[[1]](#footnote-0)

Estado del Arte de la Web: Evolución, Tecnologías Actuales y Futuras Perspectivas(marzo de 2004)

Vasquez Benitez Jose Guillermo

**Resumen - Este documento analiza la evolución de la web desde sus inicios hasta las tendencias más recientes, abarcando los diferentes paradigmas que han transformado la experiencia de los usuarios. Se explora la transición desde la Web 1.0, caracterizada por páginas estáticas y consumo de información, a la Web 2.0, que introdujo la colaboración y la interactividad a través de plataformas sociales. Además, se detalla la Web 3.0, con su enfoque en la descentralización, blockchain e inteligencia artificial para mejorar la personalización y seguridad en la navegación. Finalmente, se discuten las tecnologías que han propiciado estos cambios y los retos que presenta el futuro de la web, incluyendo aspectos regulatorios y el avance hacia una conectividad más inteligente y pervasiva.**

**Índice de Términos - Web, HTML, Web 3.0, Blockchain, Tecnologías Web**

# introducción

Este documento analiza la evolución de la web desde sus inicios hasta las tendencias más recientes, abarcando los diferentes paradigmas que han transformado la experiencia de los usuarios. La web ha pasado por diversas etapas, cada una caracterizada por avances tecnológicos significativos que han cambiado la forma en que interactuamos con la información y entre nosotros. Se explora la transición desde la Web 1.0, una fase inicial centrada en la distribución de información a través de páginas estáticas donde los usuarios eran meros consumidores de contenido, a la Web 2.0, que introdujo la colaboración y la interactividad mediante el uso de plataformas sociales, blogs, y otras herramientas que permitieron a los usuarios generar y compartir contenido de manera dinámica. Esta etapa marcó el comienzo de una web más participativa y centrada en la comunidad.

Además, se detalla la Web 3.0, la cual se caracteriza por el uso de tecnologías emergentes como la descentralización, blockchain e inteligencia artificial. Esta evolución tiene como objetivo principal mejorar la personalización de los servicios web, así como garantizar una mayor seguridad y privacidad en la navegación. La Web 3.0 también busca crear una experiencia más conectada e inteligente, permitiendo que las máquinas entiendan mejor el contexto de la información y ofrezcan respuestas más precisas y útiles a las necesidades de los usuarios.

Finalmente, se discuten las tecnologías que han propiciado estos cambios y los retos que presenta el futuro de la web, incluyendo aspectos regulatorios, la gobernanza de los datos, y el avance hacia una conectividad más inteligente y pervasiva, que incluye la integración con el Internet de las Cosas (IoT) y el desarrollo de la llamada Web 4.0, en la que se espera una interacción aún más fluida entre humanos y dispositivos inteligentes.

# Evolución de la web

## Web 1.0

La primera versión de la web, conocida como Web 1.0, se basaba en páginas HTML estáticas. Los usuarios eran consumidores de información sin la posibilidad de interactuar de forma dinámica con los contenidos. En este contexto, la web se concebía principalmente como una biblioteca de documentos a la que los usuarios accedían para buscar información específica. La comunicación era unidireccional, ya que no existía la infraestructura ni las herramientas para que los usuarios contribuyeran al contenido o interactuaran entre sí. En esta etapa, aún no habían aparecido sistemas como JavaScript para añadir interactividad a las páginas web.

La Web 1.0 surgió en un contexto en el cual el Internet era principalmente un proyecto militar conocido como ARPANET, desarrollado por la Agencia de Proyectos de Investigación Avanzados de Defensa (DARPA) de Estados Unidos. ARPANET fue un proyecto clave en la creación de lo que hoy conocemos como Internet, y sirvió como base para la futura interconexión de redes a nivel global. Este sistema, originalmente diseñado para facilitar la comunicación entre instituciones de investigación y el Departamento de Defensa, sentó las bases de la tecnología de transmisión de datos que eventualmente se convertiría en la World Wide Web.

El desarrollo de la Web 1.0 también está estrechamente vinculado al trabajo de Tim Berners-Lee, conocido como el padre de la web. En 1989, Berners-Lee, un científico del CERN (Organización Europea para la Investigación Nuclear), propuso un sistema para compartir información entre investigadores mediante el uso de hipertexto. Este sistema se convirtió en la base de la World Wide Web, que permitiría acceder a documentos y navegar entre ellos utilizando enlaces. El primer navegador web y servidor web fueron desarrollados por él, lo cual permitió la implementación de esta visión, que si bien era estática en su naturaleza, revolucionó la manera en que el mundo accedía a la información.

En la Web 1.0, el concepto de conectividad se limitaba a la disponibilidad de la información en sitios web, y aún no se vislumbraban tecnologías que permitieran la conexión de dispositivos inteligentes. La idea de integración con el Internet de las Cosas (IoT) estaba lejos de materializarse, ya que en ese momento no existían mecanismos para que los dispositivos pudieran comunicarse entre ellos o compartir datos de forma autónoma. La Web 1.0 sentó las bases para lo que más tarde se convertiría en una red de comunicación más interactiva, y eventualmente en una infraestructura conectada con dispositivos inteligentes que vemos en las etapas más avanzadas de la web.

## Web 2.0

La Web 2.0 marcó un punto de inflexión significativo en la evolución de la web, destacando por su enfoque en la colaboración y la interacción entre usuarios. A diferencia de la Web 1.0, la Web 2.0 no se limitaba a ser una fuente unidireccional de información; los usuarios comenzaron a tener un papel activo en la creación y distribución de contenidos. Este cambio fue posible gracias a una serie de nuevas tecnologías emergentes que aparecieron a principios de la década de 2000, como los lenguajes de scripting del lado del cliente y del servidor, incluidos JavaScript, AJAX (Asynchronous JavaScript and XML), y tecnologías como PHP y Ruby on Rails, que facilitaron la creación de aplicaciones web dinámicas.

El término "Web 2.0" fue popularizado por Tim O'Reilly en 2004 para describir esta nueva generación de servicios en línea que fomentaban la colaboración, la participación activa y la creación de contenido por parte de los usuarios. Plataformas como Wikipedia, YouTube, Facebook y blogs como Blogger y WordPress, surgieron como ejemplos icónicos de la Web 2.0, permitiendo que las personas no solo consumieran información, sino también la crearan y compartieran. Estas plataformas facilitaron la interacción social y la colaboración, creando un entorno de intercambio masivo de conocimiento.

Durante esta etapa, comenzaron a surgir los primeros indicios de integración con dispositivos más allá de los computadores personales, sentando las bases para el desarrollo futuro del IoT. Aunque el Internet de las Cosas aún no estaba consolidado, la evolución de las tecnologías web y la interconexión creciente entre dispositivos establecieron un entorno propicio para su aparición. Servicios como Google Maps introdujeron la integración de datos en tiempo real, un concepto que sería fundamental en el desarrollo de aplicaciones IoT en años posteriores.

La Web 2.0 también promovió el uso de APIs (Application Programming Interfaces) abiertas, lo cual permitió a los desarrolladores integrar servicios y funcionalidades de diferentes plataformas, fomentando la interoperabilidad y la creación de ecosistemas conectados. Estas tecnologías hicieron que la web fuera más interactiva, rápida y capaz de ofrecer experiencias personalizadas a los usuarios, marcando una gran diferencia respecto a la Web 1.0, que era puramente informativa y estática.

La interacción social y el aumento de las comunidades en línea fueron un sello distintivo de la Web 2.0. Los usuarios se convirtieron en parte activa del proceso de creación de contenido, transformándose en prosumidores, un término que combina las palabras "productor" y "consumidor". La estructura de la Web 2.0 permitió una mejor participación, comunicación y colaboración entre personas de diferentes partes del mundo, abriendo paso al desarrollo de redes sociales, plataformas de video y wikis, que facilitaron la democratización de la información y la conectividad global.

## Web 3.0

La Web 3.0 se caracteriza por la descentralización de los datos y la utilización de tecnologías emergentes como blockchain e inteligencia artificial. Estas tecnologías tienen como objetivo principal hacer que la web sea más inteligente y personalizada, al mismo tiempo que aseguran una mayor privacidad y seguridad para los usuarios. En lugar de depender de grandes plataformas centralizadas, la Web 3.0 busca empoderar a los usuarios, dándoles el control de sus propios datos y permitiendo una interacción directa sin intermediarios.

Una de las características más importantes de la Web 3.0 es la semántica, que busca que las máquinas entiendan mejor el contexto y el significado de la información disponible, facilitando una navegación más efectiva y precisa. Los avances en inteligencia artificial permiten que la Web 3.0 proporcione respuestas más útiles y personalizadas, basadas en un análisis profundo del comportamiento y las preferencias de los usuarios.

La descentralización es otro pilar clave de la Web 3.0. A través del uso de blockchain, se busca reducir la dependencia de intermediarios, creando un entorno más seguro y transparente para la transferencia de información y transacciones. Esta tecnología también habilita la creación de aplicaciones descentralizadas (dApps), que permiten a los usuarios interactuar de manera segura y sin censura en una red distribuida.

