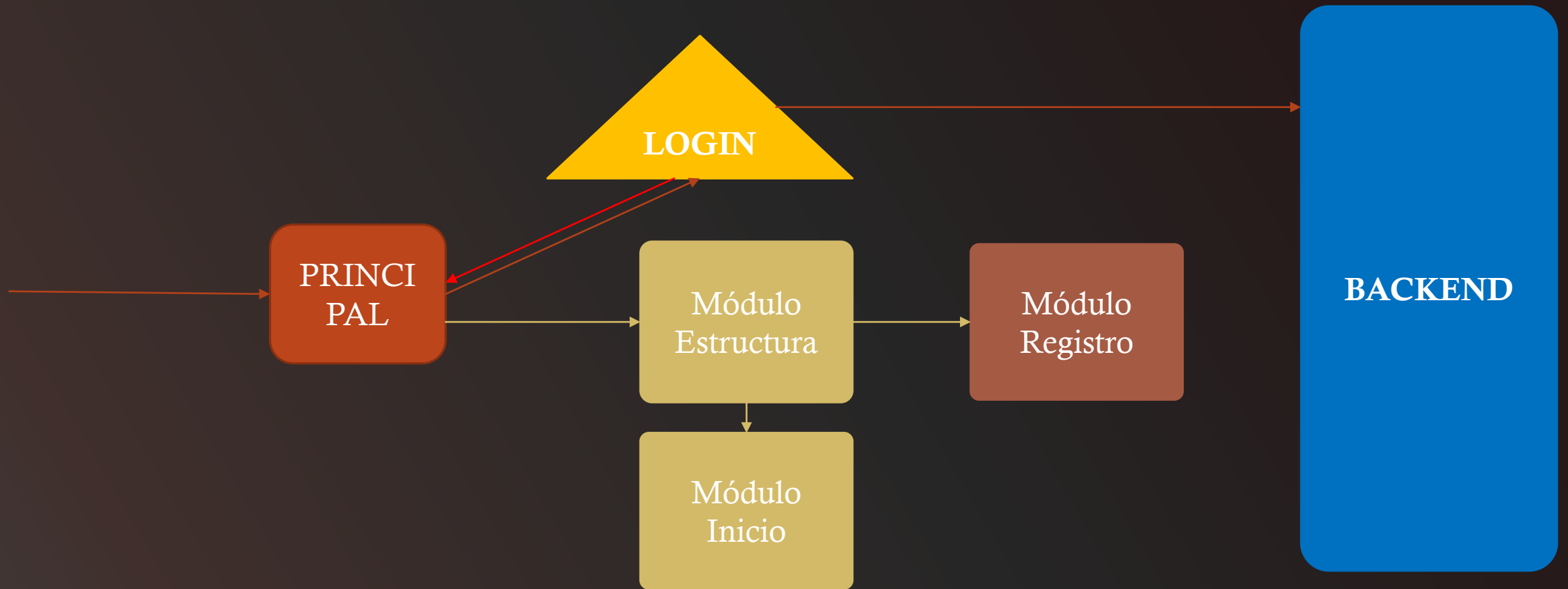




SOFTWARE BASE

ARQUITECTO DESARROLLADOR PLATAFORMA WEB NET CORE





CREANDO EL OBJETO USUARIO

usuario.ts

```
export class Usuario {  
    username: string;  
    password: string;  
    token: string  
    constructor() {}  
}
```

#generando el service para el login

ng generate service login/Login --spec=false

Salida

CREATE src/app/login/login.service.ts (134 bytes)


CREANDO LOGIN SERVICE

```
import { Injectable } from '@angular/core';  
  
@Injectable({  
  providedIn: 'root'  
})  
export class LoginService {  
  
  constructor() { }  
}
```

**IMPORTANDO LIBRERÍAS
A NIVEL DE MODULO PARA TRABAJAR CON
LOS HTTPCLIENT**


```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { AppComponent } from './app.component';
import { AppRoutingModule } from './app.routing.module';
import { HttpClientModule } from '@angular/common/http'
```

```
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule,
    FormsModule
  ],
  providers: [AutenticacionGuard],
  bootstrap: [AppComponent]
})
export class AppModule { }
```



MODIFICANDO LOGIN SERVICE

Importando librerías y componentes

```
import { Injectable } from '@angular/core';
import { HttpHeaders, HttpResponse } from '@angular/common/http';
import { HttpClient } from '@angular/common/http';
import { Observable, of } from 'rxjs';
import { catchError, map, tap } from 'rxjs/operators';

import { Usuario } from '../usuario';

const httpOptions = {
  headers: new HttpHeaders({ 'Content-Type': 'application/json' })
};

@Injectable({providedIn: 'root'})
export class LoginService {

  constructor () {}
}
```

AGREGANDO ATRIBUTOS DE LA CLASE

```
@Injectable({providedIn: 'root'})  
export class LoginService {  
  
    public servicioUrl = 'http://localhost:8080/demo'; // URL to web API  
    public usuario: Usuario;  
  
    constructor () {}  
}
```

MODIFICANDO NUESTRA CLASE PARA USAR LA LIBRERÍA DE INTERACCIÓN A TRAVÉS DEL PROTOCOLO HTTP

```
@Injectable({providedIn: 'root'})
export class LoginService {

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

    constructor (private http: HttpClient) {}

}
```

MODIFICANDO NUESTRA CLASE PARA USAR LA LIBRERÍA DE INTERACCIÓN A TRAVÉS DEL PROTOCOLO HTTP

```
@Injectable({providedIn: 'root'})
export class LoginService {

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

    constructor (private http: HttpClient) {}

}
```

