

Sistema Menù per Gelateria con Spring Boot



1. Creazione del progetto con Spring Initializr

Vai su <https://start.spring.io> e configura così:

- **Project:** Maven
- **Language:** Java
- **Spring Boot:** 3.x
- **Group:** `it.epicode`
- **Artifact:** `gelateria`
- **Name:** `gelateria`
- **Package name:** `it.epicode.gelateria`
- **Packaging:** Jar
- **Java:** 17 o superiore



Dipendenze da aggiungere

- **Spring Boot DevTools** (sviluppo più rapido)
- **Lombok** (per evitare boilerplate di getter/setter/toString)
- **Faker** (puoi aggiungerlo dopo via Maven)

Premi "**Generate**" per scaricare lo zip del progetto.



2. Struttura dei package

Nel progetto IntelliJ, crea questa struttura sotto `src/main/java/it/epicode/gelateria` :

```
gelateria/
├── GelateriaApplication.java
├── config/
│   └── GelateriaConfig.java
├── model/
│   ├── Prodotto.java
│   ├── InformazioniNutrizionali.java
│   ├── Gelato.java
│   ├── Bevanda.java
│   ├── Topping.java
│   └── Menu.java
├── runner/
│   └── MenuRunner.java
```



3. Configurazione pom.xml

Aggiungi nel `pom.xml` la dipendenza per **Faker**:

```
<dependency>
  <groupId>com.github.javafaker</groupId>
  <artifactId>javafaker</artifactId>
  <version>1.0.2</version>
</dependency>
```



4. Codice passo-passo

◆ Prodotto.java (interfaccia base)

```
package it.epicode.gelateria.model;

public interface Prodotto {
    String getNome();
    double getPrezzo();
    InformazioniNutrizionali getInfoNutrizionali();
}
```

◆ InformazioniNutrizionali.java

```
package it.epicode.gelateria.model;

import lombok.AllArgsConstructor;
import lombok.Data;

@Data
@AllArgsConstructor
public class InformazioniNutrizionali {
    private int calorie;
    private int zuccheri;
    private int grassi;
}
```

◆ Topping.java

```
package it.epicode.gelateria.model;

import lombok.AllArgsConstructor;
import lombok.Data;

@Data
@AllArgsConstructor
public class Topping {
    private String nome;
    private double prezzo;
    private InformazioniNutrizionali infoNutrizionali;
}
```

◆ Gelato.java

```
package it.epicode.gelateria.model;

import lombok.Data;
import java.util.List;

@Data
public class Gelato implements Prodotto {

    private String nome;
    private List<Topping> toppings;
    private boolean maxi;

    public Gelato(String nome, List<Topping> toppings, boolean maxi) {
        this.nome = nome;
        this.toppings = toppings;
        this.maxi = maxi;
    }
}
```

```
}
```

```
@Override
```

```
public double getPrezzo() {  
    double base = 2.0;  
    double prezzoToppings = toppings.stream().mapToDouble(Topping::getPrezzo).sum();  
    double totale = base + prezzoToppings;  
    return maxi ? totale * 1.5 : totale;  
}
```

```
@Override
```

```
public InformazioniNutrizionali getInfoNutrizionali() {  
    int calorie = 100; // base  
    int zuccheri = 15;  
    int grassi = 5;  
  
    for (Topping t : toppings) {  
        InformazioniNutrizionali info = t.getInfoNutrizionali();  
        calorie += info.getCalorie();  
        zuccheri += info.getZuccheri();  
        grassi += info.getGrassi();  
    }  
  
    if (maxi) {  
        calorie *= 1.5;  
        zuccheri *= 1.5;  
        grassi *= 1.5;  
    }  
  
    return new InformazioniNutrizionali(calorie, zuccheri, grassi);  
}
```

```
}
```

◆ Bevanda.java

```
package it.epicode.gelateria.model;

import lombok.AllArgsConstructor;
import lombok.Data;

@Data
@AllArgsConstructor
public class Bevanda implements Prodotto {

    private String nome;
    private double prezzo;
    private InformazioniNutrizionali infoNutrizionali;
}
```

◆ Menu.java

```
package it.epicode.gelateria.model;

import lombok.AllArgsConstructor;
import lombok.Data;

import java.util.List;

@Data
@AllArgsConstructor
public class Menu {

    private List<Prodotto> prodotti;

    public void stampaMenu() {
        System.out.println("=== MENU GELATERIA ===");
    }
}
```

```

prodotti.forEach(p -> {
    System.out.printf("- %s: %.2f€ | Cal: %d, Zuccheri: %dg, Grassi: %dg%n",
        p.getNome(), p.getPrezzo(),
        p.getInfoNutrizionali().getCalorie(),
        p.getInfoNutrizionali().getZuccheri(),
        p.getInfoNutrizionali().getGrassi());
});
}
}

```

5. GelateriaConfig.java (definizione dei Bean)

```

package it.epicode.gelateria.config;

import com.github.javafaker.Faker;
import it.epicode.gelateria.model.*;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

import java.util.List;

@Configuration
public class GelateriaConfig {

    Faker faker = new Faker();

    @Bean
    public Topping cioccolato() {
        return new Topping("Cioccolato", 0.5, new InformazioniNutrizionali(50, 10, 5));
    }

    @Bean

```

```

public Topping granello() {
    return new Topping("Granello", 0.4, new InformazioniNutrizionali(30, 5, 2));
}

@Bean
public Gelato coppaGolosa() {
    return new Gelato("Coppa Golosa", List.of(cioccolato(), granello()), false);
}

@Bean
public Gelato maxiDelizia() {
    return new Gelato("Maxi Delizia", List.of(cioccolato(), cioccolato(), granello()), true);
}

@Bean
public Bevanda bibitaFresca() {
    return new Bevanda(faker.beer().name(), 1.5, new InformazioniNutrizionali(80, 20, 0));
}

@Bean
public Menu menu(List<Prodotto> prodotti) {
    return new Menu(prodotti);
}
}

```

6. MenuRunner.java (CommandLineRunner)

```

package it.epicode.gelateria.runner;

import it.epicode.gelateria.model.Menu;
import org.springframework.boot.CommandLineRunner;
import org.springframework.stereotype.Component;

```



```
@Component
public class MenuRunner implements CommandLineRunner {

    private final Menu menu;

    public MenuRunner(Menu menu) {
        this.menu = menu;
    }

    @Override
    public void run(String... args) {
        menu.stampaMenu();
    }
}
```

7. GelateriaApplication.java

```
package it.epicode.gelateria;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class GelateriaApplication {
    public static void main(String[] args) {
        SpringApplication.run(GelateriaApplication.class, args);
    }
}
```

✓ Output di esempio

=== MENU GELATERIA ===

- Coppa Golosa: 2.90€ | Cal: 180, Zuccheri: 30g, Grassi: 12g
- Maxi Delizia: 4.35€ | Cal: 255, Zuccheri: 45g, Grassi: 18g
- Bud Light Lime: 1.50€ | Cal: 80, Zuccheri: 20g, Grassi: 0g