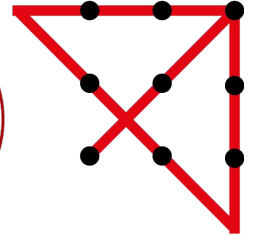
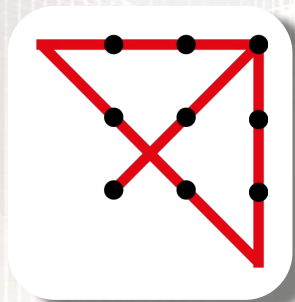


PROGETTO CAPTCHA

A.A. 2022/2023
PRODUCT BASELINE





ZUCCHETTI

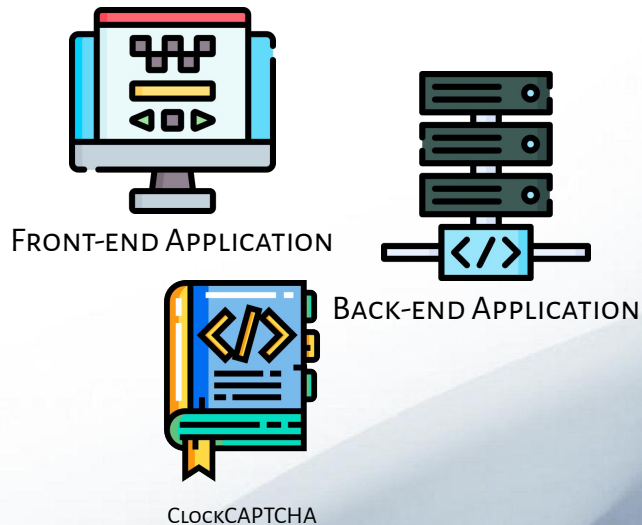
OUT OF BOUNDS

ANGELI JACOPO
BISORTOLE SIMONE
CAPUTO VALENTINA
CAZZARO MICHELE
MATTERAZZO ALBERTO
RETIS EDOARDO

01

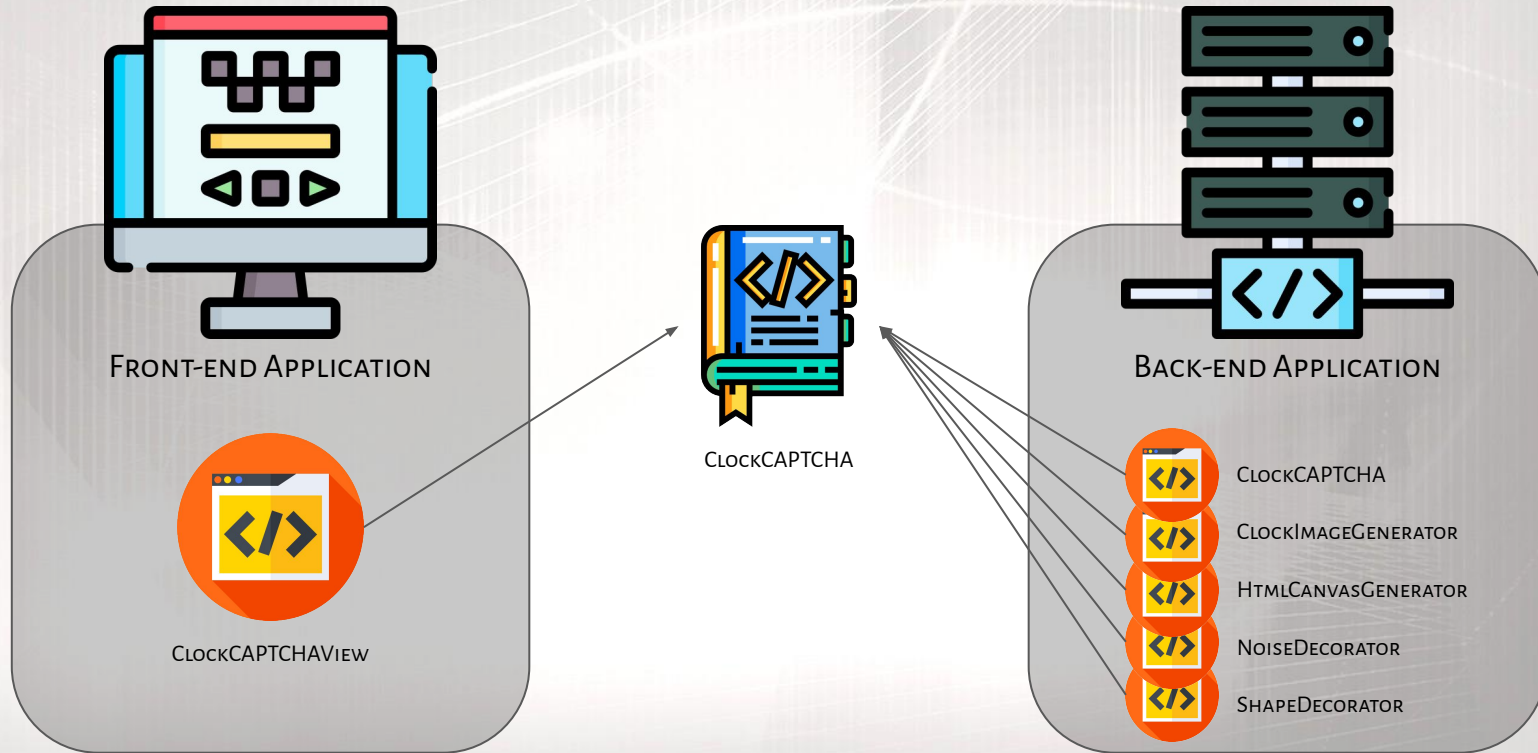
PANORAMICA DEI PRODOTTI

Analisi ad alto livello dei
tre componenti
fondamentali.



