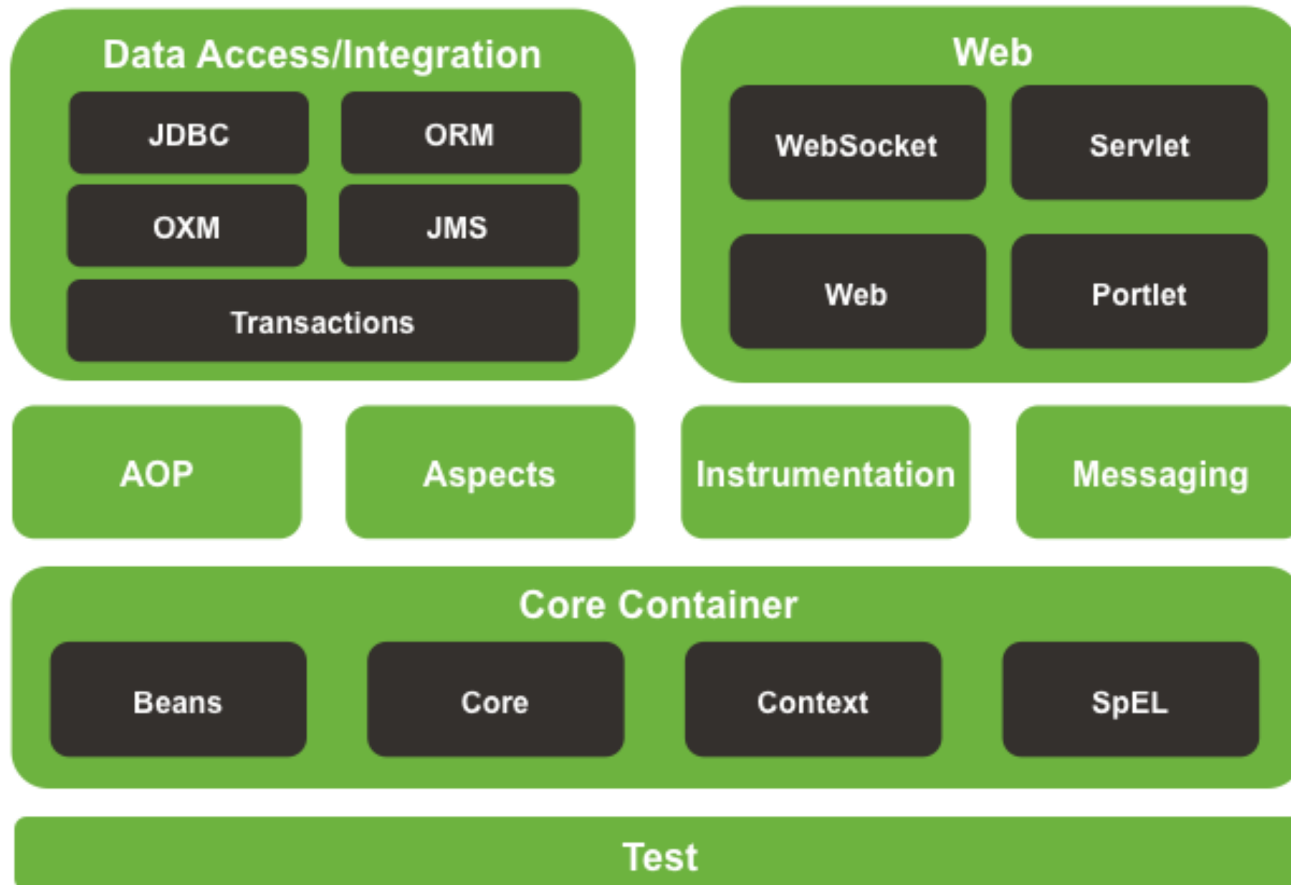
