

Revisão da Arquitetura da API de Gestão Orçamentária

Diagram owner	@Infocal Technologies
Team	<input type="checkbox"/> @Thales Borba
Informed	<input type="checkbox"/> @Infocal Technologies <input type="checkbox"/> @Isaque Arifa
Status	IN PRODUCTION
Last date updated	Dec 23, 2023
On this page	<ul style="list-style-type: none">• Objetivos• Arquitetura• Estratégia de Deployment• Requisitos• Diagramas Entidade-Relacionamento• Rest API details• SLA• Action Items• References and documentation

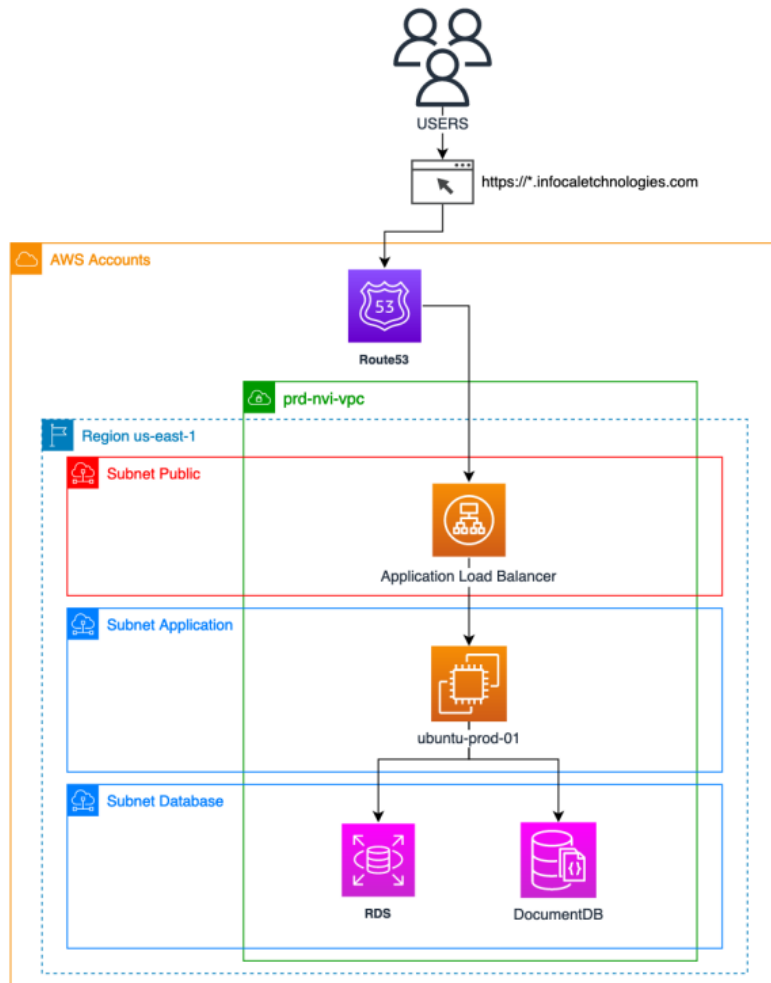
▼ Blueprint de Arquitetura Otimizada de Sistemas

Name	Description
Excelência Operacional	Capacidade de operar e monitorar a API para entregar valor ao negócio, com processos e procedimentos que suportam melhoria contínua. Isso inclui automação de implantações, monitoramento em tempo real e feedbacks rápidos para a equipe de desenvolvimento.
Segurança	Proteção de informações e sistemas através de avaliações de risco e estratégias de mitigação. Para uma API de gerenciamento de orçamento, isso significa garantir a segurança dos dados financeiros dos usuários, implementando autenticação forte, criptografia e outras medidas de segurança de dados.
Confiabilidade	A capacidade da API de se recuperar de interrupções na infraestrutura ou no serviço, adquirir recursos de computação dinamicamente para atender à demanda e mitigar interrupções, como má configurações ou problemas de rede transitórios. Isso pode incluir estratégias como redundância, backups e planejamento de desastres.
Eficiência de Desempenho	Uso eficiente dos recursos de computação para atender aos requisitos do sistema, mantendo essa eficiência à medida que a demanda muda e as tecnologias evoluem. Para uma API de gerenciamento de orçamento, isso pode envolver otimização de consultas de banco de dados, caching eficiente e escolha de arquiteturas adequadas para escalabilidade.
Otimização de Custos	Capacidade de operar sistemas para entregar valor ao negócio com o menor ponto de preço. Isso significa usar os recursos de forma eficiente, escolher as opções de computação e armazenamento mais custo-efetivas e monitorar continuamente os custos para identificar oportunidades de economia.

