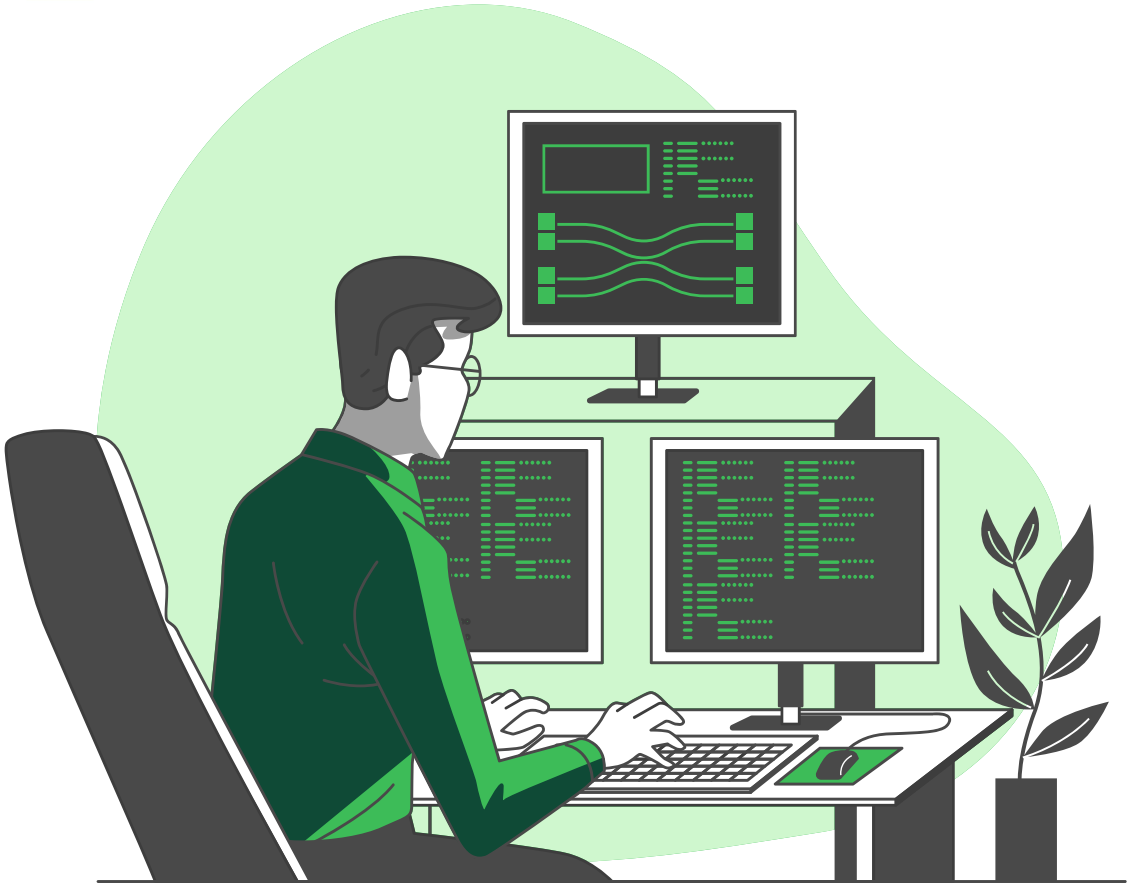
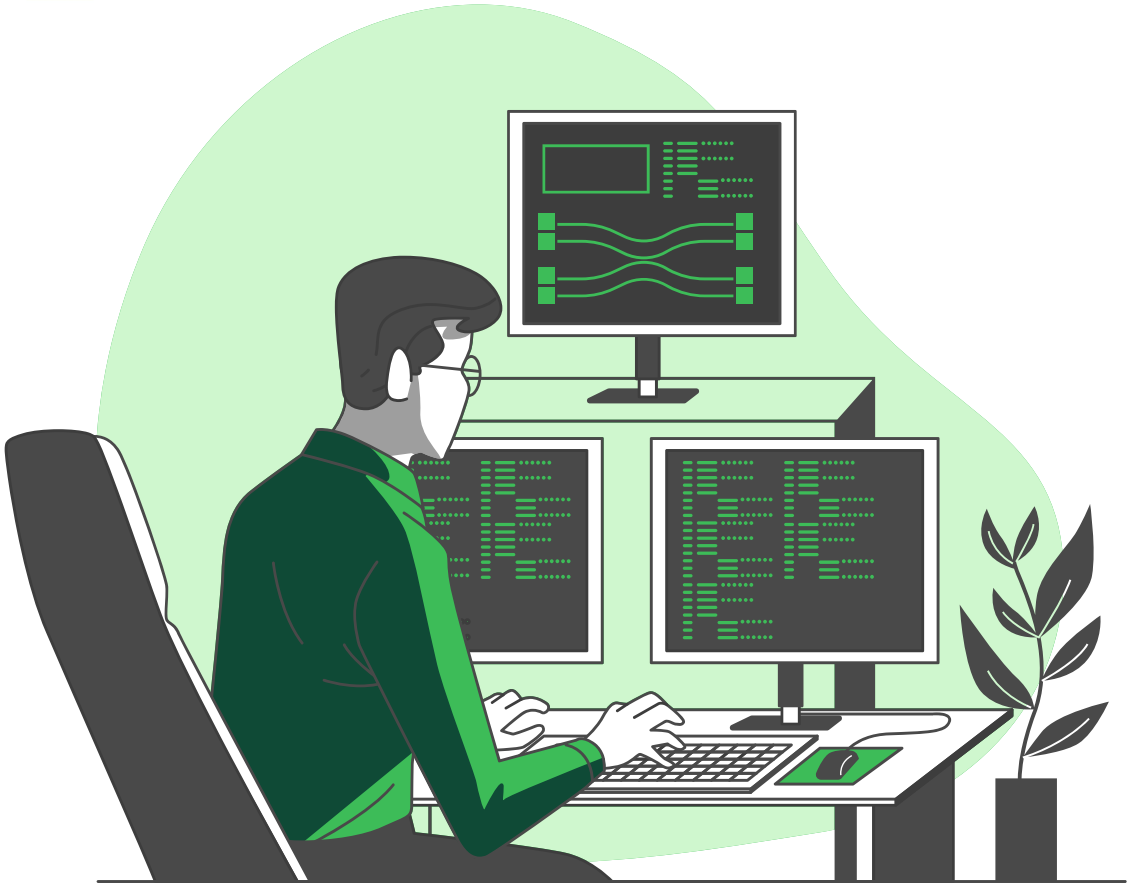


Spring

ecossistema para aplicações autossuficientes e robustas



por:

Hérson Rezende
Djair Maykon
Kennedy Andrade
Bruno de Lucas
Joab Pinheiro
João Leahy

