

Resumen

En la presente memoria se describe el diseño de un prototipo de sistema integral de domótica de índole académica. El propósito es aplicar los conocimientos adquiridos en el transcurso del posgrado. En esta **primer** etapa se implementan dos tipos de nodos y se agregarán a futuro nuevos elementos con otras funcionalidades.

En el desarrollo del trabajo se utilizaron conocimientos de diseño de páginas web, desarrollo de código backend para el servidor, implementación de bases de datos, desarrollo de sistemas embebidos, ciberseguridad y diseño electrónico para el hardware adicional.

Resumen

En la presente memoria se describe el diseño de un prototipo de sistema integral de domótica de índole académica. El propósito es aplicar los conocimientos adquiridos en el transcurso del posgrado. En esta **primera** etapa se implementan dos tipos de nodos y se agregarán a futuro nuevos elementos con otras funcionalidades.

En el desarrollo del trabajo se utilizaron conocimientos de diseño de páginas web, desarrollo de código backend para el servidor, implementación de bases de datos, desarrollo de sistemas embebidos, ciberseguridad y diseño electrónico para el hardware adicional.

v

Índice general

Resumen	i
1. Introducción general	1
1.1. Hogares inteligentes	1
1.1.1. Aplicaciones comunes	1
1.2. Motivación	2
1.3. Estado del arte	2
1.4. Objetivos y alcance	3
2. Introducción específica	5
2.1. Protocolos de comunicación	5
2.1.1. Modelo OSI	5
2.1.2. Protocolo HTTP	6
2.1.3. Protocolo MQTT	7
Calidad de servicio	8
Protocolos SSL y TLS	8
2.2. Componentes de hardware	8
2.2.1. Raspberry Pi	8
Especificaciones técnicas de la Raspberry Pi 400	9
2.2.2. Sistema en chip	10
Familia ESP32	10
2.2.3. Especificaciones de los sensores	11
2.2.4. Especificaciones de los actuadores	12
2.3. Herramientas de software	13
2.3.1. Visual studio code	13
2.3.2. Marco de desarrollo ESP	13
2.3.3. Lenguajes JavaScript y TypeScript	14
2.3.4. Angular y Ionic	14
2.3.5. Bases de datos relacionales MySQL y MariaDB	14
2.3.6. Contenedores con Docker	15
3. Diseño e implementación	17
3.1. Arquitectura del sistema	17
3.2. Modelo de datos	18
3.2.1. Tabla Dispositivos	19
3.2.2. Tabla Usuarios	20
3.2.3. Tabla Mediciones	20
3.3. Desarrollo del frontend	21
3.3.1. Rutas y páginas destacadas	23
Pantalla de login	23
Pantalla principal home	24
Pantalla de creación de dispositivo nuevo	24

v

Índice general

Resumen	i
1. Introducción general	1
1.1. Hogares inteligentes	1
1.1.1. Aplicaciones comunes	1
1.2. Motivación	2
1.3. Estado del arte	2
1.4. Objetivos y alcance	3
2. Introducción específica	5
2.1. Protocolos de comunicación	5
2.1.1. Modelo OSI	5
2.1.2. Protocolo HTTP	6
2.1.3. Protocolo MQTT	7
Calidad de servicio	8
Protocolos SSL y TLS	8
2.2. Componentes de hardware	8
2.2.1. Raspberry Pi	8
Especificaciones técnicas de la Raspberry Pi 400	9
2.2.2. Sistema en chip	10
Familia ESP32 [13]	10
2.2.3. Especificaciones de los sensores	11
2.2.4. Especificaciones de los actuadores	12
2.3. Herramientas de software	13
2.3.1. Visual studio code [21]	13
2.3.2. Marco de desarrollo ESP	13
2.3.3. Lenguajes JavaScript y TypeScript	13
2.3.4. Angular y Ionic	14
2.3.5. Bases de datos relacionales MySQL y MariaDB	14
2.3.6. Contenedores con Docker	15
3. Diseño e implementación	17
3.1. Arquitectura del sistema	17
3.2. Modelo de datos	18
3.2.1. Tabla Dispositivos	19
3.2.2. Tabla Usuarios	20
3.2.3. Tabla Mediciones	20
3.3. Desarrollo del frontend	21
3.3.1. Rutas y páginas destacadas	23
Pantalla de login	23
Pantalla principal home	24
Pantalla de creación de dispositivo nuevo	24

En la figura 1.2 se observa un kit de Matter a la izquierda y la pantalla de la aplicación de Home Assistant a la derecha.