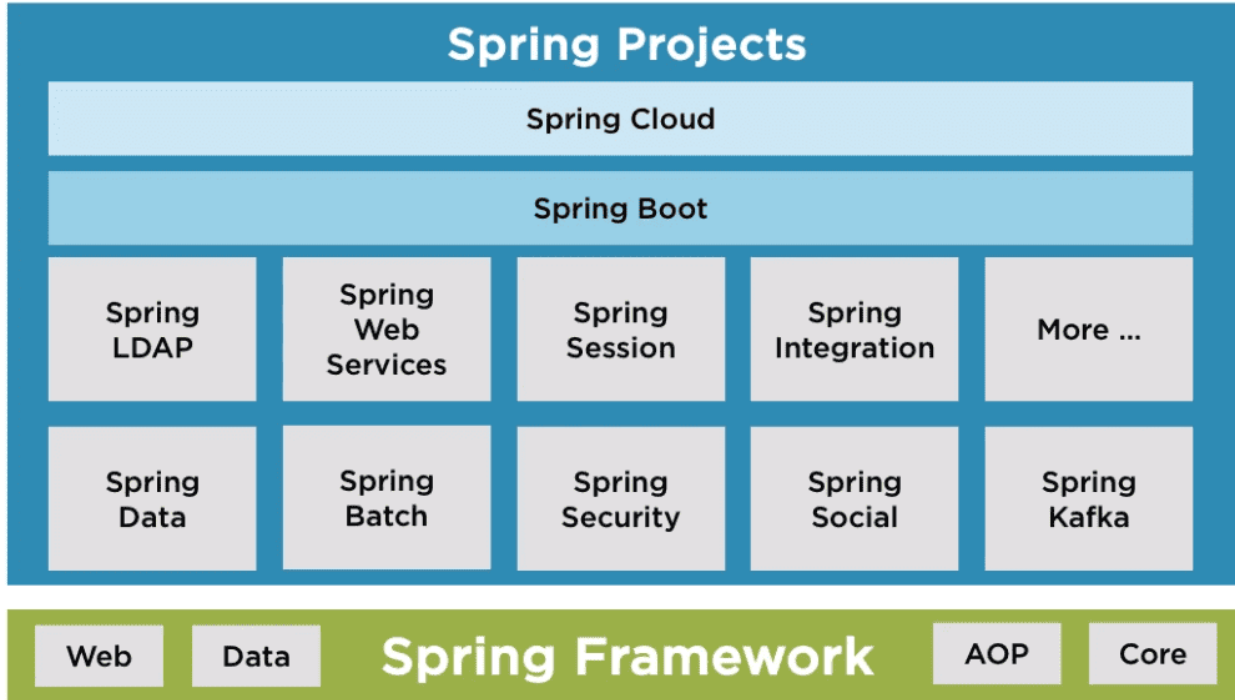