PANORAMICA DEI PRODOTTI

PRODOTTI SVILUPPATI



PRINCIPALI TECNOLOGIE UTILIZZATE



FRONT-END APPLICATION



TYPESCRIPT



ANGULAR



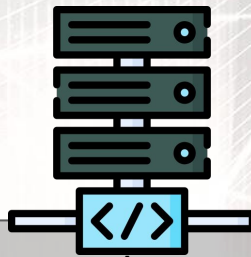
ANGULAR MATERIAL



RxJS



JSON



BACK-END APPLICATION



JAVASCRIPT

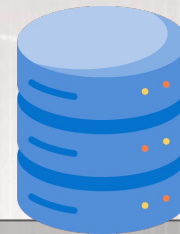


NODEJS

Express JS



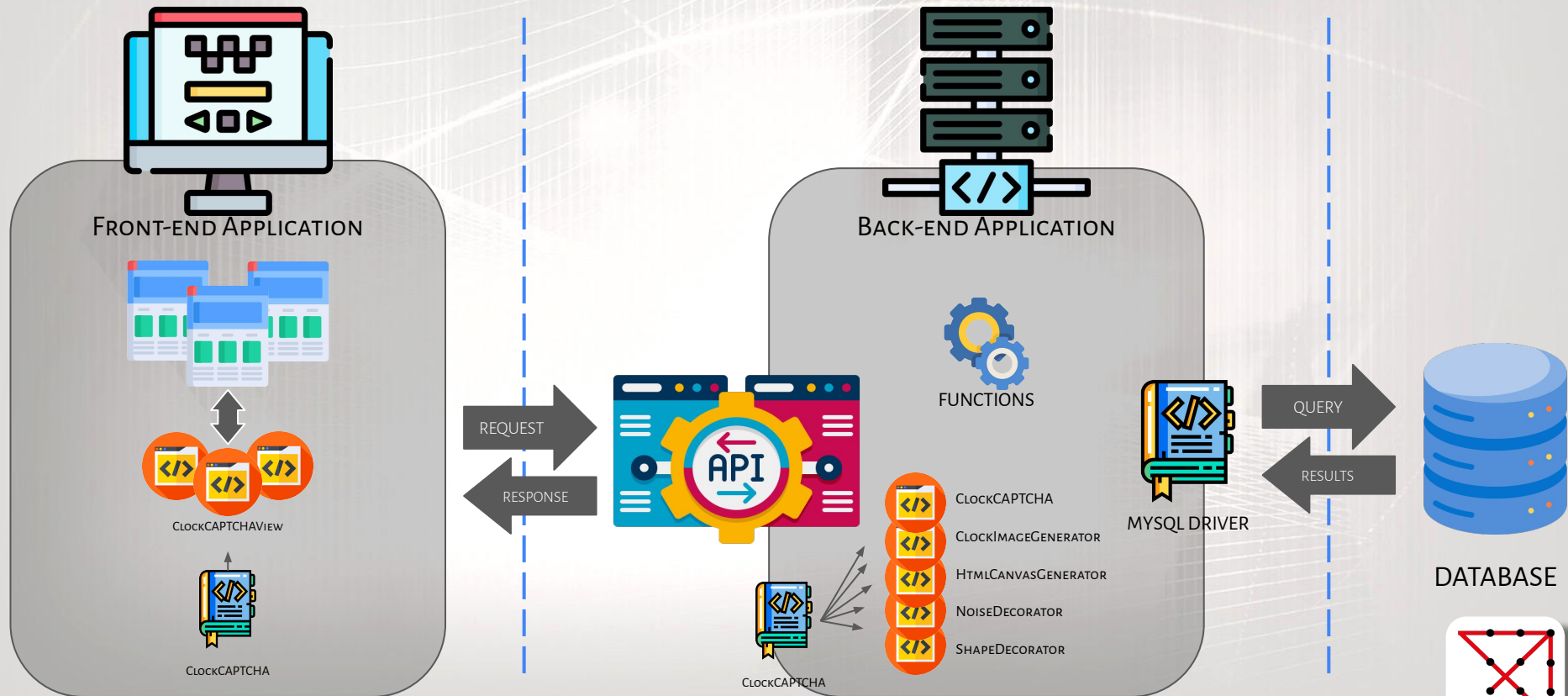
JWT



DBMS



INTERAZIONE TRA I COMPONENTI



02

FRONT-END DELL'APPLICAZIONE

Approfondimento su
applicazione di front-end.