Finalmente, la Web 3.0 también está muy relacionada con el crecimiento del Internet de las Cosas (IoT). En esta etapa, la integración de dispositivos inteligentes conectados permite que la web se convierta en un entorno más conectado, donde la información generada por diferentes dispositivos puede ser procesada y utilizada para mejorar la calidad de vida de los usuarios.

## Web 4.0.

Aunque la Web 4.0 aún no existe oficialmente, se ha comenzado a teorizar sobre sus posibles características y capacidades. La Web 4.0 es vista como una evolución natural de las etapas anteriores, en la que se espera una conexión aún más profunda e integrada entre el mundo físico y digital. Este concepto implica un entorno completamente interconectado, donde los dispositivos inteligentes, los datos y las personas interactúan de manera fluida y casi sin fricciones, gracias al uso intensivo de tecnologías avanzadas como la inteligencia artificial, el aprendizaje automático, y el IoT.

Se espera que la Web 4.0 facilite una conectividad ubicua, donde las fronteras entre el mundo real y el virtual se difuminen cada vez más. Las experiencias inmersivas, como la realidad aumentada y la realidad virtual, jugarán un papel importante en esta fase, proporcionando formas más ricas y significativas de interactuar con la información y con otros usuarios. La Web 4.0 podría describirse como una web "simbiótica", en la que la relación entre humanos y máquinas se vuelve más cooperativa e integrada.

Además, se prevé que la Web 4.0 sea más proactiva, anticipando las necesidades de los usuarios a través del análisis predictivo y la inteligencia contextual. Los sistemas inteligentes podrán analizar datos en tiempo real provenientes de una variedad de fuentes, incluyendo sensores IoT, dispositivos móviles y redes sociales, para ofrecer servicios adaptativos que respondan a las preferencias individuales de cada usuario.

Aunque la Web 4.0 todavía es una visión futura, los avances en conectividad, procesamiento de datos y tecnologías emergentes nos están llevando progresivamente hacia esa dirección. Este desarrollo promete una nueva era de interactividad e inteligencia, en la que la tecnología se convierte en un aliado constante para mejorar la calidad de vida y la eficiencia en todos los aspectos de la sociedad.



Img.1. Representación visual del avance de las diferentes clasificaciones de la web.

# Tecnologías fundamentales de la web

## HTML y CSS

HTML (HyperText Markup Language) proporciona la estructura fundamental de las páginas web, definiendo y organizando los diferentes elementos que componen un sitio, como títulos, párrafos, listas, imágenes, enlaces y otros componentes. HTML actúa como el esqueleto del contenido, permitiendo a los navegadores mostrar de manera coherente la información al usuario.

CSS (Cascading Style Sheets), por otro lado, se encarga de la presentación visual y el estilo de estos elementos HTML. CSS controla aspectos tales como los colores, las tipografías, los márgenes, los fondos y el diseño general del sitio web. Esta separación entre la estructura (HTML) y la presentación (CSS) facilita la gestión y mantenimiento del sitio, permitiendo realizar cambios de diseño sin alterar la estructura del contenido. Además, CSS ha evolucionado para incluir características avanzadas como el diseño responsivo (responsive design), que asegura que los sitios se adapten correctamente a diferentes tamaños de pantalla y dispositivos, y animaciones y transiciones que enriquecen la experiencia del usuario. Gracias a CSS Grid y Flexbox, es posible crear diseños complejos y flexibles que se adaptan fácilmente a los requisitos modernos de las interfaces de usuario.

## Javascript

JavaScript es el lenguaje de programación más utilizado para desarrollar aplicaciones web dinámicas e interactivas. Originalmente concebido para agregar interactividad básica a las páginas web, su evolución ha hecho que se convierta en una herramienta clave tanto del lado del cliente como del servidor. Gracias a su flexibilidad y la amplia comunidad de desarrolladores, JavaScript se ha transformado continuamente para adaptarse a las necesidades del desarrollo web moderno, aunque actualmente ha sido desplazado como él lenguaje de programación más usado por él lenguaje python..

La aparición de frameworks y bibliotecas como React, Vue y Svelte ha revolucionado la creación de interfaces de usuario. React, desarrollado por Meta (anteriormente Facebook), permite construir componentes reutilizables, facilitando la gestión del estado de las aplicaciones. Vue, conocido por su curva de aprendizaje amigable y su capacidad de integración progresiva, ha ganado popularidad por su simplicidad y versatilidad. Svelte, por su parte, introduce un enfoque innovador, ya que compila los componentes en código altamente optimizado, mejorando significativamente la eficiencia y el rendimiento. Estas herramientas han hecho posible crear experiencias de usuario más rápidas, fluidas y atractivas, transformando cómo interactuamos con las aplicaciones web.

## Blockchain y web 3.0

La tecnología blockchain se ha convertido en un pilar fundamental de la Web 3.0, permitiendo la descentralización de los datos y creando un entorno más seguro y transparente para los usuarios. Blockchain, inicialmente desarrollado como la tecnología subyacente del Bitcoin, ha evolucionado para ser una herramienta versátil que permite la creación de aplicaciones descentralizadas (dApps), eliminando la dependencia de intermediarios y brindando a los usuarios un mayor control sobre sus datos.

Gracias a sus características de inmutabilidad y transparencia, blockchain asegura que la información almacenada no pueda ser alterada sin el consenso de la red, lo cual incrementa la confianza en los sistemas que utilizan esta tecnología. En la Web 3.0, los contratos inteligentes (smart contracts) son una herramienta clave que permite la automatización de procesos y acuerdos de manera segura y confiable, sin necesidad de intermediarios. De esta manera, la Web 3.0, impulsada por blockchain, promete una internet más abierta y equitativa, donde los usuarios tienen el control y la propiedad de sus datos, allanando el camino para un ecosistema digital más seguro y eficiente.



Img.2. Representación visual del avance de las tecnologías de IoT, como expectativas de la web 4.0

# Aplicaciones actuales y retos

## Aplicaciones

Actualmente, la web se utiliza para una variedad inmensa de aplicaciones, que abarcan desde redes sociales hasta plataformas de comercio electrónico y servicios financieros descentralizados (DeFi). Las redes sociales, como Facebook, Instagram y Twitter, permiten a los usuarios conectarse e interactuar en tiempo real, fomentando comunidades globales. El comercio electrónico ha crecido significativamente, con plataformas como Amazon, eBay y Shopify que facilitan la compra y venta de productos en todo el mundo, brindando comodidad tanto a consumidores como a vendedores.

Por otro lado, la Web 3.0 ha permitido el surgimiento de aplicaciones financieras descentralizadas (DeFi), que eliminan la necesidad de instituciones financieras tradicionales y ofrecen servicios como préstamos, intercambios de criptomonedas y generación de ingresos pasivos, todo gestionado de forma segura mediante contratos inteligentes. Estas aplicaciones están transformando sectores clave y redefiniendo cómo las personas interactúan en el entorno digital, con un enfoque cada vez mayor en la seguridad, la accesibilidad y la descentralización.

## Retos

Entre los principales retos de la web actual se encuentran la seguridad, la privacidad y la gobernanza de los datos. La creciente digitalización y la cantidad de datos personales compartidos en línea exigen medidas cada vez más estrictas para proteger a los usuarios contra amenazas como ciberataques, fraudes y violaciones de datos. La privacidad se ha convertido en una preocupación crítica, ya que los usuarios esperan un mayor control sobre la información que comparten y cómo se utiliza, lo cual exige una regulación más eficaz y el desarrollo de tecnologías orientadas a la privacidad.

La gobernanza de los datos, por su parte, implica definir claramente quién tiene derecho a acceder, modificar o distribuir la información, y cómo se gestiona la propiedad de los datos generados. Con la adopción de tecnologías emergentes como blockchain, también surgen desafíos tecnológicos y regulatorios. La descentralización plantea problemas de escalabilidad y consumo energético, mientras que la falta de un marco regulatorio claro dificulta la adopción masiva y genera incertidumbre tanto para los desarrolladores como para los usuarios. Estos retos deben ser abordados para garantizar un desarrollo equitativo y seguro de la web en el futuro, que proteja a los usuarios y permita una adopción generalizada de nuevas tecnologías sin comprometer la seguridad o la privacidad.

# Perspectivas futuras

## Web 4.0 y la conectividad que inunda él mundo

La Web 4.0 se vislumbra como una evolución significativa que integrará de manera aún más profunda la inteligencia artificial (IA), el Internet de las Cosas (IoT) y experiencias inmersivas como la realidad aumentada (AR) y la realidad virtual (VR). Está próxima generación de la web promete una conectividad ubicua y sin fisuras, en la cual los dispositivos inteligentes, desde los electrodomésticos hasta los automóviles, estarán conectados y se comunicarán de manera fluida para ofrecer experiencias altamente personalizadas. La IA jugará un papel fundamental, anticipando las necesidades de los usuarios y adaptando los servicios en tiempo real. Las experiencias inmersivas también se verán potenciadas, transformando la manera en que interactuamos con el mundo digital, desde el entretenimiento hasta la educación y el trabajo. Esta convergencia permitirá una "web sensorial" que responde intuitivamente a los comandos de los usuarios, logrando un nivel de simbiosis entre humanos y tecnología sin precedentes.

