



*Estado do Rio de Janeiro*  
**MUNICÍPIO DE ANGRA DOS REIS**  
Secretaria de Planejamento e Parcerias  
Superintendência de Tecnologia da Informação

**MANUAL PARA MEDIÇÃO DE SOFTWARE COM BASE DE**  
**CÁLCULO EM PONTOS DE FUNÇÃO DA**  
**PREFEITURA MUNICIPAL DE ANGRA DOS REIS**

**AGOSTO – 2022**



## 1 VERSIONAMENTO

DATA	VERSÃO	RESP. TÉCNICOS	REVISADO POR
25/08/2022	1.0	Rogério de Almeida Pontes <i>Assessor de Inovação Tecnológica</i>	Andréia Schaffer Cavalcante Oliveira <i>Superintendente de Tecnologia da Informação</i>
		Lucas Soares dos Santos <i>Coordenador Técnico de Desenvolvimento e Programação de TI</i>	<b>APROVADO POR</b>
			André Luís G. A. Pimenta <i>Secretário de Planejamento e Parcerias</i>

## 2 INTRODUÇÃO E JUSTIFICATIVA

O desenvolvimento tecnológico certamente trouxe inúmeras facilidades para pessoas e empresas, principalmente nos dias atuais.

Essa constante vem sendo sustentada por um mercado desafiador.

A procura de novas tecnologias e a exigência por sistemas e serviços cada vez mais robustos, obriga a administração pública a acompanhar essa evolução.

Na prática, a administração pública, é a maior consumidora de tecnologia do país, isso porque, ela precisa oferecer as suas políticas públicas nos moldes tecnológicos atuais.

Por outro lado, os seus sistemas possuem um grau de complexidade elevado e, nem sempre é possível encontrar no mercado *softwares* prontos, licenciados e com preço fixo.

Para solucionar essa lacuna, a administração pública, em muitos casos, resolve customizar seus próprios *softwares*.

Outro desafio encontrado pela administração pública é o de entender a dimensão dessa customização, e ainda, o de precificar esse software por encomenda.



Diante do exposto e com a finalidade de promover transparência, em todo o processo de contratação, customização e entrega de *softwares sob encomenda*, a Secretaria de Planejamento e Parceria – SPP, através da Superintendência de Tecnologia da Informação – SUTIN, na cidade de Angra dos Reis/RJ, elaborou este documento.

E tem por objetivo, independente da linguagem de programação e metodologia de desenvolvimento de *softwares*, orientar aos órgãos/entidades da Prefeitura Municipal de Angra dos Reis - PMAR, como mensurar o tamanho de um *software* por encomenda, seu prazo de entrega e sua estimativa de preço.

A elaboração deste documento é fundamentada na:

IFPUG (International Function Point Users Group) entidade que dá publicidade ao tema.

ISBSG (International Software Benchmarking Standards Group) que mantém acervo público atualizado com as métricas para projetos de software.

No Conceito de Análise de Pontos de Função – APF instituído pelo Tribunal Regional do Trabalho da 9ª Região - TRT9 de Curitiba/PR.

No Roteiro SERPRO de Contagem de Pontos de Função e Estimativas do Governo Federal.

E, no Princípio da Transparência, art. 5º, incisos XXXIII da Constituição da República Federativa do Brasil de 1998.

### **3 SOFTWARE POR ENCOMENDA**

Um software é um conjunto de sequências de instruções que possui a finalidade de transmitir comandos aos componentes lógicos de um computador.

Esse conjunto de instruções controlam o funcionamento de um computador.

Havendo a necessidade de um comportamento inexistente dentro do computador é possível a sua criação, surge então a figura do software por encomenda.

Software por encomenda, nada mais é do que instruções personalizadas, construídas em uma linguagem computacional de alto nível e que permite transmitir instruções específicas para um computador.



#### **4 LINGUAGEM COMPUTACIONAL**

É a linguagem compreendida pelo computador, a mesma possui formato e um conjunto de regras sistêmicas e semânticas.

Essa linguagem é padronizada por um código fonte e traduzido (compilado) em linguagem computacional.

#### **5 CÓDIGO FONTE**

É onde se escreve as instruções a ser dada ao computador.

Quando se adquire um software por encomenda, faz-se necessário analisar se na entrega do produto final, a empresa desenvolvedora está comprometida em fornecer também o código fonte.

#### **6 COMPORTAMENTO**

Quando se encomenda um software, esperamos que ele reaja conforme a sua programação.

