

PRACTICA 2 UNIDAD 3

RECAPTCHA

Imagen de Login con icono ReCaptcha

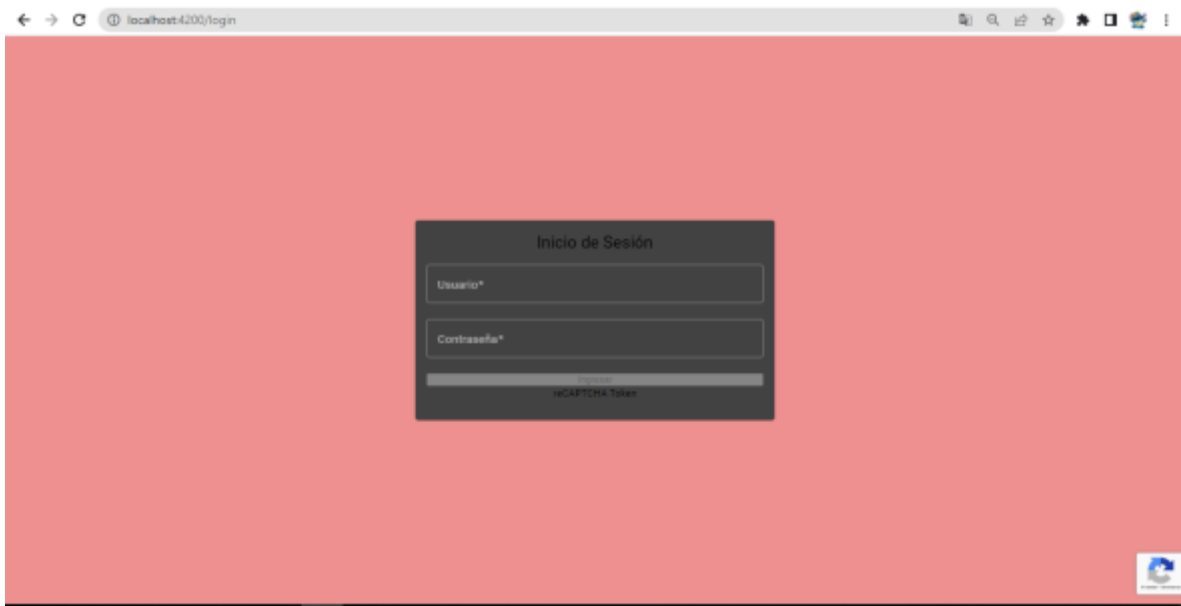
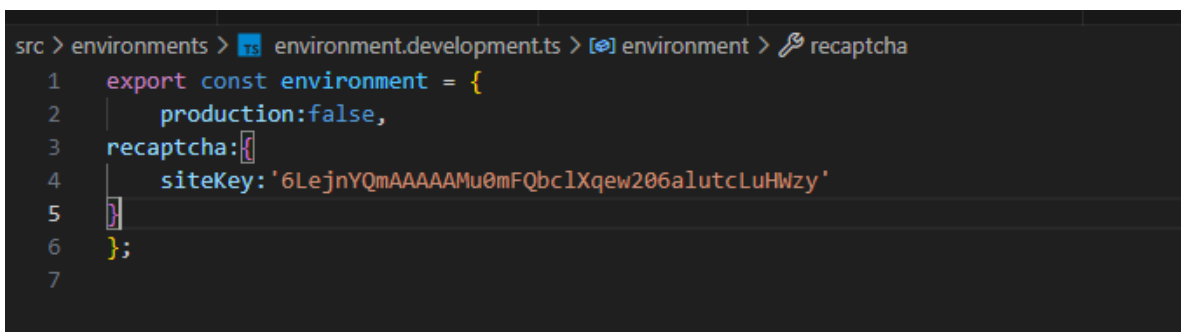
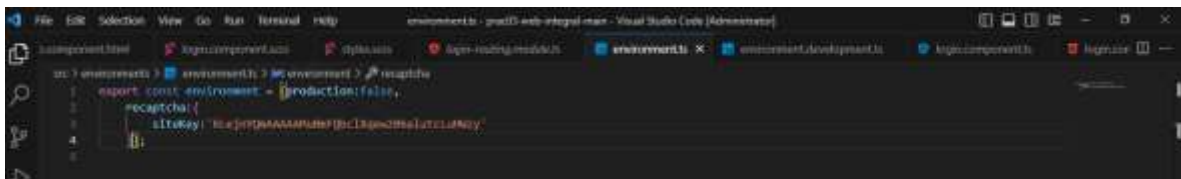


Imagen de código donde se implementa el recaptcha



```

login.component.html | login.component.scss | styles.scss | login-routing.module.ts | environments.ts | envi
src > app > pages > auth > login > login.component.ts > LoginComponent
6 import { ReCaptchaV3Service } from 'ng-recaptcha';
7
8 @Component({
9   selector: 'app-login',
10  templateUrl: './login.component.html',
11  styleUrls: ['./login.component.scss'],
12 })
13 export class LoginComponent implements OnInit {}
14
15 loginForm: any;
16 tokenVisible: boolean = true;
17 reCAPCHAToken: string = "";
18
19
20 constructor(private fb: FormBuilder, public baseForm: BaseForm, private router: Router,
21  private recaptchaV3Service: ReCaptchaV3Service) {}
22
23 ngOnInit(): void {
24   this.loginForm = this.fb.group({
25     username: ['', [Validators.required]],
26     password: ['', [Validators.required, Validators.minLength(8)]],
27   });

```

```

src > app > pages > auth > login > login.component.html > mat-grid-list
17 <mat-label>Contraseña</mat-label>
18 <input [type]="password" formControlName="password" matInput autocomplete="new-password"/>
19 <mat-error *ngIf="baseForm.isValidField(loginForm.get('password'))">
20   {{ baseForm.getErrorMessage(loginForm.get('password')) }}
21 </mat-error>
22 </mat-form-field>
23
24 <button type="submit" mat-raised-button color="primary" [disabled]="!loginForm.valid">Ingresar</button>
25 <div *ngIf="tokenVisible"> reCAPTCHA Token <br>
26   <p style="color: red">{{reCAPCHAToken}}</p>
27 </div>
28 </form>
29 </mat-card-content>
30 </mat-card>
31 </mat-grid-tile>
32 </mat-grid-list>

```

Imagen de token en pantalla Login