FIGURA 1.2. Matter y Home Assistant. ²

A nivel nacional, existen empresas que se dedican al desarrollo de sistemas de domótica de forma privada:

- Domotic [5].
- Commax [6].
- Reactor [7].

Todos estos sistemas utilizan distintos tipos de conexión como Zigbee o Wi-Fi, mensajes entre dispositivos como HTTP, MQTT y otros, y pueden alojar sus plataformas en un servidor local o en la nube.

Sin embargo, pocos de estos sistemas tienen la posibilidad de funcionar con mandos desde el lugar o sin conexión con el servidor, es decir, que sean independientes y controlables de forma local. Este proyecto tiene tal posibilidad, ya que la página web es una forma más de cambiar los parámetros de los nodos y está diseñado para decidir si cada terminal posee comandos o no.

En la tabla 1.1 pueden verse las principales características de las empresas nacionales con soluciones similares.

TABLA 1.1. Comparativa entre las distintas opciones

Empresa	Productos	Compatibilidad
Domotic	Catálogo amplio	No especifica
Commax	Catálogo amplio	Smart things y Apple Home Kit
Reactor	2 alternativas	Google Assistant e IFTTT

1.4. Objetivos y alcance

El objetivo de este trabajo es aplicar todos los conocimientos adquiridos a lo largo del posgrado en un área de interés particular, que es la domótica. En especial, el principal propósito es desarrollar un prototipo que sirva como punto de partida para un sistema más amplio y avanzado, con la particularidad de que en un futuro cercano se pueda llegar a analizar la opción de integrarlo con sistemas como Matter o Home Assistant.

²Imágenes tomadas de: <https://csa-iot.org/> y <https://www.home-assistant.io/>

En la figura 1.2 se observa un kit de Matter a la izquierda y la pantalla de la aplicación de Home Assistant a la derecha.



FIGURA 1.2. Matter y Home Assistant. ²

A nivel nacional, existen empresas que se dedican al desarrollo de sistemas de domótica de forma privada:

- Domotic [5].
- Commax [6].
- Reactor [7].

Todos estos sistemas utilizan distintos tipos de conexión como Zigbee o Wi-Fi, mensajes entre dispositivos como HTTP, MQTT y otros, y pueden alojar sus plataformas en un servidor local o en la nube.

Sin embargo, pocos de estos sistemas tienen la posibilidad de funcionar con mandos desde el lugar o sin conexión con el servidor, es decir, que sean independientes y controlables de forma local. Este proyecto tiene tal posibilidad, ya que la página web es una forma más de cambiar los parámetros de los nodos y está diseñado para decidir si cada terminal posee comandos o no.

En la tabla 1.1 pueden verse las principales características de las empresas nacionales con soluciones similares.

TABLA 1.1. Comparativa entre las distintas opciones

Empresa	Productos	Compatibilidad
Domotic	Catálogo amplio	No especifica
Commax	Catálogo amplio	Smart things y Apple Home Kit
Reactor	2 alternativas	Google Assistant e IFTTT

1.4. Objetivos y alcance

El objetivo de este trabajo es aplicar todos los conocimientos adquiridos a lo largo del posgrado en un área de interés particular, que es la domótica. En especial, el principal propósito es desarrollar un prototipo que sirva como punto de partida para un sistema más amplio y avanzado, con la particularidad de que en un futuro cercano se pueda llegar a analizar la opción de integrarlo con sistemas como Matter o Home Assistant.

²Imágenes tomadas de: <https://csa-iot.org/> y <https://www.home-assistant.io/>

TABLA 2.1. Especificaciones de Raspberry Pi 400.

Procesador	Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8 GHz
Memoria	4GB LPDDR4-3200
Conectividad	2.4 GHz and 5.0 GHz 802.11b/g/n/ac wireless LAN Bluetooth 5.0, BLE Gigabit Ethernet
Alimentación	5 V DC via USB-C

2.2.2. Sistema en chip

Un sistema en chip o SoC (del inglés *System on a Chip*) es aquel dispositivo que posee integrados todos o gran parte de los módulos que componen un sistema informático o electrónico en un único circuito integrado o chip. El diseño de estos sistemas puede estar basado en circuitos de señal digital, señal analógica y a menudo módulos o sistemas de radiofrecuencia. Un ámbito común de aplicación de la tecnología SoC son los sistemas embebidos.