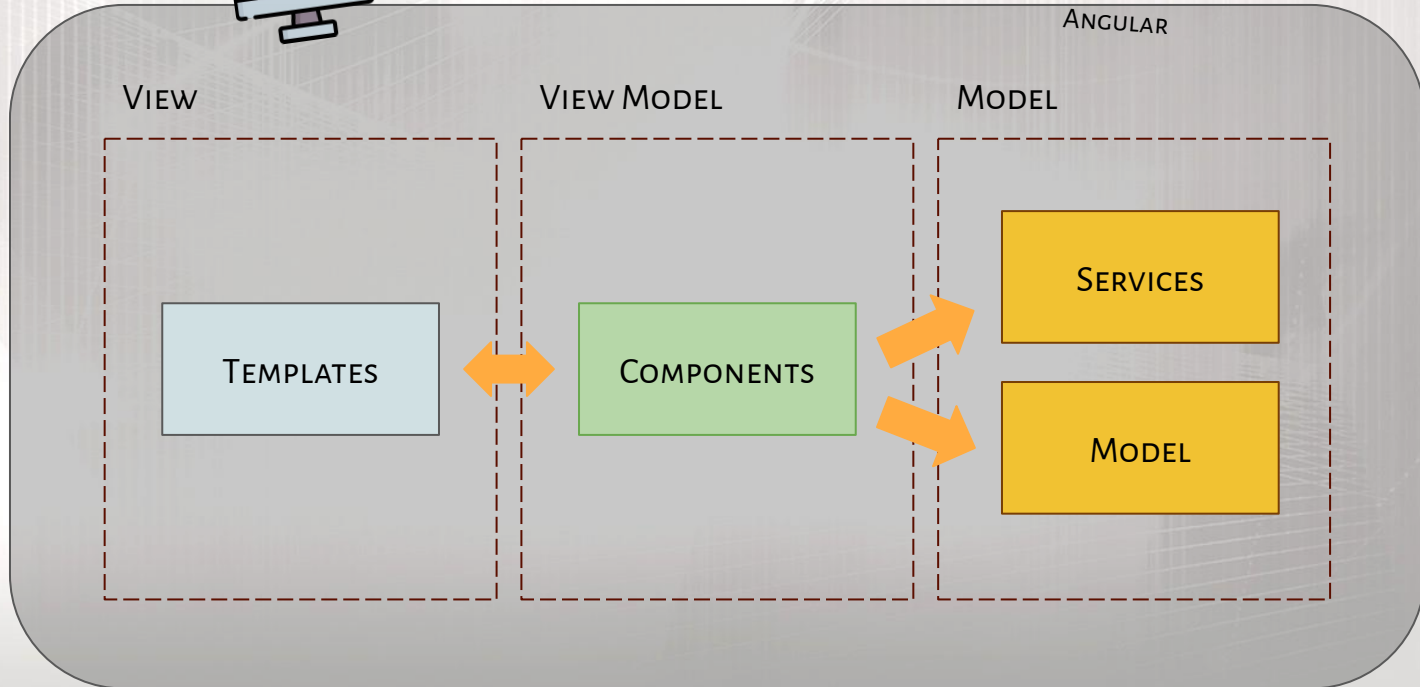
MODELLO ARCHITETTURALE



MODEL-VIEW-VIEW MODEL



ANGULAR





APPLICAZIONE DI FRONT-END

DESIGN PATTERNS



ANGULAR

SINGLETON PATTERN

SERVICES

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

LA DIRETTIVA @Injectable CON PROVIDEDIN
INDICA CHE NELLO SCOPE 'ROOT' QUINDI
L'APPLICAZIONE INTERA SI UTILIZZI UNA SINGOLA
ISTANZA.

