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

Università di Genova
DIBRIS DIPARTIMENTO
DI INFORMATICA, BIOINGEGNERIA,
ROBOTICA E INGEGNERIA DEI SISTEMI

Implementazione del Prototipo

Quali framework e quali strumenti sono stati utilizzati?

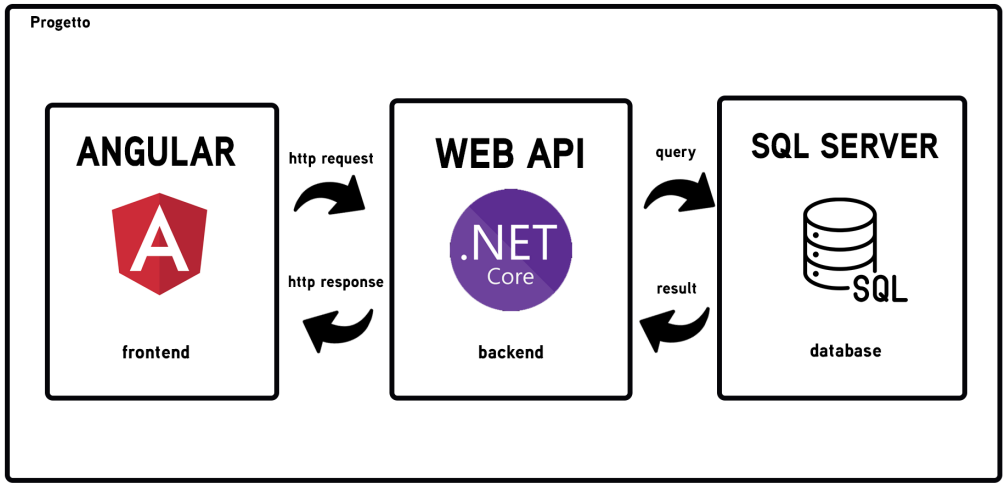
Framework e strumenti utilizzati




Angular
PrimeNG

ASP.NET

Swagger



Back-end

ASP.NET

- C#
- Multi-piattaforma
- ORM

```
namespace It.gs.backend.QueryExecutors
{
    using CSharpFunctionalExtensions;
    using It.gs.backend.Model;
    using It.gs.Repository;
    using It.gs.Repository.Model;
    using It.gs.Repository.Settings;
    using System.Data;
    using Dapper;
    using It.gs.Repository.Dapper;
    using It.gs.backend.Utilities;

    public class CoursesDataExecuteExecutor : IExecuteWithTransactionExecutor<CoursesData>
    {
        public async Task<IExecuteResult> Execute(DatabaseSettings settings, IDbConnection connection,
        IDbTransaction transaction, IExecuteInfo info)
        {
            switch(info) {
                case AddCoursesToDbExecuteInfo i:
                    return await AddCoursesExecute(settings, connection, transaction, i);
                case DeleteCoursesDataExecuteInfo i:
                    return await DeleteCoursesDataExecute(settings, connection, transaction, i);
                case UpdateCoursesDataExecuteInfo i:
                    return await UpdateCoursesDataExecute(settings, connection, transaction, i);
                default:
                    throw new NotSupportedException($"Execute with transaction for type {info.GetType().
                    FullName} not supported");
            }
        }

        private async Task<IExecuteResult> AddCoursesExecute(DatabaseSettings settings, IDbConnection
        connection, IDbTransaction transaction, AddCoursesToDbExecuteInfo info) {
            foreach(var Course in info.Courses)
                _ = await Add(settings, connection, transaction, Course);

            return await IExecuteResult.From(true);
        }
        private async Task<CoursesData> Add(DatabaseSettings settings, IDbConnection connection,
        IDbTransaction transaction, CoursesData item)
        {
            Console.WriteLine("item: ",item);
            var sql = $"INSERT INTO CoursesData (CoursesName, CoursesCapacity, CoursesType, CoursesDate)
            VALUES (@{nameof(CoursesData.CoursesName)}, @{nameof(CoursesData.CoursesCapacity)}, @{nameof
            (CoursesData.CoursesType)}, @{nameof(CoursesData.CoursesDate)})";

            var r = await connection.ExecuteAsync(sql, item, transaction);
            return item;
        }
```