Objetivos

- **Excelência Operacional**
 - Garantir que a API seja operada e monitorada de maneira eficaz para fornecer valor ao negócio.
 - Implementar processos e procedimentos que suportem a melhoria contínua.
 - Automatizar implantações, garantir monitoramento em tempo real e fornecer feedback rápido à equipe de desenvolvimento.
- **Segurança**
 - Proteger as informações e sistemas relacionados à API.
 - Realizar avaliações de risco e implementar estratégias de mitigação.
 - Garantir a segurança dos dados financeiros dos usuários por meio de autenticação forte, criptografia e outras medidas de segurança de dados.
- **Confiabilidade**
 - Assegurar que a API possa se recuperar de interrupções na infraestrutura ou no serviço.
 - Adquirir recursos de computação dinamicamente para atender à demanda.
 - Mitigar interrupções por meio de estratégias como redundância, backups e planejamento de desastres.
- **Eficiência de Desempenho**
 - Utilizar eficientemente os recursos de computação para atender aos requisitos do sistema.
 - Manter a eficiência à medida que a demanda muda e as tecnologias evoluem.
 - Otimizar consultas de banco de dados, implementar caching eficiente e escolher arquiteturas adequadas para escalabilidade.
- **Otimização de Custos**
 - Operar sistemas de maneira a entregar valor ao negócio com o menor custo possível.
 - Utilizar recursos de forma eficiente e escolher opções de computação e armazenamento mais custo-efetivas.
 - Monitorar continuamente os custos para identificar oportunidades de economia.

Arquitetura



Fluxo da Arquitetura

1. O usuário acessa alguma URL
2. O Route53 valida a URL e o domínio acessado
3. O ALB recebe a requisição e redireciona para a aplicação correta com base na URL
4. Conecta ao banco de dados correspondente e suas tabelas.

