO

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE ORIZABA

MATERIA

TECNOLOGÍAS E INTERFACES DE COMPUTADORAS

DOCENTE

NORMA RODRIGUEZ RODRIGUEZ

HORARIO

13:00 – 14:00

INTEGRANTES DEL EQUIPO

GARCIA DAMIAN BEATRIZ ANDREA
GARCIA HERNANDEZ RICARDO
GONZALEZ FRANCO XIMENA
GONZALEZ JIMENEZ ALEXIS

UNIDAD

3

NOMBRE DE LA ACTIVIDAD

ACTIVIDAD 2. CUADRO COMPARATIVO

FECHA DE ENTREGA

MARTES, 11 DE NOVIEMBRE DEL 2025

INTRODUCCIÓN

En esta actividad se presenta un análisis comparativo de tres herramientas y bibliotecas empleadas en la gestión de puertos dentro de distintos lenguajes de programación: Node-SerialPort, Johnny-Five y Boost.Asio. A través del cuadro comparativo se examinan aspectos como compatibilidad, facilidad de uso, rendimiento, ventajas, desventajas y ejemplos de código. El propósito es comprender cómo cada una permite la transferencia de datos o comunicación con dispositivos externos mediante puertos estándar, adaptándose a diferentes entornos y niveles de programación.

TECNOLÓGICO NACIONAL DE MÉXICO

ACTIVIDAD 2:

Elaborar un cuadro comparativo de las distintas herramientas de la actividad 1, incluyendo aspectos como compatibilidad, facilidad de uso, rendimiento y ejemplos de código básicos.

Aspecto	Node-SerialPort (JavaScript / Node.js)	Johnny-Five (JavaScript / Node.js)	Boost.Asio (C++)
Tipo / Naturaleza	Biblioteca para comunicación directa con puertos seriales.	Biblioteca de alto nivel para control de hardware físico (robótica e IoT).	Biblioteca C++ para I/O asíncrona, incluye comunicación por puerto serial.
Herramienta asociada	Node.js (entorno de ejecución de JavaScript).	Node.js (entorno de ejecución de JavaScript).	C++ (lenguaje y compilador, como GCC o MSVC).
Compatibilidad	Multiplataforma: Windows, Linux y macOS. Compatible con dispositivos que usen puertos COM o USB virtuales.	Multiplataforma: Windows, Linux y macOS. Compatible con Arduino, Raspberry Pi, BeagleBone, Tessel, entre otros.	Multiplataforma: Windows, Linux y macOS. Soporta múltiples arquitecturas y dispositivos industriales.
Facilidad de uso	**Alta.** Tiene instalación sencilla con npm install serialport y una API simple basada en eventos.	**Muy alta.** Controla hardware con pocas líneas de código; ideal para principiantes.	**Media-baja.** Requiere conocimientos avanzados de C++ y configuración del entorno Boost.
Rendimiento	Bueno para proyectos de comunicación serial estándar o monitoreo continuo.	Adecuado para prototipos, IoT y robótica ligera, aunque no enfocado en tiempo real.	Excelente: alto rendimiento, baja latencia y gran estabilidad en sistemas industriales.
Ventajas	- Multiplataforma y de código abierto.	- API muy sencilla y orientada a objetos.	- Comunicación síncrona y asíncrona. - Rendimiento profesional.

TECNOLÓGICO NACIONAL DE MÉXICO

	- Basada en eventos (no bloquea el programa). - Integrable con aplicaciones web o IoT.	- Amplia compatibilidad con sensores y actuadores. - Gran comunidad y abundante documentación.	- Portabilidad entre sistemas. - Muy estable y optimizada.
Desventajas	- Requiere instalación de dependencias nativas. - No apta para tareas de tiempo real crítico. - Puede necesitar permisos administrativos.	- Depende de Node-SerialPort y del firmware Firmata. - No está diseñada para sistemas industriales. - Ligeramente más lenta por su nivel de abstracción.	- Configuración inicial más compleja. - Documentación técnica avanzada. - Mayor curva de aprendizaje.

