```ts
import { of, fromEvent, interval, forkJoin, combineLatest } from 'rxjs';
import { map, filter, switchMap, mergeMap, concatMap, exhaustMap, debounceTime, distinctUntilChar
import { ajax } from 'rxjs/ajax';

/* ----------------------------------------------------
   1️⃣ map() - Transformar datos
   📌 Ejemplo real: Formatear el precio antes de mostrarlo
---------------------------------------------------- */
of(100, 200, 300)
  .pipe(map(price => `$${price.toFixed(2)}`))
  .subscribe(console.log);
// Salida: "$100.00", "$200.00", "$300.00"

/* ----------------------------------------------------
   2️⃣ filter() - Filtrar valores
   📌 Ejemplo real: Mostrar solo productos con stock
---------------------------------------------------- */
of({ name: 'Laptop', stock: 5 }, { name: 'Mouse', stock: 0 })
  .pipe(filter(product => product.stock > 0))
  .subscribe(console.log);
// Salida: { name: 'Laptop', stock: 5 }

/* ----------------------------------------------------
   3️⃣ switchMap() - Cancelar y reemplazar peticiones
   📌 Ejemplo real: Búsqueda en tiempo real con autocomplete
---------------------------------------------------- */
fromEvent(document.getElementById('searchBox')!, 'input')
  .pipe(
    map((e: any) => e.target.value),
    debounceTime(300),
    distinctUntilChanged(),
    switchMap(term => ajax.getJSON(`/api/products?q=${term}`))
  )
  .subscribe(data => console.log('Resultados:', data));
```

Copiar Edita

```
/* ------------------------------------------------
   4️⃣ mergeMap() - Ejecutar en paralelo
   📌 Ejemplo real: Procesar múltiples pedidos a la vez
------------------------------------------------ */
of(1, 2, 3)
  .pipe(mergeMap(orderId => ajax.getJSON(`/api/orders/${orderId}`)))
  .subscribe(console.log);

/* ------------------------------------------------
   5️⃣ concatMap() - Mantener el orden de ejecución
   📌 Ejemplo real: Guardar pasos de un wizard en secuencia
------------------------------------------------ */
of('paso1', 'paso2', 'paso3')
  .pipe(concatMap(step => ajax.post(`/api/saveStep`, { step })))
  .subscribe(console.log);

/* ------------------------------------------------
   6️⃣ exhaustMap() - Ignorar eventos mientras uno está activo
   📌 Ejemplo real: Prevenir doble envío de formulario
------------------------------------------------ */
fromEvent(document.getElementById('btnSubmit')!, 'click')
  .pipe(
    exhaustMap(() => ajax.post('/api/checkout', { cartId: 123 }))
  )
  .subscribe(console.log);

/* ------------------------------------------------
   7️⃣ forkJoin() - Ejecutar en paralelo y esperar todos
   📌 Ejemplo real: Cargar datos iniciales de varias APIs
------------------------------------------------ */
forkJoin({
  user: ajax.getJSON('/api/user/1'),
  cart: ajax.getJSON('/api/cart/1'),
  recommendations: ajax.getJSON('/api/recommendations')
}).subscribe(console.log);
```

```
/* ------------------------------------------------------
   8️⃣ combineLatest() - Combinar valores más recientes
   📌 Ejemplo real: Filtrar productos según rango de precio y categoría
------------------------------------------------------ */
const priceRange$ = of([10, 100]);
const category$ = of('Electrónica');
combineLatest([priceRange$, category$])
  .subscribe(([range, cat]) => console.log(`Filtrando ${cat} entre ${range[0]} y ${range[1]}`));

/* ------------------------------------------------------
   9️⃣ take() - Tomar solo X valores
   📌 Ejemplo real: Mostrar solo los primeros 3 banners
------------------------------------------------------ */
interval(1000)
  .pipe(take(3))
  .subscribe(console.log); // 0,1,2

/* ------------------------------------------------------
   🔟 catchError() - Manejar errores sin romper el stream
   📌 Ejemplo real: Mostrar mensaje si falla la API
------------------------------------------------------ */
ajax.getJSON('/api/fail')
  .pipe(
    catchError(err => {
      console.error('Error:', err);
      return of({ error: true, message: 'No se pudo cargar la data' });
    })
  )
  .subscribe(console.log);
```