



Università  
di Genova

**DIBRIS** DIPARTIMENTO  
DI INFORMATICA, BIOINGEGNERIA,  
ROBOTICA E INGEGNERIA DEI SISTEMI

# Sviluppo full-stack per un'applicazione web per la turnazione del personale

Laurea in Informatica  
A.A. 2022/23

Candidato: Enrico Pezzano

Relatrice: Marina Ribaudo

# Agenda

- Contesto
- Prototipo
- Conclusioni



# Contesto di riferimento

## Gruppo SIGLA



## Progetto

- Repository con **55mila** file
- Gestione **corsi** di formazione
- Componenti per l'**admin**
- **Testing**
- Gitlab

# Modello three-tier

## 3 livelli

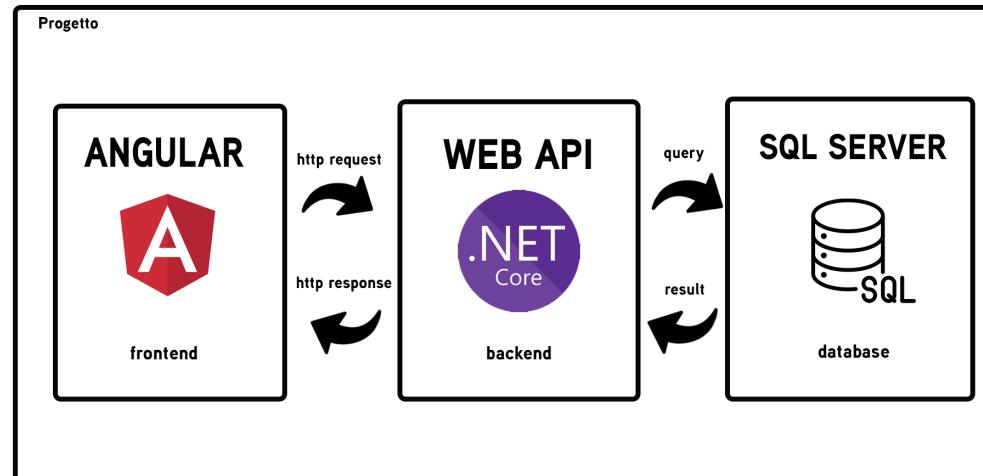
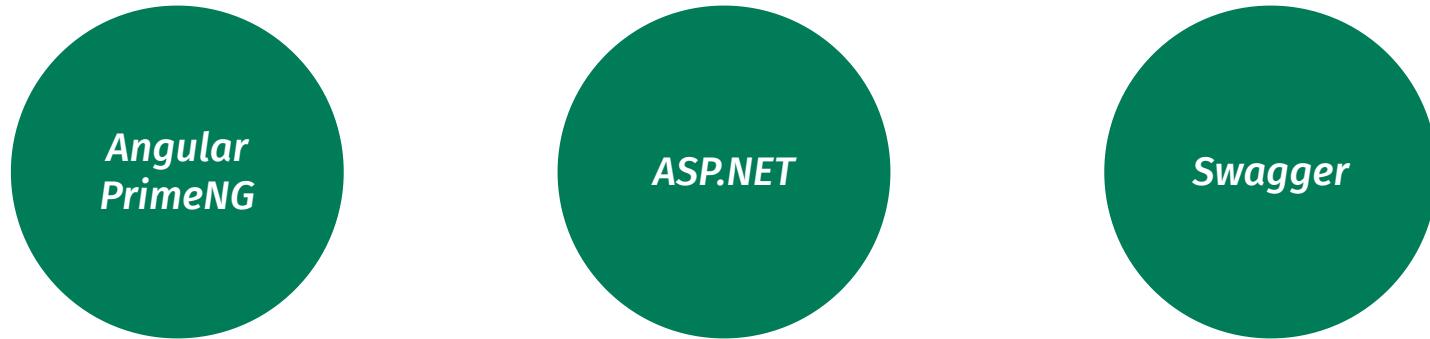
- Livello di presentazione
- Livello di applicazione
- Livello di accesso ai dati



# Implementazione del Prototipo

Quali framework e quali strumenti sono stati utilizzati?

