

IOT DEVELOPMENT PROJECT

INTERN SEBASTIAN ARRAZATE MARTINEZ



Programming technologies need to be compatible with microcontrollers.



Microcontroller app development is done at a low level, where you must work with the absence of file systems while handling registers and RAW memory. Add to that, most popular programming languages are not suitable for these goals. Java, Python, JS, and C# are languages for developing apps that are managed by an operating system (high-level development). Embedded development and simple microcontrollers don't have operating systems, and as a result, you have a very limited choice for your technology stack.

C/C++

It's flexible and is supported by the majority of compilers, including special use cases. Most importantly, it is one of the few languages that can work both with low-level development and abstractions like classes.

C++ is indeed suitable for a variety of IoT projects.



```
// 4) Inicio procedimiento recuperar lista de productos
void productos_speedlimit:recuperar(){
    ifstream lectura;
    lectura.open("productos.txt",ios-out|ios-in);
    if(lectura.is_open()){
        cout<<"\n Archivo abierto con exito..."<<endl;
        cout<<"\n Productos registrados en archivo productos.txt"<<endl;
        cout<<"-----"<<endl;
        lectura>>clave; // Inicio lectura adelantada
        int i; // Variable i de conteo
        i=1; // Valor inicial 1 para conteo
        while(!lectura.eof()){
            // Recupera datos de archivo
            lectura>>nombre_producto>>modelo_producto>>fabricante_producto>>categoria_producto>>codigo_producto>>existencias_producto>>precio_producto>>garantia_producto;
            // Imprime datos al programa
            cout<<" Datos de producto "<<i<<": "<<endl<<endl;
            cout<<"\t Clave unica de producto: "<<clave<<":"<<endl;
            cout<<"\t Nombre de producto: "<<nombre_producto<<":"<<endl;
            cout<<"\t Modelo de producto: "<<modelo_producto<<":"<<endl;
            cout<<"\t Fabricante de producto: "<<fabricante_producto<<":"<<endl;
            cout<<"\t Categoria de producto: "<<categoria_producto<<":"<<endl;
            cout<<"\t Codigo de producto: "<<codigo_producto<<":"<<endl;
            cout<<"\t Existencias en lote: "<<existencias_producto<<" unidades."<<endl;
            cout<<"\t Precio: $"<<precio_producto<<" USD."<<endl;
            cout<<"\t Garantia: "<<garantia_producto<<" meses."<<endl;
            lectura>>clave; // Fin lectura adelantada
            cout<<"-----"<<endl;
            i++; // Contador de productos
        }
    }
}
```

The IoTivity project

IoTivity is an open-source software framework enabling seamless device-to-device connectivity to address the emerging needs of the Internet of Things.



WHY IOT DEVELOPERS NEED OPEN-SOURCE SOLUTIONS

- **Time-saving and Customization**

Open source IoT infrastructure allows a developer to inherit a robust underlying communications stack and sample application code, and then focus on adding value at the application layer.

- **Expertise**

Industry organizations such as the Open Connectivity Foundation (OCF) create standards to ensure interoperability and offer open-source software to establish a proven basis for the IoT market.

- **Scalability and Reliability**

The Open Connectivity Foundation is dedicated to enabling IoT interoperability by developing industry standards that allow IoT devices to effectively communicate securely and efficiently. The strength of OCF is that its specifications are created from simple, consistent public resources. Any complex device can be created from this simple collection of open-source components, allowing developers to easily design and scale their devices.

Automotive IoT Examples (private sector)



- The major vehicle manufacturers have already started to market at least some connected car models. In addition to the benefits of monitoring, information on the technical condition of the vehicle and attention during accidents, car connectivity allows a wide variety of services to be offered for the driver and his companions. Thus, the new vehicles bring Wi-Fi connectivity and recommendation services for nearby restaurants or service stations, opening a whole range of value-added services within the cabin.



Automotive IoT Examples (business sector)

The new generation of connected vehicles brings companies a series of benefits:

- Monitoring of routes to generate efficiencies in transport times
- Vehicle tracking and traceability of the transported merchandise
- Generation of information and alerts on the need to technically verify the vehicle
- Fleet management
- Detection, alarm and rapid response to accidents, among others.

The new vehicles add to the IoT solutions that companies are already installing to improve the efficiency of their logistics and fleet monitoring tasks.