Front-end

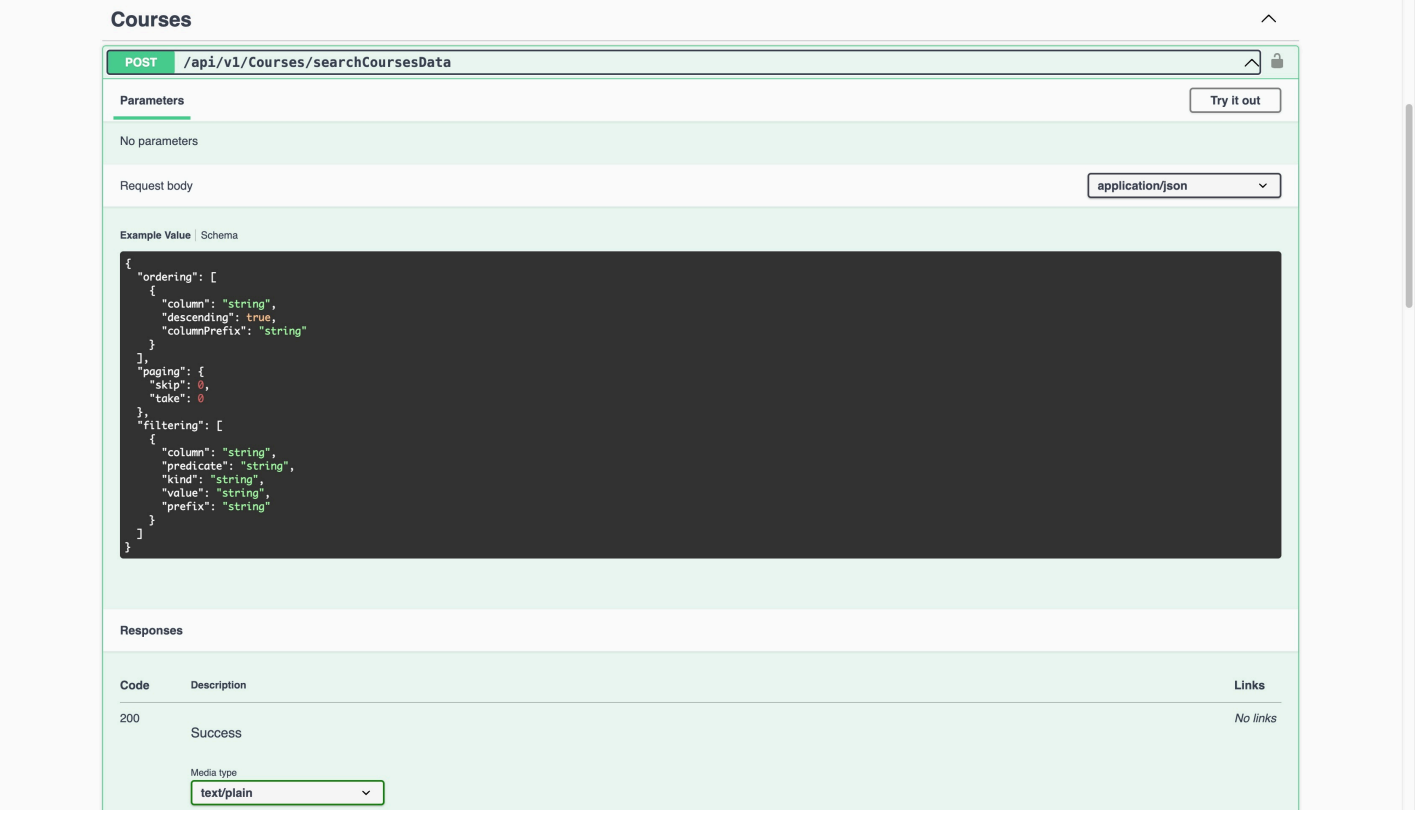
Angular e PrimeNG

- Componenti
- Servizi
- Route Guard
- Typescript
- GUI

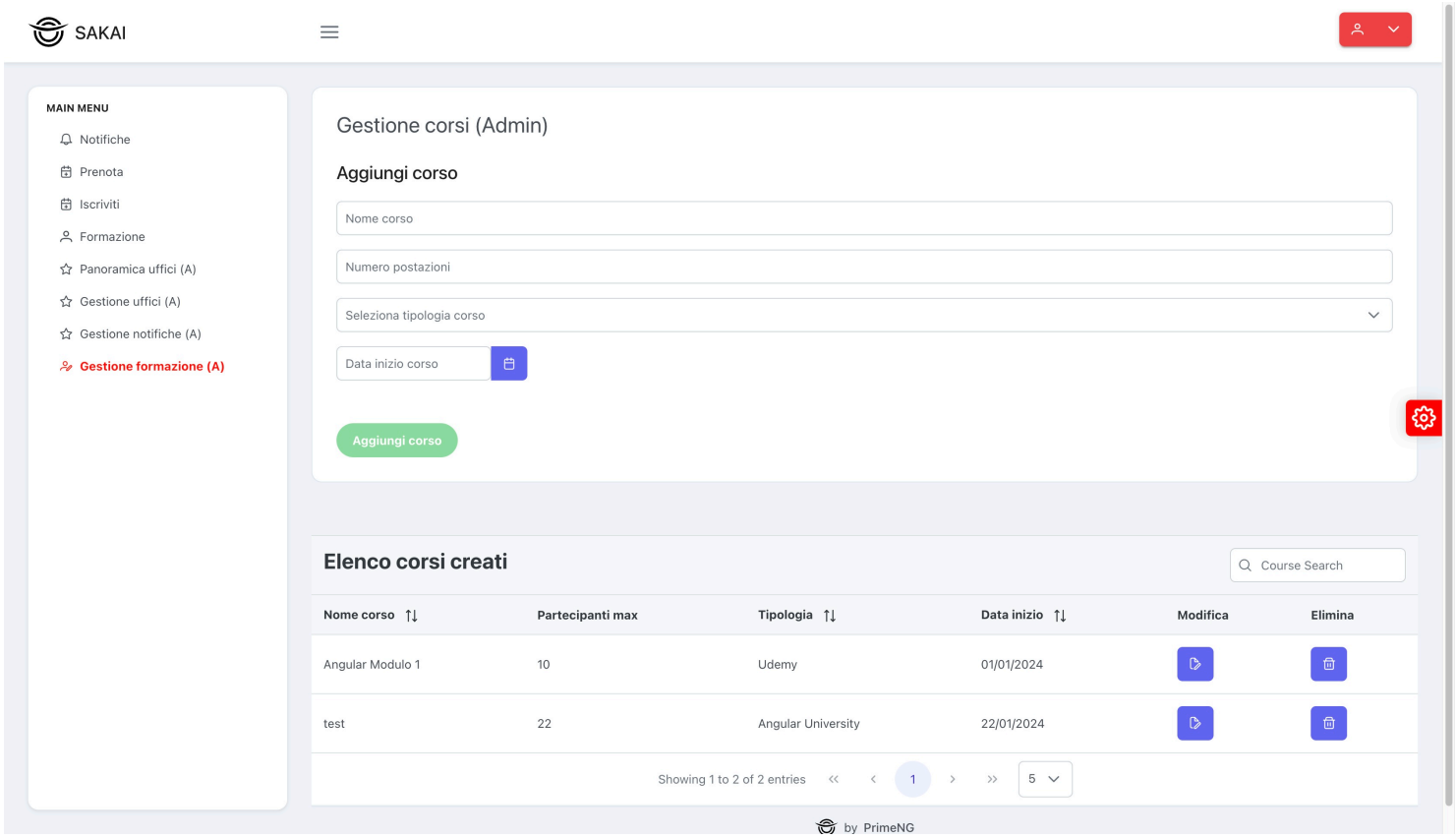
```
ngOnInit(): void {
    this.coursesData$ = this.store.select(selectCoursesData).pipe(startWith(this.route.snapshot.data.CoursesData));
    this.coursesData$.pipe(
        filter(data => !!data)
    ).subscribe((data) => {
        this.totalRecords$ = this.coursesData$.pipe(map((x) => (x ? (x[0] ? x[0].count : 0) : 0)));

        let tmp = [];
        data.forEach((course) => {
            let day:string = course.coursesDate[0] + course.coursesDate[1];
            let month:string = course.coursesDate[3] + course.coursesDate[4];
            let year:string = course.coursesDate[6] + course.coursesDate[7] + course.coursesDate[8] + course.coursesDate[9];

            tmp.push({ title: course.coursesName, date: year + '-' + month + '-' + day }); // yyy-mm-dd
            console.log(tmp);
        });

        this.calendarOptions = {
            ...this.calendarOptions,
            events: tmp
        };
    })
}
```