DEPENDENCY INJECTION

```
constructor(  
  private _router: Router,  
  private _snackBar: MatSnackBar,  
  private _sessionService: SessionService,  
  private _ccService: ClockCaptchaService  
)
```

FORTEMENTE UTILIZZATO IN ANGULAR

OBSERVER PATTERN



OBSERVABLE



BEHAVIOURSUBJECT



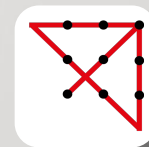
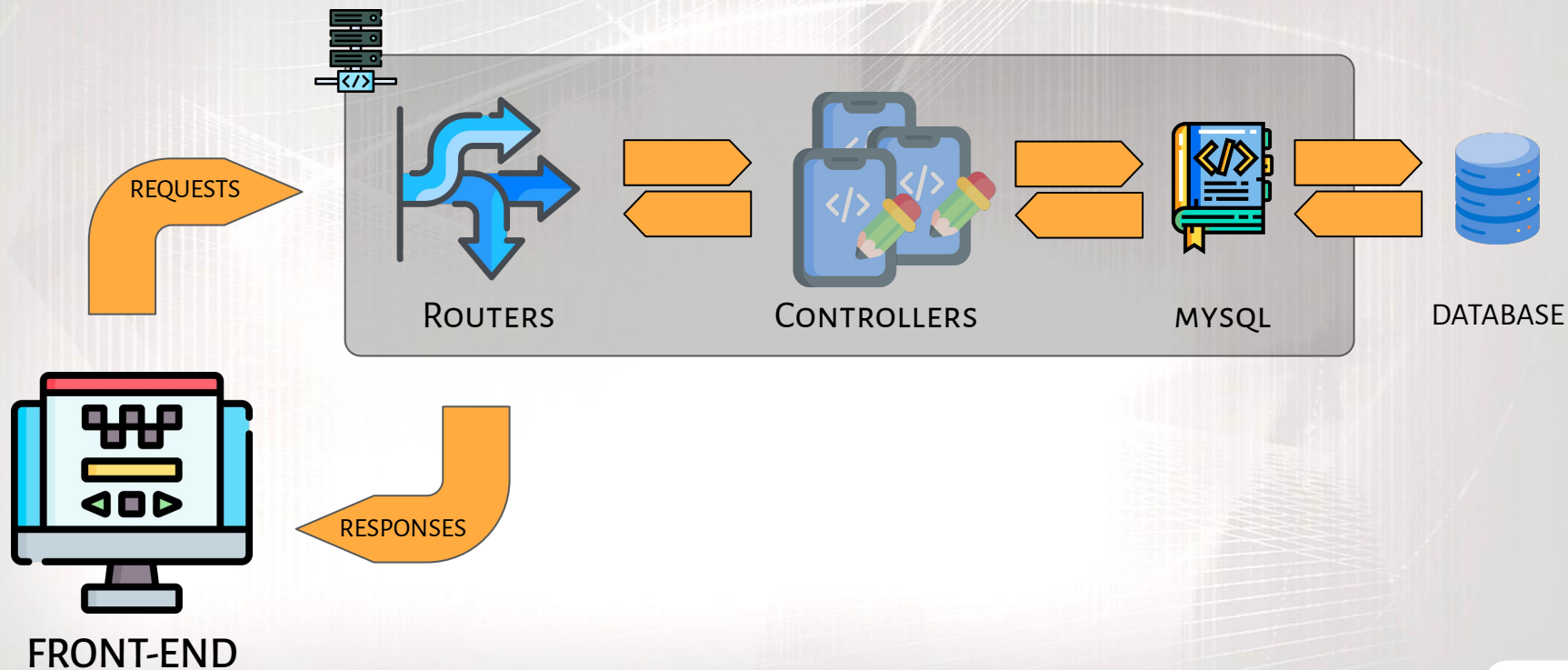
03

