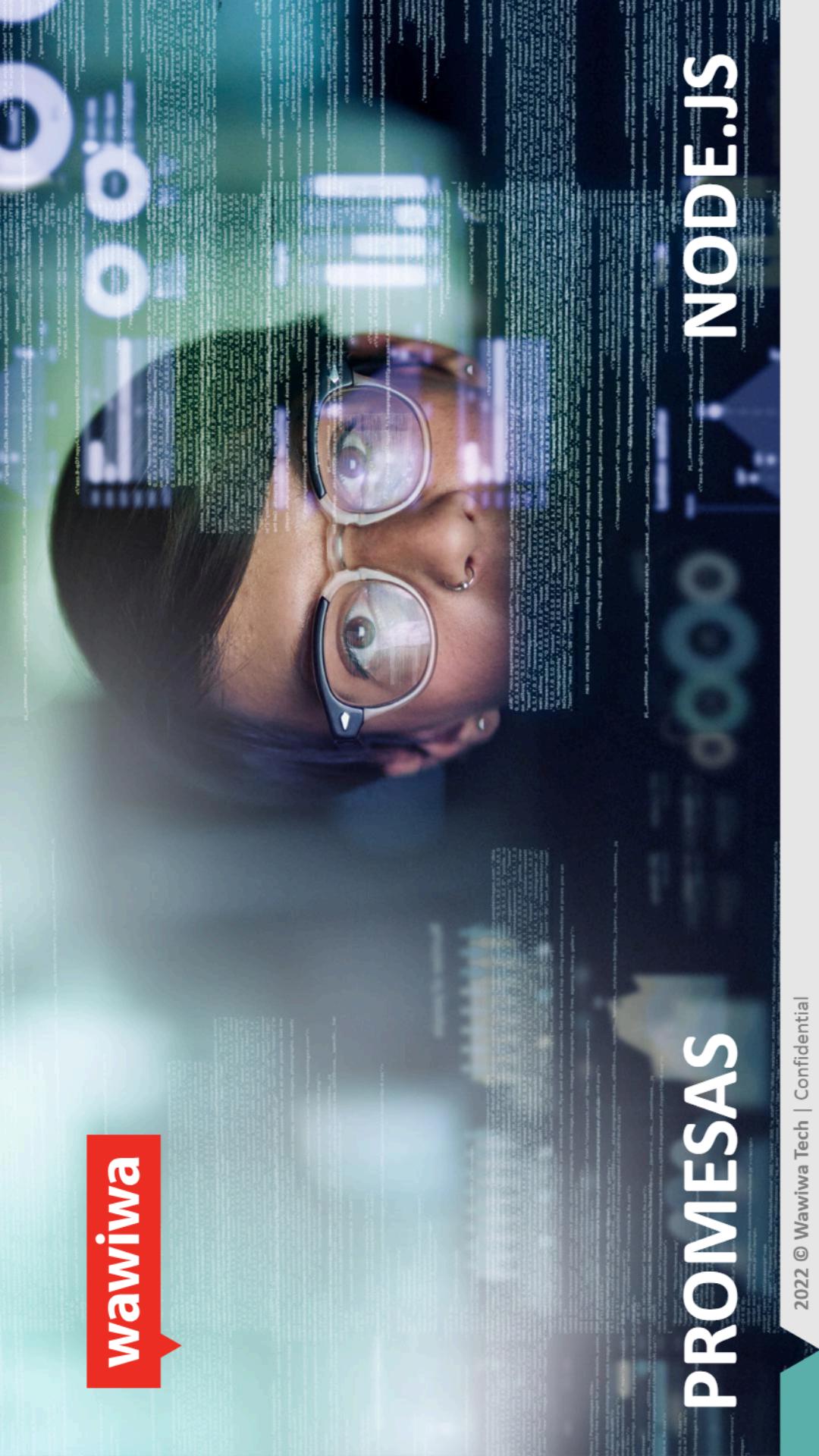


# NODE.JS

# PROMESAS

wawiwa



# callback en callback

## index1-callback-in-callback.js

```
fs.readFile('./data/data.json', 'utf-8', (err, data) => {
  if (err) return "couldn't read file";

  const product = "Ball";
  const exist = false;

  JSON.parse(data).forEach(element => {
    if (product === element.productName) exist = true;
  });

  if (exist) {
    fs.writeFile('./data/data1.json', 'Ball Exist', err => {
      if (err) console.error(err);
    });
  } else {
    fs.writeFile('./data/data1.json', 'Ball Not Exist', err => {
      if (err) console.error(err);
    });
  }
});
```

cnapo-server-without-express > promises > data > **if** data.json

**1 Ball Exist**

Callback en callback - no una buena solución

# data.json

data/data.json

```
[  
  {  
    "id": 0,  
    "productName": "Ball",  
    "image": "🟡",  
    "from": "USA",  
    "quantity": "48",  
    "price": "8",  
    "description": "Jumping yellow ball"  
  },  
  {  
    "id": 0,  
    "productName": "Heart",  
    "image": "❤️",  
    "from": "London",  
    "quantity": "45",  
    "price": "10",  
    "description": "Orange Heart"  
  }]
```

# La solución son las promises

index2-promise.js

```
const readFromFile = file => {
    return new Promise((resolve, reject) => {
        fs.readFile(file, (err, data) => {
            if(err) reject(err);
            resolve(data);
        });
    });
};

const writeDataToFile = (file, data) => {
    return new Promise((resolve, reject) => {
        fs.writeFile(file, data, err => {
            if(err) reject(err);
            else resolve(data);
        });
    });
};

};//Continue next page
```

# promesas

```
readFileData('./data/data1.json')
  .then(data => {
    data = JSON.parse(data);

    for (let i = 0; i < data.length; i++) {
      if (data[i].productName === "Ball")
        return data[i].productName;
    }

    return false;
  })
  .then(productName => {
    const output = productName ?
      `product ${productName} : 'product Not exists'` :
      `product ${productName} : 'product ${productName}'`;
    return writeDataToFile(`./data/data2.json`, output);
  })
  .then(() => console.log("data saved to file"))
  .catch(err => console.error(err));
}
```

# Async y await

index3-await-async.js

```
const readfileData = file => {
  return new Promise((resolve, reject) => {
    fs.readFile(file, (err, data) => {
      if (err) reject(err);
      else resolve(data);
    });
  });
}

const writeDataToFile = (file, data) => {
  return new Promise((resolve, reject) => {
    fs.writeFile(file, data, err => {
      if (err) reject(err);
      else resolve(data);
    });
  });
};

//Continua en la próxima página
```

# Async y await

```
const writingDataToFile = async () => {
  try {
    const data = JSON.parse(await readFileData('./data/data.json'));
    let productName = false;

    for (let i = 0; i < data.length; i++) {
      if (data[i].productName === "Ball")
        productName = data[i].productName;
    }

    const output = productName ?
      `product ${productName} saved` : 'product Not exists';

    return await writeDataToFile('./data/data2.json', output);
  } catch (e) {
    console.error(e);
  }
}; // Continúa en la próxima página
```

# Async y await

```
writingDataToFile()  
  .then(x => {  
    console.log(x); // product Ball saved  
    console.log('data saved to file!');  
  })  
  .catch(err => console.error('ERROR'));
```



wawiwa

¿Preguntas?