



# Desarrollo de software 2025

### **Propósito**

Proporcionar a los estudiantes los conocimientos, habilidades y herramientas esenciales para desarrollar software de manera efectiva y segura, centrándose en aspectos relevantes como lo son las arquitecturas y herramientas, el desarrollo de interfaces de usuario, aplicaciones orientadas a servicios, el desarrollo seguro y la realización de pruebas unitarias.

Brindar a los estudiantes pautas que le permitan desarrollar software de calidad cumpliendo las mejores prácticas para enfrentar los desafíos diarios en la industria fomentando la creación de soluciones tecnológicas seguras y eficientes.

### **Objetivos establecidos en el Diseño Curricular**

- Que el alumno conozca las arquitecturas, herramientas y patrones para el desarrollo de software.
- Que el alumno desarrolle interfaces de usuario.
- Que el alumno pueda crear soluciones de software que den respuestas a necesidades reales.
- Que el alumno sea capaz de aplicar buenas prácticas y tecnologías en el desarrollo seguro.

# Trabajo práctico integrador

**Elegir un dominio**

**Grupal**

**A través de un  
repositorio de git**



**Arquitectura Cliente  
Servidor**

**Microservicios**

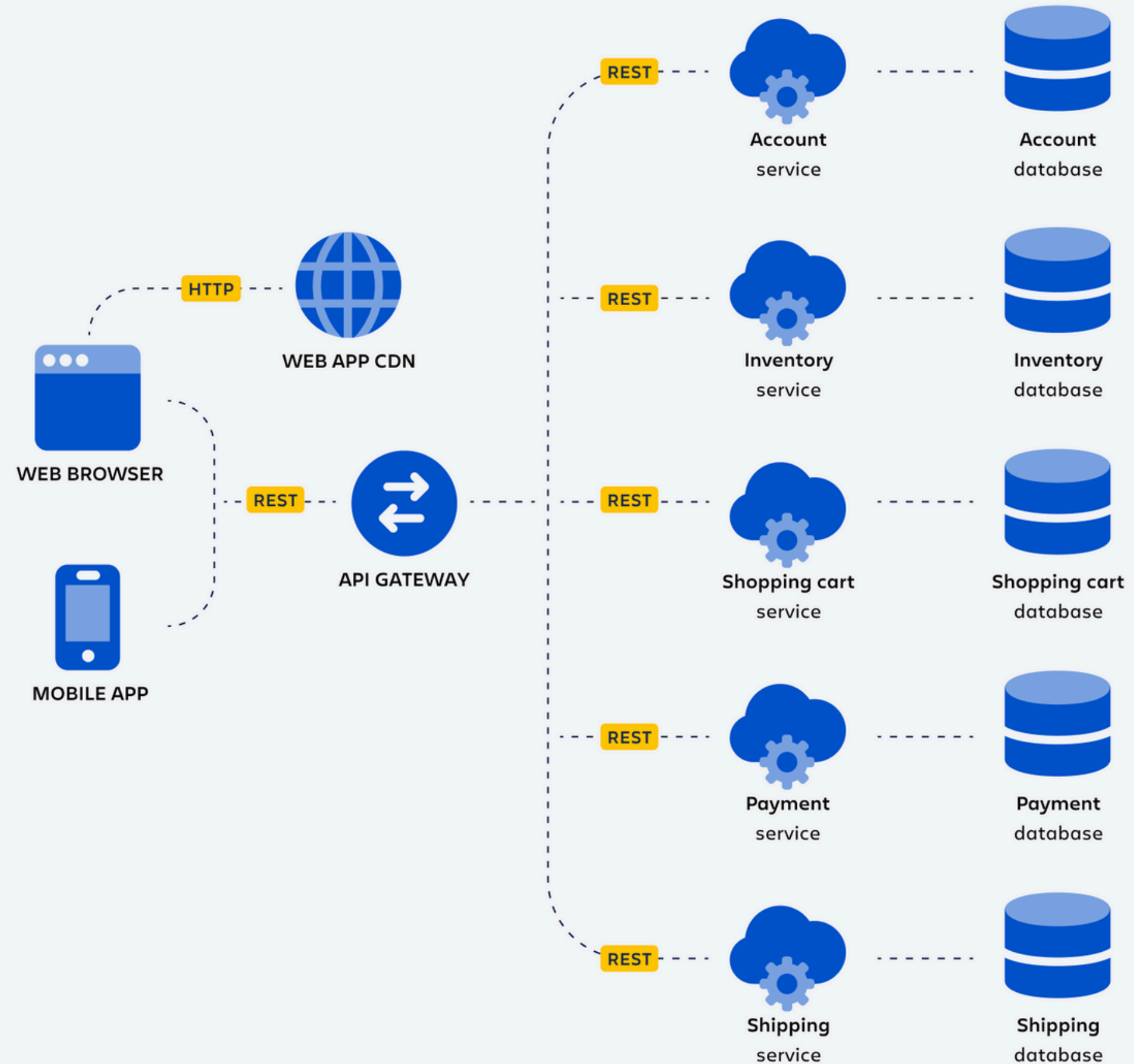
# Microservicios

## Collaboration software for software, IT and business teams

Atlassian's team collaboration software like Jira, Confluence and Trello help teams organize, discuss, and complete shared work.

