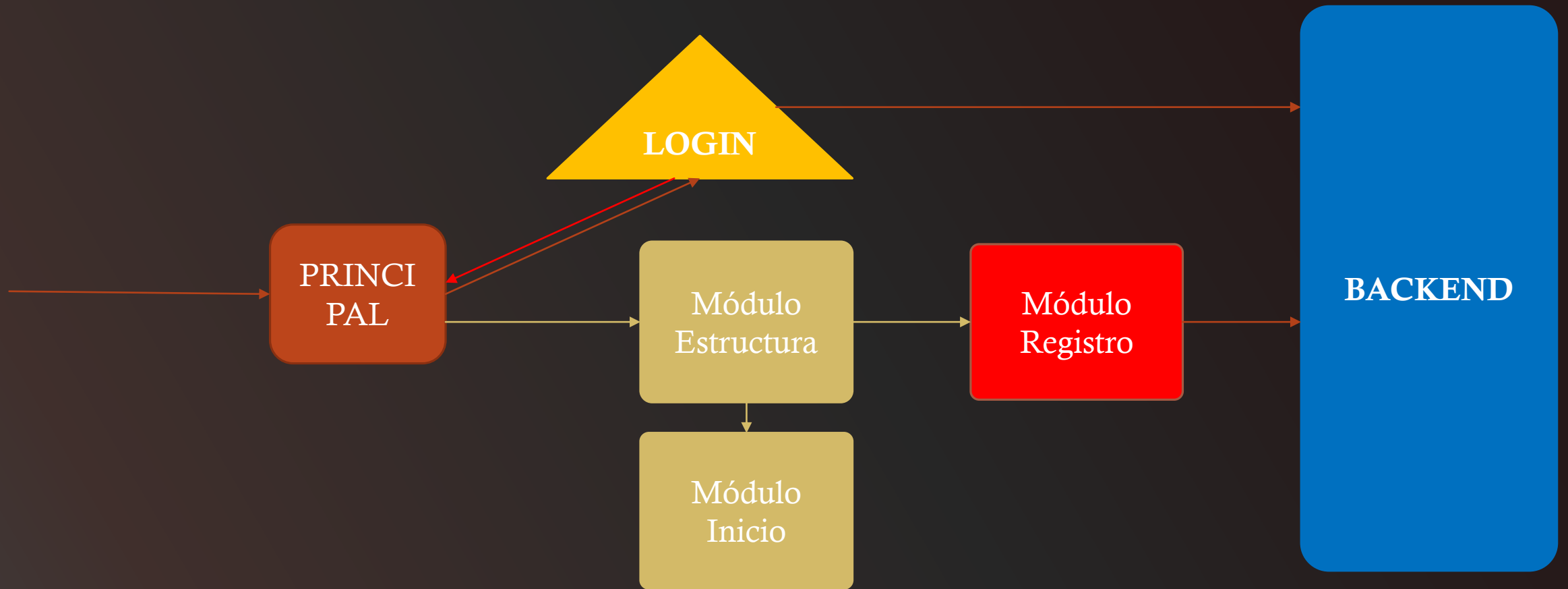




MÓDULO DE REGISTRO

ARQUITECTO DESARROLLADOR PLATAFORMA WEB NET CORE



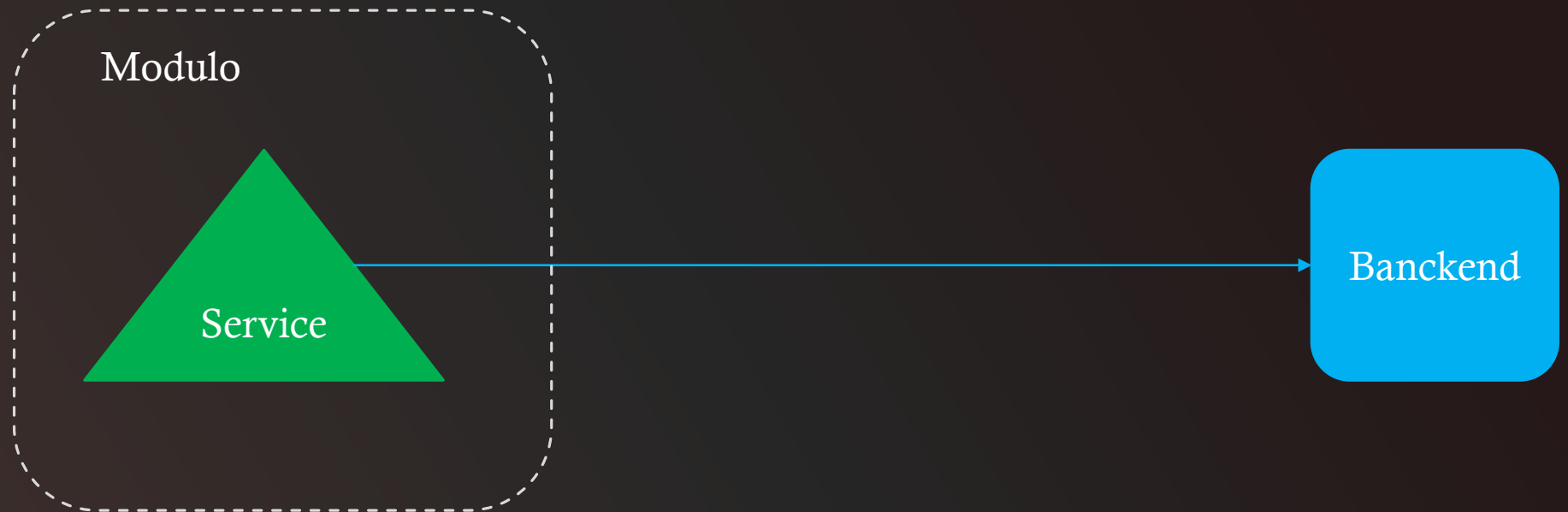


Objeto Transaccion.ts

```
export class Transaccion {  
  
    constructor(  
        public codigo: string,  
        public descripcion: string  
    ) {}  
}
```

RegistroService

SERVICE: AGREGANDO MÉTODO PARA INTERACTUAR CON EL SERVICE



AGREGANDO ATRIBUTOS E INSTANCIAS DE INTERACCIÓN A TRAVÉS DEL PROTOCOLO HTTP

```
@Injectable({ providedIn: 'root' })
```

```
export class ServicioService {
```

```
    public servicioUrl = 'http://localhost:8080/demo'; // URL to web API
```

```
    public servicios: Servicio[] = [];
```

```
    public servicio: Servicio;
```

```
    public errorMessage: string;
```

```
    constructor (private http: HttpClient) {}