BACK-END DELL'APPLICAZIONE

Analisi dettagliata del
dell'applicazione di back-end.



STRUTTURA



DESCRIZIONE DELLE API

/USERS

METODO HTTP POST

REGISTRAZIONE DI UN UTENTE AL SERVIZIO CON
CONTROLLO CAPTCHA



/CLOCK-CAPTCHA

METODO HTTP GET

RICHIESTA DEI DATI PER INIZIALIZZARE UN
OGGETTO *CLOCK-CAPTCHA-VIEW*

/SESSION

METODO HTTP POST

APERTURA DI UNA NUOVA SESSIONE

/SESSION

METODO HTTP GET

RECUPERO DATI DI SESSIONE



04

CLOCKCAPTCHA

Disamina su
implementazione della
business logic della libreria.



CLASSE ClockCAPTCHAView

ClockCAPTCHAView
<ul style="list-style-type: none">- _token: string- _spinner: NodeJS.Timer- _canvas: HTMLCanvasElement- _moduleBody: HTMLElement- _userInputElement: HTMLInputElement- _title: HTMLElement
<ul style="list-style-type: none">+constructor()+inject(container: HTMLElement): boolean+getToken(): string+getInput(): string+clear(): void+error(msg: string): void+message(msg: string): void+fill(image_src: string, token: string): void-moduleBuild(): void-done(): void-waiting(): void

VISTA DELLA CLASSE ClockCAPTCHA

CONTIENE I DATI GENERATI DA ESSA

RACCOGLIE L'INSERIMENTO UTENTE



FUNZIONAMENTO ClockCAPTCHAView

	<pre>var captchaModule : ClockCAPTCHAView = new ClockCAPTCHAView();</pre>
	<pre>//Recupero dei dati dal back-end captchaModule.fill(image, token);</pre>
	<pre>//Errore di formato captchaModule.error("Controlla il formato!");</pre>
	<pre>//Invio dati al back-end let input = captchaModule.getInput(); let token = captchaModule.getToken(); //Captcha errato captchaModule.clear();</pre>
	<pre>//Recupero dei dati dal back-end captchaModule.fill(image, token); captchaModule.error("Ops, orario scorretto!");</pre>