Podemos classificar o comportamento de um software em 4 situações distintas, são elas:

1º) Atende: é o resultado do comportamento positivo da comparação do produto final com as regras estabelecidas para a construção do software no Documento de Requisitos de Software - DRS;

2º) Defeito: é o resultado do comportamento negativo na comparação do produto final com as regras estabelecidas para a construção do software no DRS;

3º) Erro: Engano cometido por seres humanos, ou seja, não foi possível transmitir os anseios do software por encomenda para a empresa responsável pela sua implementação, gerando no ato da entrega, um comportamento inesperado;

4º) Falha: É o resultado de uma série de defeitos encontrados na entrega do software por encomenda;



## **7 DOCUMENTO DE REQUISITOS DE SOFTWARE**

O Documento de Requisitos - DRS é o documento que define o escopo de todas as funcionalidades necessárias para o seu funcionamento do software, bem como descreve os seus atributos de qualidade e tecnologias.

Esse documento é a principal peça para dimensionar os aspectos fundamentais do software, sem ele não conseguimos conhecer:

- 1º) Finalidade do software;
- 2º) Quantidade de funcionalidades ou serviços que o software vai disponibilizar;
- 3º) Comportamento;

É através do DRS que podemos mensurar a quantidade de Pontos de Função - PF para estimar os custos e prazos de entrega.

## **8 PONTOS DE FUNÇÃO**

O Ponto de Função - PF é o padrão de medição recomendado pela ISBSG para ser adotado no momento da contratação de *softwares* por encomenda.

Este padrão é utilizado nas medições de *softwares* desde 1979 e, tornou-se referência internacional pois o mesmo permite aos Técnicos de Informática, e ainda, aqueles que não dominam conhecimentos no desenvolvimento de *softwares*, estimar o produto encomendado apenas com base no DRS.

O PF também é a base para medir o software pela visão comercial, ou seja, o PF é atrelado a um valor monetário definido em contrato.

Ao encomendar um novo software, com funcionalidades a partir do zero, chamamos de funcionalidades criadas.

Para os casos de funcionalidades criadas a contraprestação é definida como "PF Cheio".

Funcionalidades criadas = PF Cheio
------------------------------------



Ao encomendar a manutenção de um software por encomenda chamamos de funcionalidades sustentadas, significa dizer que essas funcionalidades já existem e por alguma razão seu comportamento será alterado (*melhoria*).

Para funcionalidades sustentadas aplica-se um deflator, pois é compreendido que o software por encomenda não está sendo iniciado a partir do zero.

$$\text{Funcionalidades sustentadas} = \text{PF Cheio} + (\text{Deflator} = -50\%)$$

Deflacionar o PF, ou abaixar o valor do PF, é calculado por um percentual negativo sobre o valor do PF Cheio, obtém-se então o valor da funcionalidade sustentada.

E ainda, o PF permite medir prazos e a quantidade de pessoas necessárias para cada projeto.

Exemplo: Projeto orçado em 12 PFs no total.

$$1 \text{ PF} = 8 \text{ h/d} * 12 = 96 \text{ total de horas de trabalho que equivale a} \\ 12 \text{ dias ou } 2(\text{duas}) \text{ semanas e } 2(\text{dois}) \text{ dias.}$$

O exemplo acima é baseado no trabalho de apenas 1(uma) pessoa e na jornada de trabalho de 5(cinco) dias úteis semanais, porém, o total de horas e o prazo poderão sofrer alterações caso seja considerado 1(um) PF para mais ou menos de 8(oito) horas e também seja incluído mais pessoas ao projeto.

## **9 VISÃO DO USUÁRIO**

A visão do usuário representa uma descrição formal das funcionalidades apresentadas pelo software em uma linguagem simples e humanamente compreensiva.

Esta forma de medir software por encomenda é baseada exclusivamente nessa visão, desconsiderando os seguintes aspectos:



- a) Tecnologias aplicadas;
- b) Linguagem de Programação;
- c) Esforço de Implementação;
- d) Produtividade;
- e) Tempo;
- f) e Custo;

Um exemplo prático é a compra de um automóvel. O comprador não leva em consideração quantos parafusos foram necessários, quantas horas foram gastas, ou ainda, de onde veio as peças para fabricá-lo, o que realmente interessa é, se o motor é potente, se o porta-malas atende a sua família e os demais benefícios dessa aquisição.

## 10 **ESTIMADOR**

O *estimador* é a pessoa responsável pela medição do software por encomenda e a realização da leitura do DRS.

A compreensão do DRS lhe trará a *visão do usuário*, indispensável para iniciar a medição.

