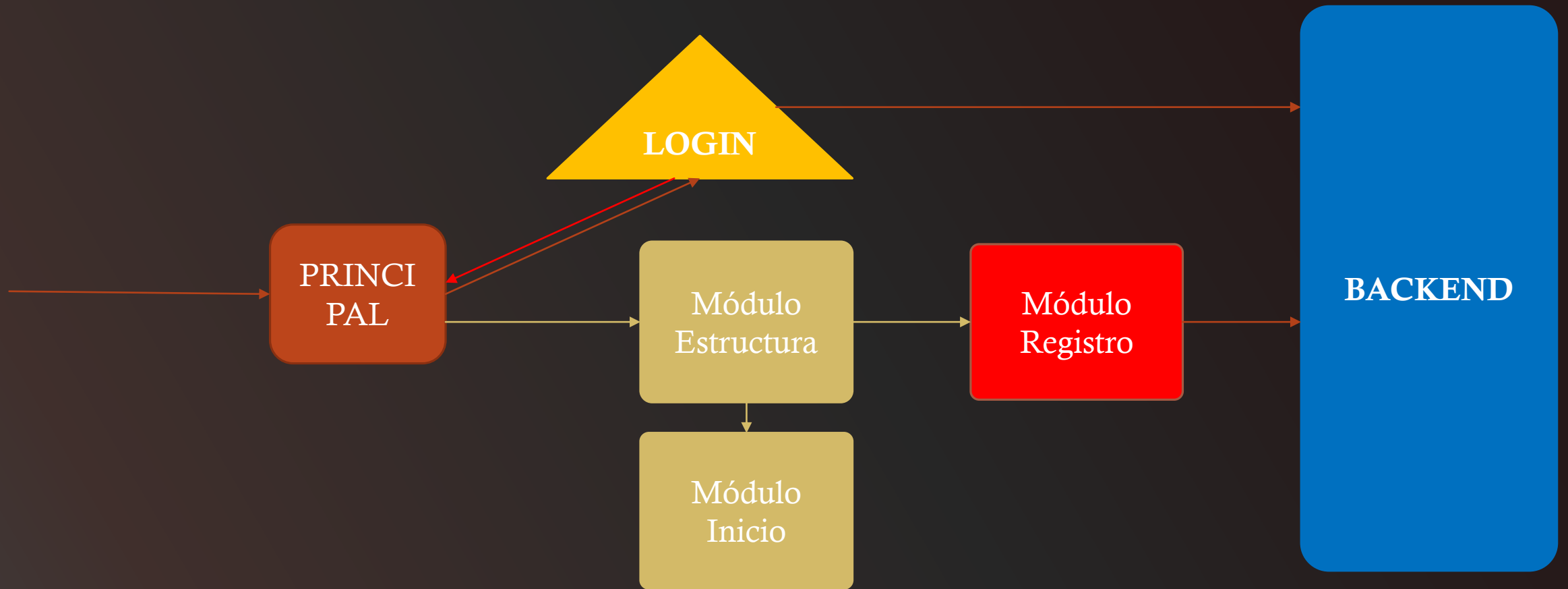




# MÓDULO DE REGISTRO:PAGAR

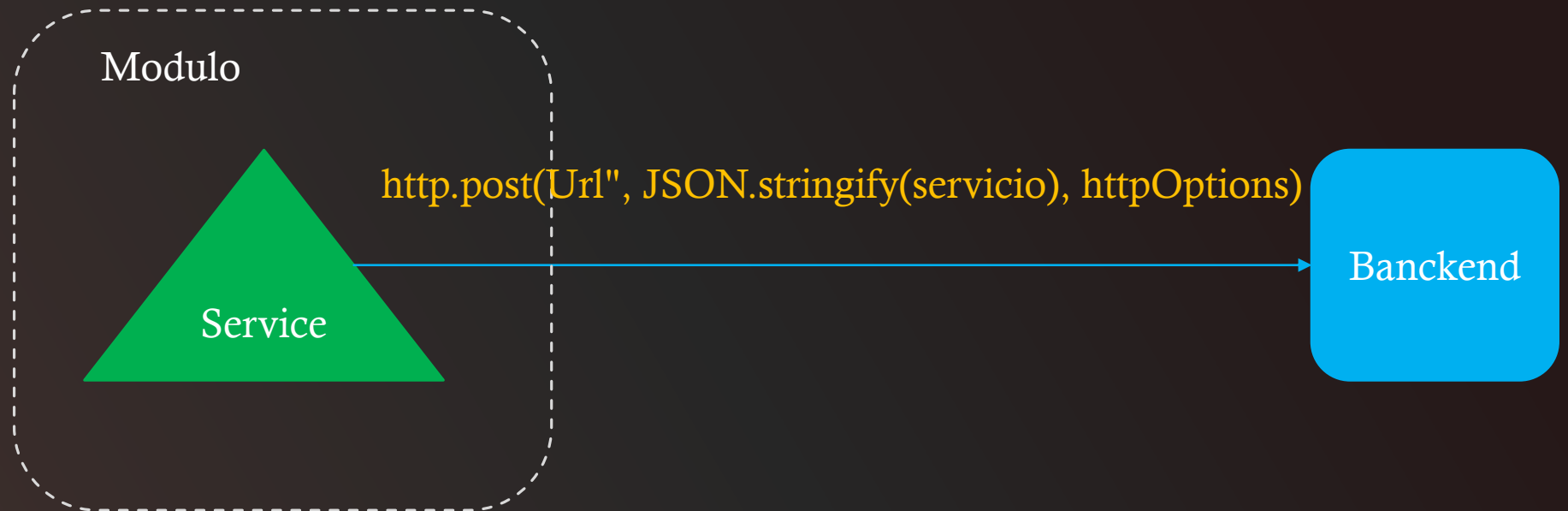
ARQUITECTO DESARROLLADOR PLATAFORMA WEB NET CORE





**RegistroService**

## SERVICE: AGREGANDO MÉTODO PARA INTERACTUAR CON EL SERVICE



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

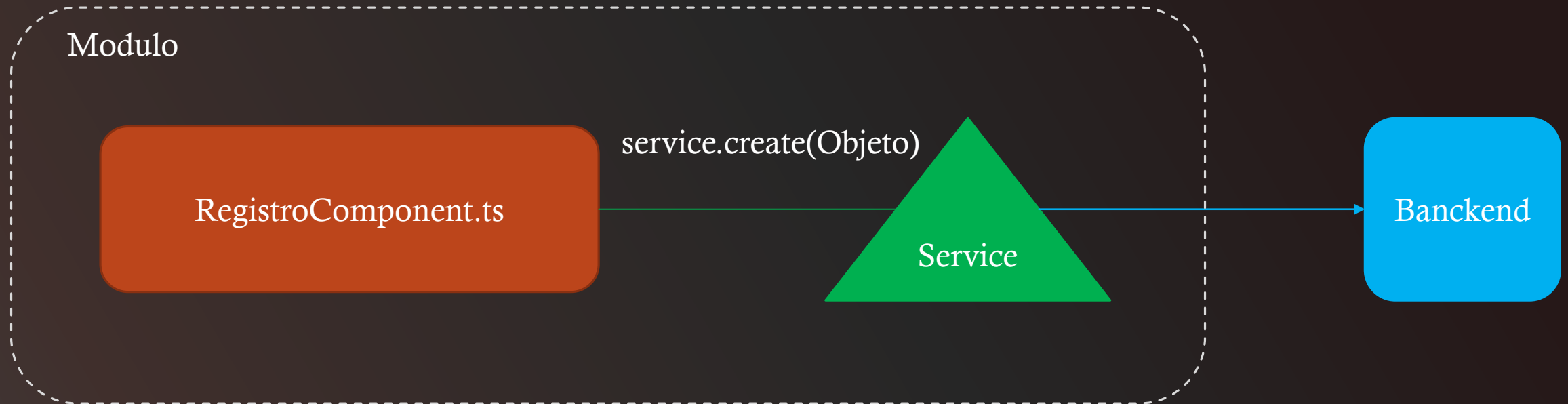
```
export class ServicioService {
```

```
....
```

```
    public create(servicio: Servicio): Observable<Transaccion> {  
        return this.http.post<Transaccion>(this.servicioUrl+"/servicio/create",  
JSON.stringify(servicio), httpOptions).pipe(  
            tap((transaccion: Transaccion) => console.log(transaccion.codigo)),  
  
            catchError(this.handleError<Transaccion>('create'))  
        );  
    }  
}
```

**RegistrarComponent.ts**

## COMPONENT: AGREGANDO MÉTODO PARA INTERACTUAR CON EL SERVICE



```
export class RegistroComponent implements OnInit {
```

```
.....
```

```
public submit() {
```

```
if(this.servicio.indSave===null){
```

```
    this.servicio.cliente = this.serviciotmp.cliente;
```

```
        this.servicio.producto = this.serviciotmp.producto;
```

```
        this.servicio.estado='0';      console.log('Saving Producto', this.servicio);
```

```
            this.create();
```

```
    }else{
```

```
        console.log('Producto update ', this.servicio);
```

```
        //this.updateProducto();
```

```
    }
```

```
    //this.resetear();
```

```
}
```

```
public create(){
```

```
    this.service.create(this.servicio).subscribe(    transaccion=> {
```

```
    console.log(transaccion);                                this.getServicios();                },
```

```
                                error=> {this.errorMessage=<any>error;}                );
```

```
    }
```

```
.....
```

```
}
```



**registro.component.html**

# TEMPLATE

Modulo

registrocomponent.html

create(objeto)

registrocomponent.ts

Service

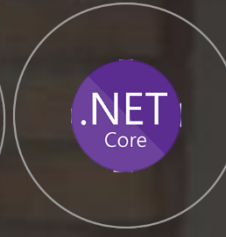
Banckend



**MODIFICANDO EL HTML PARA RECORRER LA LISTA DE DATOS**

USAR EL COMPARTIDO

**VALIDAR**



## CONSULTAS

CONVERTIMOS EL CAFÉ EN CÓDIGO