Ejemplo básico de código	javascript
	`const { Board, Led } = require("johnny-five");`
	`const board = new Board();`
	`board.on("ready", () => {`
	`  const led = new Led(13);`
	`  led.blink(1000);`
	`});`
		javascript
		`const { Board, Led } = require("johnny-five");`
		`const board = new Board();`
		`board.on("ready", () => {`
		`  const led = new Led(13);`
		`  led.blink(1000);`
		`});`
			`#include <boost/asio.hpp>`
			`#include <iostream>`
			`using namespace boost::asio;`
			`int main() {`
			`  io_context io;`
			`  serial_port port(io, "COM3");`
			`  port.set_option(serial_port_base::baud_rate(9600));`
			`  write(port, buffer("Hola desde C++\n"));`
			`  char data[100];`
			`  read(port, buffer(data, 100));`
			`  std::cout << "Recibido: " << data << std::endl;`
			`}`

Funcionamiento del código	Este código abre el puerto serial COM3, envía el mensaje *"Hola dispositivo!"* y muestra en consola los datos recibidos desde un dispositivo conectado (como un Arduino o sensor). Demuestra cómo enviar y recibir datos por un puerto estándar usando Node.js.	Este código se comunica con una placa Arduino a través del puerto USB (serial) y hace que un LED conectado al pin 13 parpadee cada segundo. Muestra cómo controlar hardware físico mediante la comunicación por puertos de forma sencilla.	Este programa en C++ abre el puerto serial COM3, envía un mensaje al dispositivo conectado y luego lee la respuesta recibida. Es un ejemplo clásico de transferencia de datos mediante un puerto estándar con alto control y rendimiento.

CONCLUSIÓN

Las tres herramientas analizadas demuestran que el manejo de puertos es esencial para la comunicación entre computadoras y dispositivos físicos. Node-SerialPort ofrece un acceso directo y sencillo desde JavaScript, Johnny-Five facilita el control de hardware mediante una interfaz más intuitiva, y Boost.Asio brinda un rendimiento profesional desde C++. En conjunto, reflejan cómo los diferentes lenguajes y bibliotecas permiten desarrollar aplicaciones eficientes de transferencia de datos a través de puertos estándar según las necesidades del proyecto.

REFERENCIAS BIBLIOGRÁFICAS

About SerialPort | Node SerialPort. (2024, November 8). https://serialport.io/docs/

Boost.Asio. (n.d.). https://www.boost.org/doc/libs/latest/doc/html/boost_asio.html

Extio Technology. (2023, july 30). *Boost.Asio: A Powerful Networking Library for C++*. Medium. https://medium.com/@extio/boost-asio-a-powerful-networking-library-for-c-e86367c2568d

Johnny-Five: the JavaScript Robotics & IoT platform. (n.d.). Johnny-Five. https://johnny-five.io/

Lab: Serial Communication with Node.js – ITP Physical Computing. (n.d.). https://itp.nyu.edu/physcomp/labs/labs-serial-communication/lab-serial-communication-with-node-js/

Mehreentahir. (n.d.). *Socket Programming in C++ using boost.asio: TCP Server and Client - CodeProject*. https://www.codeproject.com/articles/Socket-Programming-in-Cplusplus-using-boost-asio-T#comments-section

Nodejs, Arduino, Chart.js y SerialPort | Fazt Web. (n.d.). faztweb.com. https://fazt.dev/contenido/node-arduino-serialport

Serialport. (n.d.). *GitHub - serialport/node-serialport: Access serial ports with JavaScript. Linux, OSX and Windows. Welcome your robotic JavaScript overlords. Better yet, program them!* GitHub. https://github.com/serialport/node-serialport

Tahir, M. (2018, October 28). *Socket programming in C++ using Boost.Asio - TCP server and client*. https://www.c-sharpcorner.com/article/socket-programming-in-cpp-using-boost-asio-tcp-server-and-client/