Testing API

Swagger e DBeaver

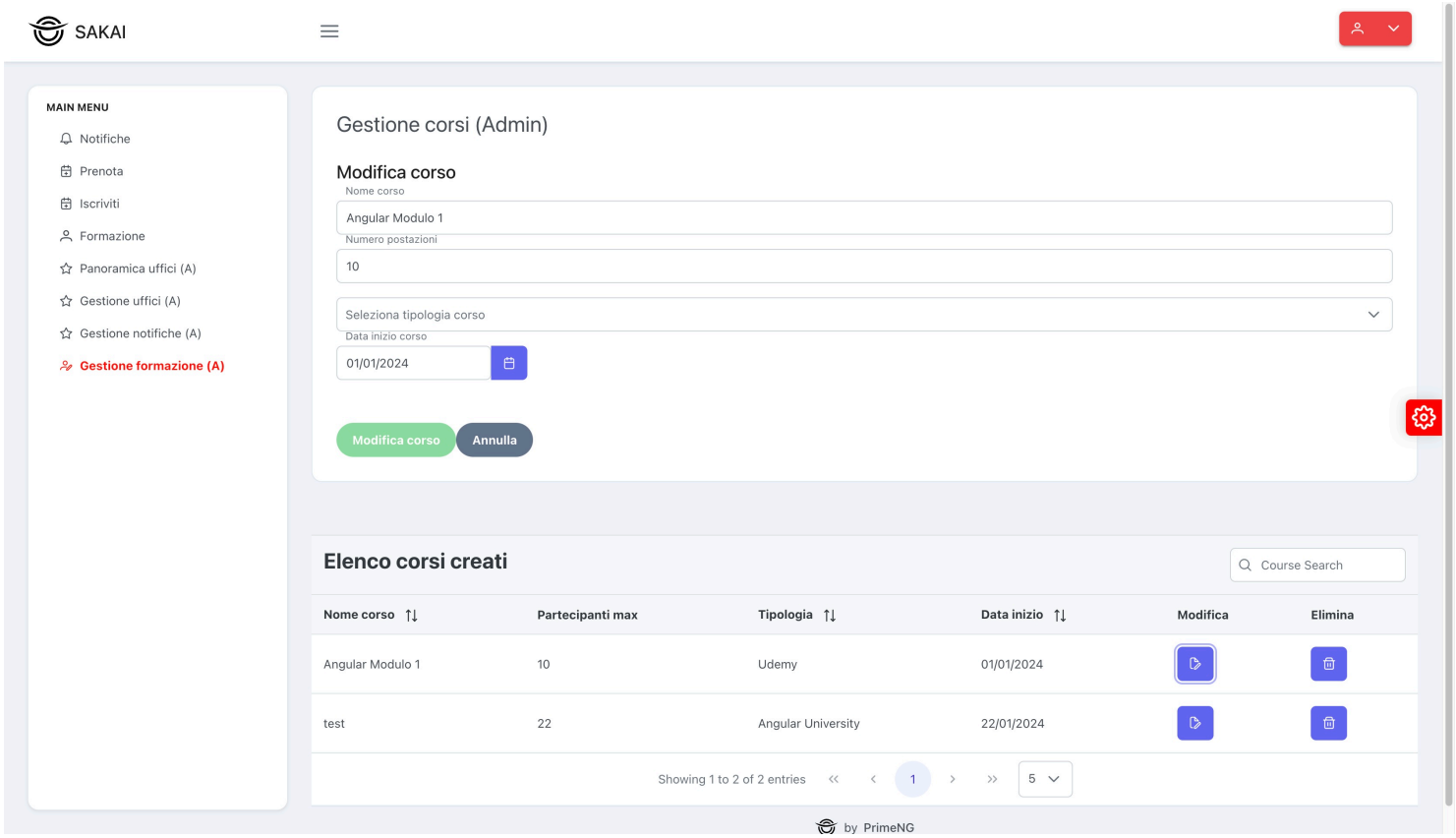


Prototipo



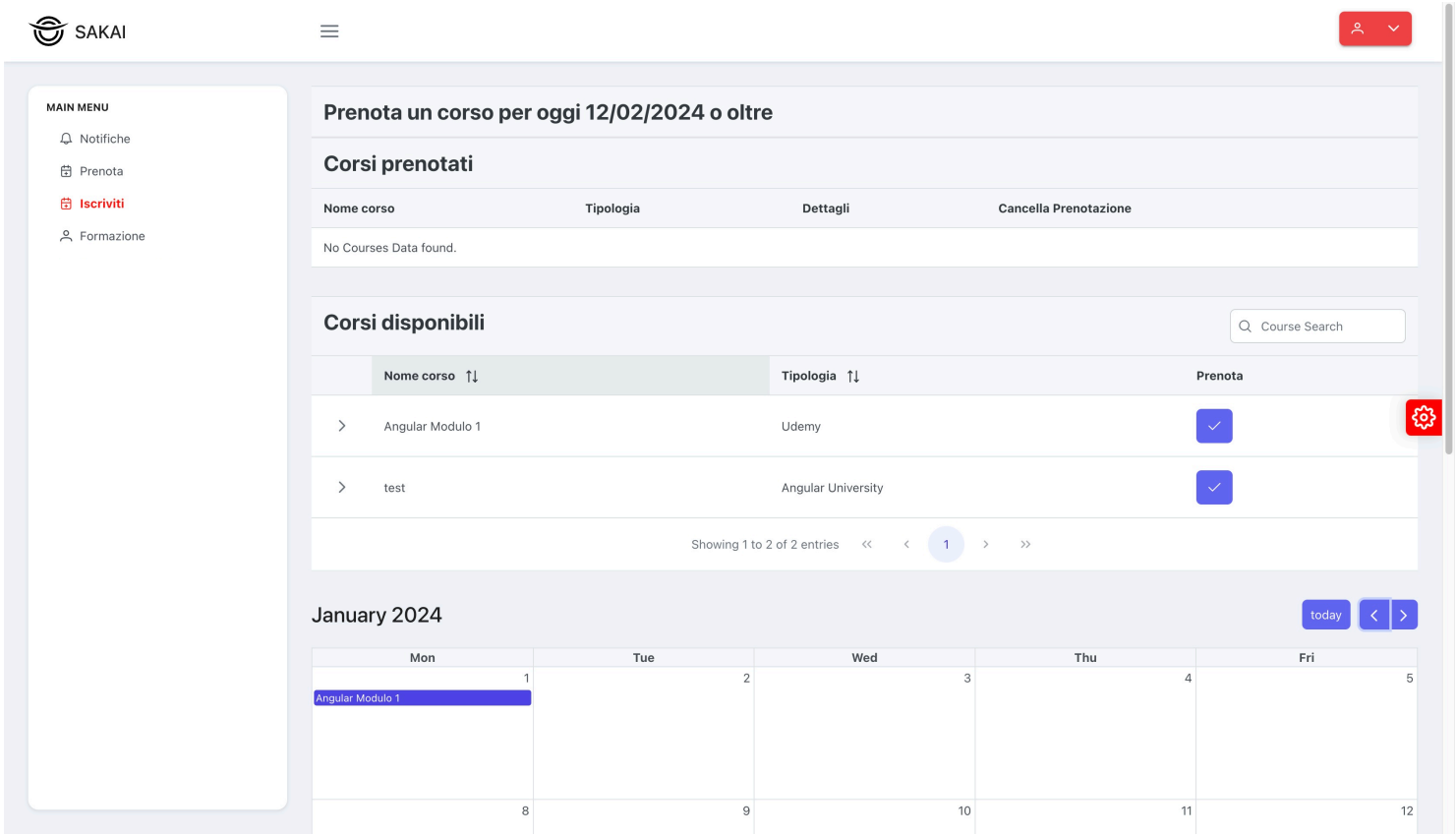
Vista amministratore - aggiunta corsi