Un SoC estándar está constituido por [12]:

- Un microcontrolador con el núcleo de la CPU. Algunos son contruidos con microprocesadores dotados de varios núcleos.
- Módulos de memoria ROM (memoria de sólo lectura), RAM (memoria de acceso aleatorio), EEPROM (memoria de sólo lectura programable y borrrable electrónicamente) y Flash (memorias de acceso muy rápido).
- Generadores de frecuencia fija.
- Componentes periféricos como contadores, temporizadores y relojes en tiempo real o RTC (en inglés *Real Time Clock*).
- Controladores de comunicación con interfaces externas normalmente estándar como USB, Ethernet, UART, o SPI.
- Controladores de interfaces analógicas, incluyendo conversores analógico a digital (ADC) y digital a analógico (DAC).
- Reguladores de **voltaje** y circuitos de gestión eficaz de la energía.

Familia ESP32

[13]

ESP32 es la denominación de una familia de chips SoC de bajo costo y consumo de energía, con tecnología Wi-Fi y Bluetooth de modo dual integrada. Emplea un microprocesador Tensilica Xtensa LX6 en sus variantes de simple y doble núcleo e incluye interruptores de antena, balun de radiofrecuencia, amplificador de potencia, amplificador receptor de bajo ruido, filtros y módulos de administración de energía. El ESP32 fue creado y desarrollado por Espressif Systems.

En la figura 2.4 se pueden ver algunos de los módulos de desarrollo que contienen ESP32.

TABLA 2.1. Especificaciones de Raspberry Pi 400.

Procesador	Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8 GHz
Memoria	4GB LPDDR4-3200
Conectividad	2.4 GHz and 5.0 GHz 802.11b/g/n/ac wireless LAN Bluetooth 5.0, BLE Gigabit Ethernet
Alimentación	5 V DC via USB-C

2.2.2. Sistema en chip

Un sistema en chip o SoC (del inglés *System on a Chip*) es aquel dispositivo que posee integrados todos o gran parte de los módulos que componen un sistema informático o electrónico en un único circuito integrado o chip. El diseño de estos sistemas puede estar basado en circuitos de señal digital, señal analógica y a menudo módulos o sistemas de radiofrecuencia. Un ámbito común de aplicación de la tecnología SoC son los sistemas embebidos.

Un SoC estándar está constituido por [12]:

- Un microcontrolador con el núcleo de la CPU. Algunos son contruidos con microprocesadores dotados de varios núcleos.
- Módulos de memoria ROM (memoria de sólo lectura), RAM (memoria de acceso aleatorio), EEPROM (memoria de sólo lectura programable y borrrable electrónicamente) y Flash (memorias de acceso muy rápido).
- Generadores de frecuencia fija.
- Componentes periféricos como contadores, temporizadores y relojes en tiempo real o RTC (en inglés *Real Time Clock*).
- Controladores de comunicación con interfaces externas normalmente estándar como USB, Ethernet, UART, o SPI.
- Controladores de interfaces analógicas, incluyendo conversores analógico a digital (ADC) y digital a analógico (DAC).
- Reguladores de **tensión** y circuitos de gestión eficaz de la energía.

Familia ESP32 [13]

ESP32 es la denominación de una familia de chips SoC de bajo costo y consumo de energía, con tecnología Wi-Fi y Bluetooth de modo dual integrada. Emplea un microprocesador Tensilica Xtensa LX6 en sus variantes de simple y doble núcleo e incluye interruptores de antena, balun de radiofrecuencia, amplificador de potencia, amplificador receptor de bajo ruido, filtros y módulos de administración de energía. El ESP32 fue creado y desarrollado por Espressif Systems.

En la figura 2.4 se pueden ver algunos de los módulos de desarrollo que contienen ESP32.

⁴Imagen tomada de: <https://www.electrodaddy.com/esp32/>



FIGURA 2.4. Módulos de la familia ESP32. ⁴

En la tabla 2.2 pueden verse las características de hardware más importantes del microcontrolador ESP32, utilizado para el desarrollo de los dispositivos.

TABLA 2.2. Especificaciones técnicas del módulo ESP32 [14].