AGREGAR MÉTODOS PARA OBTENER EL USUARIO Y GUARDARLO EN CONTEXTO

```
@Injectable({providedIn: 'root'})
export class LoginService {
  ....
  public getToken(usuario: Usuario):Observable<any> { return this.http.post<any>(this.servicioUrl,
JSON.stringify(usuario), httpOptions).pipe(  map((res: HttpResponse<any>)=>{    console.log("DATA =
"+res.headers.get("Authorization"));    if(res.headers.has("Authorization"))    {      console.log("DATA =
"+res.headers.get("Authorization"));      usuario.token=res.headers.get("Authorization").slice(7);    }
return usuario;    }  ),  catchError(this.handleError)  ) }

  private handleError( error:any){    console.log("createOrder error",error );    return throwError("lanzando erro: " +
error);  }
}
}
```

LoginComponent

MODIFICACIÓN

AGREGANDO EL SERVICE DE LOGIN PARA TRABAJAR SOLO CON EL COMPONENTE

```
import { Component, OnInit } from '@angular/core';
import { Usuario } from './usuario';
import { Router, ActivatedRoute, ParamMap } from '@angular/router';
import { LoginService } from './login.service';
```

```
@Component({
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {
  model: Usuario = new Usuario();
  errorMessage:string;
  loading:boolean = false;
  constructor(private router: Router, private service:LoginService) { }
  ngOnInit() {
  }

}
```

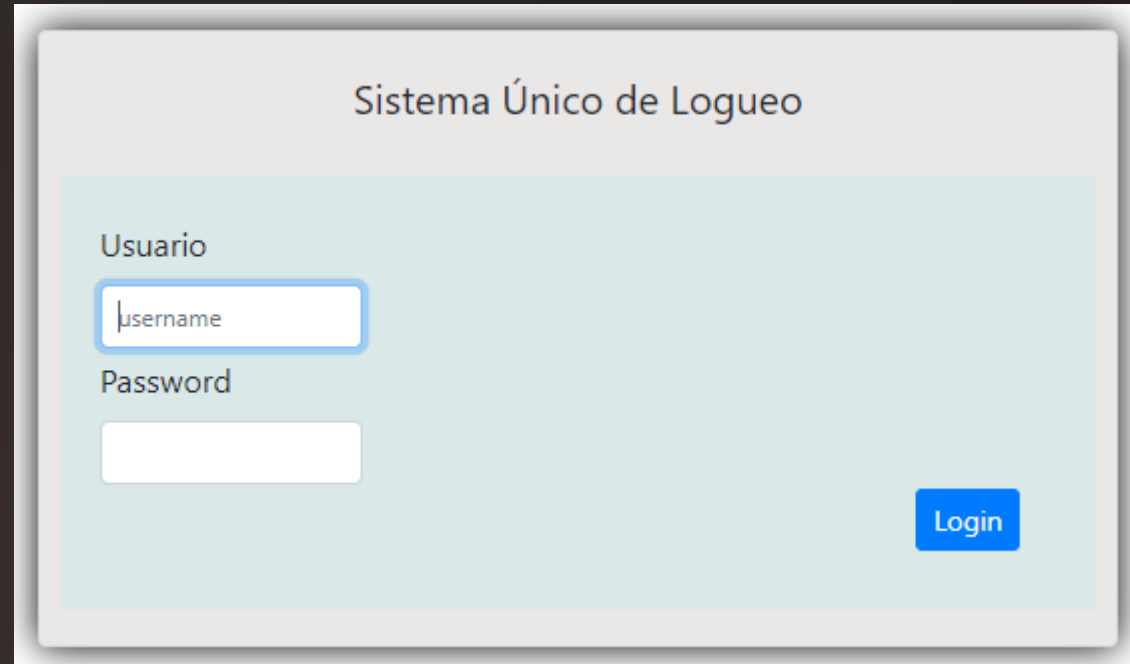
AGREGANDO EL SERVICE DE LOGIN PARA TRABAJAR SOLO CON EL COMPONENTE

```
.....
@Component({
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {
  .....
  ngOnInit() {
  }
  public login(){
this.service.getToken(this.usuario).subscribe(
usuario=>{
    console.log("usuario",usuario);
    sessionStorage.setItem("user",usuario);
    this.redirectLogin();
  },
error=>{
  this.errorMessage=<any>error;    console.log("error login",this.errorMessage);
this.router.navigate(['/mlogin']);
}
);
}
```

logincomponent.html

```
class="form-actions floatRight">
    <button class="btn btn-primary btn-sm" (click)="login()"
[disabled]="loading">Login</button>
</div>
```

AGREGANDO EL SERVICE DE LOGIN PARA TRABAJAR SOLO CON EL COMPONENTE



The image shows a login interface with a light blue background and a white border. At the top, the title "Sistema Único de Logueo" is centered in a dark blue font. Below the title, the label "Usuario" is positioned above a text input field containing the placeholder text "username". The input field has a blue border. Below the username field, the label "Password" is positioned above an empty text input field. In the bottom right corner of the form, there is a blue button with the text "Login" in white.

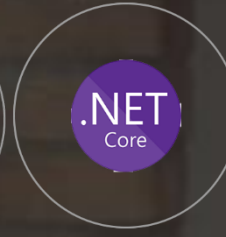
Sistema Único de Logueo

Usuario

username

Password

Login



CONSULTAS

CONVERTIMOS EL CAFÉ EN CÓDIGO