O estimador deve procurar nos requisitos informações relevantes, a exemplo:

- a) O que justifica a criação desse software?
- b) Qual a serventia deste software?
- c) O que este software precisa fazer exatamente?
- d) Quais são os tipos de usuário que utilizarão este software?
- e) De onde o software recebe informações, como ele é alimentado?
- f) Que tipo de informações o software armazena?

Dentre outras.

Após essa abstração, o estimador está apto a mensurar os serviços existentes no software.

Um exemplo prático, para se ter uma visão geral até aqui, é a encomenda de software para uma Agência de Locação de Veículos, ao analisar o DRS o estimador verificou os seguintes serviços no sistema:



### SOFTWARE DE GERENCIAMENTO E LOCAÇÃO DE VEÍCULOS

Serviço de Autenticação	Serviço de Locação
Login (área restrita)	Consulta Carros Alugados
Incluir Usuário	Consulta Data de aluguel
Excluir Usuário	Consulta Cliente com Carro Alugado
Alterar Usuário	Consulta Cliente que mais Aluga
Lista de Usuários	Realiza Reserva de Veículo
	Registra Locação de Veículo
	Finaliza Locação de Veículo

Serviço de Cadastro de Clientes
Incluir Cliente
Excluir Cliente
Alterar Cliente
Lista de Clientes

Serviço de Cadastro de Veículos
Incluir Veículo
Excluir Veículo
Alterar Veículo
Lista de Veículo

Figura 1

## 11 SERVIÇOS

Um serviço pode ser definido como a menor unidade de atividade de processamento dentro de um *software*, além disso, ele possui uma determinada finalidade.

Os serviços podem ser facilmente compreendidos na perceptiva da visão do usuário.

Um software, poderá apresentar um ou mais serviços, e ainda, um serviço pode estar dividido em pequenas partes.

Este é o princípio básico das funcionalidades dentro de um software.

## 12 FUNCIONALIDADES

As funcionalidades do software são tarefas que um usuário, seja ele administrador do sistema ou não, poderá executar através de uma interface (tela), a



exemplo, uma consulta e emissão de guias, controle de débitos, emissão de certidões, etc.

É importante ressaltar que todas essas tarefas são facilmente executáveis através dos botões presentes na interface.

### **13 SERVIÇOS E FUNCIONALIDADES**

Neste momento podemos afirmar que um serviço pode estar dividido em pequenas partes, essas pequenas partes são conhecidas como funcionalidades.

A exemplo, um serviço de cadastro de clientes, ele pode ser dividido em, a exemplo, 4 funcionalidades distintas, note:

- a) Adicionar um cliente;
- b) Editar um cliente;
- c) Excluir um cliente;
- d) E, listar os clientes cadastrados;

### **14 AMBIENTE**

Quando estamos em casa, podemos afirmar que fazemos parte de um ambiente, ou quando participamos de uma celebração em uma igreja, também estamos em um ambiente.

Assim são os *softwares*, eles funcionam a partir de um ambiente.

Não vamos nos aprofundar no tema pelo fato de ser ele um tanto complexo, porém, a título de compreensão para continuidade desse manual, podemos definir que *ambiente* é, um conjunto de ferramentas criadas para auxiliar na criação, no desenvolvimento e na publicação de um software.

Significa dizer que, para um software ser fabricado e disponibilizado, ele necessariamente funciona através de um ambiente tecnológico próprio.

### **15 FRONTEIRA DA APLICAÇÃO**

Já sabemos um pouco sobre ambiente, agora vamos entender acerca das suas fronteiras.



Todos os recursos que um software necessita para funcionar corretamente estão dentro de seu ambiente.

Tendo como base o exemplo anterior de cadastro de clientes, podemos afirmar que esse software utiliza um tipo de banco de dados para armazenar tais informações.

E quando uma informação não estiver presente dentro do ambiente do referido software?

Digamos que, no cadastro de clientes, o software busca a informação do cep do cliente no ambiente dos Correios para certificar-se que o dado informado está correto.

Diante do escopo entendemos que existe uma dependência no software, ele depende de uma informação que está fora de suas fronteiras.

Logo, a Fronteira da Aplicação - FA define o que é interno e externo para um *software* dentro de um ambiente de desenvolvimento.