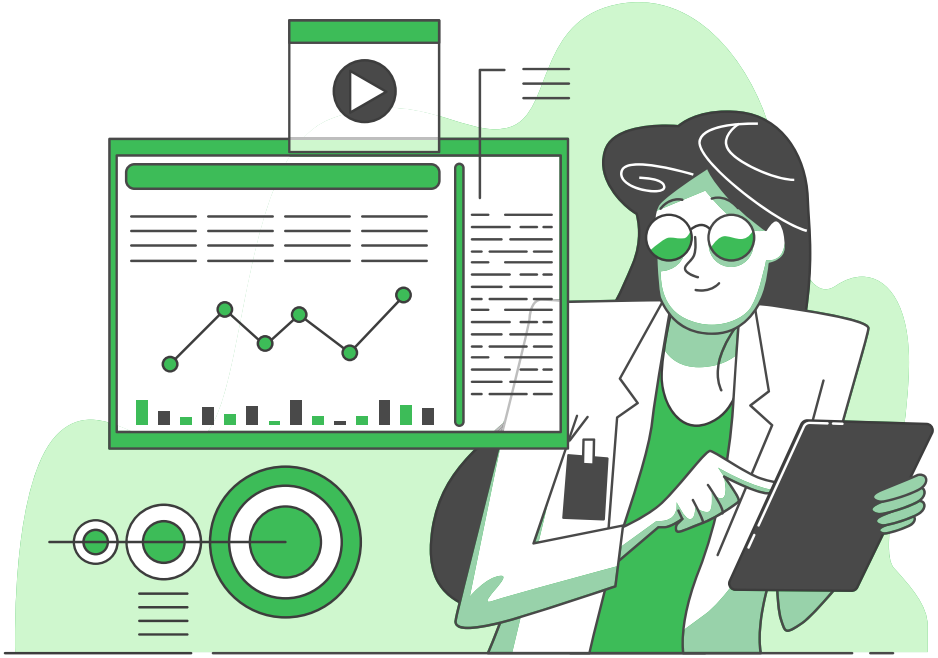
Conteúdos

01 Introdução

02 Fundamentos

03 Principais Recursos

04 Desenvolvendo Aplicações



Conteúdos

05 Ecossistema e Integrações

06 Arquitetura e Padrões

07 Casos de Uso

01
Introdução

O que é Spring

- O Spring é um ecossistema de desenvolvimento de aplicações Java voltado para o desenvolvimento de aplicações robustas e escaláveis. Ele fornece um conjunto abrangente de recursos e abstrações para lidar com as complexidades do desenvolvimento de software
- No início, o Spring Framework possuía diversos recursos integrados em uma mesma plataforma, porém o projeto cresceu tanto que alguns de seus recursos foram copiados por outros projetos. A partir daí, surgiu a ideia de modularizar as principais funcionalidades do Spring Framework e assim facilitar a integração com outros projetos que utilizavam outros frameworks. Com essa modularização surgiu o projeto Spring.
- Particularmente adequado para o desenvolvimento de aplicativos web, serviços RESTful e microsserviços, mas também pode ser usado em uma variedade de outras aplicações empresariais.

Por que usar o Spring?

- Abordagem abrangente: O ecossistema Spring oferece uma ampla gama de bibliotecas e frameworks para atender a diferentes necessidades no desenvolvimento de aplicações. O ecossistema Spring abrange várias áreas, como desenvolvimento de aplicações web, acesso a bancos de dados, segurança, integração com serviços da web, gerenciamento de configurações, entre outros.
- Integração perfeita: O Spring se integra facilmente com outras tecnologias e frameworks populares, permitindo que você aproveite os melhores recursos de cada um.
- Modularidade e flexibilidade: O ecossistema Spring é modular, permitindo que você escolha e utilize apenas os módulos necessários para o seu projeto.
- Comunidade ativa: O Spring possui uma comunidade de desenvolvedores muito ativa e engajada em todo o mundo. Além disso, é amplamente adotado por empresas em todo o mundo, desde startups até grandes organizações.

02
Fundamentos

Como funciona o Spring?

- **Inversão de Controle (IoC)**: Trata-se da interrupção do fluxo de execução de um código retirando, de certa forma, o controle sobre ele e delegando-o para uma dependência ou container. O principal propósito é minificar o acoplamento do código.
- **Injeção de Dependência (DI)**: É a proveniência de instâncias de classes que um objeto precisa sem que este instancie por si mesmo com o propósito de evitar o acoplamento de código numa aplicação.
- **Programação Orientada a Aspectos (AOP)**: é uma técnica de programação que permite modularizar aspectos transversais de um sistema, como por exemplo, logging, transações, segurança e caching, separando-os da lógica de negócio e mantendo o código mais organizado e manutenível.

03
Principais Recursos

O que o Spring nos oferece?

- **Injeção de dependencias:**
  - Utilizando o construtor da classe (Constructor Injection);

```
@Service
public class ExampleService {

        private ExampleRepository exampleRepository;

        public ExampleService(ExampleRepository exampleRepository) {
                this.exampleRepository = exampleRepository;
        }
}
```

O que o Spring nos oferece?

- **Injeção de dependencias:**
  - Utilizando o método setter (Setter Injection).;

```
@Service
public class ExampleService {

        private ExampleRepository exampleRepository;

        @Autowired
        public void setExampleRepository(ExampleRepository
exampleRepository) {

        this.exampleRepository = exampleRepository

        }
}
```

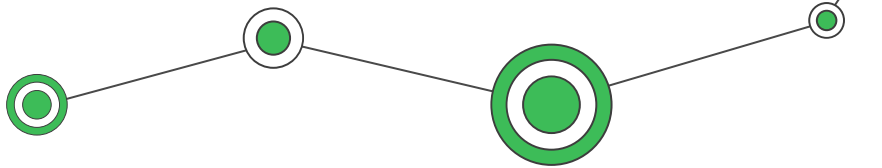
O que o Spring nos oferece?

- **Injeção de dependencias:**
  - Anotação @Autowired;

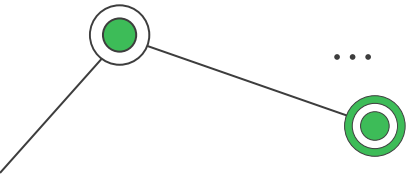
```
@Service
public class ExampleService {

        @Autowired
        private ExampleRepository exampleRespository;

        // construtor, getters e setters

}
```

O que o Spring nos oferece?

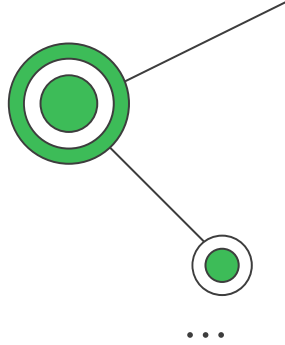
- Anottations
- Bean Scopes
- Annotation-based Container Configuration
- Environment Abstraction
- Spring Expression Language (SpEL)
- AOP Proxies



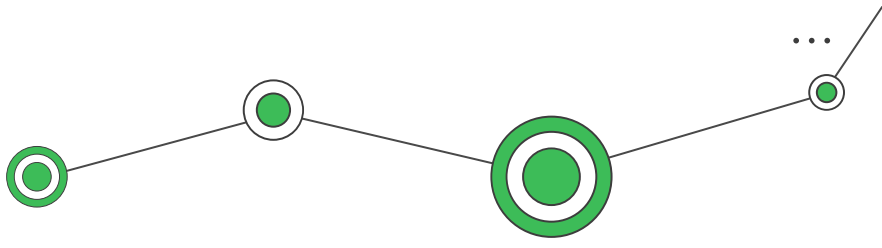
04
Desenvolvendo Aplicações



Como começar?



- https://start.spring.io/



Desenvolvimento de endpoints RESTful

- O desenvolvimento de APIs RESTful é essencial para criar aplicativos modernos e interoperáveis
- O Spring Boot facilita a criação de endpoints RESTful, permitindo expor funcionalidades do aplicativo por meio de URLs simples e padronizadas

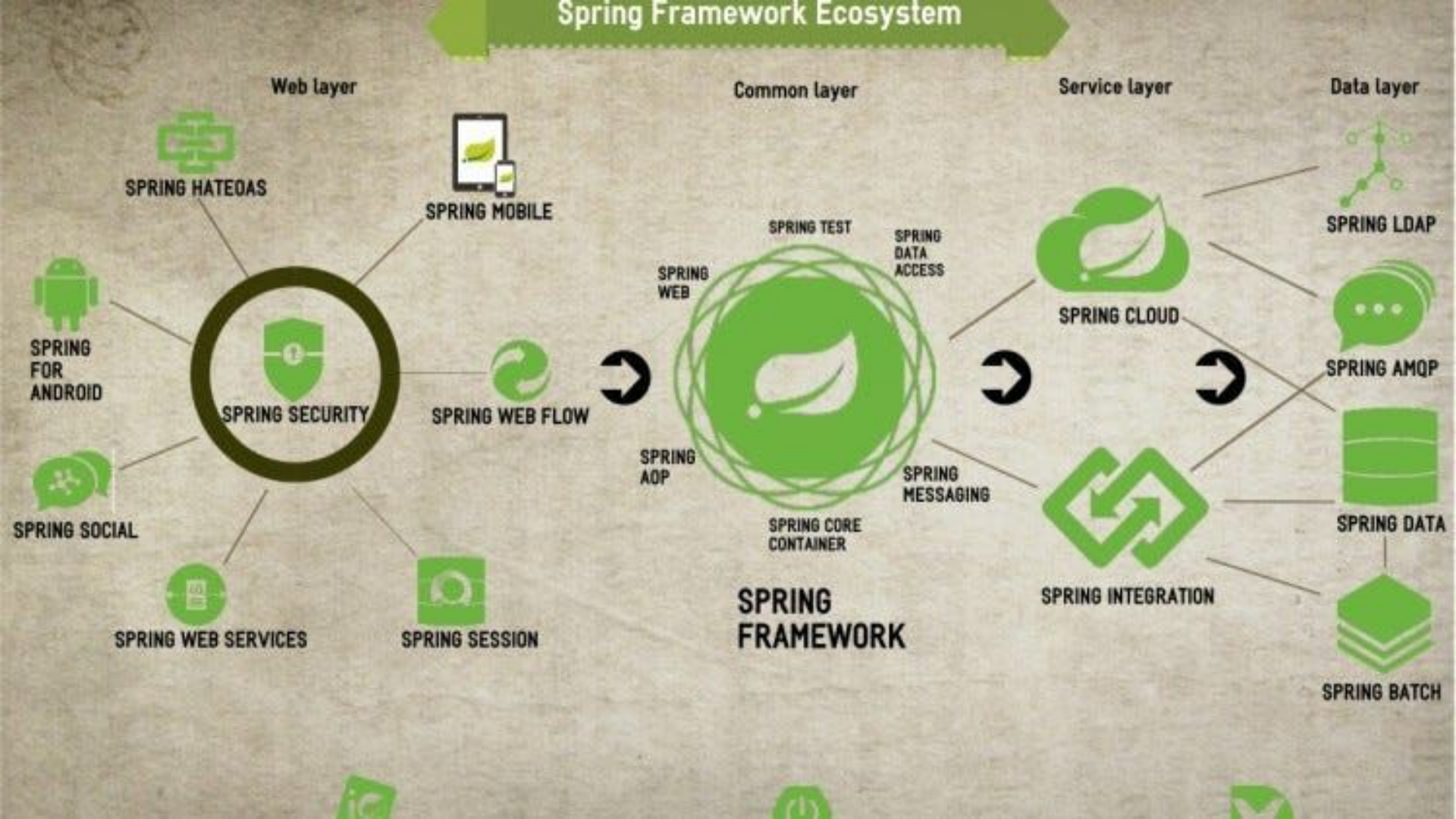
```java
@RestController
@RequestMapping("/api")
public class ExemploController {

    @GetMapping("/recurso")
    public ResponseEntity<String> obterRecurso() {
        String recurso = "Este é um exemplo de resposta RESTful.";
        return ResponseEntity.ok(recurso);
    }
}
```