Característica	ESP32
Núcleo	Xtensa® dual-core 32-bit LX6 @240 MHz
Flash	0 MB, 2 MB o 4 MB (dependiendo la versión)
Protocolo Wi-Fi	802.11 b/g/n, 2.4 GHz

2.2.3. Especificaciones de los sensores

A continuación, se listan sensores utilizados con sus principales características:

- DHT22: sensor de temperatura y humedad relativa ambiente [15].
 - Rango de temperatura: -40 a 80 grados Celsius.
 - Resolución: 0,1 grado Celsius.
 - Comunicación: serie, bus de 1 hilo, 40 bits por trama.
- KY-040: encoder rotativo con interruptor. 20 pulsos por vuelta [16].

En la figura 2.5 se muestra la imagen del encoder a la izquierda y del sensor de temperatura a la derecha con sus correspondientes esquemas de conexión.

⁴Imagen tomada de: <https://www.electrodaddy.com/esp32/>



FIGURA 2.4. Módulos de la familia ESP32. ⁴

En la tabla 2.2 pueden verse las características de hardware más importantes del microcontrolador ESP32, utilizado para el desarrollo de los dispositivos.

TABLA 2.2. Especificaciones técnicas del módulo ESP32 [14].

Característica	ESP32
Núcleo	Xtensa® dual-core 32-bit LX6 @240 MHz
Flash	0 MB, 2 MB o 4 MB (dependiendo la versión)
Protocolo Wi-Fi	802.11 b/g/n, 2.4 GHz

2.2.3. Especificaciones de los sensores

A continuación, se listan sensores utilizados con sus principales características:

- DHT22: sensor de temperatura y humedad relativa ambiente [15].
 - Rango de temperatura: -40 a 80 grados Celsius.
 - Resolución: 0,1 grado Celsius.
 - Comunicación: serie, bus de 1 hilo, 40 bits por trama.
- KY-040: encoder rotativo con interruptor. 20 pulsos por vuelta [16].

En la figura 2.5 se muestra la imagen del encoder a la izquierda y del sensor de temperatura a la derecha con sus correspondientes esquemas de conexión.

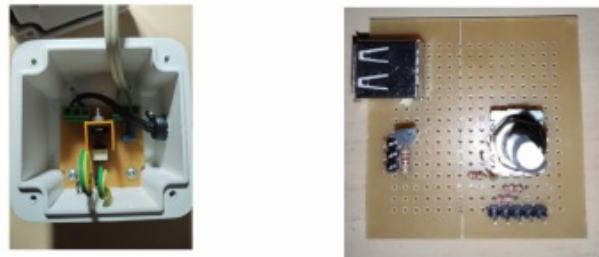


FIGURA 2.6. Actuadores utilizados.

2.3. Herramientas de software

2.3.1. Visual studio code

[21]

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para múltiples sistemas operativos. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. Es personalizable y tiene la posibilidad de instalar extensiones para agregar lenguajes, depuradores y herramientas para el desarrollo de código. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

Algunos de los lenguajes de programación que soporta son: C, C++, Dockerfile, Git-commit, HTML, JSON, Java, JavaScript, PHP, Python, Ruby, Rust, SQL, Shell script, TypeScript y Visual Basic entre otros.

2.3.2. Marco de desarrollo ESP

El marco de desarrollo de ESP, o ESP-IDF (en inglés *ESP IoT Development Framework*) es un entorno completo de programación para desarrollar sistemas embebidos para dispositivos ESP. Es desarrollado por Espressif y se puede descargar como una extensión de Visual studio code. El lenguaje de programación es C e incluye herramientas para cargar el código desarrollado al chip y depurar el programa en tiempo real.

El lenguaje C es un lenguaje de programación de propósito general de tipos de datos estáticos, débilmente tipado. Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez, dispone de construcciones del lenguaje que permiten un control a bajo nivel. Uno de los objetivos de diseño del lenguaje C es que solo sean necesarias unas pocas instrucciones en lenguaje máquina para traducir cada elemento del lenguaje, sin que haga falta un soporte intenso en tiempo de ejecución [22].

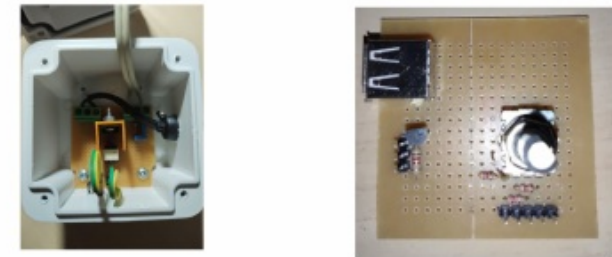


FIGURA 2.6. Actuadores utilizados.