🚀 Estratégia de Deployment

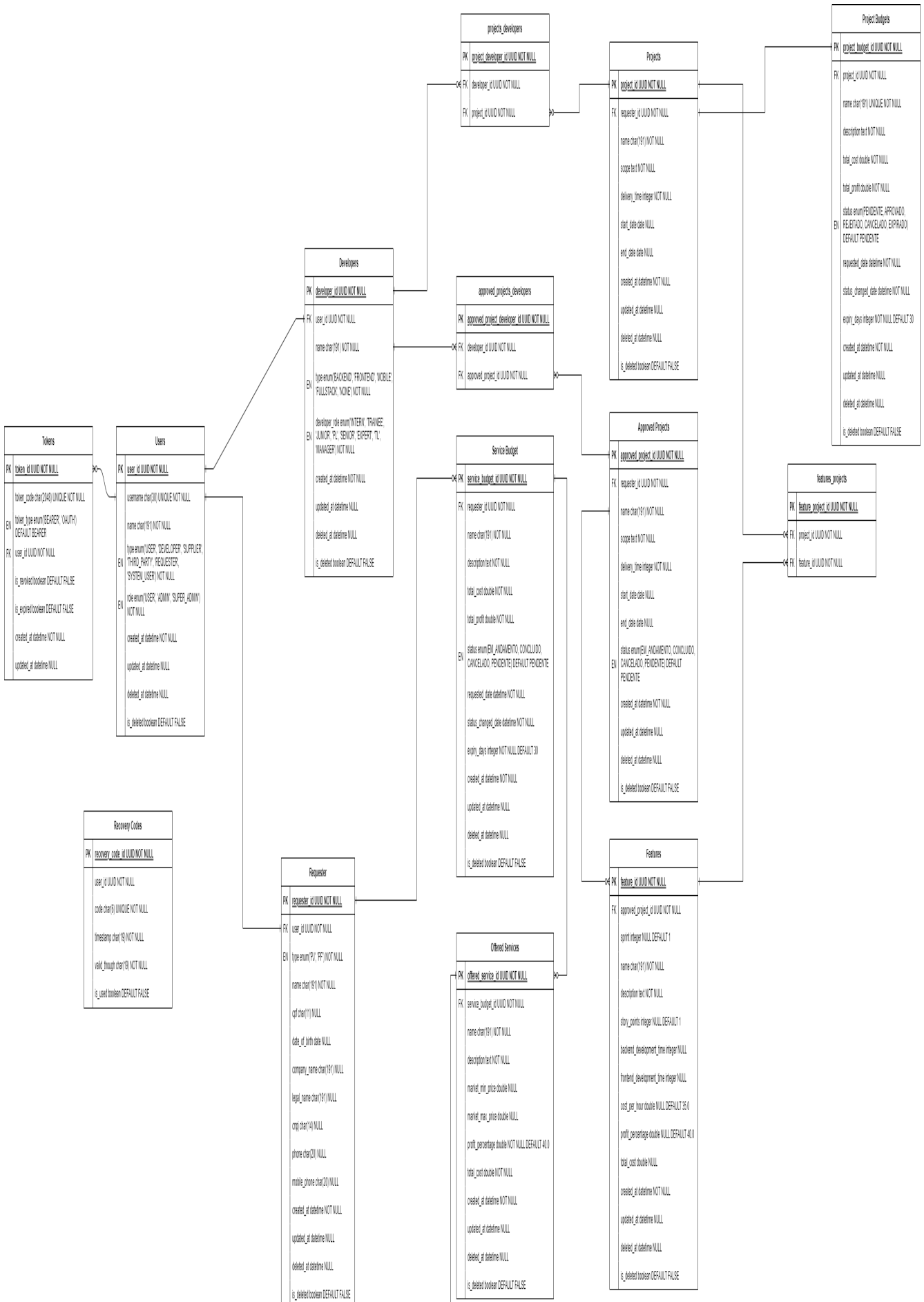
Deployment region: us-east-1

Requisitos

- **Proteção de Rotas:**

- Uma rota `admin` pode ser acessada tanto por um admin quanto por um super admin, uma rota `private` pode ser acessada apenas pelo administrador do sistema (super admin).
- **Developer:**
 - Para criar um desenvolvedor, este precisa ter um usuário ao qual será associado.
- **Requester:**
 - Para criar um *requester*, este precisa ter um usuário ao qual será associado.
 - Para solicitar um orçamento de projeto ou serviços, o requester precisa existir e estar devidamente associado a um usuário
- **Project Budget:**
 - O valor total do projeto é calculado com base no valor individual de cada feature presente no projeto pela média da porcentagem de lucro de todas as features.
 - Para a criação de um orçamento, é mandatório ter ao menos uma feature vinculada ao projeto
- **Feature:**
 - Cada feature possui um tempo (em minutos) de desenvolvimento para o backend e frontend separadamente, seu custo por hora e a porcentagem de lucro
 - O custo de cada feature é obtido convertendo o o tempo de desenvolvimento em horas, multiplicando pelo custo/hora e somado ao lucro.
- **Service Budget:**
 - Para criar um orçamento de serviço, um usuário envia um nome e uma descrição do(s) serviço(s) solicitado(s), um status de PENDENTE é atribuído à solicitação automaticamente, após devidamente precificado pelo administrativo é gerado um Service Budget para o conjunto de serviços ou serviço solicitado e o usuário recebe uma notificação por todos os meios de comunicação cadastrados (SMS/E-mail).
 - Para criar um serviço novo, podemos receber também uma lista de observações caso o usuário queira colocar detalhes sobre o serviço solicitado, a lista de observações deve ser criada antes, persistida e vinculada ao serviço antes de salvá-lo.

Diagramas Entidade-Relacionamento



Offered Service Observations	
PK	offered_service_observation_id UUID NOT NULL
FK	offered_service_id UUID NOT NULL
	observation_text NOT NULL
	observation_type enum (SERVICE_OFFERED, SERVICE_PRICE, SERVICE_QUALITY, SERVICE_DELIVERY, SERVICE_SUPPORT, SERVICE_DOCUMENTATION, SERVICE_TRAINING, SERVICE_MAINTENANCE, SERVICE_OTHER, SERVICE_NOT_APPLICABLE) NULL
	created_at datetime NOT NULL
	updated_at datetime NULL
	deleted_at datetime NULL
	is_deleted boolean DEFAULT FALSE

Rest API details

Authentication

POST /api/v1/auth/register

```

1 {
2   "username": "jonhhy_test",
3   "password": "password",
4   "email": "jonhhy_test@email.com"
5 }
```

POST /api/v1/auth/authenticate

```

1 {
2   "username": "S_ADMIN",
3   "password": "123456789"
4 }
```

Developer

POST /api/v1/admin/developers → Criar desenvolvedor (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

```

1 {
2   "name": "João do Teste",
3   "type": "BACKEND",
4   "developerRole": 3,
5   "userId": "8aa16ca9-a879-4de2-9ff4-18a171740fb6"
6 }
```

PUT /api/v1/admin/developers/{id} → Atualizar desenvolvedor (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

```

1 {
2   "name": "João do Teste",
3   "type": "BACKEND",
4   "developerRole": 3,
5   "userId": "8aa16ca9-a879-4de2-9ff4-18a171740fb6"
6 }
```

DELETE /api/v1/s_admin/developers/{id} → Deletar desenvolvedor (SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

GET /api/v1/admin/developers → Buscar todos os desenvolvedores (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

GET /api/v1/admin/developers/{id} → Buscar desenvolvedor (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

User - Budgets

POST /api/v1/user/budgets/projects → Criar uma nova solicitação de orçamento para projeto

Authorization Header: Bearer token

```
1  {
2    "requesterId": "c59b87c3-8961-4e7a-87e6-6f231b9ec7ae",
3    "project": {
4      "name": "Meu Projeto",
5      "scope": "Escopo do Projeto",
6      "deliveryTime": 15,
7      "startDate": "2024-01-01",
8      "endDate": "2024-01-15"
9    },
10   "features": {
11     "existingFeatureIds": ["1b3a6062-84e0-4d3f-a8e1-058d6cc095e7", "2b4b709a-24c2-4d29-b8a3-0f0d5e8e65a0"],
12     "newFeatures": [
13       {
14         "name": "Nova Feature 1",
15         "description": "Descrição da Nova Feature 1"
16       },
17       {
18         "name": "Nova Feature 2",
19         "description": "Descrição da Nova Feature 2"
20       }
21     ]
22   },
23   "projectBudget": {
24     "name": "Orçamento do Projeto",
25     "description": "Descrição do Orçamento"
26   }
27 }
```