```

```
}
```

SERVICE: AGREGANDO CONTROL DE EXCEPCIONES

```
@Injectable({ providedIn: 'root' })
```

```
export class ServicioService {
```

```
.....
```

```
    private handleError<T> (operation = 'operation', result?: T) {  
    return (error: any): Observable<T> => {  
        console.error(error); // log to console instead  
    return of(result as T);  
    };  
}
```


SERVICE: AGREGANDO MÉTODO PARA OBTENER LISTA DE DEUDAS

```
@Injectable({ providedIn: 'root' })
```

```
export class ServicioService {
```

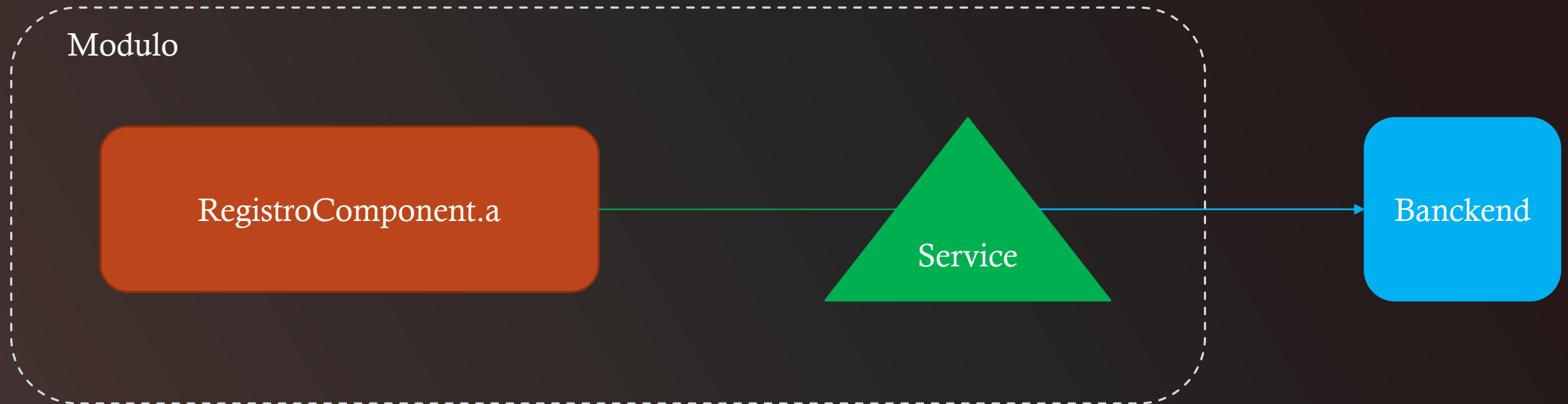
```
.....
```

```
    public getServicios(): Observable<Servicio[]> {  
    return this.http.get<Servicio[]>(this.servicioUrl+"/servicios").pipe(  
        catchError(this.handleError('getServicios', [])) );  
    }
```

```
}
```