DIAGRAMMA DELLA CLASSI

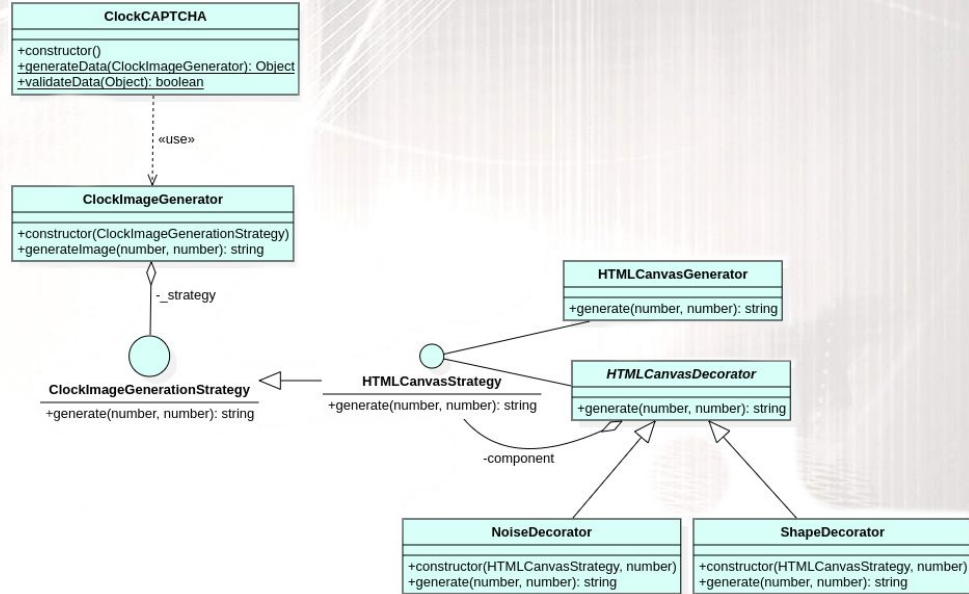
**LA PARTE DI BACK-END HA DUE
FUNZIONI:**

**GENERARE DEI DATI E LA LORO
RAPPRESENTAZIONE VISIVA, RESTITUENDO
LA COPPIA**

{informazione_criptata, sua_rappresentazione_visiva}

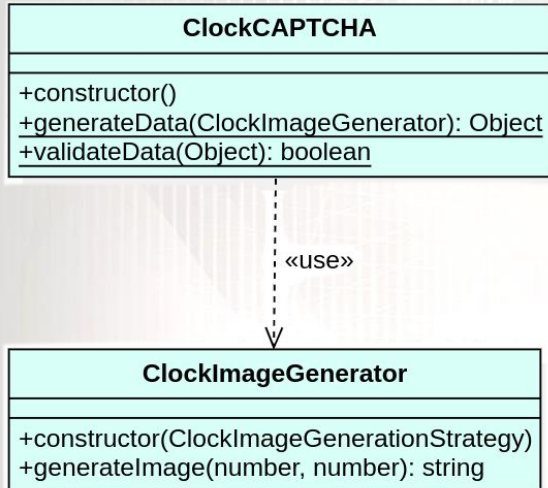
**VALIDARE LA NATURA DELL'UTENTE
VERIFICANDO LA COPPIA**

{input_utente, informazione_criptata}



DESIGN PATTERNS

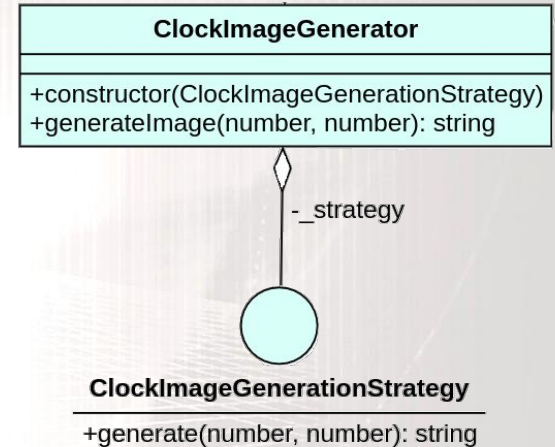
DEPENDENCY INJECTION



L'UTILIZZO DEL *DEPENDENCY INJECTION* FAVORISCE LA MODULARITÀ DEL CODICE, FACILITANDO LA FASE DI TEST E MIGLIORANDO LA MANUTENIBILITÀ DEL PRODOTTO.