## **16 CLASSIFICAÇÃO DAS FUNCIONALIDADES**

Vamos classificar neste momento o que existe dentro e fora da FA, são eles:

- a) ALI - Arquivo Lógico Interno: Corresponde a um agrupamento de serviços relacionados logicamente, reconhecidos pelos usuários, os quais são armazenados dentro da FA.
- b) AIE - Arquivo de Interface Externa: é um grupo de serviços, reconhecido pelo usuário, indispensáveis para o funcionamento do *software*. Esses serviços são consumidos pela aplicação e normalmente são mantidos por terceiros, a exemplo, um determinado *software* consulta o CEP através do serviço fornecido pelos Correios. Sua lógica é mantida fora da FA, porém, recebe tratamento especial da medição do software.

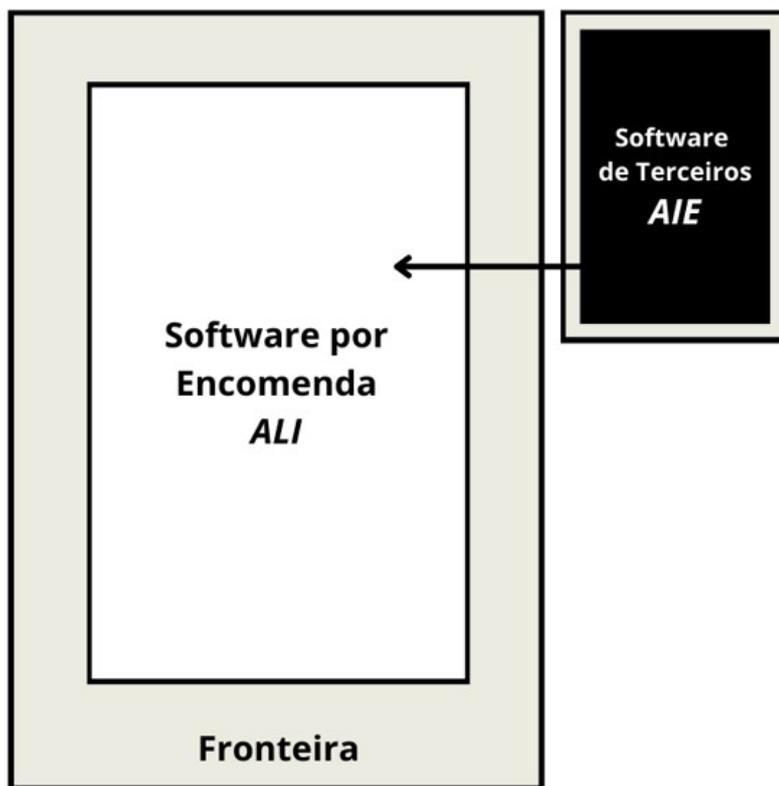


Figura 2

## 17 CLASSIFICAÇÃO DAS ENTRADAS E SAÍDAS

Outra classificação fundamental para o sucesso da medição é o reconhecimento das entradas e saídas da ALI, elas afetam diretamente no resultado final, são elas:

- a) EE – Entrada Externa: Refere-se a uma ou mais informações recebidas e tratadas pelo *software*, normalmente alimentadas por um usuário, a exemplo, o preenchimento do CPF em um campo de formulário.

O ALI recebe uma informação, no exemplo o CPF, de fora da FA.

E por fim, esse registro é tratado pelo ALI.



- b) **SE – Saída Externa:** São processos que enviam informações do ALI para fora da FA.

Normalmente, esse processamento é fruto de uma interação de usuário ou de aplicações externas (AIE), esses fazem a solicitação ou requisição, gerando um resultado lógico.

É importante ressaltar que esse resultado só é possível após a solicitação ser tratada dentro da ALI, significa dizer que existe um processamento lógico e matemático para gerar a saída.

Tendo como base o exemplo acima, após o preenchimento do campo de CPF e enviado a informação o ALI identifica que o CPF informado está errado e requer do solicitante a sua correção.

- c) **CE – Consultas Externa:** São informações enviadas pelo *software* para fora da FA e que não precisam ser tratadas dentro da ALI.

A exemplo, podemos citar páginas estáticas de um site, tais como: Política de Privacidade, Quem Somos, Conheça a Nossa Empresa, Termo de Uso, etc. Elas são exibidas sem a necessidade adicional de cálculos ou processamentos e, essas informações são raramente atualizadas.