## Estandarización y regulación

Con el aumento de la descentralización, se espera que se desarrollen nuevas normativas que aseguren la interoperabilidad, la seguridad y la privacidad de los datos. A medida que blockchain y las aplicaciones descentralizadas (dApps) se vuelven más comunes, es crucial establecer estándares que faciliten la integración y el intercambio de información entre diferentes plataformas y dispositivos, garantizando al mismo tiempo la protección de los usuarios. La estandarización será clave para asegurar que todos los actores tecnológicos operen bajo un marco común, evitando fragmentaciones que puedan limitar el potencial de la Web 4.0. Además, la regulación debe evolucionar para abordar desafíos como la propiedad de los datos, la protección contra fraudes, y los derechos de los usuarios en un entorno digital cada vez más complejo. Los legisladores tendrán el reto de crear políticas equilibradas que promuevan la innovación, protejan los derechos de los usuarios y aseguren un uso ético y transparente de las tecnologías emergentes.



Img.3. Representación visual de dispositivos IoT con conexión web en nuestra era moderna

# Proceso de implementación Raspberry Pi

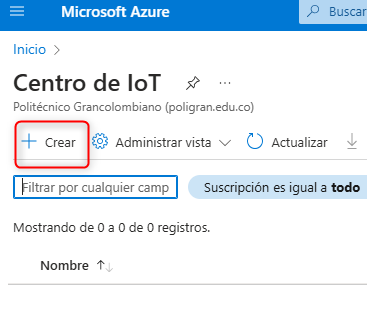
## creación de cuenta AWS for students

Para crear una cuenta Azure for Students, dirígete a la página oficial en<https://azure.microsoft.com/en-us/free/students/>. Esta cuenta ofrece acceso gratuito a varios servicios de Azure sin necesidad de ingresar una tarjeta de crédito, aunque es limitada en cuanto al número de consultas y recursos disponibles, adecuándose a los requerimientos de los estudiantes. Es necesario contar con una cuenta de correo electrónico educativa válida (generalmente con extensión .edu u otra proporcionada por la universidad), que permita verificar tu estatus como estudiante. Luego, sigue las instrucciones de registro para acceder a los beneficios disponibles para estudiantes.

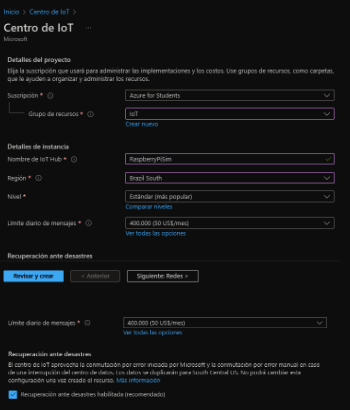
## Configuración del espacio de trabajo



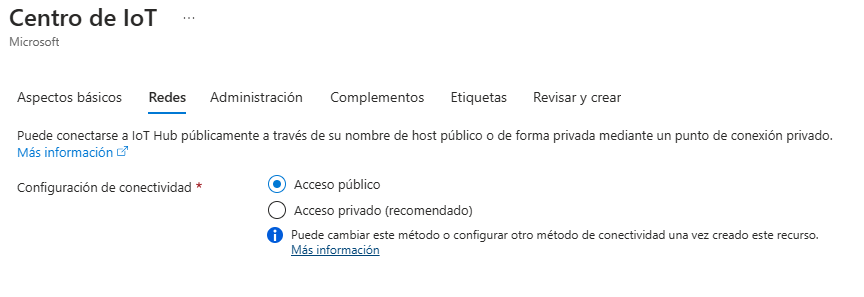
Luego, en la página principal de Azure, usa el buscador para encontrar el "Centro de IoT" y selecciona la opción para crear un nuevo espacio de trabajo. Una vez dentro, comienza la configuración del entorno de trabajo.



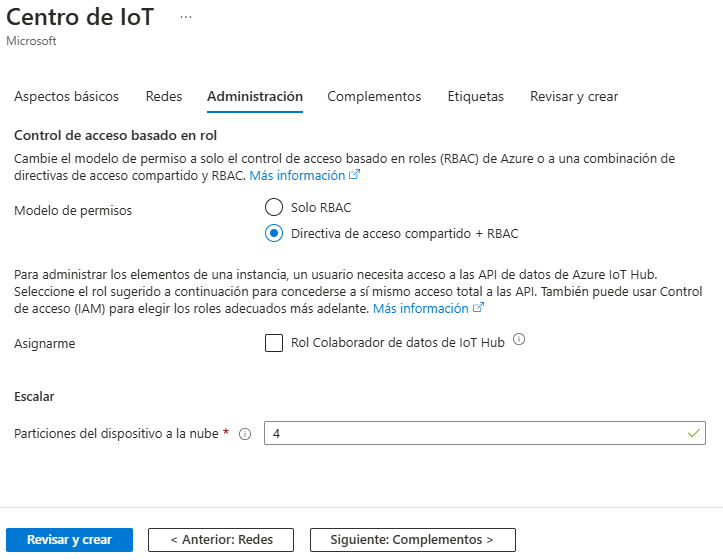
En la configuración del espacio de trabajo, selecciona la región más cercana para reducir la latencia y minimizar errores. Activa la opción de recuperación ante desastres para garantizar la seguridad de los proyectos en caso de fallos en los servidores de Azure.



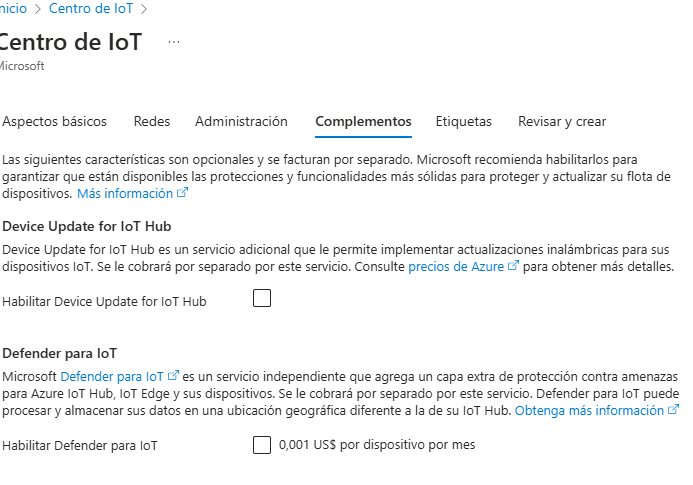
Continúa configurando las redes necesarias para el Centro de IoT, asegurándose de que toda la infraestructura esté preparada para la comunicación adecuada entre los dispositivos y servicios.



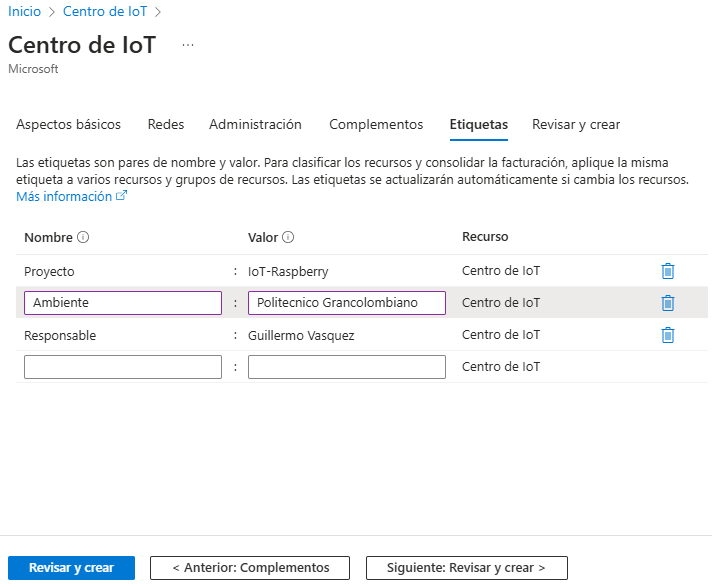
En la sección de administración de acceso, define el método de control de acceso. Puedes optar por "Solo RBAC" para dar permisos específicos a los usuarios, o por la opción de "Directiva de Acceso Compartido + RBAC", que es más flexible y permite un control más detallado. Selecciona esta última opción y marca la casilla del "Rol Colaborador de Datos de IoT Hub" para asignar automáticamente permisos de lectura y escritura. A continuación, ajusta el escalado de particiones en la nube. Para esta aplicación pequeña, cuatro particiones serán suficientes para garantizar un buen rendimiento sin excesos innecesarios.



En la pestaña siguiente se ofrecen complementos de seguridad y actualizaciones inalámbricas para los dispositivos. Dado que la aplicación es sencilla, estos complementos no se habilitarán para evitar una complejidad innecesaria.