RegistrarComponent.ts

SERVICE: AGREGANDO MÉTODO PARA INTERACTUAR CON EL SERVICE



```
import { Component, OnInit } from '@angular/core';
import { Cliente,Producto,Servicio } from './app.servicio';
import { ServicioService } from './servicio.service';
@Component({
  selector: 'app-registro',
  templateUrl: './registro.component.html',
  styleUrls: ['./registro.component.css'],
  providers: [ServicioService]
})
export class RegistroComponent implements OnInit {

  public getServicios(){
    this.service.getServicios().subscribe(
      servicios=> {
        console.log(servicios);
        this.servicios=servicios;
        this.serviciotmp=servicios[0];
      },
      error=> {this.errorMessage=<any>error;}
    );
  }

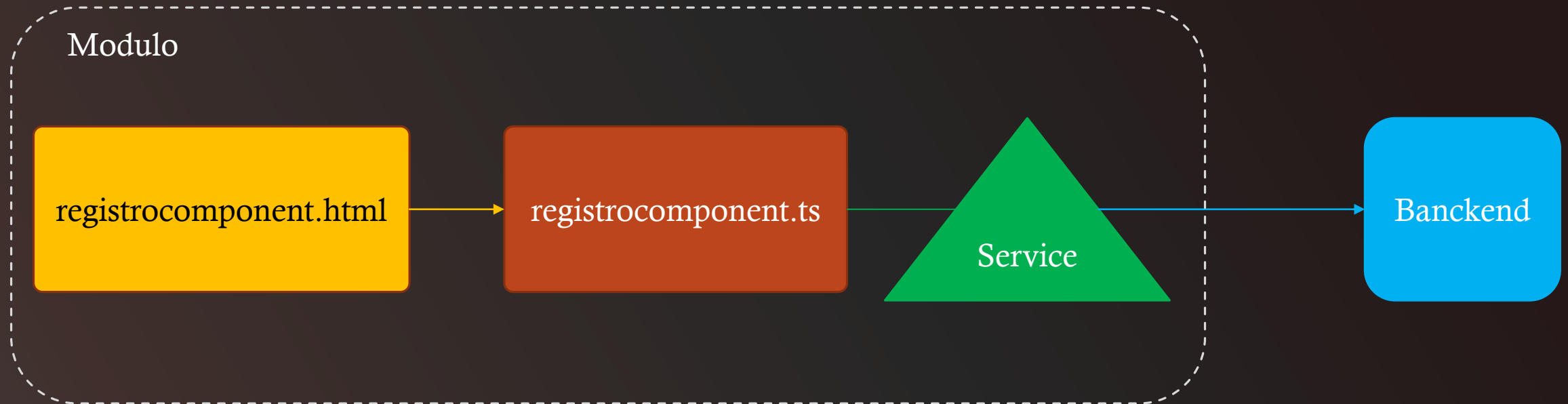
}
```

SERVICE: AGREGANDO EN EL INIT LA OBTENCIÓN DE LA LISTA DE DEUDAS

```
export class RegistroComponent implements OnInit {  
  
  ngOnInit() {  
    this.servicio = new Servicio();  
    this.serviciotmp = new Servicio();  
    this.servicio.indSave=null;  
    this.getServicios();  
  }  
  
}
```

registro.component.html

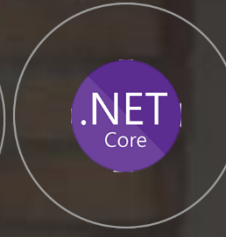
SERVICE: AGREGANDO MÉTODO PARA INTERACTUAR CON EL SERVICE



MODIFICANDO EL HTML PARA RECORRER LA LISTA DE DATOS

```
<tr *ngFor="let p of servicios">
  <td> {{p.codigo}} </td>
  <td> {{p.cliente.nombres}} </td>
  <td> {{p.cliente.codigo}} </td>
  <td> {{p.producto.descripcion}} </td>
  <td> <p *ngIf="p.estado=='0'">PENDIENTE</p><p *ngIf="p.estado=='1'">PAGADO</p></td>
  <td>
    <button type="button" class="btn btn-success custom-width" (click)="update(p)" >Pagar</button>
    <button type="button" class="btn btn-warning custom-width" (click)="delete(p.codigo)" >Eliminar</button>
  </td>
</tr>
```


VALIDAR



CONSULTAS

CONVERTIMOS EL CAFÉ EN CÓDIGO

