TypeScript - Interfaces

Objetivos: Demostrar la tipificación estructural (duck typing), Crear una interfaz e implementarla en una clase, Diferenciar los alias de tipo de las interfaces.

Ejercicio 2.1

Dado el siguiente codigo:

```
function agregarAlCarro(item: { id: number, titulo: string, idVariante: number }) {
  console.log('[Ejercicio 2.1]', `Agregando "${item.titulo}" al carro de compras.`);
}

agregarAlCarro({ id: 1, titulo: 'Zapatos de cuero' });
```

1 Crea una interfaz ‘CartItem‘ y reemplaza el tipo de parametros con ella

2 Hacer idVariante opcional

Ejercicio 2.2

Dado el siguiente codigo:

```
class Persona {
  constructor(public nombre: string, public edad: number) { }
}

const juan = new Persona('Juan', 31);

console.log('[Ejercicio 2.2]', `${juan.nombre} tiene ${juan.edad} anios.`);
```

1 Cree e implemente una interfaz en ‘Persona‘ para asegurarse de que siempre tenga acceso a las propiedades miembros ‘nombre‘ y ‘edad‘

Ejercicio 2.3

Dado el siguiente codigo:

```
// [no editar] (pretender que esto proviene de una version externa de la
// biblioteca `foo.d.ts`)
interface Ciudad {
  nombre: string;
}
// [/no editar]

const montreal = {
  coords: {
```



```
    latitud: 42.332,
    longitud: -73.324,
  },
  nombre: 'Montreal',
};

const tampa = {
  coords: {
    latitud: '27.9478',
    longitud: '-82.4584',
  },
  nombre: 'Tampa',
};

function informacionCiudad(ciudad: Ciudad) {
  const coords =
    `(${ciudad.coords.latitud.toFixed(3)}, ${ciudad.coords.longitud.toFixed(3)})`;
  return `${ciudad.nombre.toUpperCase()} se encuentra en ${coords}.`;
}

console.log('[Ejercicio 2.3]',
  `${informacionCiudad(montreal)} \n\n ${informacionCiudad(tampa)}`);
```

1 Cree una interfaz 'Coords' que tenga las propiedades numéricas 'latitud' y 'longitud'

2 Extienda la interfaz existente 'Ciudad' (sin modificarla en línea) agregando una propiedad 'coords' de tipo 'Coords'

3 Corregir lo que está mal con 'tampa'

Ejercicio 2.4

El propósito de este ejercicio es simplemente ilustrar el uso de 'readonly':

```
interface EsquemaUsuario {
  readonly id: number;
  nombre: string;
}

class Usuario implements EsquemaUsuario {
  constructor(public nombre: string, readonly id: number) { }
}

const usuario = new Usuario('Perro', 1);

console.log(usuario.id); // legible

usuario.nombre = 'Harold'; // asignable
usuario.id = 5; // no asignable

console.log('[Ejercicio 2.4]', `Usuario:`, usuario)
```