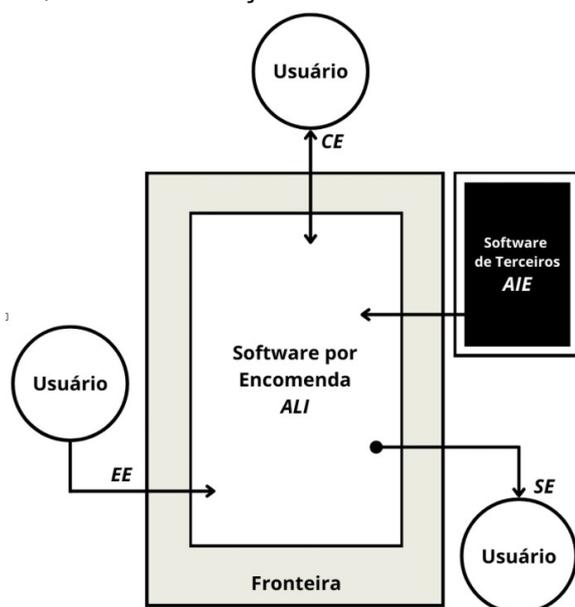


Figura 3



É considerado também EE quando os dados são enviados não por pessoas, mas por outras aplicações, a exemplo, o AIE envia uma informação para o ALI.

## 18 COMPLEXIDADES NA VISÃO DO USUÁRIO

Conforme aprendemos, a visão do usuário é a ferramenta principal para a medição em PF, note:

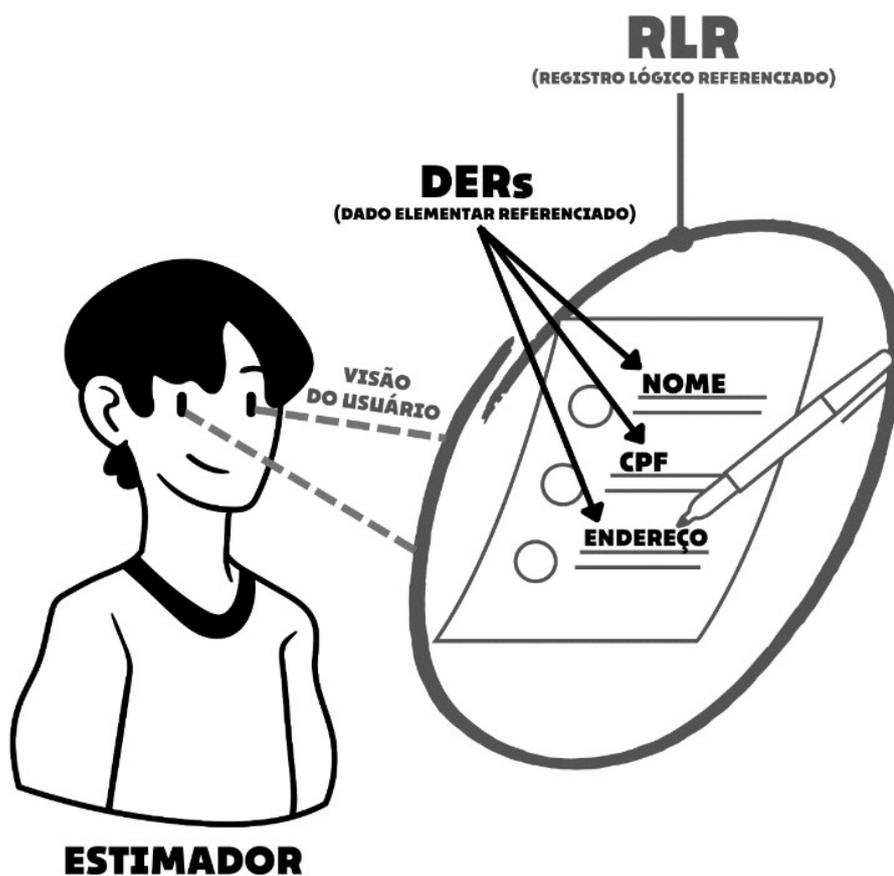


Figura 4

Como base na Figura 4 podemos medir a complexidade através de um entendimento amplo do software, temos então:

- RLR – Registro Lógico Referenciado: é o serviço em si. Nessa perspectiva verificamos quantos serviços a Interface do software possui.