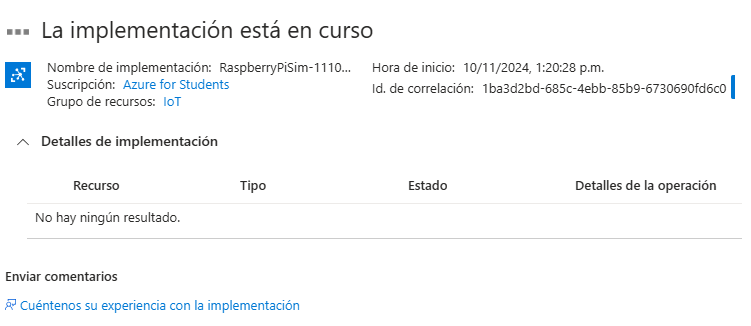


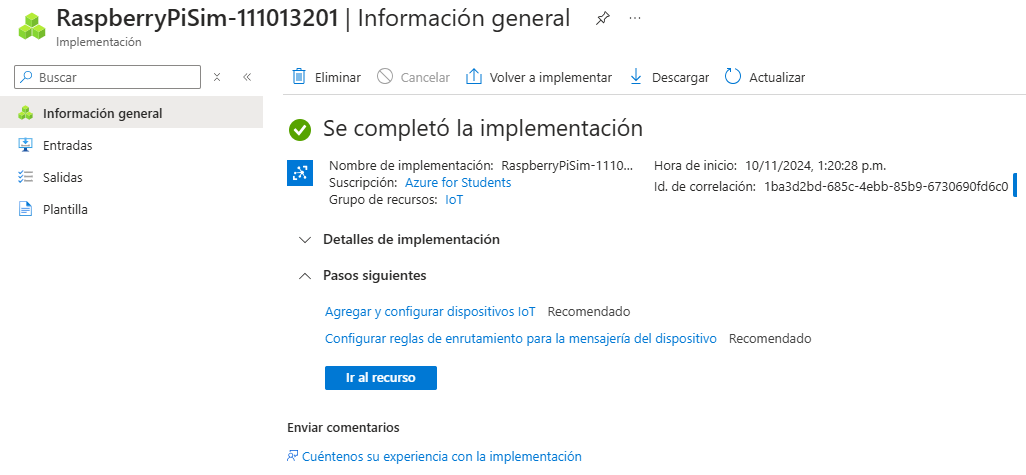
Después, asigna etiquetas al proyecto para facilitar su búsqueda y administración. Estas etiquetas son útiles cuando hay múltiples proyectos, ya que ayudan a organizar elementos y llevar un mejor control de la facturación.



Finalmente, haz clic en "Revisar y crear". Revisa la configuración proporcionada por la plataforma y, si todo está correcto, se selecciona "Crear" para iniciar la implementación del centro de IoT. La implementación comenzará y deberás esperar a que se complete para empezar a trabajar con el entorno creado.







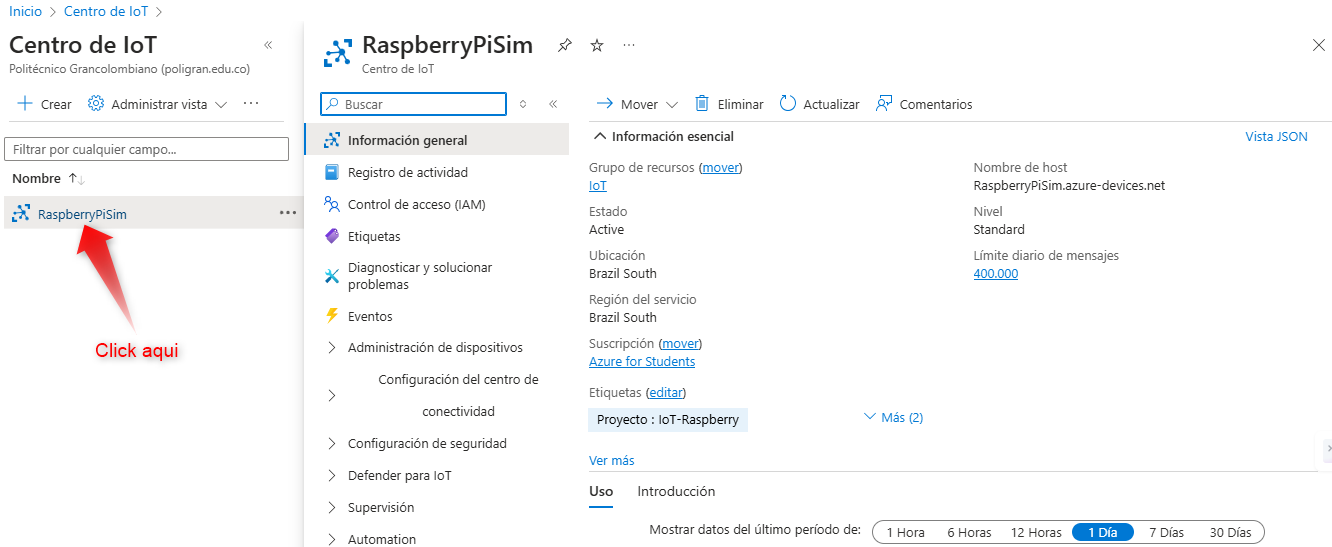
## creación de componente RaspberryPi

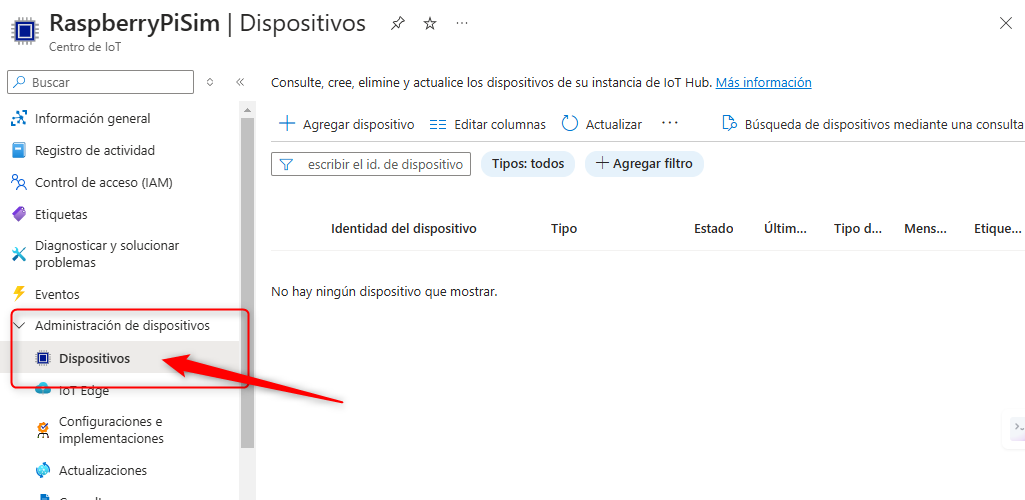
Espacio de trabajo en él centro de IoT

Regresando al Centro de IoT, veremos el espacio de trabajo que hemos creado. Solo necesitamos hacer clic sobre él para acceder y empezar a configurar



El siguiente paso es configurar el dispositivo Raspberry Pi. Para ello, nos dirigimos a la sección de administración de dispositivos y seleccionamos "Dispositivos".

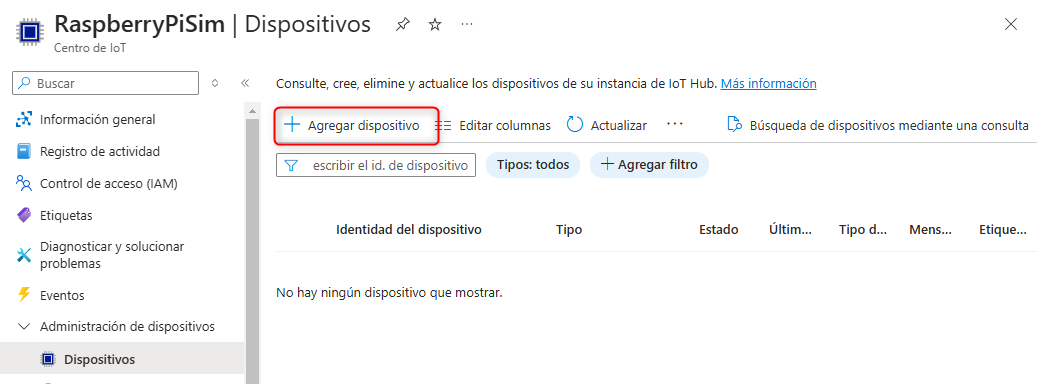




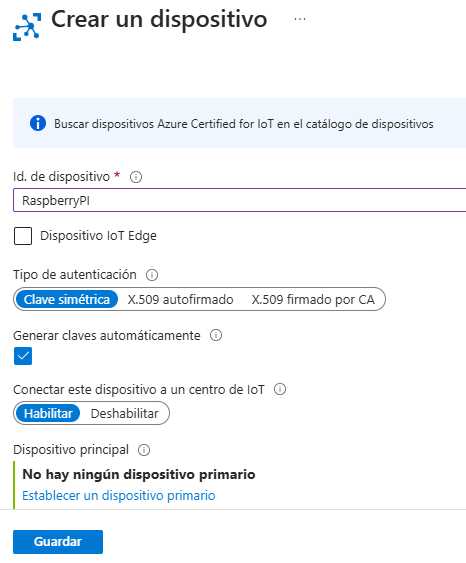
Agregamos, un nuevo dispositivo, asignándole un nombre y luego guarda la configuración. Después de unos segundos, el dispositivo se agregará y estará disponible en el sistema, listo para comenzar con las pruebas y configuraciones adicionales.