Testes Automatizados

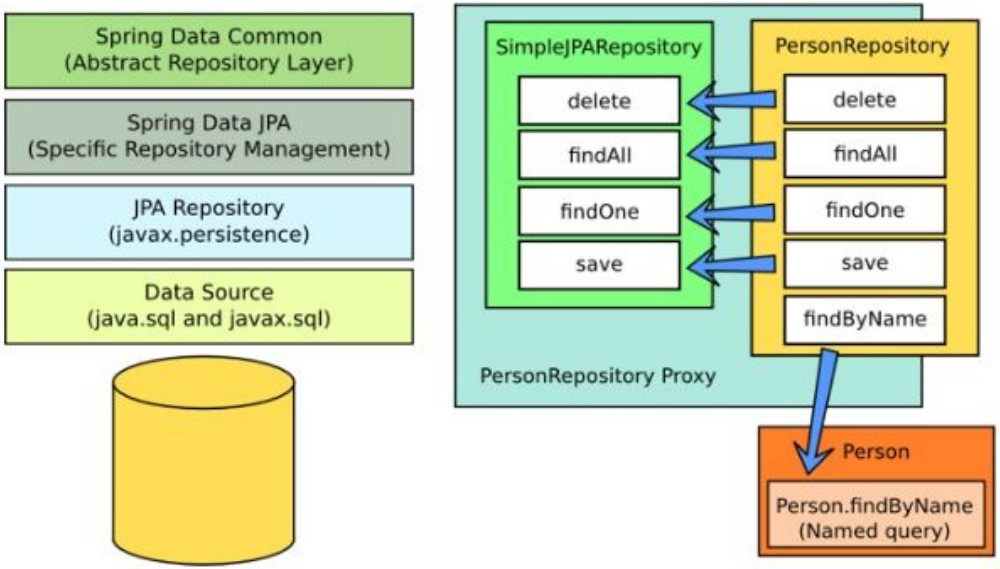
- Os testes automatizados são fundamentais para garantir a qualidade do software.
- O Spring Boot oferece suporte para escrever testes automatizados, incluindo testes de unidade e testes de integração.

```java
@RunWith(SpringRunner.class)
@SpringBootTest
public class ExemploControllerTest {

    @Autowired
    private MockMvc mockMvc;

    @Test
    public void testObterRecurso() throws Exception {
        mockMvc.perform(get("/api/recurso"))
            .andExpect(status().isOk())
            .andExpect(content()
                .string("Este é um exemplo de resposta RESTful."));
    }
}
```

05
Ecossistemas e Integrações

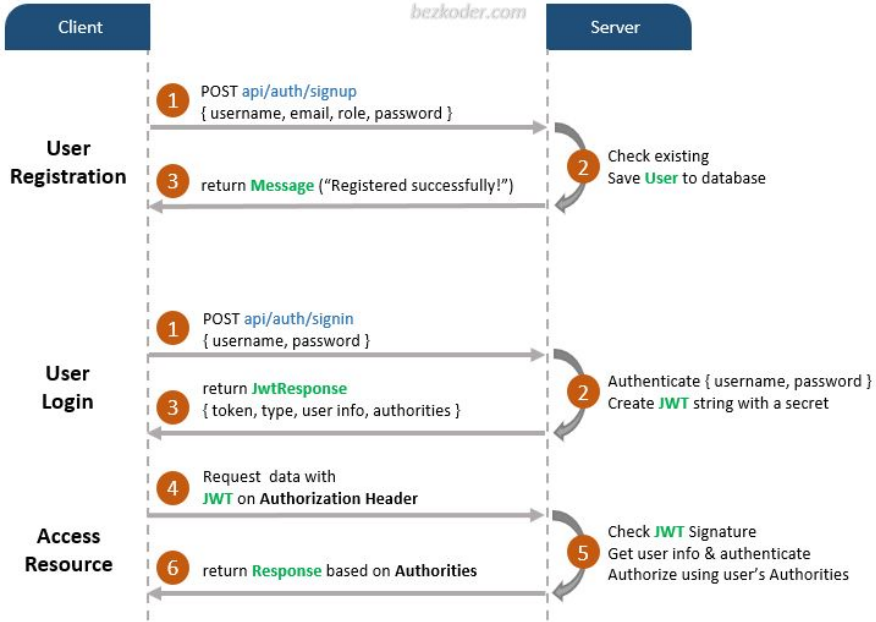

Spring Framework Ecosystem
Web layer
Common layer
Service layer
Data layer
SPRING HATEOAS
SPRING MOBILE
SPRING FOR ANDROID
SPRING SECURITY
SPRING WEB FLOW
SPRING SOCIAL
SPRING WEB SERVICES
SPRING SESSION
SPRING TEST
SPRING DATA ACCESS
SPRING WEB
SPRING AOP
SPRING MESSAGING
SPRING CORE CONTAINER
SPRING FRAMEWORK
SPRING CLOUD
SPRING INTEGRATION
SPRING LDAP
SPRING AMQP
SPRING DATA
SPRING BATCH

Spring Data


Spring Data Common
(Abstract Repository Layer)
Spring Data JPA
(Specific Repository Management)
JPA Repository
(javax.persistence)
Data Source
(java.sql and javax.sql)
SimpleJPARepository
delete
findAll
findOne
save
PersonRepository
delete
findAll
findOne
save
findByName
PersonRepository Proxy
Person
Person.findByName
(Named query)

```java
@Entity
public class Produto {

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

    private String nome;
    private BigDecimal preco;

    // getters e setters
}

@Repository
public interface ProdutoRepository extends JpaRepository<Produto, Long> {

    List<Produto> findByNome(String nome);
}
```

