

Trabajo Práctico: Introducción a TypeScript

Ejercicio 1: Convertir un archivo JavaScript (app.js) a TypeScript (app.ts) añadiendo tipos de datos adecuados para mejorar la robustez del código.

```
// app.js

function calcularArea(largo, ancho) {
  return largo * ancho;
}

function saludar(nombre, edad) {
  if (edad >= 18) {
    return `Hola ${nombre}, eres mayor de edad.`;
  } else {
    return `Hola ${nombre}, eres menor de edad.`;
  }
}

const nombreProyecto = "Proyecto";
let largo = 15;
let ancho = 7;
let edadUsuario = 25;
const nombreUsuario = "Ana";

const area = calcularArea(largo, ancho);
console.log(`${nombreProyecto} tiene un área de ${area}.`);
console.log(saludar(nombreUsuario, edadUsuario));

// Ejemplo de datos mixtos
const datosUsuario = ["Carlos", 30];
```

Ejercicio 2: Configuración de un Proyecto TS

1. Crear una carpeta mi-primer-proyecto-ts y arrastrarla a VS
2. Inicializa un nuevo proyecto de Node.js. En la terminal de VS inicia el ambiente

```
npm init -y
```

```
npm install typescript --save-dev
```

```
npx tsc --init
```

3. Configura tu archivo tsconfig.json

```
{
  "compilerOptions": {
    "target": "es6",
    "module": "commonjs",
    "strict": true,
    "outDir": "./dist",
    "rootDir": "./src"
  },
  "include": ["src/**/*"],
  "exclude": ["node_modules"]
}
```

4. Crea un archivo `src/index.ts`:

```
// src/index.ts

// Tipos básicos
let nombre: string = "Ana";
let edad: number = 30;
let esEstudiante: boolean = true;

// Imprime los valores
console.log(`Nombre: ${nombre}, Edad: ${edad}, Estudiante: ${esEstudiante}`);
```

5. Ejecuta el código en una terminal

`npx tsc`

6. Ejecuta el código compilado

`node dist/index.js`

Ejercicio 3: Probar el funcionamiento del ejercicio 1