Prototipo



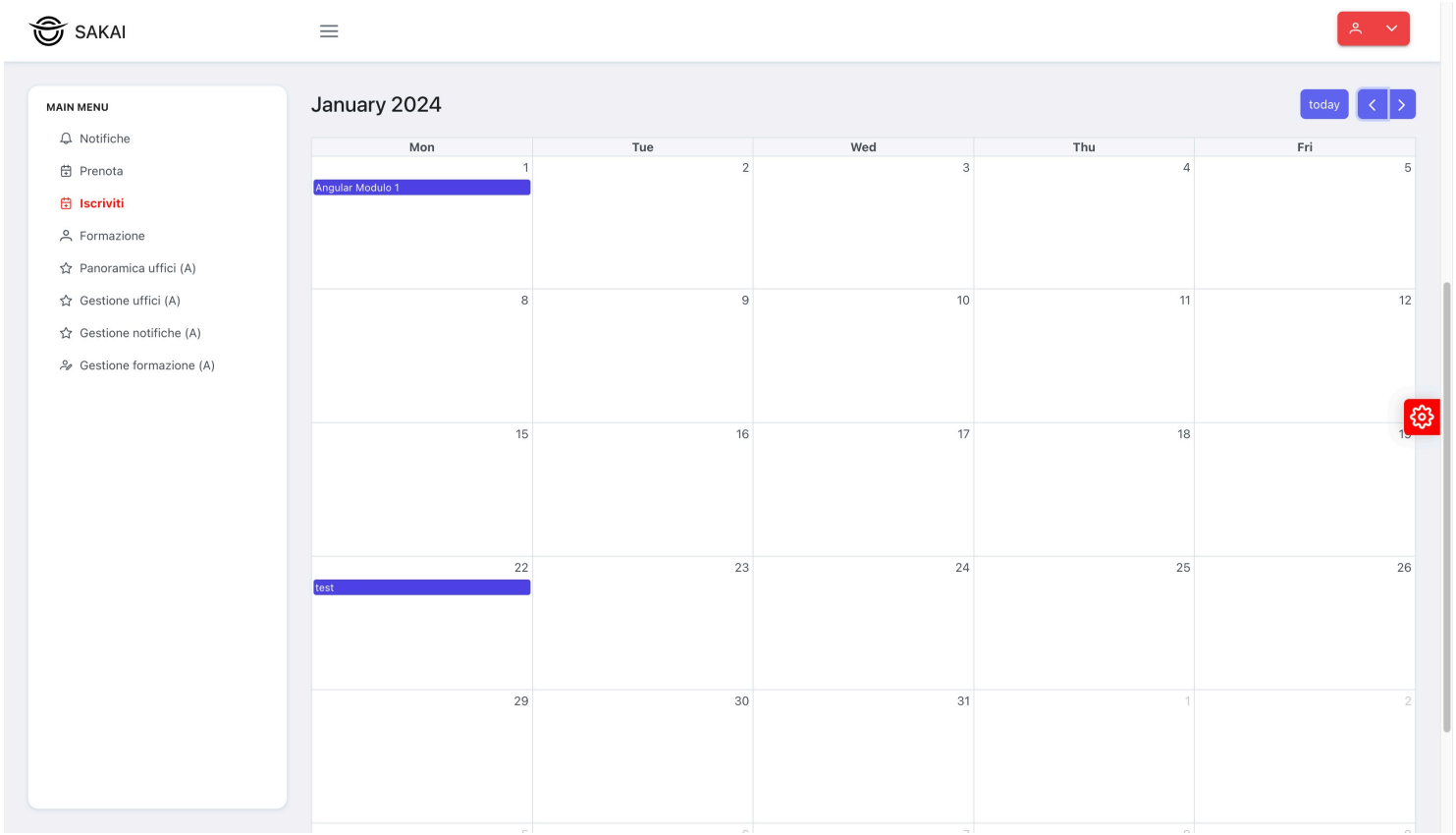
Vista amministratore - modifica corsi

Prototipo



Vista utente base

Prototipo



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

Università di Genova
DIBRIS DIPARTIMENTO
DI INFORMATICA, BIOINGEGNERIA,
ROBOTICA E INGEGNERIA DEI SISTEMI

Grazie per l'attenzione

UniGe

DIBRIS