



TECNICATURA SUPERIOR EN **Telecomunicaciones**

Espacio: Proyecto Integrador I – TST – 2024

Alumno: Dario Arriola

Profesor: Gonzalo Vera

Objetivos:

1. Comprender los Fundamentos del IoT: Familiarizarse con los conceptos básicos del Internet de las Cosas, incluyendo su importancia, aplicaciones y el impacto potencial en la sociedad y la industria.
2. Introducción a Git y GitHub: Aprender los fundamentos de Git y GitHub como herramientas esenciales para el desarrollo colaborativo de software, especialmente en proyectos de IoT.
3. Exploración de Herramientas de Desarrollo: Comenzar a utilizar herramientas de desarrollo clave como Visual Studio Code (VsCode), comprendiendo su importancia en la programación y desarrollo de proyectos.

TRABAJO PRÁCTICO N° 1

PARTE 1

A) Definición y Aplicaciones del IoT:

El Internet de las Cosas (IoT) se refiere a la interconexión de dispositivos físicos a través de internet, permitiendo que se comuniquen entre sí y con otros sistemas para recopilar, analizar y actuar sobre datos en tiempo real. Estos dispositivos incluyen desde sensores y actuadores hasta electrodomésticos, vehículos y maquinaria industrial, todos equipados con tecnología que les permite enviar y recibir datos. El IoT está revolucionando diversas industrias al proporcionar información valiosa que optimiza procesos, mejora la eficiencia y facilita nuevas formas de interacción entre el mundo físico y digital.

Tres aplicaciones prácticas del IoT en diferentes industrias incluyen:

Domótica: En el sector de los hogares inteligentes, el IoT permite el control y automatización de dispositivos como luces, termostatos, cerraduras de puertas y sistemas de seguridad. Esto no solo mejora la comodidad y la eficiencia energética, sino que también aumenta la seguridad al permitir el monitoreo remoto del hogar.

Industria 4.0: En la manufactura, el IoT juega un papel crucial en la transformación hacia la "Industria 4.0", donde las máquinas y sistemas de producción están interconectados para monitorear y optimizar procesos en tiempo real. Esto reduce tiempos de inactividad, mejora la calidad del producto y reduce costos operativos.

Salud: En la atención médica, el IoT facilita el monitoreo remoto de pacientes mediante dispositivos wearables que recopilan datos como frecuencia cardíaca, presión arterial y niveles de glucosa. Estos datos pueden ser enviados a los profesionales de la salud para un monitoreo constante y una intervención temprana si se detectan anomalías.

B) Impacto del IoT:

El IoT tiene el potencial de transformar significativamente la vida cotidiana y los negocios en el futuro. En la vida diaria, las personas podrían disfrutar de hogares más inteligentes, ciudades más seguras y transportes más eficientes. Los datos en tiempo real permitirán a las empresas tomar decisiones más informadas, mejorar la experiencia del cliente y crear nuevos modelos de negocio basados en servicios conectados.

Un problema actual que podría resolverse con el IoT es la gestión de residuos en las ciudades. Actualmente, la recolección de basura a menudo sigue rutas fijas y horarios predeterminados, lo que puede resultar ineficiente y costoso. Una solución basada en IoT podría incluir la instalación de sensores en los contenedores de basura que midan su nivel de llenado en tiempo real. Estos datos se enviarían a un sistema central que optimizaría las rutas de recolección,

Proyecto Integrador I - TST - 2024 Dario Arriola

asegurando que los camiones recojan los residuos solo cuando los contenedores estén llenos, ahorrando tiempo, combustible y reduciendo la huella de carbono.