Payload de resposta (ProjectCreationResponseDTO):

```

1 {
2   "requesterId": "a1b2c3d4-5678-90e1-f2a3b4c5d6e7",
3   "project": {
4     "projectId": "c59b87c3-8961-4e7a-87e6-6f231b9ec7ae",
5     "name": "Nome do Projeto",
6     "scope": "Escopo do Projeto",
7     "deliveryTime": 30,
8     "startDate": "01/01/2024 12:00:00",
9     "endDate": "01/02/2024 12:00:00",
10    "developers": [],
11    "createdAt": "01/01/2024 12:00:00"
12  },
13  "features": [
14    {
15      "featureId": "d8e9f0a1-b2c3-4d5e-6f7a-8b9c0d1e2f3",
16      "name": "Nome da Feature",
17      "description": "Descrição da Feature",
18      "createdAt": "01/01/2024 12:00:00"
19    }
20  ],
21  "projectBudget": {
22    "projectBudgetId": "f2a3b4c5-d6e7-8f9a-0b1c2d3e4f5",
23    "name": "Orçamento do Projeto",
24    "description": "Descrição do Orçamento",
25    "status": "IN_PROGRESS",
26    "requestedDate": "01/01/2024 12:00:00",
27    "expiryDays": 30,
28    "isDeleted": false,
29    "createdAt": "01/01/2024 12:00:00"
30  }
31 }

```

POST /api/v1/user/budgets/services → Criar uma nova solicitação de orçamento para serviços

Authorization Header: Bearer token

```

1 {
2   "requesterId": "c59b87c3-8961-4e7a-87e6-6f231b9ec7ae",
3   "services": {
4     "existingServiceIds": [
5       "1b3a6062-84e0-4d3f-a8e1-058d6cc095e7",
6       "2b4b709a-24c2-4d29-b8a3-0f0d5e8e65a0"
7     ],
8     "newServices": [
9       {
10        "serviceName": "service 1",
11        "serviceDescription": "description 1",
12        "observations": [
13          {
14            "observation": "Alguma observation",
15            "observationType": "SERVICE_REQUEST"
16          }
17        ]
18      },
19      {
20        "serviceName": "service 2",
21        "serviceDescription": "description 2",

```



```

22         "observations": [
23             {
24                 "observation": "Alguma observation",
25                 "observationType": "SERVICE_REQUEST"
26             }
27         ]
28     }
29 ]
30 },
31 "serviceBudget": {
32     "name": "Orçamento de Serviços",
33     "description": "Descrição do Orçamento"
34 }
35 }

```

Payload de resposta (ServiceCreationResponseDTO):

```

1  {
2      "requesterId": "a1b2c3d4-5678-90e1-f2a3b4c5d6e7",
3      "services": [
4          {
5              "offeredServiceId": "6d7e8f9a-0b1c-2d3e-4f5g-6h7i8j9k0l1",
6              "serviceName": "Nome do Serviço 1",
7              "serviceDescription": "Descrição do Serviço 1",
8              "observations": [
9                  {
10                     "offeredServiceObservationId": "a1b2c3d4-5678-90e1-f2a3b4c5d6e7",
11                     "observation": "Observação do Serviço 1",
12                     "observationType": "TIPO_OBSERVACAO",
13                     "isDeleted": false,
14                     "createdAt": "01/01/2024 12:00:00"
15                 }
16             ],
17             "isDeleted": false,
18             "createdAt": "01/01/2024 12:00:00"
19         },
20         {
21             "offeredServiceId": "b2c3d4e5-6f7a-8b9c-0d1e2f3g4h5",
22             "serviceName": "Nome do Serviço 2",
23             "serviceDescription": "Descrição do Serviço 2",
24             "observations": [
25                 {
26                     "offeredServiceObservationId": "c4d5e6f7-a8b9-0c1d-2e3f-4g5h6i7j8k9l0",
27                     "observation": "Observação do Serviço 2",
28                     "observationType": "TIPO_OBSERVACAO",
29                     "isDeleted": false,
30                     "createdAt": "01/01/2024 12:00:00"
31                 }
32             ],
33             "isDeleted": false,
34             "createdAt": "01/01/2024 12:00:00"
35         }
36     ],
37     "serviceBudget": {
38         "serviceBudgetId": "d4e5f6a7-8b9c-0d1e-2f3g-4h5i6j7k8l9",
39         "name": "Orçamento do Serviço",
40         "description": "Descrição do Orçamento do Serviço",
41         "status": "IN_PROGRESS",

```

```
42     "requestedDate": "01/01/2024 12:00:00",
43     "expiryDays": 30,
44     "isDeleted": false,
45     "createdAt": "01/01/2024 12:00:00"
46   }
47 }
```

PATCH /api/v1/user/budgets/{id}/approve → Aprovar um pedido de orçamento

Authorization Header: Bearer token

Body: N/A

PATCH /api/v1/user/budgets/{id}/reject → Rejeitar um pedido de orçamento

Authorization Header: Bearer token

Body: N/A

PATCH /api/v1/user/budgets/{id}/cancel → Cancelar um pedido de orçamento

Authorization Header: Bearer token

Body: N/A

GET /api/v1/user/budgets/{id} → Buscar um pedido de orçamento existente (de projetos e de serviços)

Authorization Header: Bearer token

Body: N/A

Obs.: No payload de resposta, colocar o tipo de orçamento que é: serviço ou projeto com base em qual entidade foi recuperado

GET /api/v1/user/budgets → Buscar todos os pedidos de orçamento existentes (de projetos e de serviços)

Authorization Header: Bearer token

Body: N/A

Obs.: No payload de resposta, colocar em cada orçamento o tipo de orçamento que é: serviço ou projeto com base em qual entidade foi recuperado