# O Spring



- Abstrair configurações para foco em regra de negócio
- Modularizado
- Projetos podem ter subProjetos

# O Spring boot

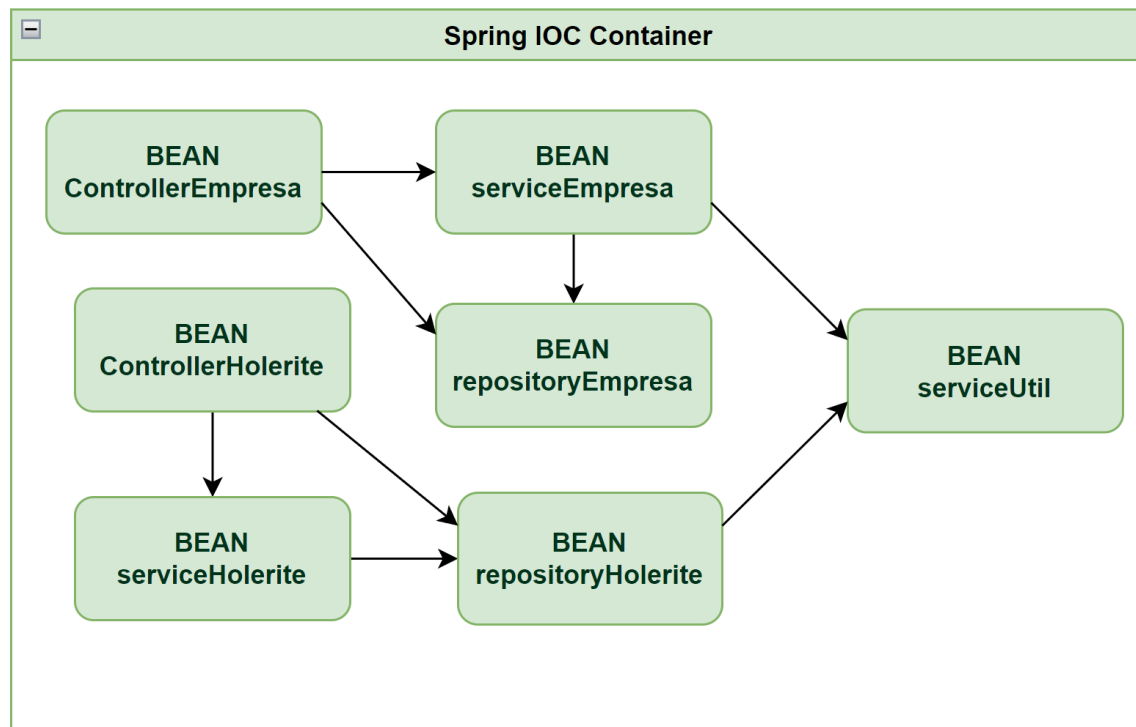
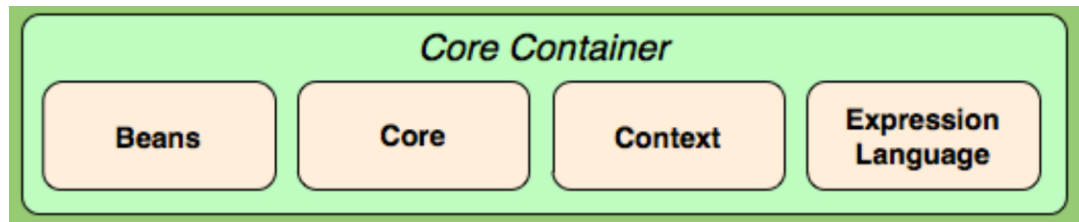


- Simplifica a organização dos projetos Spring
- Configura usando "visão opinativa"
- Servidor embarcado

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

# Spring core/loC container



- Responsável por lidar com a **inversão de controle e injeção de dependência entre Beans**
- O container gerencia os Beans

# ORM



HIBERNATE



```
@Entity
public class Usuario {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @OneToMany(cascade = CascadeType.ALL, mappedBy = "usuario")
    private List<Experiencia> experiencias = new ArrayList<>();

    @OneToOne(cascade = CascadeType.ALL)
    @JoinColumn(name = "links_id")
    private LinksDisponiveis linksDisponiveis;

    @OneToOne(cascade = CascadeType.ALL)
    @JoinColumn(name = "soft_skills_id")
    @JsonIgnore
    private SoftSkills softSkills;

    @OneToOne(cascade = CascadeType.ALL)
    @JoinColumn(name = "hard_skills_id")
    @JsonIgnore
    private HardSkills hardSkills;
}
```

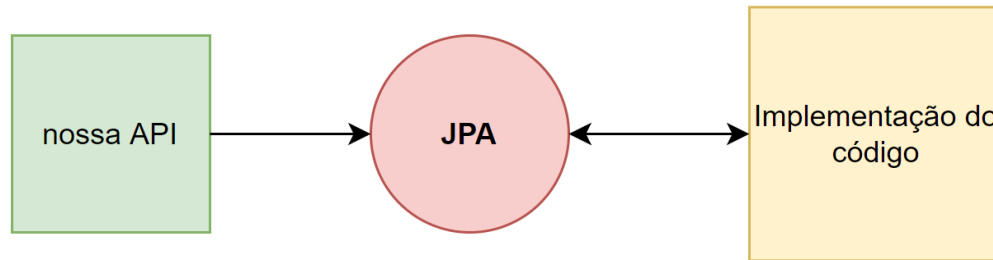
- Mapeia classes em entidades do banco de dados