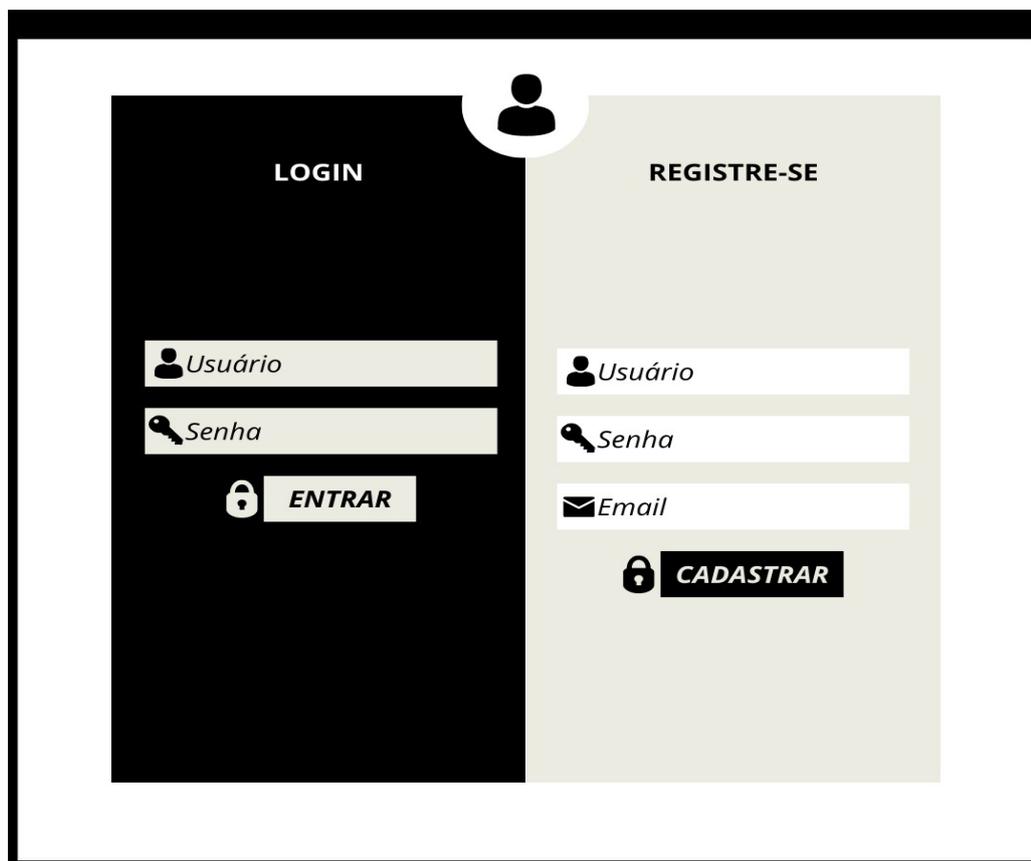


Figura 5

Com base na Figura 5, notamos na visão do usuário que, ela possui 2 (dois) RLRs, sendo 1 (um) o serviço de Login e 1 (um) o serviço de Registro de Usuários.

Os RLRs estão dispostas na mesma Interface, porém, visualmente percebemos que são serviços diferentes.

- b) DER – Dado Elementar Referenciado, são atributos dos RLRs, é a observação detalhada de um grupo existente na Interface do software.

Para medir a quantidade de DERs de uma RLR é preciso somar as entradas de um serviço, a exemplo:



Exemplo - Figura 6

A quantidade de RLRs é a soma dos serviços de Login e Registro de Usuários

**Total de RLRs: 2 (dois)**

Para saber a quantidade de DERs somamos os campos dos serviços de Login e de Registro de Usuários usando a visão do usuário assim:

Serviço de Login 2 (duas) entradas = Usuário + Senha = **2 (dois) DERs**

Serviço de Registro 3 (três) entradas = Usuário + Senha + Email: **3 (dois) DERs**

**Total de DERs: 5 (cinco)**



Figura 6

Na tabela a seguir, podemos contar EXATAMENTE o número de DERs e RLRs em uma aplicação com base na Figura 6.



### EXEMPLO DE DEFINIÇÃO DA COMPLEXIDADE DE CADA FUNÇÃO DE DADOS

		DERs		
		De 1 a 19	De 20 a 50	Maior que 50
RLRs	1	Baixa	Baixa	Média
	De 2 a 5	Baixa	Média	Alta
	Maior que 5	Média	Alta	Alta

Figura 7

Na verdade, isto só é relevante quando a quantidade está nos limites das faixas da tabela de complexidade. Além disso, uma eventual classificação incorreta da complexidade da função afeta de forma significativa o resultado final da medição.

O mais importante é identificar corretamente a quantidade de funções (tanto de dados como de transações) para definir a complexidade do software.

## 19 FUNÇÕES DE DADOS LÓGICOS E FUNÇÕES TRANSACIONAIS

Agora que entendemos como mensurar as horas trabalhadas por um desenvolvedor em um projeto de software por encomenda e também como verificar a sua complexidade, esboçaremos, de forma prática, o processo para aferir os PFs.

Sabemos que, tanto as *Funções de Dados Lógicos* quanto as *Funções Transacionais* definem o que é externo à aplicação.

