## Objetivos del Taller

El objetivo de este Taller es entregar a los desarrolladores las herramientas y conocimientos necesarios para dominar TypeScript, permitiéndoles crear aplicaciones modernas, eficientes y de fácil mantenimiento. A lo largo del Taller, los participantes aprenderán a utilizar TypeScript para potenciar el desarrollo con un sistema de tipado estático, programación orientada a objetos y buenas prácticas de codificación.



El enfoque estará en enseñar cómo integrar TypeScript en aplicaciones reales, tanto en el desarrollo frontend con frameworks como React y Vue 3, como en el backend utilizando Node.js. Los participantes explorarán cómo mejorar la calidad del código, reducir errores en tiempo de ejecución y facilitar la colaboración en equipos gracias al tipado explícito y a herramientas modernas.

Además, el Taller cubrirá aspectos avanzados como interfaces, genéricos, modularización y decoradores, que permiten resolver problemas comunes de desarrollo y construir soluciones escalables. Se realizarán prácticas orientadas a la creación de APIs RESTful, sistemas modulares y proyectos full-stack, utilizando las características más relevantes del lenguaje.

Con un enfoque altamente práctico y progresivo, los participantes no solo adquirirán habilidades técnicas, sino también la capacidad de aplicar TypeScript en proyectos reales, enfrentando los desafíos del desarrollo moderno con confianza. Al finalizar el Taller, los participantes podrán desarrollar aplicaciones robustas, adaptables y con una arquitectura de código que favorezca el crecimiento y la sostenibilidad del proyecto.

## Contenidos



**Módulo 1: Introducción a TypeScript**
- ¿Qué es TypeScript y en qué se diferencia de JavaScript?
- Instalación y configuración del entorno de desarrollo.
- Uso del compilador tsc y archivo de configuración tsconfig.json.
- Primeros pasos: transpilar código TypeScript a JavaScript.

**Módulo 2: Tipos Básicos**
- Tipos primitivos: string, number, boolean.
- Arrays, tuplas y enums.
- Tipos especiales: any, unknown, void, null y undefined.
- Inferencia de tipos y tipado explícito.

**Módulo 3: Funciones y Objetos**
- Declaración de funciones y tipos de retorno.
- Parámetros obligatorios, opcionales y con valores por defecto.
- Sobrecarga de funciones.
- Tipado de objetos y estructura de datos.

**Módulo 4: Objetos y Tipos Personalizados en TypeScript**
- Creación de interfaces y type aliases.
- Uso de tipos compuestos: unión e intersección.
- Interfaces con propiedades opcionales y de solo lectura.
- Ejemplos prácticos: Definición de modelos de datos.

**Módulo 5: Depuración de Errores y el Archivo tsconfig.json**
- Configuración avanzada con tsconfig.json.
- Opciones comunes: strict, target, module y rootDir.
- Herramientas de depuración en editores como VSCode.
- Identificación y solución de errores en tiempo de compilación.

**Módulo 6: Características de ES6 y JavaScript2015 Disponibles en TypeScript**
- Declaración de variables con let y const.
- Funciones de flecha (arrow functions).
- Desestructuración de objetos y arreglos.
- Plantillas de cadena (template strings).
- Promesas y async/await.

**Módulo 7: Clases en TypeScript**
- Creación de clases y objetos.
- Constructores, propiedades y métodos.
- Modificadores de acceso: public, private, protected.
- Herencia, clases abstractas y polimorfismo.
- Getters y setters.

**Módulo 8: Interfaces**
- Declaración y uso de interfaces.
- Extensión de interfaces (extends).
- Interfaces en clases.
- Ejercicios prácticos: Definición de contratos y validaciones de objetos.

**Módulo 9: Namespaces**

- **¿Qué son los namespaces y para qué se usan?**
- **Creación de namespaces.**
- **Modularización del código.**
- **Ejemplo práctico: Organización de librerías internas.**

**Módulo 10: Genéricos (Generics)**

- **Introducción a genéricos en funciones, clases e interfaces.**
- **Ventajas del uso de <T> y tipado dinámico.**
- **Restricciones y parámetros de tipo.**
- **Ejemplo práctico: Creación de estructuras de datos reutilizables.**

**Módulo 11: Decoradores**

- **Introducción a los decoradores y su propósito.**
- **Decoradores de clase, método y propiedad.**
- **Configuración necesaria en tsconfig.json.**
- **Ejercicio práctico: Implementación de decoradores personalizados.**



Las Bellotas 199 Oficina 62, Providencia, Santiago
+569 8800 3239 | Web: www.xpertis.cl

## Metodología

- **70% práctico y 30% teórico.**
- Ejercicios reales con soluciones aplicadas a proyectos modernos.
- Desarrollo de una aplicación final que integre los conceptos aprendidos.
- Acompañamiento en la resolución de dudas y problemas.

## Estrategia Metodológica

- Proyectos Modulares y Progresivos
- Resolución de Problemas Reales
- Trabajo Colaborativo y Discusión Técnica
- Revisión y Optimización del Código
- Simulación de Escenarios en Producción
- Retroalimentación Constante

## Dirigido a

Desarrolladores de aplicaciones

## Requisitos participantes

Conocimientos en JavaScript o en algún lenguaje de programación.

## Generalidades

- Duración 36 horas cronológicas
- Taller cerrado.