# Framework e strumenti utilizzati





# Back-end

## ASP.NET

- C#
- Multi-piattaforma
- ORM

```
1  namespace It.gs.backend.QueryExecutors
2  {
3      using CSharpFunctionalExtensions;
4      using It.gs.backend.Model;
5      using It.gs.Repository;
6      using It.gs.Repository.Model;
7      using It.gs.Repository.Settings;
8      using System.Data;
9      using Dapper;
10     using It.gs.Repository.Dapper;
11     using It.gs.backend.Utilities;
12
13     public class CoursesDataExecuteExecutor : IExecuteWithTransactionExecutor<CoursesData>
14     {
15         public async Task<IExecuteResult> Execute(DatabaseSettings settings, IDbConnection connection,
16             IDbTransaction transaction, IExecuteInfo info)
17         {
18             switch(info)
19             {
20                 case AddCoursesToDbExecuteInfo i:
21                     return await AddCoursesExecute(settings, connection, transaction, i);
22                 case DeleteCoursesDataExecuteInfo i:
23                     return await DeleteCoursesDataExecute(settings, connection, transaction, i);
24                 case UpdateCoursesDataExecuteInfo i:
25                     return await UpdateCoursesDataExecute(settings, connection, transaction, i);
26                 default:
27                     throw new NotSupportedException($"Execute with transaction for type {info.GetType().
28                         FullName} not supported");
29             }
30
31         private async Task<IExecuteResult> AddCoursesExecute(DatabaseSettings settings, IDbConnection
32             connection, IDbTransaction transaction, AddCoursesToDbExecuteInfo info)
33         {
34             foreach(var Course in info.Courses)
35             {
36                 _ = await Add(settings, connection, transaction, Course);
37             }
38             return await IExecuteResult.From(true);
39         }
40         private async Task<CoursesData> Add(DatabaseSettings settings, IDbConnection connection,
41             IDbTransaction transaction, CoursesData item)
42         {
43             Console.WriteLine("item: " + item);
44             var sql = $"INSERT INTO CoursesData (CoursesName, CoursesCapacity, CoursesType, CoursesDate)
45             VALUES (@{nameof(CoursesData.CoursesName)}, @{nameof(CoursesData.CoursesCapacity)}, @{nameof(
46             CoursesData.CoursesType)}, @{nameof(CoursesData.CoursesDate)})";
47
48             var r = await connection.ExecuteAsync(sql, item, transaction);
49             return item;
50         }
51     }
52 }
```



# Front-end

## Angular e PrimeNG

- Componenti
- Servizi
- Route Guard
- Typescript
- GUI

```
148  ngOnInit(): void {
149    this.coursesData$ = this.store.select(selectCoursesData).pipe(startWith(this.route.snapshot.data.CoursesData));
150    this.coursesData$.pipe(
151      filter(data => !data)
152    ).subscribe((data) => {
153      this.totalRecords$ = this.coursesData$.pipe(map((x) => (x ? (x[0] ? x[0].count : 0) : 0)));
154
155      let tmp = [];
156      data.forEach((course) => {
157        let day:string = course.coursesDate[0] + course.coursesDate[1];
158        let month:string = course.coursesDate[3] + course.coursesDate[4];
159        let year:string = course.coursesDate[6] + course.coursesDate[7] + course.coursesDate[8] + course.coursesDate[9];
160
161        tmp.push({ title: course.coursesName, date: year + '-' + month + '-' + day }); // yyyy-mm-dd
162        console.log(tmp);
163      });
164
165      this.calendarOptions = {
166        ...this.calendarOptions,
167        events: tmp
168      };
169    })
170 }
```



# Testing API

## Swagger e DBeaver

The screenshot shows the Swagger UI interface for a REST API endpoint. The endpoint is `POST /api/v1/Courses/searchCoursesData`. The request body is defined as follows:

```
{  "ordering": [    {      "column": "string",      "descending": true,      "columnPrefix": "string"    }  ],  "paging": {    "skip": 0,    "take": 0  },  "filtering": [    {      "column": "string",      "predicate": "string",      "kind": "string",      "value": "string",      "prefix": "string"    }  ]}
```

The responses section shows a single 200 Success response with a media type of `text/plain`.