GET /api/v1/user/budgets/by-status → Buscar todos os pedidos de orçamento pelo status (de projetos e de serviços)

Authorization Header: Bearer token

Request Param: status (PENDING, APPROVED, REJECTED, CANCELLED, EXPIRED)

Body: N/A

Obs.: No payload de resposta, colocar em cada orçamento o tipo de orçamento que é: serviço ou projeto com base em qual entidade foi filtrado os itens

GET /api/v1/user/budgets/by-date → Buscar todos os pedidos de orçamento pela data de criação (de projetos e de serviços)

Authorization Header: Bearer token

Request Param: creationDate (no formato 2023-10-17)

Body: N/A

Obs.: No payload de resposta, colocar em cada orçamento o tipo de orçamento que é: serviço ou projeto com base em qual entidade foi filtrado os itens

GET /api/v1/user/budgets/by-date-range → Buscar todos os pedidos de orçamento pelo range de data de criação (de projetos e de serviços)

Authorization Header: Bearer token

Request Param: initialDate (no formato 2023-10-17) e finalDate (no formato 2023-10-17)

Body: N/A

Obs.: No payload de resposta, colocar em cada orçamento o tipo de orçamento que é: serviço ou projeto com base em qual entidade foi filtrado os itens

Admin - Budgets

POST /api/v1/admin/budgets/projects → Criar uma nova solicitação de orçamento para projeto

Authorization Header: Bearer token

```
1 {
2   "requesterId": "c59b87c3-8961-4e7a-87e6-6f231b9ec7ae",
3   "project": {
4     "name": "Meu Projeto",
5     "scope": "Escopo do Projeto",
6     "deliveryTime": 15,
7     "startDate": "2024-01-01",
8     "endDate": "2024-01-15"
9   },
10  "features": {
11    "existingFeatureIds": [
12      "1b3a6062-84e0-4d3f-a8e1-058d6cc095e7",
13      "2b4b709a-24c2-4d29-b8a3-0f0d5e8e65a0"
14    ],
15    "newFeatures": [
16      {
17        "name": "Nova Feature 1",
18        "description": "Descrição da Nova Feature 1",
19        "storyPoints": 12,
20        "backendDevelopmentTime": 60,
21        "frontendDevelopmentTime": 0
22      },
23      {
24        "name": "Nova Feature 2",
25        "description": "Descrição da Nova Feature 2",
26        "storyPoints": 12,
27        "backendDevelopmentTime": 60,
28        "frontendDevelopmentTime": 0
29      }
30    ]
31  }
32 }
```

```
30     ]
31   },
32   "projectBudget": {
33     "name": "Orçamento do Projeto",
34     "description": "Descrição do Orçamento",
35     "expiryDays": 10
36   }
37 }
```

Payload de resposta:

```
1  {
2    "project": {
3      "projectId": "c59b87c3-8961-4e7a-87e6-6f231b9ec7ae",
4      "requester": {
5        "requesterId": "a1b2c3d4-5678-90e1-f2a3b4c5d6e7",
6        "name": "Nome do Requester",
7        "fiscalType": "PJ",
8        "companyName": "Nome da Empresa",
9        "cnpj": "12345678901234",
10       "cpf": "98765432109",
11       "phone": "1234567890",
12       "mobilePhone": "987654321"
13     },
14     "name": "Nome do Projeto",
15     "scope": "Escopo do Projeto",
16     "deliveryTime": 30,
17     "startDate": "01/01/2024 12:00:00",
18     "endDate": "01/02/2024 12:00:00",
19     "developers": [],
20     "isDeleted": false,
21     "createdAt": "01/01/2024 12:00:00"
22   },
23   "features": [
24     {
25       "featureId": "d8e9f0a1-b2c3-4d5e-6f7a-8b9c0d1e2f3",
26       "name": "Nome da Feature",
27       "description": "Descrição da Feature",
28       "storyPoints": 8,
29       "backendDevelopmentTime": 120,
30       "frontendDevelopmentTime": 90,
31       "costPerHour": 50.0,
32       "profitPercentage": 40.0,
33       "totalCost": 14700.00,
34       "isDeleted": false,
35       "createdAt": "01/01/2024 12:00:00"
36     }
37   ],
38   "projectBudget": {
39     "projectBudgetId": "f2a3b4c5-d6e7-8f9a-0b1c2d3e4f5",
40     "name": "Orçamento do Projeto",
41     "description": "Descrição do Orçamento",
42     "totalCost": 14700.00,
43     "totalProfit": 5880.00,
44     "status": "IN_PROGRESS",
45     "requestedDate": "01/01/2024 12:00:00",
46     "statusChangedDate": "01/01/2024 12:00:00",
47     "expiryDays": 30,
```

```
48     "isDeleted": false,
49     "createdAt": "01/01/2024 12:00:00"
50   }
51 }
```

POST /api/v1/admin/budgets/services → Criar uma nova solicitação de orçamento para serviços