L'UTILIZZO DEL *STRATEGY PATTERN* FACILITA L'INTRODUZIONE DI NUOVE TECNOLOGIE E RELATIVE METODOLOGIE AL PRODOTTO

STRATEGY PATTERN

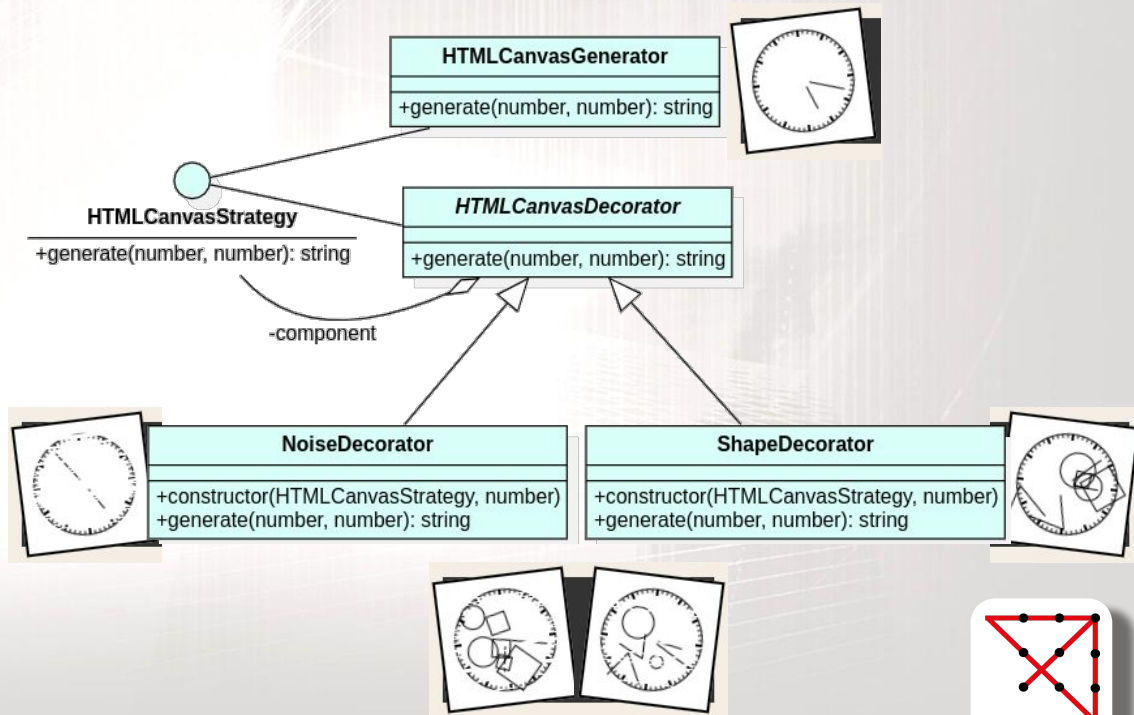


DESIGN PATTERNS

DECORATOR PATTERN

L'UTILIZZO DEL *Decorator Pattern* CONSENTE LA CREAZIONE DI IMMAGINI CON GRADO E TIPO DI DISTURBO PERSONALIZZABILE

PER UTILIZZARE IL *Pattern* SI PROCEDE PRIMA A CREARE PRIMA LA STRATEGIA BASE CHE GENERA L'OROLOGIO ANALOGICO, POI APPLICANDOCI SOPRA I DISTURBATORI INTERESSATI



05

DATABASE

Considerazioni sul database
del progetto.



STRUTTURA E GESTIONE

IL DATABASE VIENE GESTITO CON DRIVERS *MySQL* E PRESENTA LE SEGUENTI TABELLE:

“USERS”: CONTENENTE LE INFORMAZIONI RELATIVE AGLI UTENTI

“APIKEYS”: CONTENENTE LE CHIAVI SEGRETE DELLE APPLICAZIONI A CUI È CONSENTITO L'UTILIZZO DEL *BACK-END*

“BLACKLIST”: CONTENENTE TUTTI I TOKEN GENERATI MEDIANTE *CLOCKCAPTCHA* CHE SONO STATI UTILIZZATI ALMENO UNA VOLTA

users	
PK	<u>email</u>
	username
	firstname
	lastname
	password

apikeyes	
PK	<u>secretkey</u>
	host
	appName

blacklist	
PK	<u>OTT</u>
	used



06

APPLICAZIONE IN ESECUZIONE

Applicazione in uso.



GRAZIE PER L'ATTENZIONE