**Creación del dispositivo Raspberry Pi Sim en él entorno**

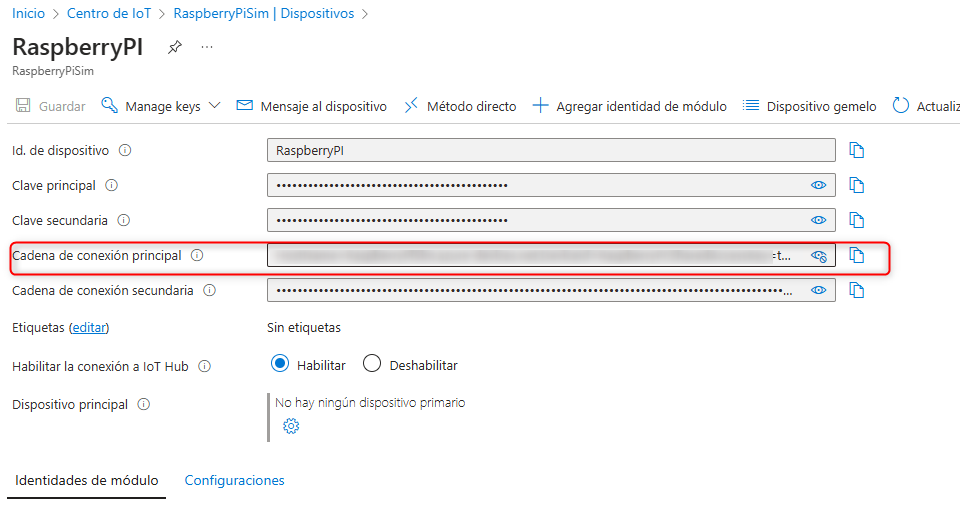
Solo debemos dar en este sitio agregar nuevo dispositivo, como se aprecia la imagen



Concluimos la creación dando guardar.



Hacemos clic sobre él y accedemos

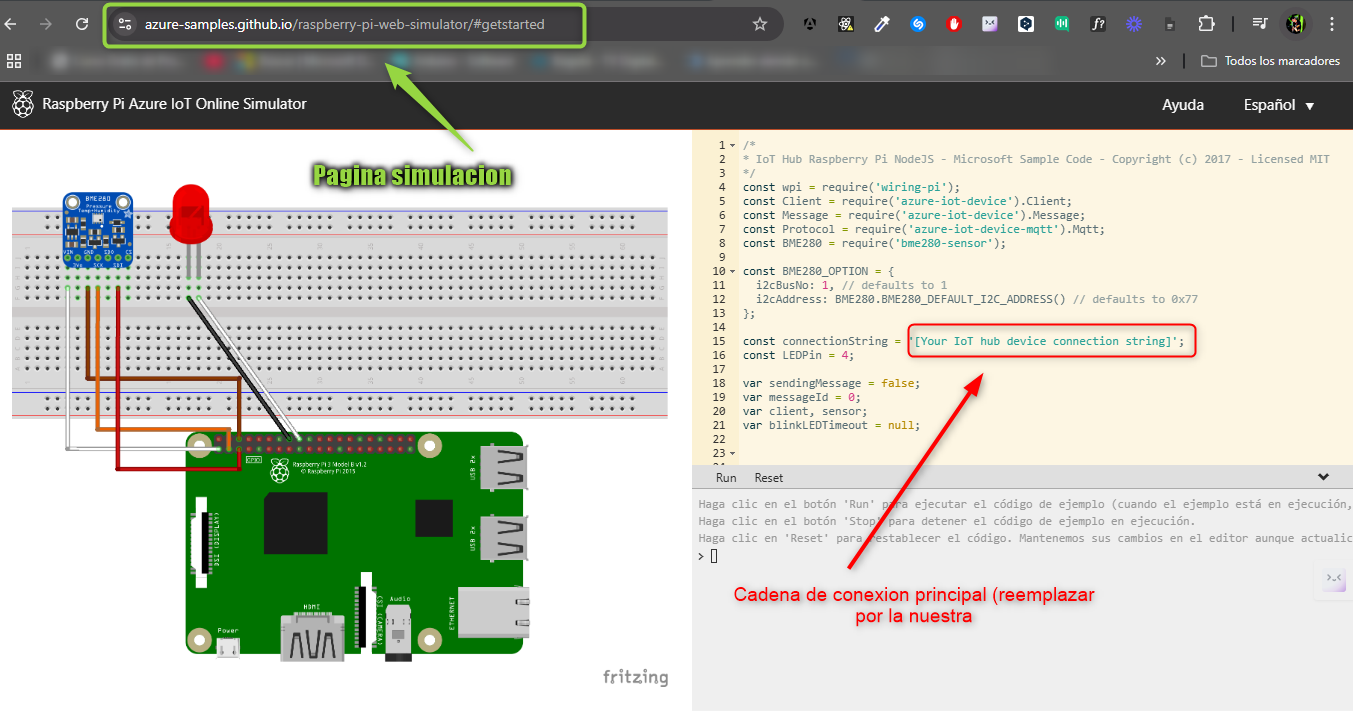


## Simulación mediante cadena de conexión principal

La **cadena de conexión principal:** Esta cadena de conexión actúa como una credencial que autentica y permite que el dispositivo se conecte de forma segura al IoT Hub, contiene varios datos importantes como

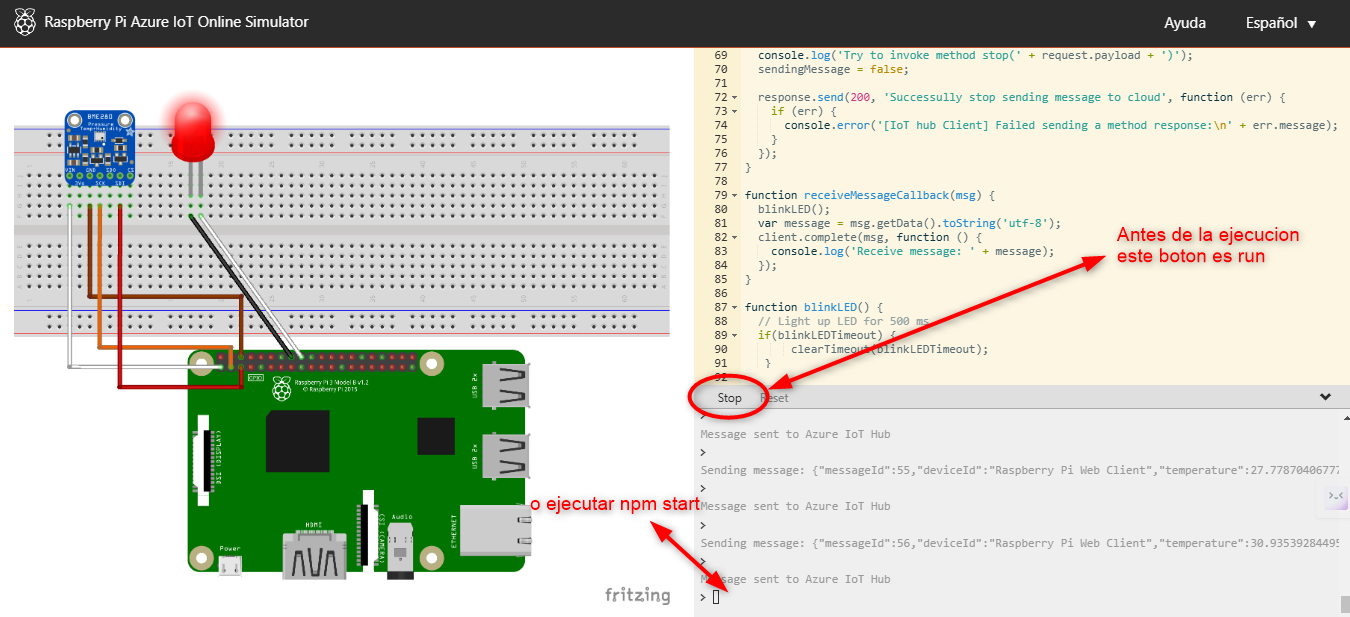
* **HostName**: Es el nombre del IoT Hub al que el dispositivo se va a conectar. Esto identifica al IoT Hub específico en la nube de Azure.
* **DeviceId**: Es el identificador único del dispositivo dentro del IoT Hub. En este caso, el dispositivo se llama RaspberryPI.
* **SharedAccessKey**: Es una clave de acceso compartido que se utiliza para autenticar la conexión del dispositivo al IoT Hub. Esta clave asegura que sólo los dispositivos autorizados puedan conectarse al IoT Hub.

