La no repository já tem o metodo chamado save

\*\* Lembrar que no repositório não recebemos um objeto e sim um DTO

Então precisamos converter o objeto para DTO

@PostMapping // Estou dizendo que o tipo de dado que vai ter que chegar tem que ser um POST  
public void cadastrar(@RequestBody DadosCadastroMedicos dados) { // RequestBody quer dizer: Spring pegue os dados do corpo da requisição!  
 repository.save(new Medico(null, dados.nome(), dados.email(), dados.crm(), new Endereco()));

Ficaria mais ou menos dessa forma, como o ID o banco que vai autoincrementar de 1 em 1, podemos colocar ele como nulo, e depois passar o resto dos dados, mas tem outra forma de fazer mais fácil...

Podemos criar um construtor dentro do medico.

@PostMapping // Estou dizendo que o tipo de dado que vai ter que chegar tem que ser um POST

@Transactional // Esse metodo é porque vamos ter uma transação em escrita entao precisamos ter.  
public void cadastrar(@RequestBody DadosCadastroMedicos dados) { // RequestBody quer dizer: Spring pegue os dados do corpo da requisição!  
 repository.save(new Medico(dados));  
}

Para passar esses dados.

public Medico(DadosCadastroMedicos dados) {  
 this.nome = dados.nome();  
 this.email = dados.email();  
 this.crm = dados.crm();  
 this.especialidade = dados.especialidade();  
 this.endereco = new Endereco(dados.endereco());  
  
}

La na classe endereço precisamos criar um construtor que recebe os dados.

public Endereco(DadosEndereco dados) {  
 this.logradouro = dados.logradouro();  
 this.bairro = dados.bairro();  
 this.cep = dados.cep();  
 this.numero = dados.numero();  
 this.complemento = dados.complemento();  
 this.cidade = dados.cidade();  
 this.uf = dados.uf();  
  
 }  
}

Agora no nosso método cadastrar podemos fazer assim

@PostMapping // Estou dizendo que o tipo de dado que vai ter que chegar tem que ser um POST  
 public void cadastrar(@RequestBody DadosCadastroMedicos dados) { // RequestBody quer dizer: Spring pegue os dados do corpo da requisição!  
 repository.save(new Medico(dados));  
 }  
}