# Prototipo

The screenshot shows the Sakai Admin interface for course management. The top navigation bar includes the Sakai logo, a main menu icon, and a user profile icon. The left sidebar, titled "MAIN MENU", lists various administrative functions: Notifiche, Prenota, Iscriviti, Formazione, Panoramica uffici (A), Gestione uffici (A), Gestione notifiche (A), and Gestione formazione (A). The main content area is titled "Gestione corsi (Admin)" and contains a "Aggiungi corso" (Add Course) form with fields for Nome corso, Numero postazioni, Selezione tipologia corso (with a dropdown menu), Data inizio corso (with a date picker icon), and an "Aggiungi corso" button. Below this is a "Elenco corsi creati" (List of created courses) table with columns: Nome corso, Partecipanti max, Tipologia, Data inizio, Modifica, and Elimina. Two entries are listed: "Angular Modulo 1" (10 participants, Udemy, 01/01/2024) and "test" (22 participants, Angular University, 22/01/2024). A "Course Search" input field is also present. At the bottom, a footer note says "by PrimeNG".

Nome corso ↑↓	Partecipanti max	Tipologia ↑↓	Data inizio ↑↓	Modifica	Elimina
Angular Modulo 1	10	Udemy	01/01/2024		
test	22	Angular University	22/01/2024		

Vista amministratore - aggiunta corsi

# Prototipo

The screenshot shows the Sakai Admin interface for course management. The top navigation bar includes the Sakai logo, a main menu icon, and a user profile icon. The left sidebar, titled "MAIN MENU", lists various administrative functions: Notifiche, Prenota, Iscriviti, Formazione, Panoramica uffici (A), Gestione uffici (A), Gestione notifiche (A), and Gestione formazione (A). The "Gestione formazione (A)" option is highlighted in red. The main content area is titled "Gestione corsi (Admin)". It displays a "Modifica corso" (Edit Course) form for "Angular Modulo 1". The form fields include "Nome corso" (Angular Modulo 1), "Numero postazioni" (10), "Selezione tipologia corso" (dropdown menu), and "Data inizio corso" (01/01/2024). Below the form are two buttons: "Modifica corso" (green) and "Annulla" (blue). To the right of the form is a red gear icon. The bottom section is titled "Elenco corsi creati" (List of created courses) and shows a table with two entries:

Nome corso	Partecipanti max	Tipologia	Data inizio	Modifica	Elimina
Angular Modulo 1	10	Udemy	01/01/2024		
test	22	Angular University	22/01/2024		

Below the table is a "Course Search" input field and a pagination control showing "Showing 1 to 2 of 2 entries" with page numbers 1, 2, and 5.

Vista amministratore - modifica corsi

# Prototipo

The screenshot shows a Sakai-based course booking interface. At the top left is the Sakai logo. On the right are user profile and navigation icons. A sidebar on the left contains a 'MAIN MENU' with links: Notifiche, Prenota, Iscriviti (highlighted in red), and Formazione.

**Prenota un corso per oggi 12/02/2024 o oltre**

**Corsi prenotati**

Nome corso	Tipologia	Dettagli	Cancella Prenotazione
No Courses Data found.			

**Corsi disponibili**

Nome corso ↑↓	Tipologia ↑↓	Prenota
> Angular Modulo 1	Udemy	<input checked="" type="button"/>
> test	Angular University	<input checked="" type="button"/>

Showing 1 to 2 of 2 entries << < 1 > >>

**January 2024**

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1 Angular Modulo 1	2	3	4	5	6	7
8	9	10	11	12	13	14

today < >

Vista utente base

# Prototipo

The screenshot shows a calendar interface for January 2024. The top navigation bar includes the SAKAI logo, a menu icon, and user account information. Below the header is a sidebar titled "MAIN MENU" containing links such as Notifiche, Prenota, Iscriviti (highlighted in red), Formazione, Panoramica uffici (A), Gestione uffici (A), Gestione notifiche (A), and Gestione formazione (A). The main area displays a grid of days from Monday to Friday. A blue bar labeled "Angular Modulo 1" spans from Monday, January 1st, to Tuesday, January 2nd. Another blue bar labeled "test" spans from Monday, January 22nd, to Tuesday, January 23rd. A red square icon with a gear symbol is positioned over the Friday, January 19th, cell. The bottom of the calendar features horizontal scroll bars.

Vista calendario

# Cosa ho imparato durante il tirocinio?

- Cosa sono i framework e la loro importanza
- **Angular**, per il front-end
- **ASP.NET**, per il back-end
- Sincronizzazione front-end/back-end





**DIBRIS** DIPARTIMENTO  
DI INFORMATICA, BIOINGEGNERIA,  
ROBOTICA E INGEGNERIA DEI SISTEMI

# Grazie per l'attenzione

**UniGe**  

---

**DIBRIS**