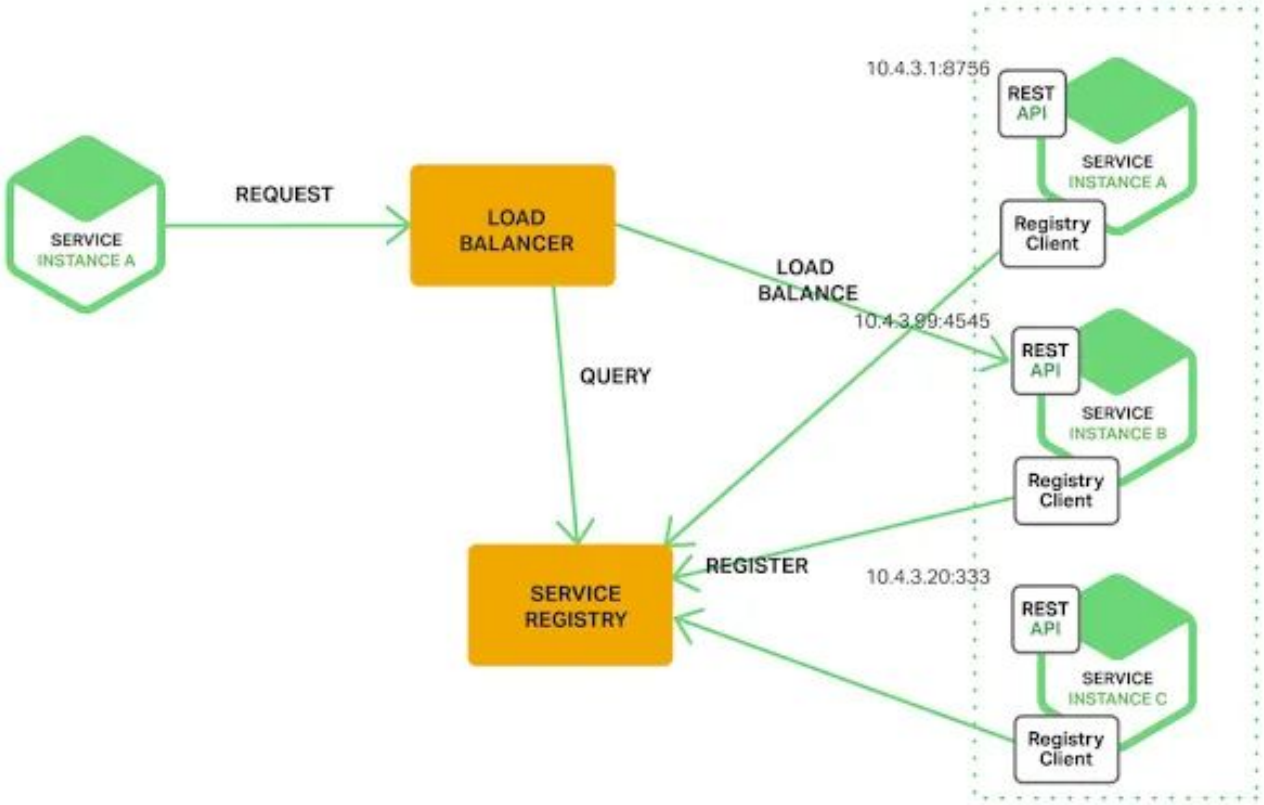
Spring Security




```java
@Configuration
@EnableWebSecurity
public class WebSecurityConfig {

	@Bean
	public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
		http
			.authorizeHttpRequests((requests) -> requests
				.requestMatchers("/", "/home").permitAll()
				.anyRequest().authenticated()
			)
			.formLogin((form) -> form
				.loginPage("/login")
				.permitAll()
			)
			.logout((logout) -> logout.permitAll());

		return http.build();
	}

	@Bean
	public UserDetailsService userDetailsService() {
		UserDetails user =
			 User.withDefaultPasswordEncoder()
				.username("user")
				.password("password")
				.roles("USER")
				.build();

		return new InMemoryUserDetailsManager(user);
	}
}
```

```java
@Configuration
@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
            .authorizeRequests()
                .antMatchers("/api/recurso").authenticated()
                .anyRequest().permitAll()
            .and()
            .formLogin()
                .permitAll();
    }
```

Spring Cloud

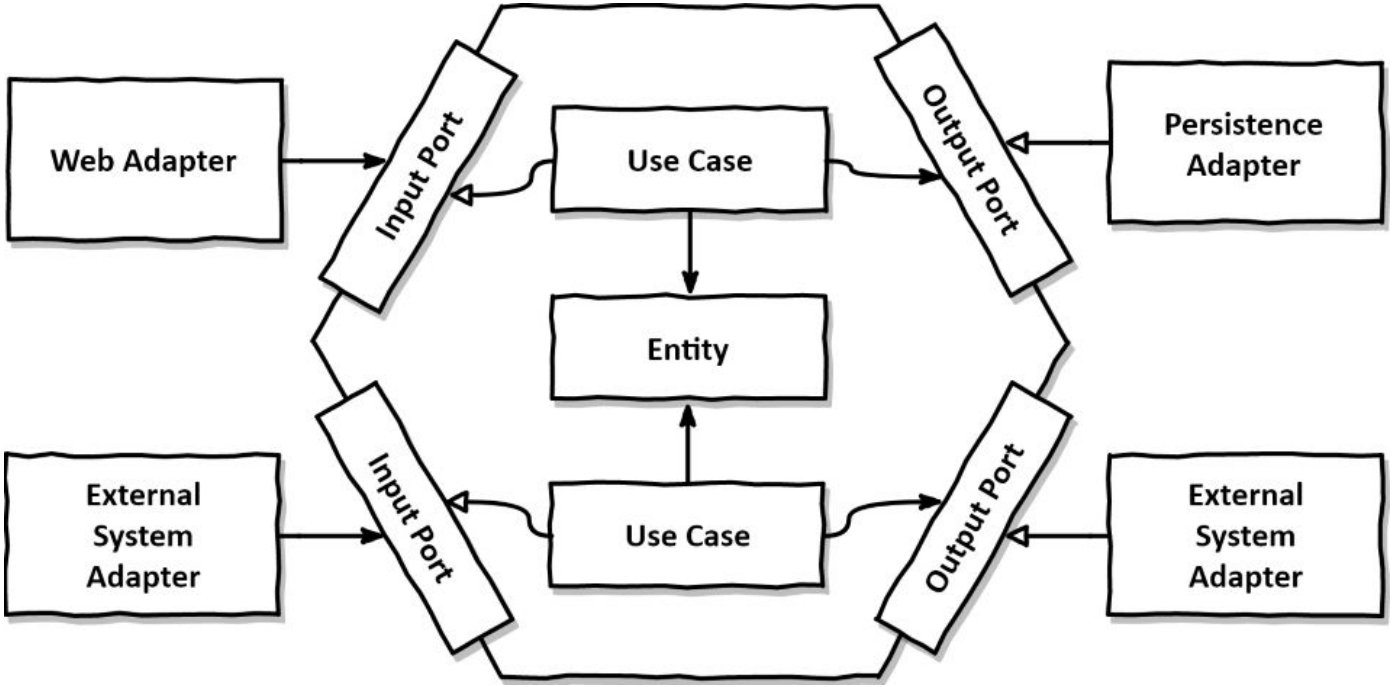


06
Arquiteturas e Padrões

Arquiteturas e Padrões

- SpringMVC
  - Artigo DevMedia sobre **SpringMVC**.

Arquitetura Hexagonal



Arquitetura Hexagonal

- Classes são divididas em dois principais grupos:
    - Classes de domínio e Classes de infraestrutura.
- Garante que as **classes de domínio não vão depender de classes relacionadas com infraestrutura, tecnologias ou sistemas externos**.
- Garante a construção de sistemas que favorecem a reusabilidade de código, alta coesão, baixo acoplamento, independência de tecnologia e que são mais fáceis de serem testados.
- https://www.devmedia.com.br/desenvolvendo-uma-aplicacao-web-com-spring-boot-e-spring-mvc/34122

Arquitetura Hexagonal

Fique por dentro com Código Fonte TV!

07

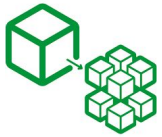
Casos de Uso

Usos comuns e setores que se beneficiam do Spring

Desenvolvimento Web

Recursos abrangentes para criar aplicativos web escaláveis

* Desenvolvimento de controladores
* Injeção de dependência
* Integração com frameworks de visualização



Microservices

Spring Cloud

* Facilita a criação, configuração e descoberta de serviços
* Fornece recursos para balanceamento de carga, tolerância a falhas e monitoramento de microservices

APIs RESTful

* Suporte para mapeamento de endpoints
* Gestão de versões da API
* Serialização e desserialização de objetos JSON
* Autenticação e autorização

Usos comuns e setores que se beneficiam do Spring

Integração de sistemas

* Suporte para integração de serviços SOAP, RESTful e JMS
* Suporte a integração de bancos de dados, mensageria e sistemas de arquivos

Aplicativos empresariais

* Oferece recursos para controle transacional
* Gerenciamento de cache
* Segurança
* Agendamento de tarefas
* Entre outros

Setor financeiro

Capacidade de lidar com requisitos complexos

* Transações seguras
* Integração com sistemas legado
* Conformidade regulatória

Empresas que usam Spring

Airbnb (Desenvolvimento Web)

Site para encontrar e reservar acomodações

PayPal (API RESTful)

Troca de informações de forma segura entre dois sistemas de computador

Netflix (Microservices)

Criação de um aplicativo com componentes independentes, habilitando a inovação

Cisco (Integração de Sistemas)

União de sistemas e softwares de estoque, pagamentos, entre outros de maneira mais rápida e eficiente entre os setores

Volkswagen (App Empresarial)

Controle transacional, gerenciamento de cache e integração de sistemas

Citibank (Setor financeiro)

Recursos de segurança avançada, controle transacional confiável e integração com sistemas de pagamento e serviços bancários

Atividade

Link para a atividade sobre a apresentação:

GitHub

Obrigado!