Eles marcam claramente a FA, entre o software que está sendo medido e o usuário.

Podemos definir as *Funções de Transações* com EEs, SEs e CEs, elas passam para dentro e para fora da aplicação, justificando o porquê no nome.

As tabelas da Figura 8 estão baseadas nas métricas da ISBSG, sendo utilizadas pelos agentes públicos e empresas privadas. Essas métricas definem os PFs para as *Funções de Transações*.



### COMPLEXIDADE FUNCIONAL DAS EE

		DERs		
		De 1 a 4	De 5 a 15	Maior que 15
ALRs	De 0 a 1	Baixa	Baixa	Média
	2	Baixa	Média	Alta
	Maior que 2	Média	Alta	Alta

### COMPLEXIDADE FUNCIONAL DAS SE e CE

		DERs		
		De 1 a 5	De 6 a 19	Maior que 19
ALRs	De 0 a 1	Baixa	Baixa	Média
	De 2 a 3	Baixa	Média	Alta
	Maior que 3	Média	Alta	Alta

NOTA: Uma CE tem no mínimo 1 ALR.

Figura 8

Já as *Funções de Dados Lógicos* são os ALIs e AIEs, elas nos auxiliam a identificação dos dados lógicos referenciados que são mantidos dentro e fora da FA.

### TAMANHO DAS FUNÇÕES DE DADOS

		TIPO	
		ALI	AIE
Complexidade Funcional	Baixa	7	5
	Média	10	7
	Alta	15	15

Figura 9



## 20 EXEMPLO DE MEDIÇÃO POR PONTO DE FUNÇÃO

Analisando a Figura 10 pela visão do usuário.

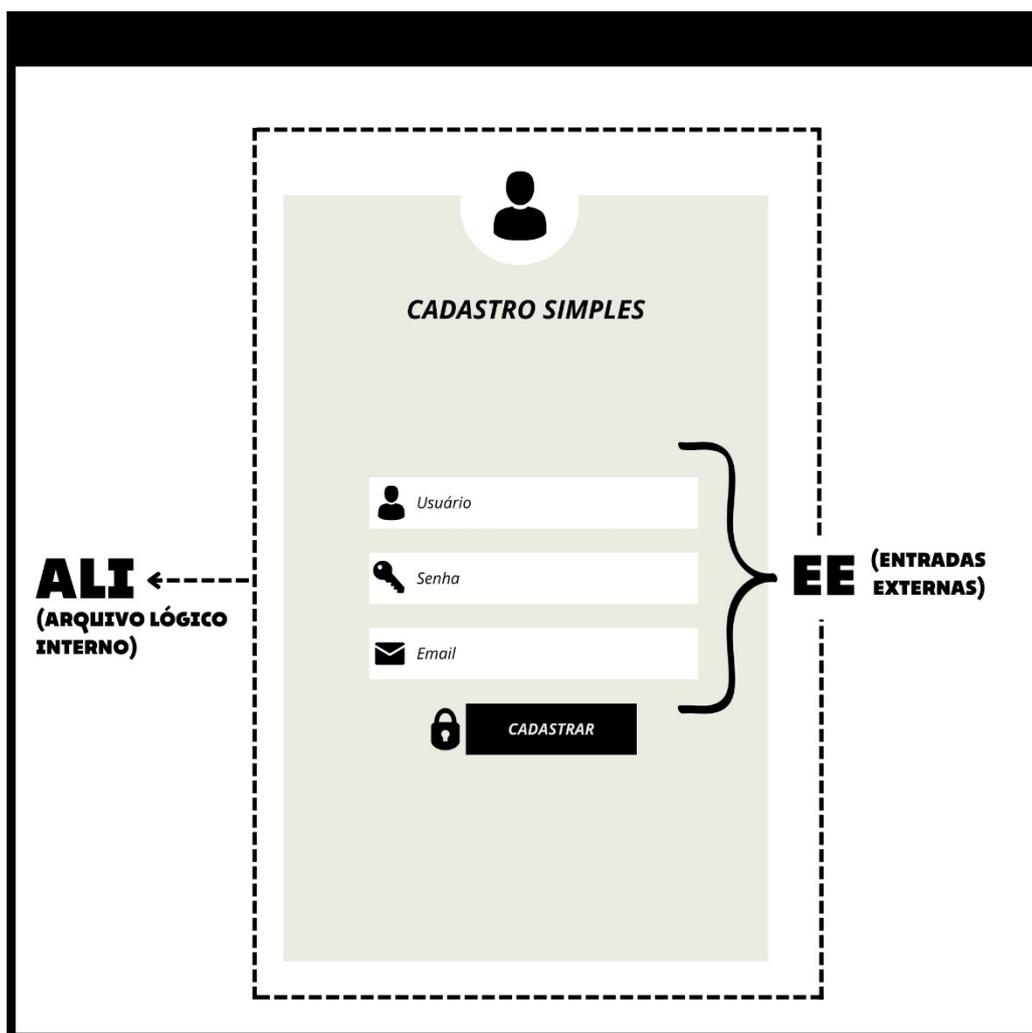


Figura 10



Temos:

<p><u>Funções de Transações:</u> 3 (três) EE = (Usuário, Senha e E-mail) 0 (zero) SE 0 (zero) CE</p> <p><u>Funções de Dados:</u> 1 (um) ALI 0 (zero) AIE</p> <p>Complexidade: Se temos 1 (um) ALI a complexidade é “baixa” = 7(sete) PFs</p>
--

De posse dos dados, basta preencher a tabela e verificar a quantidade de PFs.

**CONTAGEM DE PONTOS DE FUNÇÃO - EXEMPLO DE MEDIÇÃO**

TIPO DE FUNÇÃO	COMPLEXIDADE FUNCIONAL	TOTAL DA COMPLEXIDADE	SOMA DAS COMPLEXIDADES
ALI	1 BAIXA x 7	7	7 PFs
	MÉDIA x 10		
	ALTA x 15		
AIE	BAIXA x 5	9	9 PFs
	MÉDIA x 7		
	ALTA x 10		
EE	3 BAIXA x 3	9	9 PFs
	MÉDIA x 4		
	ALTA x 6		
SE	BAIXA x 4		
	MÉDIA x 5		
	ALTA x 7		
CE	BAIXA x 3		
	MÉDIA x 4		
	ALTA x 6		
<b>TOTAL:</b>			<b>16 PFs</b>

Figura 11



TABELA PARA MEDIÇÃO DE SOFTWARE COM BASE DE CÁLCULO EM PONTOS DE FUNÇÃO DA PREFEITURA MUNICIPAL DE ANGRA DOS REIS

Complexidade ALI e AIE		
1 a 19 DERS	20 a 50 DERS	51 ou mais DERS
1 RLR	Baixa	Média
2 a 5 RLRs	Baixa	Alta
6 ou mais RLRs	Média	Alta

Complexidade EE		
1 a 4 DERS	5 a 15 DERS	16 ou mais DERS
0 a 1 RLR	Baixa	Média
2 RLRs	Baixa	Alta
3 ou mais RLRs	Média	Alta

Complexidade SE		
1 a 5 DERS	6 a 19 DERS	20 ou mais DERS
0 a 1 RLR	Baixa	Média
2 a 3 RLRs	Baixa	Alta
4 ou mais RLRs	Média	Alta

Complexidade CE		
1 a 5 DERS	6 a 19 DERS	20 ou mais DERS
0 a 1 RLR	Baixa	Média
2 a 3 RLRs	Baixa	Alta
4 ou mais RLRs	Média	Alta

ALI Complexidade Funcional	
Baixa	7
Média	10
Alta	15

AIE Complexidade Funcional	
Baixa	5
Média	7
Alta	10

EE Complexidade Funcional	
Baixa	4
Média	5
Alta	6

SE Complexidade Funcional	
Baixa	3
Média	4
Alta	6

CE Complexidade Funcional	
Baixa	3
Média	4
Alta	6



Estado do Rio de Janeiro  
**MUNICÍPIO DE ANGRA DOS REIS**  
Secretaria de Planejamento e Parcerias  
Superintendência de Tecnologia da Informação

Data da Medição:	
Nome do Estimador:	
Matrícula do Estimador:	
Nome do Software:	
Nome da Serviço:	
Observações da Medição:	

### TABELA DE CONTAGEM DE PONTOS DE FUNÇÃO

TIPO DE FUNÇÃO	COMPLEXIDADE FUNCIONAL		TOTAL DA COMPLEXIDADE	SOMA DAS COMPLEXIDADES
ALI	BAIXA	x 7		
	MÉDIA	x 10		
	ALTA	x 15		
AIE	BAIXA	x 5		
	MÉDIA	x 7		
	ALTA	x 10		
EE	BAIXA	x 3		
	MÉDIA	x 4		
	ALTA	x 6		
SE	BAIXA	x 4		
	MÉDIA	x 5		
	ALTA	x 7		
CE	BAIXA	x 3		
	MÉDIA	x 4		
	ALTA	x 6		
<b>TOTAL:</b>				