 [atlassian.com](https://atlassian.com)

### STOREFRONT WEB APP





**Node**  
**Typescript**  
**Javascript**

**NO es un lenguaje de  
programación !!!**

# Node.js

Node.js, es un entorno en tiempo de ejecución basado en JavaScript



# Frontend y Backend

# Tópicos

## JAVASCRIPT



Lenguaje de programación

Podemos utilizar **Typescript**

## MOTOR V8 DE GOOGLE



Diseñado para ejecutar JavaScript  
en los navegadores

Motor subyacente de node

## FRAMEWORKS/ BIBLIOTECAS



Express  
NestJS  
Multer  
Joi  
TypeOrm

## MODULOS



global (console, process,  
setTimeout, buffer)  
**fs**  
**os**  
**path**  
...

# Valores Primitivos

- undefined
- null
- boolean
- number - bigint
- string

# Objetos globales

Date  
JSON  
Map  
Function  
ArrayBuffer

**STRINGS**

Manipular una secuencia de caracteres.

**ARRAYS**

Objetos tipo lista de alto nivel

# String

indexOf / lastIndexOf

Localizar un substring

replace / replaceAll

Reemplazar un substring

substring

Cortar una cadena

trim

Eliminar los espacios

split

Convierte el string en un  
array

includes / match

Evaluar un substring

# String

toLowerCase

Minúsculas

toUpperCase

Mayúsculas

repeat

Repetir n veces

startsWith / endsWith

Evalua un substring al inicio o  
al final , devuelve true

# Array

filter

Retorna el array

find / findLast

Retorna un elemento

findIndex / findLastIndex

Retorna un entero / índice

includes

Retorna true/false

some

Retorna true/false

map

Crea un nuevo array  
modificado

reduce

Ejecuta una funcion y retorna  
un único valor

splice

Eliminar o agregar elementos

# Array

## push

Agregar un elemento al final  
Modifica el array original

## unshift

Agregar un elemento al inicio  
Modifica el array original

## pop

Elimina el último elemento de  
un array  
Modifica el array original

## sort

Ordenar el array

## reverse

Revertir el array

## forEach

Ejecuta una función una vez  
por cada elemento  
No modifica el array



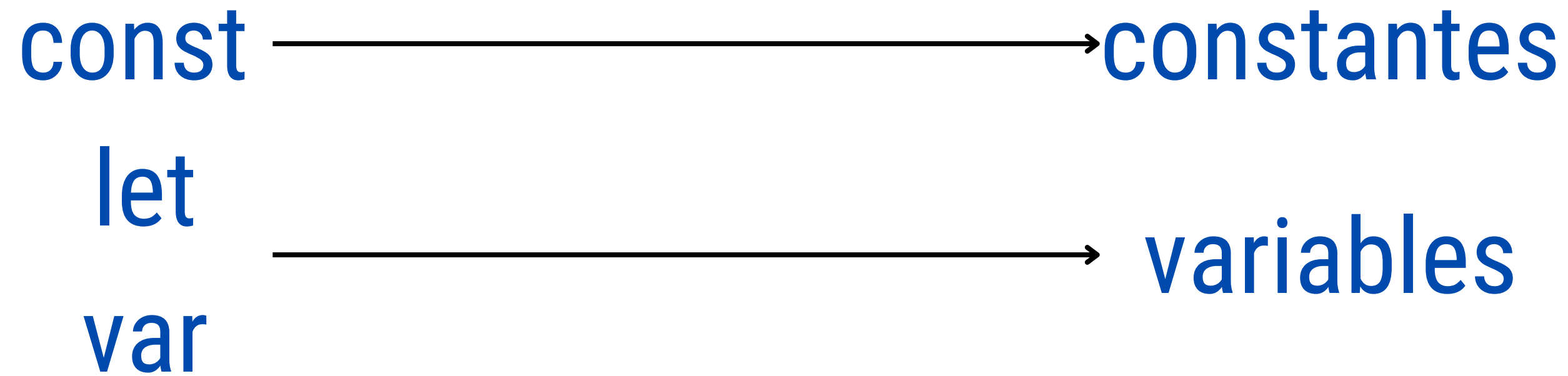
# Array / String

concat() slice() includes() indexOf()

```
// Definimos un array
const array: number[] = [];
// []
array.push(1);
// [1]

// Definimos un string
const str = "Hola mundo!";
// 'Hola mundo!'
const substr=str.substring(0, 4)
// 'Hola'
// str => 'Hola mundo!'
```