Understanding the Problem

Neptune

Neptune is the farthest planet from the Sun

...

Mercury

Mercury is the smallest planet in the Solar System

...

Jupiter

Jupiter is the biggest planet in the Solar System

...

Our Solutions

01 Mercury

Mercury is the closest planet to the sun

02 Venus

Venus is the second planet from the sun

03 Jupiter

Jupiter is the biggest planet of them all

04 Neptune

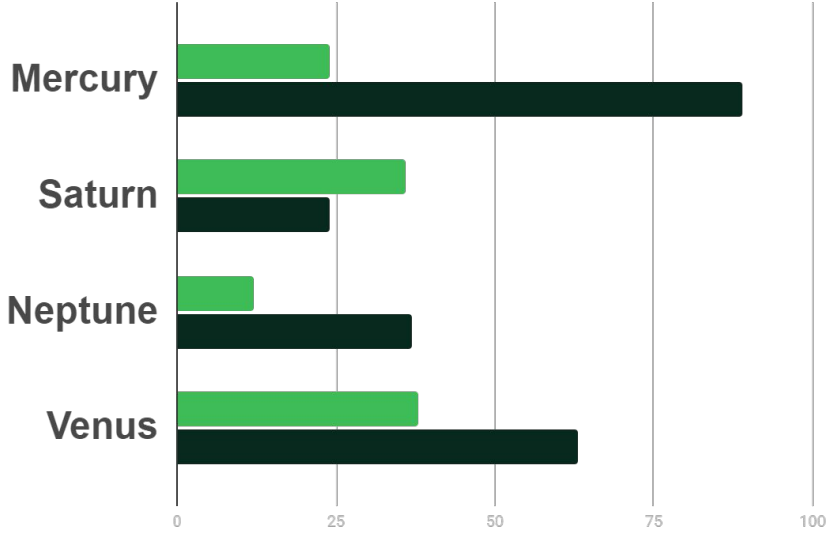
Neptune is the farthest planet from the Sun

Main Competitors

	Mercury	Jupiter	Venus	Neptune
	✗	✗	✗	✗
	✗	✓	✗	✓
	✓	✗	✓	✗
	✓	✓	✗	✗

Market Research

Evolution



To modify this graph, click on it, follow the link, change the data and paste the resulting graph here

$20K

Gross Revenue in the Sector

10%

Sector Growth

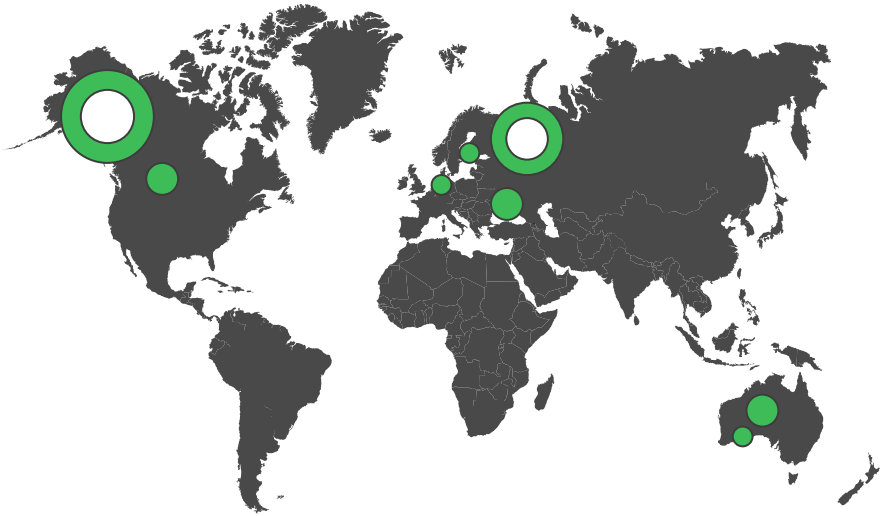
“This is a quote, words full of wisdom that someone important said and can make the reader get inspired.”

—Someone Famous

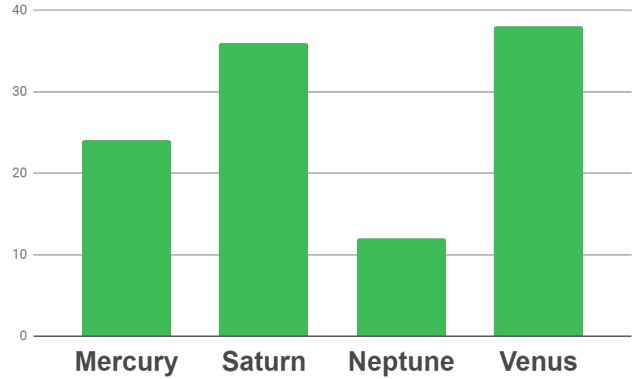
Awesome Words

Analysis

Outreach



Top Rated Values



To modify this graph, click on it, follow the link, change the data and paste the new graph here