Authorization Header: Bearer token

```
1  {
2    "requesterId": "c59b87c3-8961-4e7a-87e6-6f231b9ec7ae",
3    "services": {
4      "existingServiceIds": [
5        "1b3a6062-84e0-4d3f-a8e1-058d6cc095e7",
6        "2b4b709a-24c2-4d29-b8a3-0f0d5e8e65a0"
7      ],
8      "newServices": [
9        {
10         "serviceName": "service 1",
11         "serviceDescription": "description 1",
12         "observations": [
13           {
14             "observation": "Alguma observation",
15             "observationType": "SERVICE_REQUEST"
16           }
17         ],
18         "marketMinPrice": 3100.00,
19         "marketMaxPrice": 6000.00
20       },
21       {
22         "serviceName": "service 2",
23         "serviceDescription": "description 2",
24         "observations": [
25           {
26             "observation": "Alguma observation",
27             "observationType": "SERVICE_REQUEST"
28           }
29         ],
30         "marketMinPrice": 3100.00,
31         "marketMaxPrice": 6000.00
32       }
33     ]
34   },
35   "serviceBudget": {
36     "name": "Orçamento de Serviços",
37     "description": "Descrição do Orçamento"
38     "expiryDays": 15
39   }
40 }
```

Payload de resposta:

```
1  {
2    "requesterId": "a1b2c3d4-5678-90e1-f2a3b4c5d6e7",
3    "services": [
4      {
5        "offeredServiceId": "6d7e8f9a-0b1c-2d3e-4f5g-6h7i8j9k0l1",
```

```

6     "serviceName": "Nome do Serviço 1",
7     "serviceDescription": "Descrição do Serviço 1",
8     "marketMinPrice": 3100.00,
9     "marketMaxPrice": 6000.00,
10    "profitPercentage": 40.00,
11    "totalCost": 2480.00,
12    "observations": [
13      {
14        "offeredServiceObservationId": "a1b2c3d4-5678-90e1-f2a3b4c5d6e7",
15        "observation": "Observação do Serviço 1",
16        "observationType": "TIPO_OBSERVACAO",
17        "isDeleted": false,
18        "createdAt": "01/01/2024 12:00:00"
19      }
20    ],
21    "isDeleted": false,
22    "createdAt": "01/01/2024 12:00:00"
23  },
24  {
25    "offeredServiceId": "b2c3d4e5-6f7a-8b9c-0d1e2f3g4h5",
26    "serviceName": "Nome do Serviço 2",
27    "serviceDescription": "Descrição do Serviço 2",
28    "marketMinPrice": 5000.00,
29    "marketMaxPrice": 10800.00,
30    "profitPercentage": 40.00,
31    "totalCost": 4000.00,
32    "observations": [
33      {
34        "offeredServiceObservationId": "c4d5e6f7-a8b9-0c1d-2e3f-4g5h6i7j8k9l0",
35        "observation": "Observação do Serviço 2",
36        "observationType": "TIPO_OBSERVACAO",
37        "isDeleted": false,
38        "createdAt": "01/01/2024 12:00:00"
39      }
40    ],
41    "isDeleted": false,
42    "createdAt": "01/01/2024 12:00:00"
43  }
44 ],
45 "serviceBudget": {
46   "serviceBudgetId": "d4e5f6a7-8b9c-0d1e-2f3g-4h5i6j7k8l9",
47   "name": "Orçamento do Serviço",
48   "description": "Descrição do Orçamento do Serviço",
49   "totalCost": 6480.00,
50   "totalProfit": 2592.00,
51   "status": "IN_PROGRESS",
52   "requestedDate": "01/01/2024 12:00:00",
53   "statusChangedDate": "01/01/2024 12:00:00",
54   "expiryDays": 30,
55   "isDeleted": false,
56   "createdAt": "01/01/2024 12:00:00"
57 }
58 }

```

PUT /api/v1/admin/budgets/projects/update/{id} → Atualizar uma solicitação de orçamento para projeto

Authorization Header: Bearer token

```

1 {
2   "requesterId": "c59b87c3-8961-4e7a-87e6-6f231b9ec7ae",
3   "project": {
4     "name": "Meu Projeto",
5     "scope": "Escopo do Projeto",
6     "deliveryTime": 15,
7     "startDate": "2024-01-01",
8     "endDate": "2024-01-15"
9   },
10  "features": {
11    "existingFeatureIds": [
12      "1b3a6062-84e0-4d3f-a8e1-058d6cc095e7",
13      "2b4b709a-24c2-4d29-b8a3-0f0d5e8e65a0"
14    ],
15    "newFeatures": [
16      {
17        "name": "Nova Feature 1",
18        "description": "Descrição da Nova Feature 1",
19        "storyPoints": 12,
20        "backendDevelopmentTime": 60,
21        "frontendDevelopmentTime": 0
22      },
23      {
24        "name": "Nova Feature 2",
25        "description": "Descrição da Nova Feature 2",
26        "storyPoints": 12,
27        "backendDevelopmentTime": 60,
28        "frontendDevelopmentTime": 0
29      }
30    ]
31  },
32  "projectBudget": {
33    "name": "Orçamento do Projeto",
34    "description": "Descrição do Orçamento",
35    "expiryDays": 10
36  }
37 }

```

Obs.1: É possível enviar dados parciais e atualizar somente as *features* do orçamento (removendo ou adicionando novas, porém não é possível editar uma *feature* por este endpoint) ou atualizar apenas os dados do **“projectBudget”** / do **“project”**.