# Declaraciones

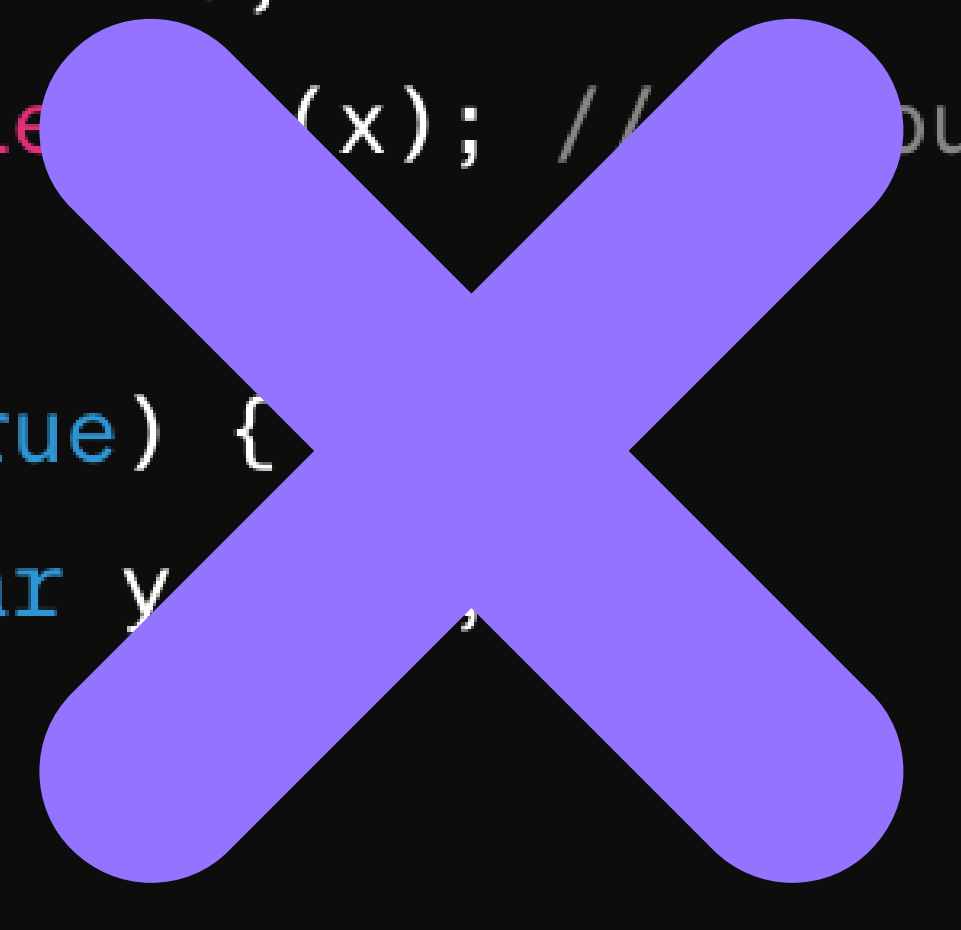


# var / let

```
var x = 10;  
console.log(x); // Output: 10  
  
if (true) {  
    var y = 20;  
}  
  
console.log(y); // Output: 20
```

```
let a = 30;  
console.log(a); // Output: 30  
  
if (true) {  
    let b = 40;  
    console.log(b); // Output: 40  
}  
  
console.log(b); // Error: b is not defined
```

# var / let



```
var x = 10;  
console.log(x); // Output: 10  
  
if (true) {  
    var y = 20;  
}  
  
console.log(y); // Output: 20
```

```
let a = 30;  
console.log(a); // Output: 30  
  
if (true) {  
    let b = 40;  
    console.log(b); // Output: 40  
}  
  
console.log(b); // Error: b is not defined
```

# const

```
const PI = 3.14;  
console.log(PI); // Output: 3.14  
  
PI = 3.14159; // Error: Assignment to constant variable.  
  
const arr = [1, 2, 3];  
arr.push(4);  
console.log(arr); // Output: [1, 2, 3, 4]
```

# Typescript

- Tipado estático
- Clases
- Interfaces
- Enumeraciones
- Facilita la programación orientada a objetos
- Decoradores

# Detectar errores en el proceso de compilación

```
function sum(num1,num2) {  
  return num1 + num2  
}  
  
console.log(sum(1, 2)); 3  
  
console.log(sum(1, "Hola")); '1Hola'
```

```
function sum(num1:number, num2:number):number {  
  return num1 + num2;  
}
```

```
console.log(sum(1, 2)); 3
```

```
console.log(sum(1, "Hola")); '1Hola'
```

No se puede asignar un argumento de tipo "string" al parámetro de tipo "number". ts(2345)



# Ejercicios Prácticos

Documentación:  
<https://developer.mozilla.org/>



Node.js — Download  
Node.js®

### **Node.js — Download Node.js®**

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

 [nodejs.org](https://nodejs.org/)

<https://nodejs.org/>



<https://acesse.one/desarrollo-frvm>