2.3. Herramientas de software

2.3.1. Visual studio code [21]

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para múltiples sistemas operativos. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. Es personalizable y tiene la posibilidad de instalar extensiones para agregar lenguajes, depuradores y herramientas para el desarrollo de código. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

Algunos de los lenguajes de programación que soporta son: C, C++, Dockerfile, Git-commit, HTML, JSON, Java, JavaScript, PHP, Python, Ruby, Rust, SQL, Shell script, TypeScript y Visual Basic entre otros.

2.3.2. Marco de desarrollo ESP

El marco de desarrollo de ESP, o ESP-IDF (en inglés *ESP IoT Development Framework*) es un entorno completo de programación para desarrollar sistemas embebidos para dispositivos ESP. Es desarrollado por Espressif y se puede descargar como una extensión de Visual studio code. El lenguaje de programación es C e incluye herramientas para cargar el código desarrollado al chip y depurar el programa en tiempo real.

El lenguaje C es un lenguaje de programación de propósito general de tipos de datos estáticos, débilmente tipado. Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez, dispone de construcciones del lenguaje que permiten un control a bajo nivel. Uno de los objetivos de diseño del lenguaje C es que solo sean necesarias unas pocas instrucciones en lenguaje máquina para traducir cada elemento del lenguaje, sin que haga falta un soporte intenso en tiempo de ejecución [22].

2.3.3. Lenguajes JavaScript y TypeScript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz

2.3.3. Lenguajes JavaScript y TypeScript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque también se utiliza del lado del servidor. Todos los navegadores modernos interpretan el código en este lenguaje integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del modelo de objeto de documento, o DOM (en inglés *Document Object Model*). Es el único lenguaje de programación que entienden de forma nativa los navegadores [23].

TypeScript es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, su antecesor, que esencialmente añade tipos estáticos y objetos basados en clases. Es usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor, o extensiones para programas. Extiende la sintaxis de su antecesor, por tanto cualquier código en el lenguaje original existente debería funcionar sin problemas. Está pensado para grandes proyectos, los cuales a través de un compilador de TypeScript se traducen a código JavaScript [24].

2.3.4. Angular y Ionic

Angular es un marco de desarrollo (framework en inglés) de ingeniería de software de código abierto mantenido por Google, que sirve para desarrollar aplicaciones web de estilo aplicación de una sola página (en inglés *single page application* o SPA) y aplicación web progresiva (en inglés *Progressive Web App* o PWA). Sirve tanto para versiones móviles como de escritorio. Ofrece soluciones robustas, escalables y optimizadas para lograr un estilo de codificación homogéneo y de gran modularidad. Su desarrollo se realiza por medio de TypeScript o JavaScript. En este último se ofrecen diversas herramientas adicionales al lenguaje como tipado estático o decoradores [25].

El marco de desarrollo Ionic es un kit de desarrollo de software (en inglés *Software Development Kit* o SDK) de frontend de código abierto para desarrollar aplicaciones híbridas basado en tecnologías web (HTML, CSS y JS). Es decir, un framework que nos permite desarrollar aplicaciones multiplataforma desde una única base de código. Posee la capacidad de integrarse con otros marcos populares como Angular, React y Vue. Su principal característica es que permite desarrollar y desplegar aplicaciones híbridas, que funcionan en múltiples plataformas, como iOS nativo, Android, escritorio y la web (como una aplicación web progresiva) con una única base de código [26].

2.3.5. Bases de datos relacionales MySQL y MariaDB

Una base de datos relacional es un conjunto de una o más tablas estructuradas en registros (filas) y campos (columnas), que se vinculan entre sí por un campo en común. MySQL es un sistema de administración de bases de datos relacionales de código abierto desarrollado por Oracle. Se considera como la base de datos de código abierto más utilizada en el mundo. Posee cuatro funciones básicas que se conocen con la sigla CRUD: *create* (crear), *read* (leer), *update* (actualizar) y *delete* (borrar). Estas funciones son las que se aplican a los nuevos registros que se

de usuario y páginas web dinámicas, aunque también se utiliza del lado del servidor. Todos los navegadores modernos interpretan el código en este lenguaje integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del modelo de objeto de documento, o DOM (en inglés *Document Object Model*). Es el único lenguaje de programación que entienden de forma nativa los navegadores [23].