Obs.2: Lembra de, durante a atualização, recalculer o totalCost e o totalProfit do **“projectBudget”** com base em todas as *features* presentes no orçamento.

PUT /api/v1/admin/budgets/services/update/{id} → Atualizar uma solicitação de orçamento de serviços

Authorization Header: Bearer token

```

1 {
2   "requesterId": "c59b87c3-8961-4e7a-87e6-6f231b9ec7ae",
3   "services": {
4     "existingServiceIds": [
5       "1b3a6062-84e0-4d3f-a8e1-058d6cc095e7",
6       "2b4b709a-24c2-4d29-b8a3-0f0d5e8e65a0"
7     ],
8     "newServices": [
9       {

```

```

10     "serviceName": "service 1",
11     "serviceDescription": "description 1",
12     "observations": [
13         {
14             "observation": "Alguma observation",
15             "observationType": "SERVICE_REQUEST"
16         }
17     ],
18     "marketMinPrice": 3100.00,
19     "marketMaxPrice": 6000.00
20 },
21 {
22     "serviceName": "service 2",
23     "serviceDescription": "description 2",
24     "observations": [
25         {
26             "observation": "Alguma observation",
27             "observationType": "SERVICE_REQUEST"
28         }
29     ],
30     "marketMinPrice": 3100.00,
31     "marketMaxPrice": 6000.00
32 }
33 ]
34 },
35 "serviceBudget": {
36     "name": "Orçamento de Serviços",
37     "description": "Descrição do Orçamento"
38     "expiryDays": 15
39 }
40 }

```

Obs.: É possível enviar dados parciais e atualizar somente os serviços do orçamento (removendo ou adicionando novos, porém não é possível editar um serviço por este endpoint) ou atualizar apenas os dados do **"serviceBudget"**.

PATCH /api/v1/admin/budgets/{id}/discount → Aplica um desconto a um orçamento existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Request Param: discount_percentage (no formato 10.25 - que indica, 10,25%)

GET /api/v1/admin/budgets/{id} → Buscar detalhes de um orçamento (de serviços ou projetos)

Authorization Header: Bearer token

Body: N/A

GET /api/v1/admin/budgets → Buscar todos os pedidos de orçamento existentes (de projetos e de serviços)

Authorization Header: Bearer token

Body: N/A

GET /api/v1/admin/budgets/by-creation-date-range → Buscar todos os pedidos de orçamento existentes pelo range de data de criação (de projetos e de serviços)

Authorization Header: Bearer token

Request Param: initialDate (no formato 2023-10-17) e finalDate (no formato 2023-10-17)

Body: N/A

GET /api/v1/admin/budgets/by-creation-date → Buscar todos os pedidos de orçamento existentes pela data de criação (de projetos e de serviços)

Authorization Header: Bearer token

Request Param: creationDate (no formato 2023-10-17)

Body: N/A

GET /api/v1/admin/budgets/by-status-changed-date-range → Buscar todos os pedidos de orçamento existentes pelo range de data de alteração de status (de projetos e de serviços)

Authorization Header: Bearer token

Request Param: initialDate (no formato 2023-10-17) e finalDate (no formato 2023-10-17)

Body: N/A

GET /api/v1/admin/budgets/by-status-changed-date → Buscar todos os pedidos de orçamento existentes pela data de alteração de status (de projetos e de serviços)

Authorization Header: Bearer token

Request Param: statusChangedDate (no formato 2023-10-17)

Body: N/A

DELETE /api/v1/s_admin/budgets/cancel/{id} → Cancelar um pedido de orçamento (de projetos e de serviços)

Authorization Header: Bearer token

Body: N/A

Feature

POST /api/v1/admin/features → Criar uma nova feature (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

```
1 {
2   "name": "Admin Feature",
3   "description": "Admin Feature Description",
4   "storyPoints": 7,
5   "backendDevelopmentTime": 12,
6   "frontendDevelopmentTime": 10
7 }
```

storyPoints → deverá aceitar apenas os valores do padrão de Fibonacci

POST /api/v1/user/features → Criar uma nova feature (USER)

Authorization Header: Bearer token

```
1 {
2   "name": "User Feature",
3   "description": "User Feature Description"
4 }
```

PUT /api/v1/admin/features/{id} → Atualizar uma feature existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

```
1 {
2   "name": "Minha Feature",
3   "description": "Descrição da Minha Feature",
4   "storyPoints": 5,
5   "backendDevelopmentTime": 10,
6   "frontendDevelopmentTime": 8
7 }
```

PATCH /api/v1/admin/features/update-prices/{id} → Atualizar os preços de uma feature existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

```
1 {
2   "costPerHour": 25.0,
3   "profitPercentage": 15.0
4 }
```

GET /api/v1/admin/features/{id} → Buscar uma feature existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

GET /api/v1/admin/features → Buscar todas as features existentes (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

Project

POST /api/v1/admin/projects/{id}/features → Cria uma nova feature e a adiciona a um projeto existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

```
1 {
2   "name": "Minha Feature",
3   "description": "Descrição da Minha Feature",
4   "storyPoints": 5,
5   "backendDevelopmentTime": 10,
```

```
6 "frontendDevelopmentTime": 8
7 }
```