En la pagina <https://azure-samples.github.io/raspberry-pi-web-simulator/#getstarted> en donde apreciamos él simulador en él código de la derecha, en la línea 15 copiaremos este código y lo utilizaremos allí en esa línea 15

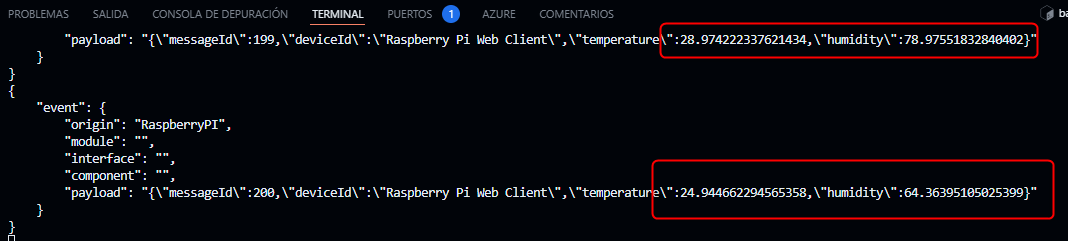


Ya estaría configurado él dispositivo para recibir datos, pero no podríamos verlos amenos que le demos al botón de run o ejecutemos él comando:

*npm start*



Es claro que se debe cambiar CONNECTION\_STRING= por él valor de nuestra cadena de coneccion principal, y así obtendremos los datos del sensor



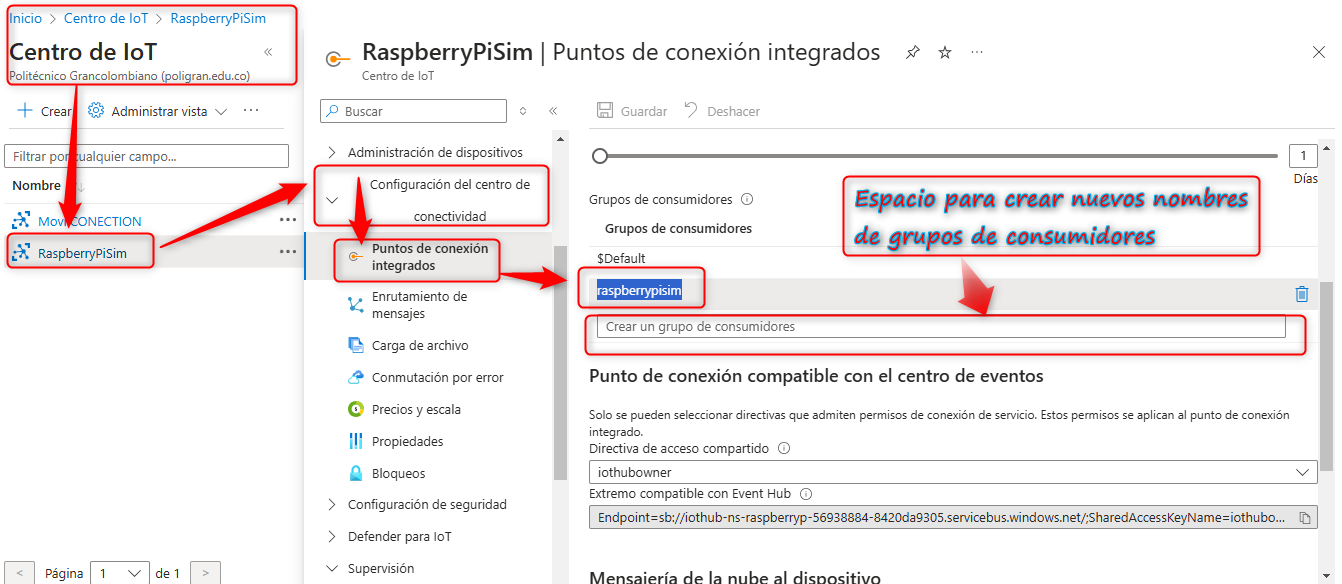
# 

VII. IMPLEMENTACIÓN DE VISUALIZACIÓN DE

DATOS EN TIEMPO REAL DESDE HUB DE IoT

DE AZURE.

El siguiente paso será configurar y crear él grupo de consumidores, para ello debemos ir al IoT Hub dentro de azure y acceder a nuestro dispositivo en este caso [RaspberryPiSim](https://portal.azure.com/#@poligran.edu.co/resource/subscriptions/5cb4281d-15c8-4644-b52e-235fbd41e449/resourceGroups/IoT/providers/Microsoft.Devices/IotHubs/RaspberryPiSim) , luego ir a configuración del centro de conectividad, y en él espacio grupo de consumidores, darle nombre a nuestro grupo, para este caso lo llamaremos similar al entorno raspberrypisim (sim con minúscula), se debe recordar este nombre ya que más tarde lo usaremos en la configuración del socket.



Debemos clonar él repositorio de git que entrega por defecto Azure en su documentación, esto lo podemos hacer usando él editor de código VSC (visual studio code) creando una carpeta en cualquier parte de nuestro PC, y accediendo a esa ruta, es decir si tienes la carpeta creada, en C-desktop-carpetacreada, debes guardar esa ruta, abrir la terminal y acceder a ella mediante él comando:

*cd c:\desktop\carpetacreada*

El siguiente paso será ejecutar él comando para clonar él repositorio que entrega Microsoft en su documentación de azur desde la terminal:

<https://learn.microsoft.com/es-es/azure/iot-hub/iot-hub-live-data-visualization-in-web-apps#prerequisites>

Él comando es él siguiente:

git clone https://github.com/usuario/web-apps-node-iot-hub-data-visualization.git

Luego ir a la carpeta ejecutando también desde la terminal:

cd web-apps-node-iot-hub-data-visualization

Esto nos llevará a la terminal a la carpeta del Proyecto

Muy bien ahora deberemos configurar él entorno mediante diferentes comandos en la terminal como npm, ejecutando:

npm install

Se usa también un framework de node.js para crear él socket o servidor web (Él sockets un punto de comunicación, que permite que nuestro código, se conecte a los datos, que está generando él simulador, y leyendo Azure), ejecutando:

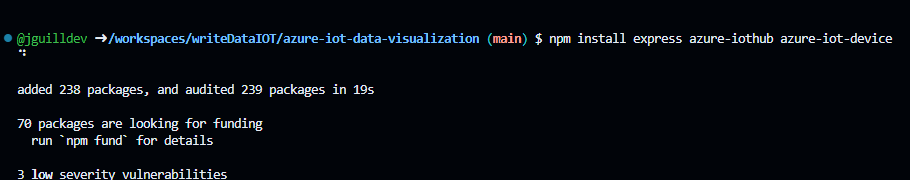
npm install express

Para interactuar con él IoT Hub se debe instalar azure-iothub, ejecutando:

npm install azure-iothub

También él azure-iot-device, que permite la comunicación con él dispositivo simulado, en este caso un raspberryPi, ejecuta desde la terminal:

npm install azure-iot-device



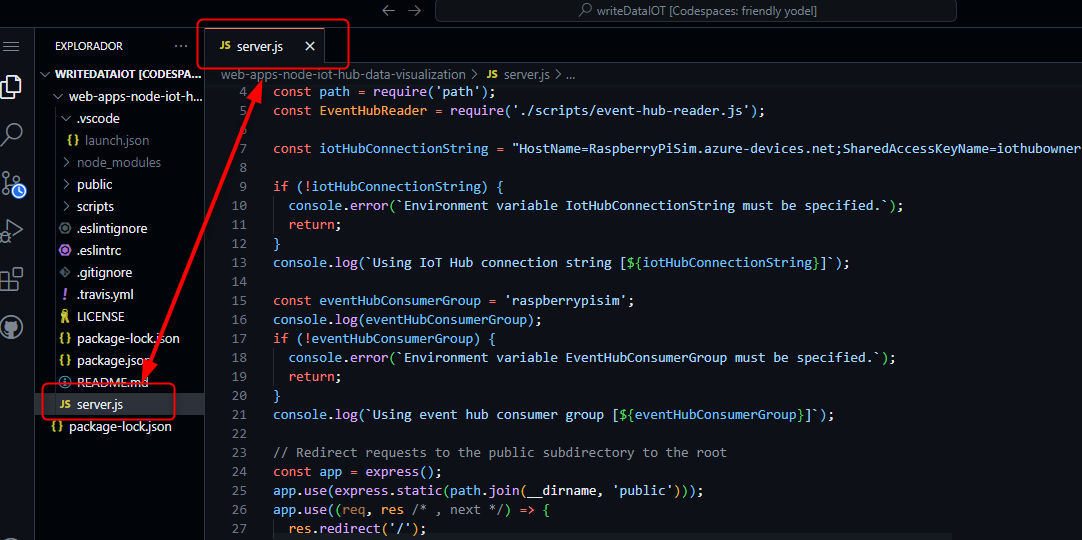
Él protocolo MQTT es él mas usado para enviar mensajes en entornos de IoT, así que también ejecutaremos:

npm install azure-iot-device-mqtt

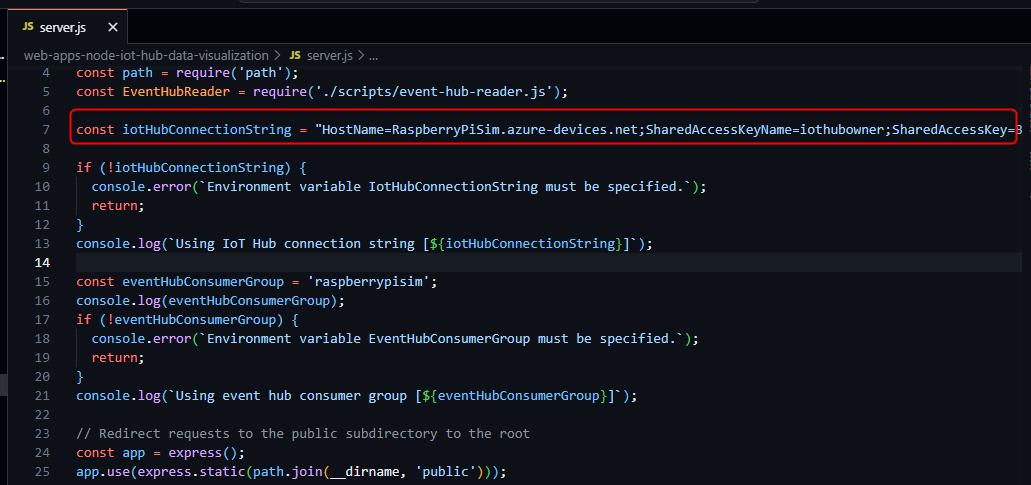
Ejecutar socket.io para la inicialización del socket desde la terminal:

*npm install socket.io*

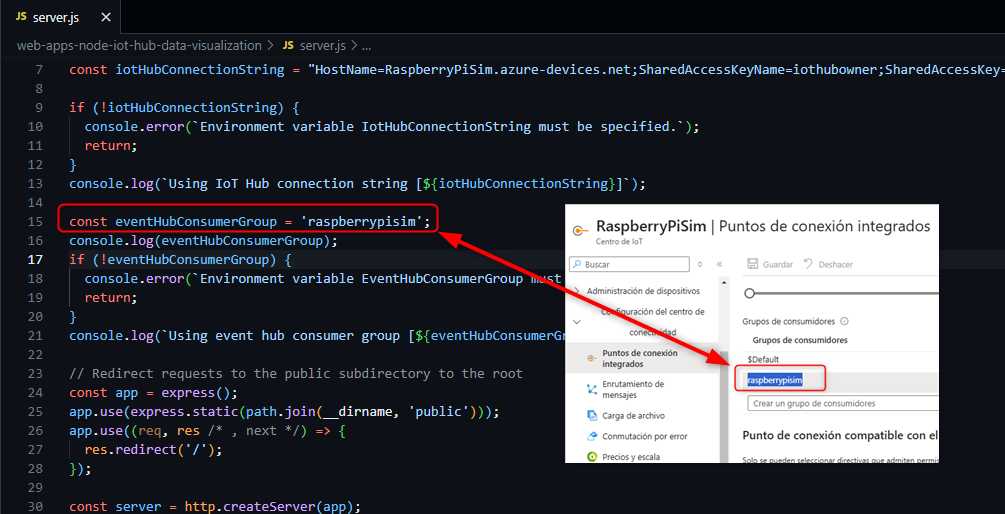
El siguiente paso será modificar él archivo, server.js que está en la carpeta clonada del repositorio de git



Igual que en él código de la simulación debemos reemplazar en la constante de este código la línea const iotHubConnectionString, por la cadena de conexión principal que usamos en él simulador del Raspberry Pi



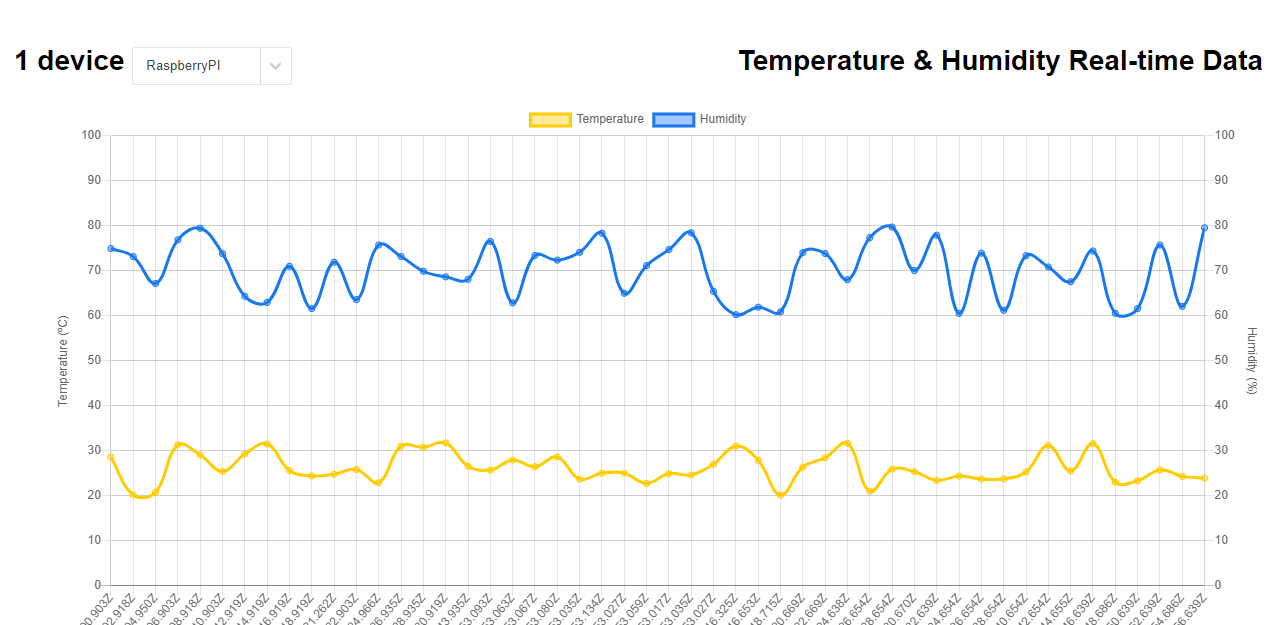
Además también la línea que se refiere al grupo de consumo configurado previamente que nosotros llamamos raspberrypisim, en la línea 15 del server.js



Finalmente ejecutamos desde la terminal usando él comando

node server.js

Esto permitirá que se escuche él puerto 3000, y se dibuje las señales actuales, que empieza a recibir este socket, del sensor de temperatura y humedad del Raspberry Pi Simulado

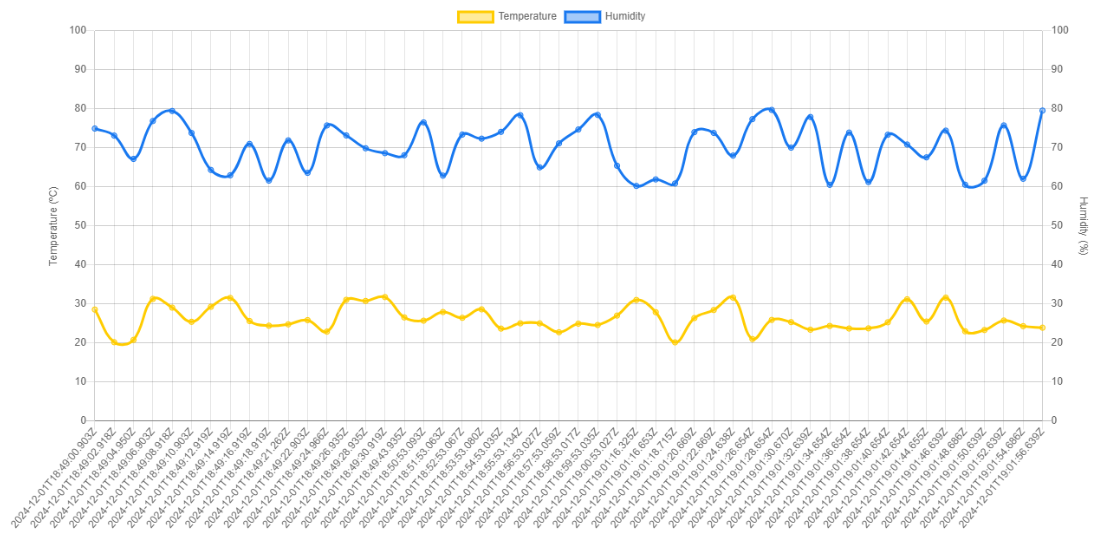


IX. RESUMEN DE DATOS DE LA GRÁFICA

La gráfica que observamos representa la evolución de la temperatura y la humedad a lo largo del tiempo, mostrando cómo ambas variables cambian de forma más o menos regular. La temperatura, representada en amarillo, se mueve entre unos **20°C y 30°C**, sin cambios demasiado drásticos. Podemos notar que sube y baja de manera cíclica, con picos que no llegan a ser extremos.