333,000.00

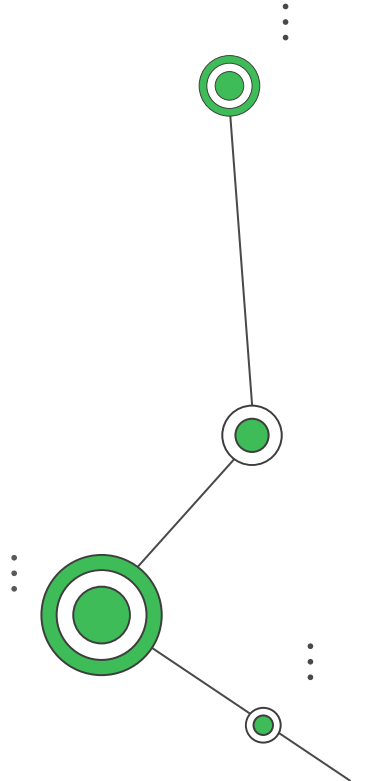
earths is the sun's mass

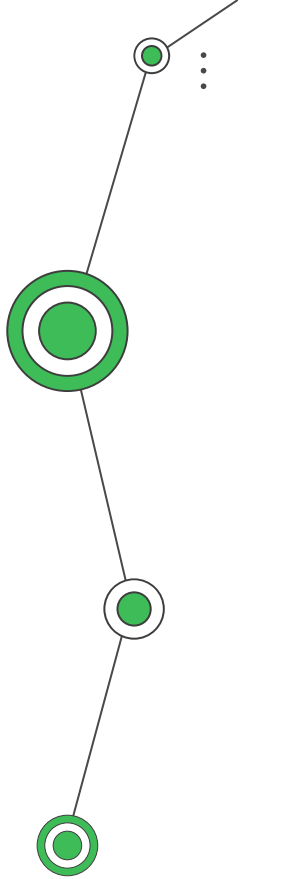
24h 37m 23s

is Jupiter's rotation period

386,000 km

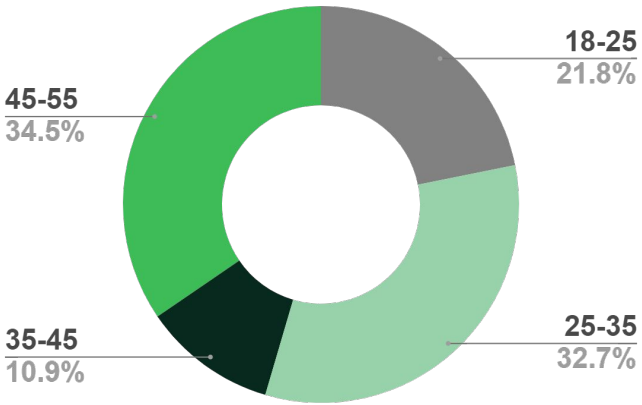
is the distance to the Moon





Target

Age



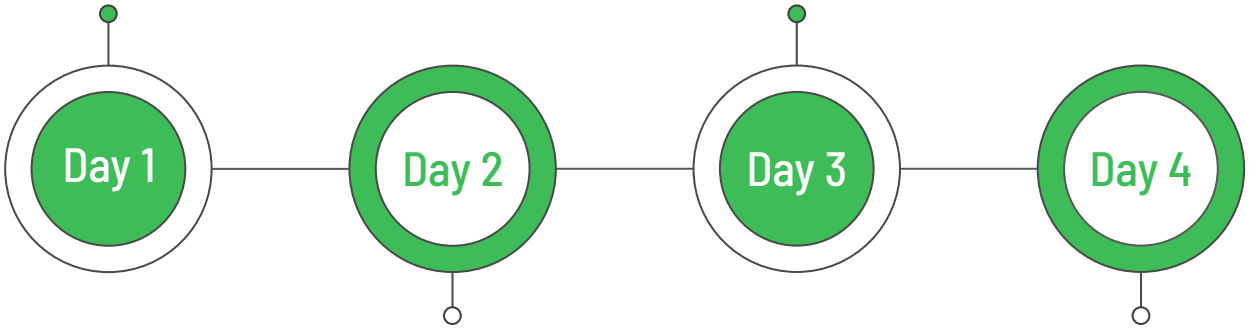
To modify this graph, click on it, follow the link, change the data and paste the new graph here

Gender

Interests

- Security
- Usability
- Interface

Our Process


Mercury is the closest planet to the Sun
Day 1
Day 2
Jupiter is the biggest planet in the Solar System
Venus is the second planet from the Sun
Day 3
Day 4
Neptune is the farthest planet from the Sun

Whoa!

This could be the part of the presentation where you can introduce yourself, write your email...

Our Consultants

John James

Venus is the second planet from the Sun

Jane Patterson

Mercury is the closest planet to the Sun

Joe Doe

Neptune is the farthest planet from the Sun

Our Partners

Mercury

Mercury is the closest planet to the Sun

Jupiter

Jupiter is the biggest planet of them all

Neptune

Neptune is the farthest planet from the Sun

Saturn

Saturn is the ringed one and a gas giant

Mars

Despite being red, Mars is actually a cold place

Venus

Venus is the second planet from the Sun

Testimonials

Helena James

“Mercury is the closest planet to the Sun”

Jenna Doe

“Jupiter is the biggest planet of them all “

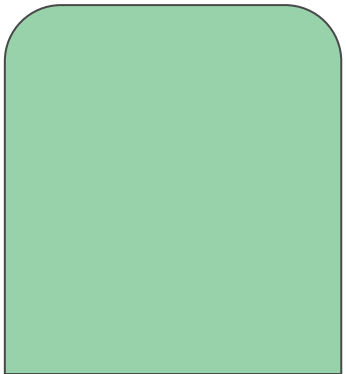
Mary Patterson

“Neptune is the farthest planet from the Sun”

Awards

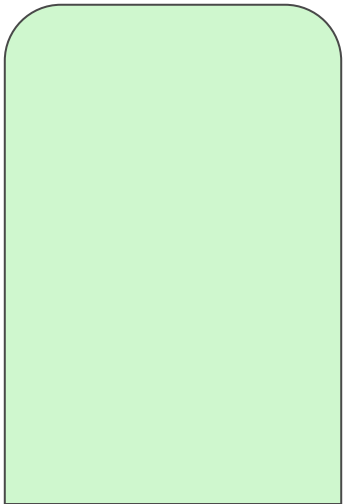
Venus

Venus is the second planet from the Sun



Mercury

It’s the closest planet to the Sun

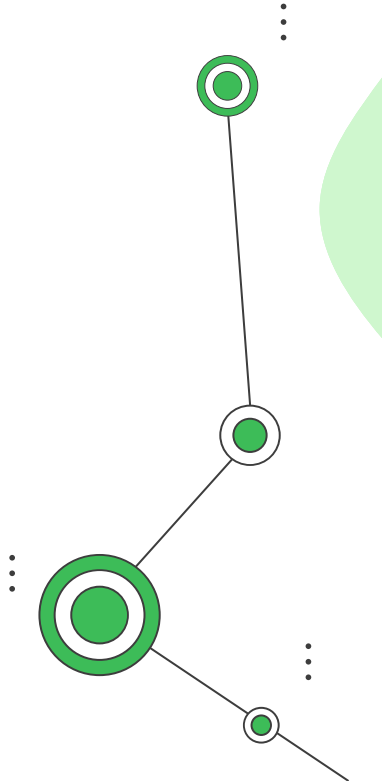


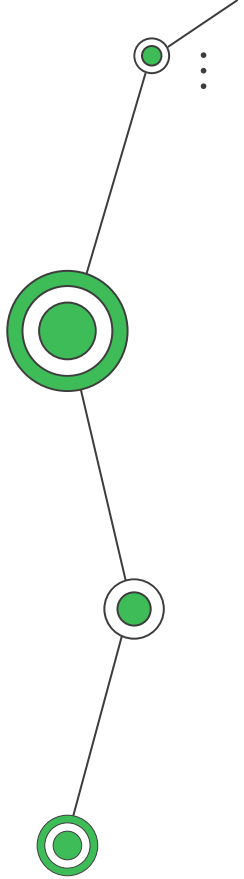
Jupiter

Jupiter is the biggest planet of them all

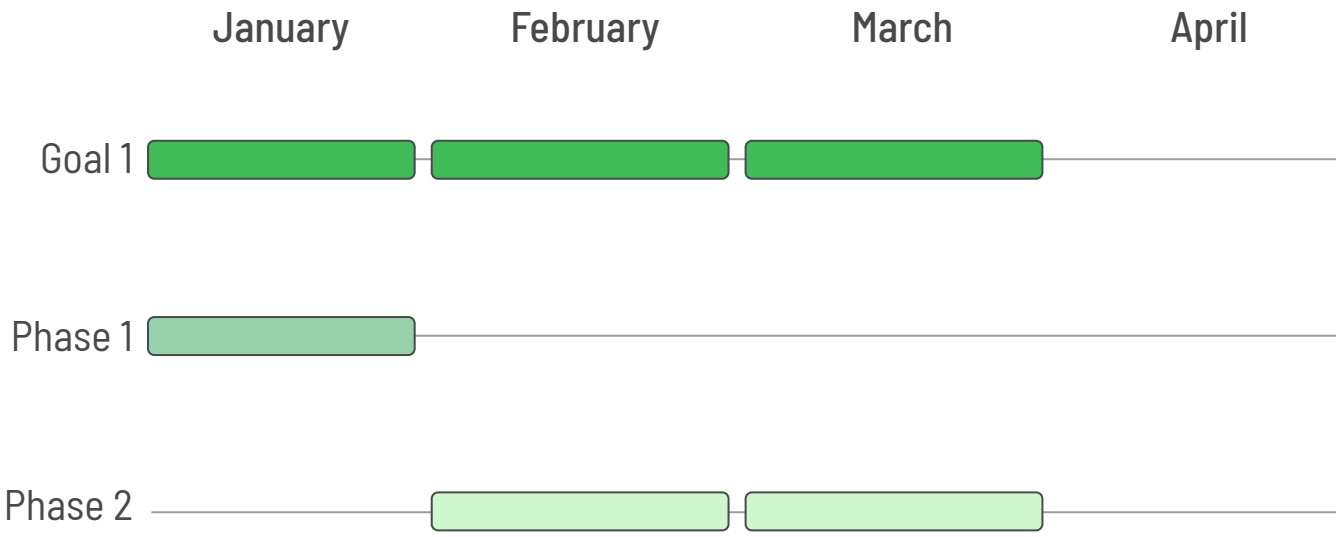
498,300,000

Big numbers catch your
audience's attention





Upcoming Goals



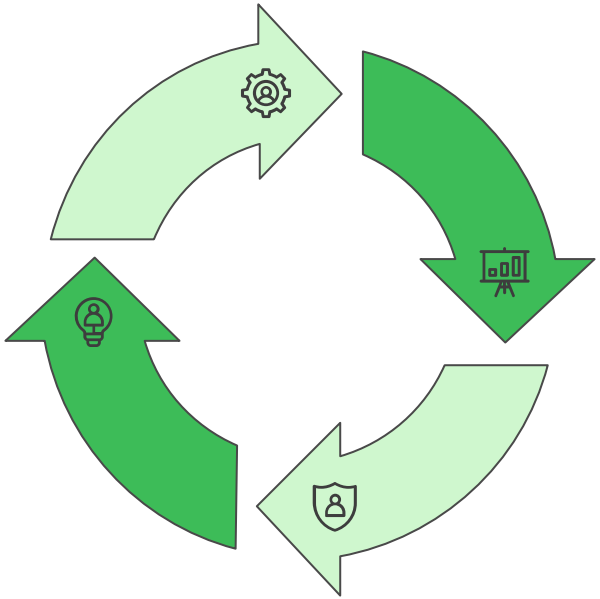
Infographics Make Your Idea Understandable...