TypeScript es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, su antecesor, que esencialmente añade tipos estáticos y objetos basados en clases. Es usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor, o extensiones para programas. Extiende la sintaxis de su antecesor, por tanto cualquier código en el lenguaje original existente debería funcionar sin problemas. Está pensado para grandes proyectos, los cuales a través de un compilador de TypeScript se traducen a código JavaScript [24].

2.3.4. Angular y Ionic

Angular es un marco de desarrollo (framework en inglés) de ingeniería de software de código abierto mantenido por Google, que sirve para desarrollar aplicaciones web de estilo aplicación de una sola página (en inglés *single page application* o SPA) y aplicación web progresiva (en inglés *Progressive Web App* o PWA). Sirve tanto para versiones móviles como de escritorio. Ofrece soluciones robustas, escalables y optimizadas para lograr un estilo de codificación homogéneo y de gran modularidad. Su desarrollo se realiza por medio de TypeScript o JavaScript. En este último se ofrecen diversas herramientas adicionales al lenguaje como tipado estático o decoradores [25].

El marco de desarrollo Ionic es un kit de desarrollo de software (en inglés *Software Development Kit* o SDK) de frontend de código abierto para desarrollar aplicaciones híbridas basado en tecnologías web (HTML, CSS y JS). Es decir, un framework que nos permite desarrollar aplicaciones multiplataforma desde una única base de código. Posee la capacidad de integrarse con otros marcos populares como Angular, React y Vue. Su principal característica es que permite desarrollar y desplegar aplicaciones híbridas, que funcionan en múltiples plataformas, como iOS nativo, Android, escritorio y la web (como una aplicación web progresiva) con una única base de código [26].

2.3.5. Bases de datos relacionales MySQL y MariaDB

Una base de datos relacional es un conjunto de una o más tablas estructuradas en registros (filas) y campos (columnas), que se vinculan entre sí por un campo en común. MySQL es un sistema de administración de bases de datos relacionales de código abierto desarrollado por Oracle. Se considera como la base de datos de código abierto más utilizada en el mundo. Posee cuatro funciones básicas que se conocen con la sigla CRUD: *create* (crear), *read* (leer), *update* (actualizar) y *delete* (borrar). Estas funciones son las que se aplican a los nuevos registros que se quieran crear y a los ya existentes que se deseen leer, actualizar y borrar. Este tipo de bases de datos posee una arquitectura cliente-servidor, siendo el cliente el que hace las solicitudes de datos y el servidor el que posee dichos datos y responde a dicha solicitud.

quieran crear y a los ya existentes que se deseen leer, actualizar y borrar. Este tipo de bases de datos posee una arquitectura cliente-servidor, siendo el cliente el que hace las solicitudes de datos y el servidor el que posee dichos datos y responde a dicha solicitud.

MariaDB es una versión modificada de MySQL. Fue creada por el equipo de desarrollo original de MySQL debido a problemas de licencia y distribución después de que Oracle Corporation adquiriera MySQL. MariaDB adopta los archivos de definición de tablas y datos de MySQL y también usa protocolos de cliente, API de cliente, puertos y sockets idénticos. Con ello se pretende que los usuarios de MySQL puedan cambiar a MariaDB sin problemas [27].

2.3.6. Contenedores con Docker

Los contenedores son una forma de virtualización del sistema operativo. Un solo contenedor se puede usar para ejecutar cualquier aplicación, desde un microservicio o un proceso de software a una aplicación de mayor tamaño. Dentro de un contenedor se encuentran todos los ejecutables, el código binario, las bibliotecas y los archivos de configuración necesarios. Sin embargo, en comparación con los métodos de virtualización de máquinas o servidores, los contenedores no contienen imágenes del sistema operativo. Esto los hace más ligeros y portátiles, con una sobrecarga significativamente menor. En implementaciones de aplicaciones de mayor tamaño, se pueden poner en marcha varios contenedores como uno o varios clústeres de contenedores. Estos clústeres se pueden gestionar mediante un orquestador de contenedores, como Kubernetes o Docker Compose [28].

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Usar Docker para crear y gestionar contenedores puede simplificar la creación de sistemas altamente distribuidos, permitiendo que múltiples aplicaciones, las tareas de los trabajadores y otros procesos funcionen de forma autónoma en una única máquina física o en varias máquinas virtuales [29]. En la figura 2.7 se puede ver de forma gráfica el funcionamiento de un contenedor Docker.