Por otro lado, la humedad, en azul, muestra un rango algo más amplio, variando entre **60% y 80%**. A diferencia de la temperatura, la humedad tiene una tendencia a ser un poco más volátil, pero sigue un patrón similar, también con picos y caídas que se repiten en el tiempo. Este comportamiento sincronizado podría reflejar una relación directa entre ambas variables: cuando la temperatura sube, la humedad tiende a fluctuar de forma parecida. Esto puede ser un indicio de cómo las condiciones del ambiente influyen en ambas de manera simultánea.

Aunque las dos variables cambian constantemente, no hay una sensación de caos. La estabilidad relativa de ambos parámetros sugiere que, a pesar de sus altibajos, el sistema en general se encuentra en equilibrio, sin grandes desvíos de los valores esperados.



IX. CONCLUSIÓN

Durante nuestro recorrido desde la configuración del arte móvil hasta la integración en Azure, hemos aprendido mucho sobre la colaboración y la superación de obstáculos técnicos. Comenzamos con una visión sobre cómo las tecnologías móviles podían conectarse al Internet de las Cosas, y luego nos movimos hacia un entorno más robusto y escalable usando Azure IoT Hub. Esta transición mostró la capacidad de estos sistemas para evolucionar, desde el uso básico hasta una infraestructura completa que ofrece monitoreo en tiempo real y análisis de datos.

Trabajar en la visualización de los datos también fue una parte reveladora de este proyecto. Lograr ver reflejada la temperatura y la humedad en gráficos en tiempo real no solo fue un logro técnico, sino también una muestra de cómo el hardware, el software y la nube pueden conectarse para generar información útil. Las gráficas mostraban las fluctuaciones naturales de temperatura y humedad, y esto nos llevó a entender el poder del monitoreo continuo. La capacidad de detectar patrones a través de los datos recolectados fue una validación del valor de tener un sistema de IoT funcionando de manera integrada.

El uso de Azure como plataforma de manejo de datos ha demostrado ser una elección que da lugar a flexibilidad y posibilidades de crecimiento. Empezamos desde cero, clonando un repositorio, instalando las dependencias necesarias, y configurando cada componente. Esta infraestructura permite que el sistema pueda adaptarse fácilmente, añadiendo más dispositivos o expandiendo el rango de datos que queremos observar. También nos enseñó que cada configuración, cada parámetro que ingresamos, tiene un papel crucial en el éxito final de la integración.

References

1. C. Galindo, A. Hernández y J. Martínez, “Implementación de sistemas de monitoreo ambiental con Raspberry Pi para redes IoT,” *Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 12, no. 3, pp. 148-155, 2019. Este artículo en español discute aplicaciones específicas de la Raspberry Pi en el monitoreo ambiental, una de las áreas clave de IoT.
2. M. Richardson y S. Wallace, *Getting Started with Raspberry Pi*, 3rd ed., O'Reilly Media, 2016. Este libro es una guía introductoria que incluye aplicaciones prácticas de la Raspberry Pi en proyectos de IoT, ideal para usuarios que comienzan a explorar sus capacidades.
3. R. Morales, “Comparación de plataformas de IoT: Raspberry Pi y Arduino en aplicaciones de domótica y control,” *Ciencia y Tecnología para el Desarrollo*, vol. 5, no. 2, pp. 200-207, 2020. Este estudio en español analiza las diferencias entre la Raspberry Pi y el Arduino en términos de rendimiento y consumo energético, proporcionando una comparativa útil para proyectos IoT.
4. S. A. Osman, “Raspberry Pi and the Internet of Things: A Review,” *International Journal of Computer Science and Network Security*, vol. 20, no. 4, pp. 120-127, 2020. Este artículo en inglés ofrece una revisión del uso de la Raspberry Pi en IoT, abordando sus aplicaciones y desafíos técnicos.
5. F. J. García y M. P. Pérez, “Diseño de un sistema de riego automatizado utilizando Raspberry Pi y sensores de humedad,” *Ingeniería e Investigación*, vol. 34, no. 1, pp. 123-130, 2021. Este artículo en español presenta un caso práctico de uso de la Raspberry Pi en agricultura inteligente, uno de los campos de aplicación destacados en IoT.
6. S. Monk, *Programming the Raspberry Pi: Getting Started with Python*, 2nd ed., McGraw-Hill Education, 2015. Este libro proporciona una guía sobre programación en Python para la Raspberry Pi, un lenguaje comúnmente usado en proyectos IoT.
7. P. Mellado y D. Ruiz, “Análisis de la seguridad en dispositivos IoT: Aplicaciones y vulnerabilidades en Raspberry Pi,” *Revista Electrónica de Computación*, vol. 18, no. 4, pp. 45-53, 2020. Este artículo en español explora los aspectos de seguridad en la Raspberry Pi, destacando los riesgos y soluciones en el contexto de IoT.
8. J. Johnson y M. Smith, “Edge Computing with Raspberry Pi for Real-Time IoT Applications,” *Journal of IoT Research*, vol. 10, no. 2, pp. 85-94, 2021. Este artículo en inglés analiza el uso de la Raspberry Pi en computación en el borde, un área emergente en IoT que permite procesar datos en tiempo real.
9. L. Casas y F. Díaz, “Plataformas de desarrollo para IoT: Raspberry Pi, Arduino y ESP8266,” *Avances en Computación e Ingeniería*, vol. 8, no. 3, pp. 99-110, 2018. Este artículo en español compara la Raspberry Pi con otras plataformas IoT populares, proporcionando una visión integral de sus ventajas y limitaciones.
10. E. Upton y G. Halfacree, *Raspberry Pi User Guide*, 4th ed., Wiley, 2016. Este libro en inglés es una guía detallada sobre el uso de la Raspberry Pi, sus capacidades técnicas y ejemplos de proyectos prácticos, útil para comprender sus aplicaciones en IoT.
11. Microsoft Docs, "Visualización de datos en tiempo real en aplicaciones web con Azure IoT Hub", Microsoft Learn, 2023. Este recurso en español explica los pasos necesarios para visualizar datos de IoT Hub en una aplicación web, incluyendo los prerrequisitos y configuraciones requeridas.
12. Global Azure Spain, "Getting Started with Azure IoT", YouTube, 2022. Este video, en español, proporciona una introducción detallada al uso de Azure IoT, destacando sus aplicaciones prácticas y ejemplos.
13. Taygan Rifat, "Azure IoT Hub Overview", YouTube, 2023. Video en inglés donde se explora el concepto de IoT Hub y cómo se puede usar con Raspberry Pi, proporcionando un excelente complemento visual para comprender el entorno de desarrollo.
14. Microsoft Azure IoT Live Stream, "IoT Hub Visualización en tiempo real", YouTube, 2023. Este video, en español, muestra cómo conectar dispositivos IoT a Azure IoT Hub y visualizar los datos en tiempo real, ideal para comprender los conceptos de integración en aplicaciones web.
15. GitHub Repository, "Node.js Azure IoT Hub Data Visualization Web App", GitHub. Este repositorio contiene el proyecto web para la visualización de datos de IoT Hub utilizando Node.js, configurado para Azure
16. D. Cruz y A. Romero, “Guía para el uso de Azure IoT Hub con Raspberry Pi”, Blog de IoT para Desarrolladores, 2022. Este artículo describe los pasos básicos para configurar Azure IoT Hub con una Raspberry Pi, proporcionando una base sólida para principiantes.
17. J. Gómez, “Conexión de la Raspberry Pi a Azure IoT”, Blog de Tecnologías, vol. 9, pp. 23-29, 2022. Se abordan las principales dificultades que los desarrolladores pueden enfrentar al conectar una Raspberry Pi a Azure IoT, con soluciones prácticas y sugerencias para evitar errores comunes.
18. F. Morales, “Desarrollo de Proyectos IoT en Azure: Buenas Prácticas”, IoT en el Hogar, 2021. Este recurso en español proporciona buenas prácticas para diseñar soluciones IoT que involucren Azure y Raspberry Pi, cubriendo desde la conexión inicial hasta la visualización de datos.
19. M. Reyes y L. Vargas, “Visualización de datos con Node.js y Azure”, Revista de Desarrollo Web, 2021. En este artículo se explica cómo utilizar Node.js junto con Azure para visualizar datos provenientes de dispositivos IoT, utilizando ejemplos claros y específicos con Raspberry Pi.

1. Vasquez Benitez Jose Guillermo

   jgvasquez@poligran.edu.co [↑](#footnote-ref-0)