Mercury

Mercury is the closest planet to the Sun

Jupiter

Jupiter is the biggest planet of them all



Neptune

Neptune is the farthest planet from the Sun

Saturn

Saturn is the ringed one and a gas giant

...And the Same Goes for Tables

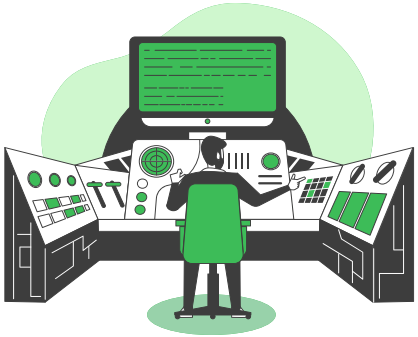
	Mass (earths)	Diameter (earths)	Gravity (earths)
Mercury	0.06	0.38	0.38
Mars	0.11	0.53	0.38
Saturn	95.2	9.4	1.16

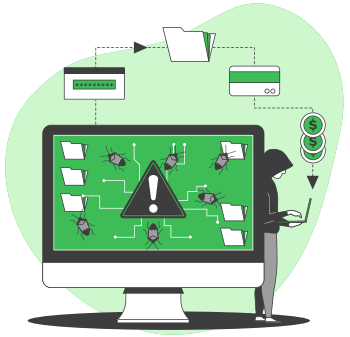
Thanks!

Do you have any questions?

youremail@freepik.com
+91 620 421 838
yourcompany.com

Alternative Resources





Find more illustrations like these on Stories by Freepik

Resources

Vectors

- Programming
- Development
- Mission impossible
- Social networking
- Segmentation
- Site stats
- Responsive
- Meeting
- Analysis
- Cyber attack
- Secure server
- Control panel

Photos

- Close-up of woman using a laptop
- Side view of a happy woman holding laptop on grey background

Icons

- Project Management Icon

Find more illustrations like these on
Stories by Freepik

Fonts & colors used

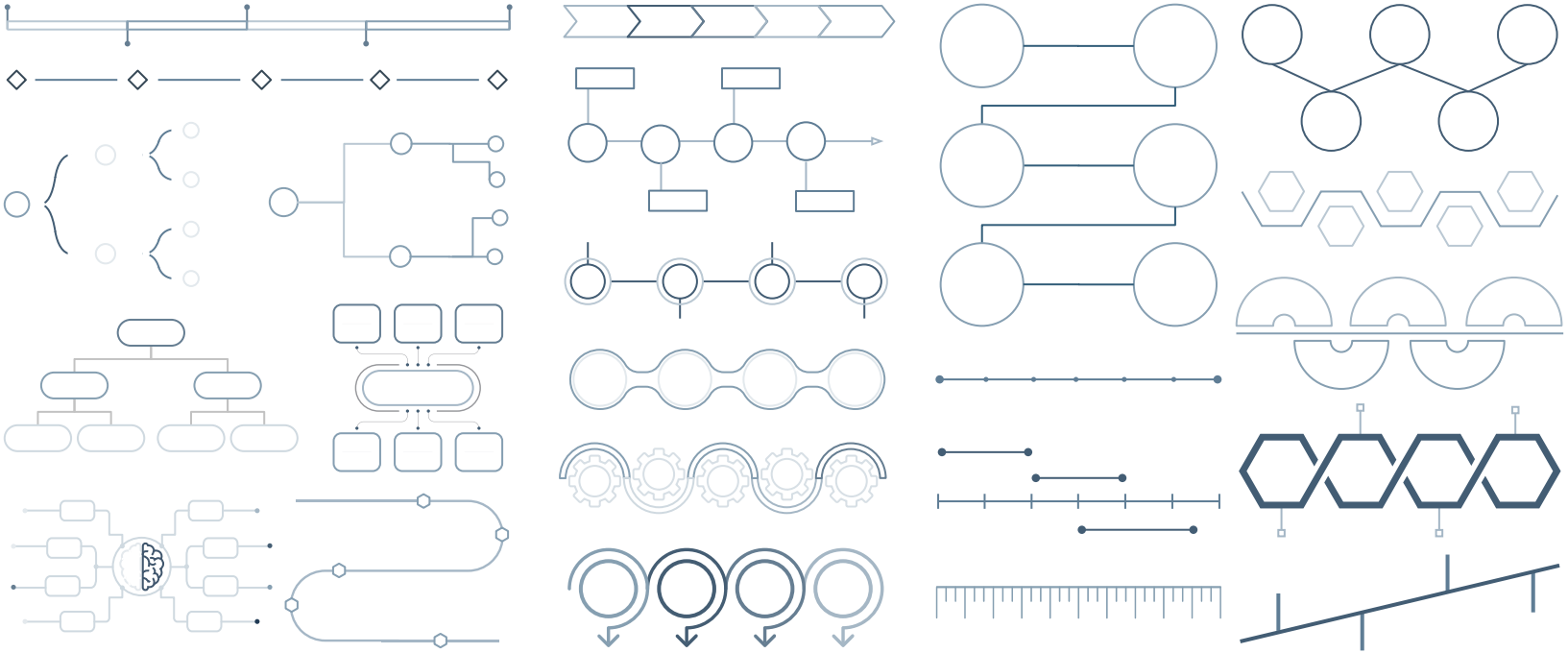
This presentation has been made using the following fonts:

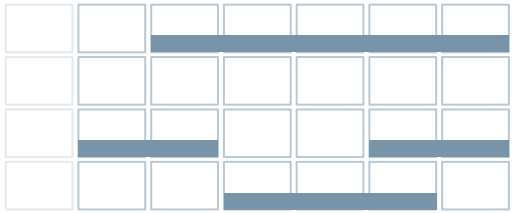
Fjalla One
(https://fonts.google.com/specimen/Fjalla+One)

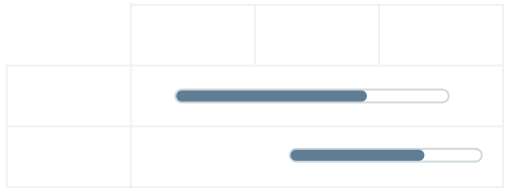
Barlow Condensed
(https://fonts.google.com/specimen/Barlow+Condensed)

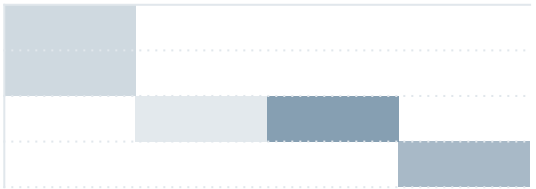
Use our editable graphic resources...

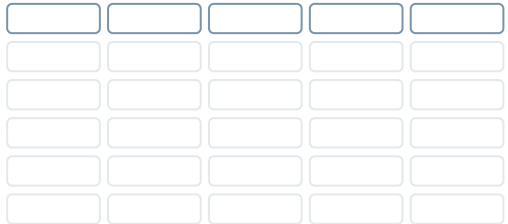
You can easily resize these resources without losing quality. To change the color, just ungroup the resource and click on the object you want to change. Then, click on the paint bucket and select the color you want. Group the resource again when you're done. You can also look for more infographics on Slidesgo.