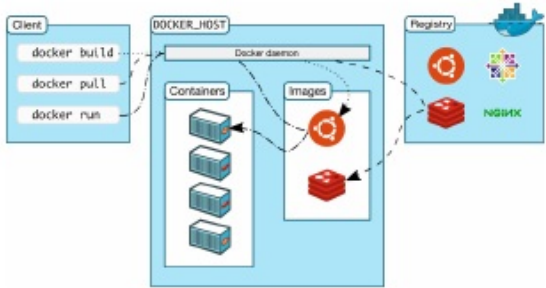


FIGURA 2.7. Contenedor Docker. ⁵

⁵Imagen tomada de: <https://algodaily.com/lessons/what-is-a-container-a-docker-tutorial>

MariaDB es una versión modificada de MySQL. Fue creada por el equipo de desarrollo original de MySQL debido a problemas de licencia y distribución después de que Oracle Corporation adquiriera MySQL. MariaDB adopta los archivos de definición de tablas y datos de MySQL y también usa protocolos de cliente, API de cliente, puertos y sockets idénticos. Con ello se pretende que los usuarios de MySQL puedan cambiar a MariaDB sin problemas [27].

2.3.6. Contenedores con Docker

Los contenedores son una forma de virtualización del sistema operativo. Un solo contenedor se puede usar para ejecutar cualquier aplicación, desde un microservicio o un proceso de software a una aplicación de mayor tamaño. Dentro de un contenedor se encuentran todos los ejecutables, el código binario, las bibliotecas y los archivos de configuración necesarios. Sin embargo, en comparación con los métodos de virtualización de máquinas o servidores, los contenedores no contienen imágenes del sistema operativo. Esto los hace más ligeros y portátiles, con una sobrecarga significativamente menor. En implementaciones de aplicaciones de mayor tamaño, se pueden poner en marcha varios contenedores como uno o varios clústeres de contenedores. Estos clústeres se pueden gestionar mediante un orquestador de contenedores, como Kubernetes o Docker Compose [28].

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Usar Docker para crear y gestionar contenedores puede simplificar la creación de sistemas altamente distribuidos, permitiendo que múltiples aplicaciones, las tareas de los trabajadores y otros procesos funcionen de forma autónoma en una única máquina física o en varias máquinas virtuales [29]. En la figura 2.7 se puede ver de forma gráfica el funcionamiento de un contenedor Docker.

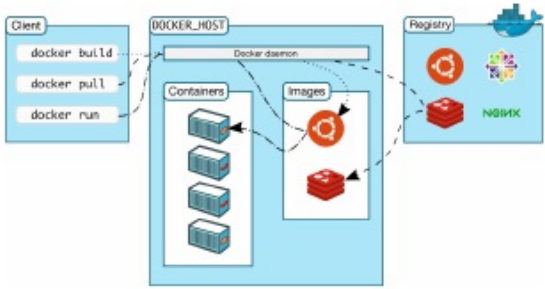


FIGURA 2.7. Contenedor Docker. ⁵

Docker Compose es una herramienta para definir y ejecutar aplicaciones Docker de varios contenedores. Se utiliza un archivo YAML para configurar los servicios de su aplicación, y luego, con un solo comando, se crean e inician todos los servicios desde su configuración. Tiene comandos para gestionar todo el ciclo de vida de una aplicación con los que se pueden [30]:

⁵Imagen tomada de: <https://algodaily.com/lessons/what-is-a-container-a-docker-tutorial>

Docker Compose es una herramienta para definir y ejecutar aplicaciones Docker de varios contenedores. Se utiliza un archivo YAML para configurar los servicios de su aplicación, y luego, con un solo comando, se crean e inician todos los servicios desde su configuración. Tiene comandos para gestionar todo el ciclo de vida de una aplicación con los que se pueden [30]:

- Iniciar, detener y reconstruir servicios.
- Ver el estado de los servicios en ejecución.
- Transmitir la salida del registro de los servicios en ejecución.
- Ejecutar un comando único en un servicio.

- Iniciar, detener y reconstruir servicios.
- Ver el estado de los servicios en ejecución.
- Transmitir la salida del registro de los servicios en ejecución.
- Ejecutar un comando único en un servicio.