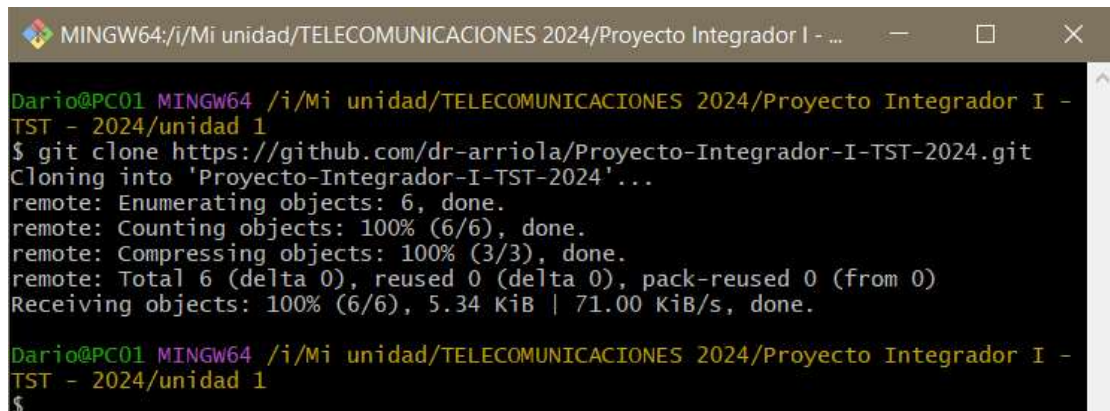
PARTE 2

Informe de Aprendizaje: Introducción a Git y GitHub

Informe de Aprendizaje:

Durante el tutorial de Git y GitHub, aprendí los conceptos básicos y las operaciones fundamentales que facilitan el desarrollo colaborativo de software. Comencé por entender cómo inicializar un repositorio Git en mi máquina local, lo que me permitió convertir una carpeta normal en un repositorio controlado por versiones.

<https://github.com/dr-arriola/Proyecto-Integrador-I-TST-2024>



```
MINGW64:/i/Mi unidad/TELECOMUNICACIONES 2024/Proyecto Integrador I - ...
Dario@PC01 MINGW64 /i/Mi unidad/TELECOMUNICACIONES 2024/Proyecto Integrador I - TST - 2024/unidad 1
$ git clone https://github.com/dr-arriola/Proyecto-Integrador-I-TST-2024.git
Cloning into 'Proyecto-Integrador-I-TST-2024'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), 5.34 KiB | 71.00 KiB/s, done.
Dario@PC01 MINGW64 /i/Mi unidad/TELECOMUNICACIONES 2024/Proyecto Integrador I - TST - 2024/unidad 1
$
```

A continuación, aprendí a hacer commits, que son esencialmente capturas de los cambios realizados en el código. Es importante acompañar cada commit con un mensaje claro que describa qué cambios se hicieron y por qué. Esto facilita el seguimiento del historial del proyecto y la colaboración con otros desarrolladores.

```
Dario@PC01 MINGW64 /i/Mi unidad/TELECOMUNICACIONES 2024/Proyecto Integrador I -  
TST - 2024/unidad 1/Proyecto-Integrador-I-TST-2024 (main)  
$ git commit -m "clone el repositorio en una carpeta local"  
[main dc20b8e] clone el repositorio en una carpeta local  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 desktop.ini  
  
Dario@PC01 MINGW64 /i/Mi unidad/TELECOMUNICACIONES 2024/Proyecto Integrador I -  
TST - 2024/unidad 1/Proyecto-Integrador-I-TST-2024 (main)  
$
```

El siguiente paso fue aprender sobre los comandos push y pull, que son esenciales para sincronizar los cambios entre el repositorio local y el remoto en GitHub. Push envía mis commits al servidor de GitHub, mientras que pull descarga y aplica los cambios realizados por otros en mi copia local del proyecto.

Finalmente, el tutorial abordó el uso de ramas (branches) y pull requests. Crear ramas me permite trabajar en nuevas características o correcciones sin afectar el código principal. Una vez que los cambios están listos, los puedo integrar de vuelta al proyecto principal mediante un pull request, lo cual facilita la revisión y colaboración entre equipos.

Este tutorial me ha proporcionado una base sólida en el uso de Git y GitHub, herramientas fundamentales para cualquier desarrollador de software en proyectos colaborativos.