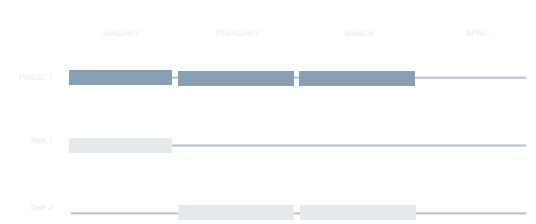


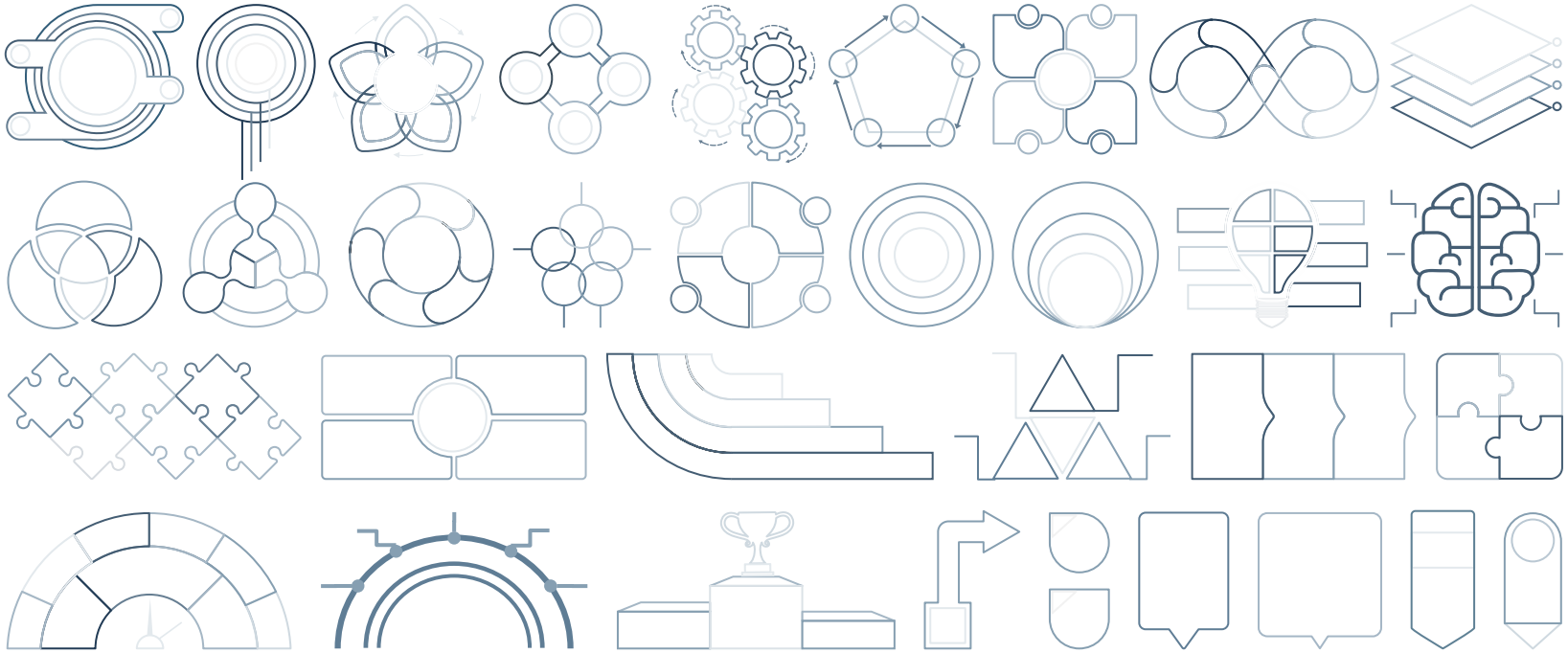


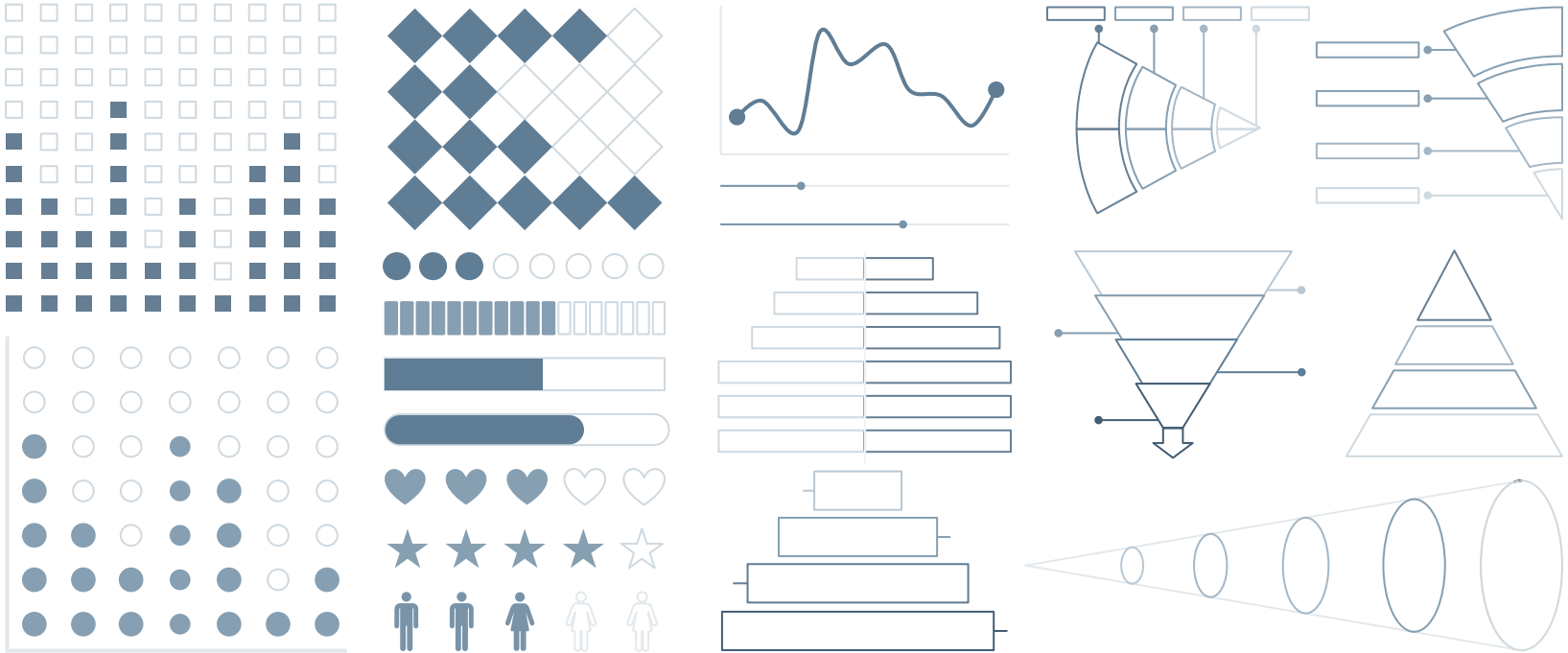







JANUARY
FEBRUARY
MARCH
APRIL
PHASE 1
Task 1
Task 2





...and our sets of editable icons

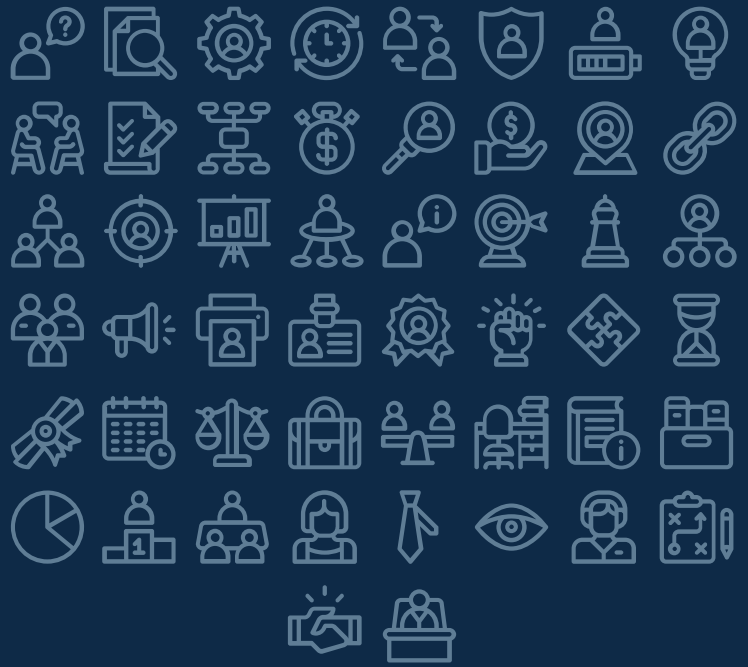
You can resize these icons without losing quality.
You can change the stroke and fill color; just select the icon and click on the paint bucket/pen.
In Google Slides, you can also use Flaticon's extension, allowing you to customize and add even more icons.

Educational Icons

Medical Icons

Business Icons

Teamwork Icons



Help & Support

Avatar Icons

Creative Process Icons

Performing Arts Icons

Nature Icons

SEO & Marketing Icons


slidesgo