

Prueba Técnica AXPE - Parte teórica

Actualmente veo un problema de es la legibilidad/mantenibilidad del código en caso de que la lista de servicios aumente en un futuro. Siendo esta la estructura actual:

```
getTotal () {  
  let total = 0;  
  
  this.services.forEach(service, index => {  
    let multimediaContent = service.getMultimediaContent ();  
  
    if (typeof service == StreamingService) {  
      total += multimediaContent.streamingPrice;  
    } else if (typeof service == DownloadService) {  
      total += multimediaContent.downloadPrice;  
    }  
  
    if (typeof multimediaContent == PremiumContent) {  
      total += multimediaContent.additionalFee;  
    }  
  })  
  
  return total;  
}
```

Con el aumento de servicios puede llevar a tener una estructura tipo:

```
if ():  
  else if ():  
  else if ():  
  else if ():  
  else if ():  
  else if ():  
  ...
```

Esto programáticamente puede llevar a introducir errores y en el caso en que esto ocurra, el resultado de getTotal() sería 0, con el riesgo para la plataforma que esto podría generar.

Si hiciéramos una estructura que mapease el tipo de servicio al método que devuelve el precio, podríamos usar un ErrorHandler, de tal manera que cada vez que se introdujese un error, la estructura de cálculo de precio total seguiría en pie, tan solo el servicio fallido quedaría excluido. En el ErrorHandler definiríamos que tipo de respuesta esperamos frente a ese error. Ejemplo muy simplificado:

```
SERVICE_PRICE_GETTER_MAP = {  
  StreamingService: MultimediaContent.streamingPrice,  
  DownloadService: MultimediaContent.downloadPrice,  
  ...  
}
```

```
if typeof service not in SERVICE_PRICE_GETTER_MAP:  
  ErrorHandler  
else:  
  total += SERVICE_PRICE_GETTER_MAP(typeof service)()
```