The screenshot shows a database IDE interface. On the left, a tree view displays the 'usuario' table with its columns: id, email, link\_foto, localizacao, nivel\_senioridade, nome, profissao\_atual, resumo, senha, squad, interesses\_id, hard\_skills\_id, links\_id, and soft\_skills\_id. On the right, a 'Result Grid' shows the first three rows of data.

	id	email
1	1	lauralr@fcamara.com
2	2	mariajp@fcamara.com
3	3	robertost@fcamara.com

# JPA

- Simplesmente uma API que defini como ORM deve implementar recursos que a JPA expõe



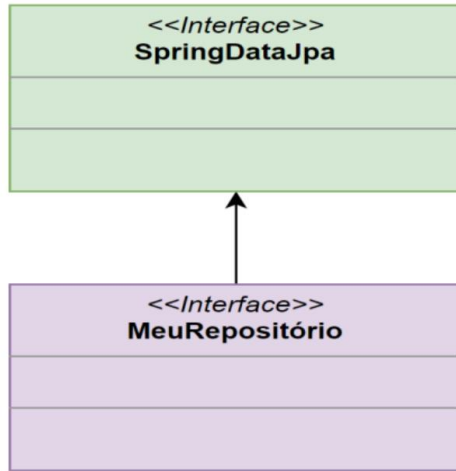
```
@Entity
public class Usuario {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

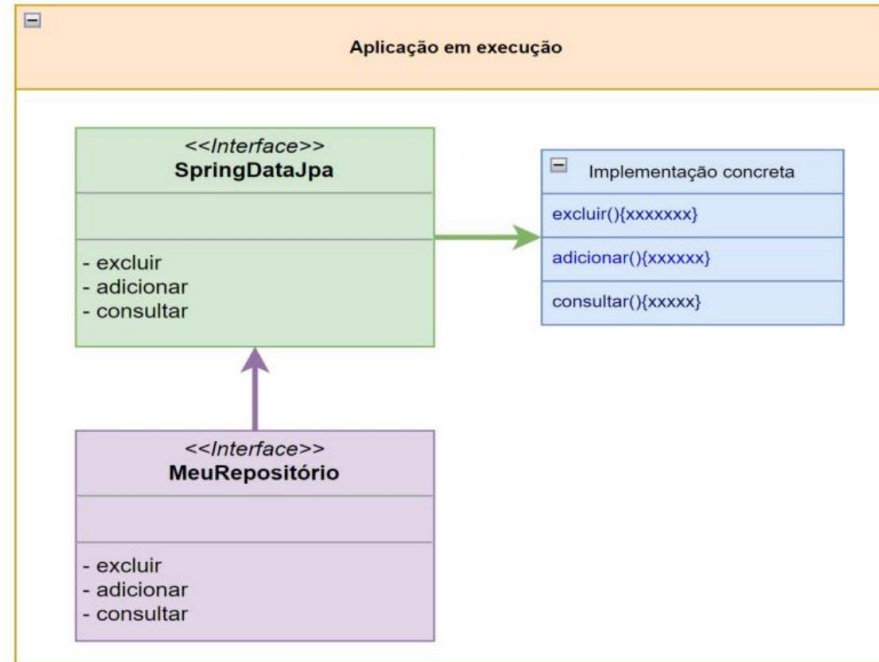
    private String nome;

    @Column(name = "link_foto")
    private String linkFoto;
```

# Spring Data JPA



- Fornece implementações para ações simples de interação com banco de dados



- Cria implementações automáticas pra consultas
- Paginação e etc