PUT /api/v1/admin/projects/{id}/add-feature → Adiciona uma feature existente a um projeto (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

```
1 {
2   featureId: "1b3a6062-84e0-4d3f-a8e1-058d6cc095e7"
3 }
```

DELETE /api/v1/s_admin/projects/{id}/features/{featureId} → Remove uma feature de um projeto (SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

GET /api/v1/admin/projects/{id}/features → Retorna as features de um projeto existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

PATCH /api/v1/admin/projects/{id}/developers → Adiciona desenvolvedores a um projeto existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

```
1 {
2   "developerIds": [
3     "1b3a6062-84e0-4d3f-a8e1-058d6cc095e7"
4   ]
5 }
```

DELETE /api/v1/admin/projects/{id}/developers → Remove desenvolvedores de um projeto existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

```
1 {
2   "developerIds": [
3     "1b3a6062-84e0-4d3f-a8e1-058d6cc095e7"
4   ]
5 }
```

USER

PATCH /api/v1/s_admin/users/{id}/role → Atualiza a role de um usuário existente (SUPER ADMIN)

Authorization Header: Bearer token

```
1 {
2   "userId": "1b3a6062-84e0-4d3f-a8e1-058d6cc095e7"
3 }
```

REQUESTER

POST /api/v1/admin/requesters → Cria um requisitante para um usuário existente (ADMIN e SUPER ADMIN)

POST /api/v1/user/requesters → Cria um requisitante para um usuário existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Pessoa Jurídica

```
1 {
2   "userId": "c59b87c3-8961-4e7a-87e6-6f231b9ec7ae",
3   "name": "Exemplo Nome",
4   "fiscalType": "PJ",
5   "companyName": "Exemplo Empresa",
6   "cnpj": "00.000.000/0001-00",
7   "phone": "(00) 0000-0000",
8   "mobilePhone": "(00) 90000-0000"
9 }
```

Pessoa Física

```
1 {
2   "userId": "c59b87c3-8961-4e7a-87e6-6f231b9ec7ae",
3   "name": "Exemplo Nome",
4   "fiscalType": "PF",
5   "cpf": "000.000.000-00",
6   "phone": "(00) 0000-0000",
7   "mobilePhone": "(00) 90000-0000"
8 }
```

PUT /admin/requesters/{id} → Atualiza um requisitante existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Pessoa Jurídica

```
1 {
2   "name": "Exemplo Nome",
3   "companyName": "Exemplo Empresa",
4   "cnpj": "00.000.000/0001-00",
5   "phone": "(00) 0000-0000",
6   "mobilePhone": "(00) 90000-0000"
7 }
```

Pessoa Física

```
1 {
2   "name": "Exemplo Nome",
3   "cpf": "000.000.000-00",
4   "phone": "(00) 0000-0000",
}
```

```
5     "mobilePhone": "(00) 90000-0000"
6 }
```

DELETE /s_admin/requesters/{id} → Deleta um requisitante existente (SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

GET /admin/requesters/{id} → Busca um requisitante existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

GET /admin/requesters → Busca todos os requisitantes existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

SERVICE

POST /api/v1/admin/services → Adicionar um novo serviço (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

```
1 {
2     "serviceName": "service 1",
3     "serviceDescription": "description 1",
4     "observations": [
5         {
6             "observation": "Alguma observation",
7             "observationType": "SERVICE_REQUEST"
8         }
9     ],
10    "marketMinPrice": 3100.00,
11    "marketMaxPrice": 6000.00
12 }
```

PUT /api/v1/admin/services → Atualizar um serviço existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

```
1 {
2     "serviceName": "service 1",
3     "serviceDescription": "description 1",
4     "observations": [
5         {
6             "observation": "Alguma observation",
7             "observationType": "SERVICE_REQUEST"
8         }
9     ],
10    "marketMinPrice": 3100.00,
```

```
11     "marketMaxPrice": 6000.00
12 }
```

DELETE /api/v1/s_admin/services/{id} → Remover um serviço existente (SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

GET /api/v1/admin/services/{id} → Buscar um serviço existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

GET /api/v1/admin/services → Listar todos os serviços existentes (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

POST /api/v1/admin/services/{id}/observations → Adicionar uma observação a um serviço existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

```
1 {
2     "observation": "Alguma observation",
3     "observationType": "SERVICE_REQUEST"
4 }
```

PUT /api/v1/admin/services/{id}/observations/{obsId} → Atualizar uma observação de um serviço existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

```
1 {
2     "observation": "Alguma observation que está sendo modificada",
3     "observationType": "SERVICE_DOCUMENTATION"
4 }
```

DELETE /api/v1/s_admin/services/{id}/observations/{obsId} → Remover uma observação de um serviço existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

GET /api/v1/admin/services/{id}/observations → Listar todas as observações de um serviço existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

GET /api/v1/admin/services/{id}/observations/{obsId} → Buscar uma observação específica de um serviço existente (ADMIN e SUPER ADMIN)

Authorization Header: Bearer token

Body: N/A

SLA

Requests Per Second (RPS):

Latency: < 1s

Action Items

	Action	Description	Owner	Due date	Jira ticket
1	<input type="checkbox"/>			Jan 1, 2024	
2					

References and documentation

-
-