

Documentação da Extração - Microsserviço de Consultas

1. Bounded Context Escolhido

Contexto Selecionado: GESTÃO DE CONSULTAS MÉDICAS

Justificativa da Seleção:

- Alta taxa de mudança nas regras médicas
- Ajustes frequentes em políticas de agendamento
- Requisitos regulatórios em constante evolução

Escalabilidade:

- Alta carga no agendamento exige escalabilidade independente
- Serviço de consultas pode ser escalado horizontalmente

Criticidade:

- Funcionalidade essencial do sistema de saúde
- Impacto direto na experiência do paciente
- Requisitos de alta disponibilidade e auditoria

2. Context Map Simple (PlantUML)

```
@startuml
!define RECTANGLE class

package "Monólito Original" {
    [Gestão de Usuários] as Users
    [Gestão de Horários] as Schedules
    [Gestão de Consultas] as ConsultasOld
}

package "Microsserviços" {
    [Auth Service] as Auth
    [User Service] as UserService
    [Consulta Service] as ConsultaService
}

Users --> Auth : Customer/Supplier
```

Documentação da Extração - Microserviço de Consultas

```
Users --> UserService : Customer/Supplier
Schedules --> ConsultaService : Customer/Supplier
ConsultaService --> Auth : Customer/Supplier
ConsultaService --> UserService : Customer/Supplier
```

```
package "Anti-Corruption Layer" {
    [MonolitoACL] as ACL1
    [ConsultaServiceACL] as ACL2
}
```

```
ConsultaService --> ACL1 : uses
UserService --> ACL2 : uses
@enduml
```

3. Linguagem Ubíqua e Modelagem Tática

Consulta (Aggregate Root):

```
@Entity
public class Consulta {
    private Long id;
    private LocalDateTime horario;
    private Long pacienteId;
    private Long profissionalId;
    private StatusConsulta statusConsulta;
    private String observacoes;
}
```

StatusConsulta (Value Object):

```
public enum StatusConsulta {
    AGENDADA,
    CANCELADA,
    CONCLUIDA
}
```

HorarioDTO (ACL):

```
public class HorarioDTO {
    private Long id;
    private LocalDateTime dataHoraInicio;
```

Documentação da Extração - Microserviço de Consultas

```
private LocalDateTime dataHoraFim;
private boolean disponivel;
private Long profissionalId;
}
```

4. Contrato do Microserviço

GET /api/consultas:

Listar todas as consultas com paginação (Autenticação obrigatória)

```
{
  "content": [
    {
      "id": 1,
      "horario": "2025-07-08T14:00:00",
      "pacienteId": 3,
      "profissionalId": 2,
      "statusConsulta": "AGENDADA",
      "observacoes": "Primeira consulta"
    }
  ],
  "totalElements": 1,
  "totalPages": 1,
  "size": 20,
  "number": 0
}
```

5. Integração com o Monólito

Exemplo de busca de horários disponíveis via RestTemplate:

```
private List<HorarioDTO> buscarHorariosDisponiveisDoMonolito(Long profissionalId) {
    String url = monolitoBaseUrl + "/horarios/profissionais/" + profissionalId +
"/horarios?size=100";

    ResponseEntity<PageResponseDTO<HorarioDTO>> response = restTemplate.exchange(
        url, HttpMethod.GET, new HttpEntity<>(new HttpHeaders()),
        new ParameterizedTypeReference<PageResponseDTO<HorarioDTO>>() {}
    );
    return response.getBody().getContent();
}
```

Documentação da Extração - Microserviço de Consultas

6. Benefícios da Extração

Técnicos:

- Isolamento de falhas
- Escalabilidade independente
- Deploy independente

Organizacionais:

- Equipes especializadas
- Velocidade de desenvolvimento
- Tecnologias específicas

Operacionais:

- Monitoramento granular
- Resiliência
- Manutenibilidade

7. Próximos Passos

- Event Sourcing
- Saga Pattern
- API Gateway
- Service Mesh
- Distributed Tracing
- Health Checks e Metrics