```
contato = zoho.crm.getRecordById("Contacts", contatoId);
info contato;
```

Json Formatter, utilizado para ver o Jason

1-Pegando o Nome dos contatos:

```
contato = zoho.crm.getRecordById("Contacts", idContato);
nome = contato.get("Full_Name");
info nome;
```

2-Pegando o telefone:

```
contato = zoho.crm.getRecordById("Contacts", idContato);
telefone = contato.get("Phone");
nome = contato.get("Full_Name");
info nome+" "+telefone;
```

3-Pegando todo o Jason de orçamentos:

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
info orcamento;
```

4-Pegando em Orçamento o CEP de cobrança:

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
cep = orcamento.get("Billing_Code");
info cep;
```

5-Pegando a tabela de itens cotados detalhados dentro de Orçamento:

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
itensCotados  = orcamento.get("Product_Details");
info itensCotados;
```

6-Usando forEach dentro da tabela de itens cotados para mostrar cada item:

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
// info orcamento;

itensCotados  = orcamento.get("Product_Details");

for each  item in itensCotados.
{
	info item;
}

// listaProdutos = orcamento.get("product");
// info listaProdutos;
```

7-Usando forEach para pegar informações específicas dentro da tabela;

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
// info orcamento;

itensCotados  = orcamento.get("Product_Details");

for each  item in itensCotados
{
	nomeItem = item.get("product").get("name");

	info nomeItem;
}
```

8-Pegando informações específicas e somando montante Total usando forEach;

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
// info orcamento;

itensCotados  = orcamento.get("Product_Details");

montanteFinal =0;
for each  item in itensCotados
{

	nomeItem = item.get("product").get("name");
	quantidadeItem = item.get("quantity");
	precoItem = item.get("total_after_discount");
	montanteItem = item.get("net_total");

//	precoFinal = quantidadeItem*precoItem+;

	montanteFinal = montanteItem+montanteFinal;
```

```
        info "Nome: "+nomeItem+" "+ "  Quantidade: " +quantidadeItem +"Preço do Item: "+precoItem
+" Montante: "+montanteItem;
}

info " Preço do montante final: " +montanteFinal;
```

9-Pegando informações específicas e somando montante Total usando forEach; teste 002

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
// info orcamento;

itensCotados  = orcamento.get("Product_Details");

montanteFinal =0;
quantidadeFinalItens =0;
for each  item in itensCotados
{

        nomeItem = item.get("product").get("name");
        quantidadeItem = item.get("quantity");
        precoItem = item.get("total_after_discount");
        montanteItem = item.get("net_total");

   quantidadeFinalItens =quantidadeItem+quantidadeFinalItens;

        montanteFinal = montanteItem+montanteFinal;

        info "Nome: "+nomeItem ;
        info  " Quantidade:" +quantidadeItem ;
        info "Preço do Item: "+precoItem ;
        info  "montanteItem" +montanteItem;
        info "_____";
}

info " Preço do montante final: " +montanteFinal ;
info  "Quantidade total de itens vendidos: "+quantidadeFinalItens;
```

10-Pegando informações específicas e somando montante Total usando forEach; teste 003

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
```

```
// info orcamento;

geralProduto = orcamento.get("Product_Details");
// info geralProduto;

 totalMontante =0;

for each item in geralProduto

{
        nomeProduto = item.get("product").get("name");
        precoUnitario = item.get("unit_price");
        quantidadeProdutos = item.get("quantity");
        montante= item.get("total");

        totalMontante = montante+totalMontante;

        precoFinal= precoUnitario*quantidadeProdutos;

        info  " Nome Produto: "+nomeProduto;
        info  " Preço Unitario: " + precoUnitario;
        info  " Quantidade de Produtos: " +quantidadeProdutos;
        info   "Preço Final Produtos: "+precoFinal;
        info "_____";
}

info "Total Montante: "+totalMontante;
```

11-Adicionando um novo item na lista de itens cotados dentro de orcamento :

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
produtos = orcamento.get("Product_Details");
listaProduto = List();
produtoNovo = {"product":{"id":"5284804000000387069"},"quantity":3,"list_price":57};
for each produto in produtos
{
        mapaProduto = Map();
        mapaProduto.put("quantity",produto.get("quantity"));
        mapaProduto.put("list_price", produto.get("list_price"));
        mapaIdProduto = {"id":produto.get("product").get("id")};
        mapaProduto.put("product", mapaIdProduto);
//      info mapaProduto;
        listaProduto.add(mapaProduto);
}
listaProduto.add(produtoNovo);
info listaProduto;
mapaAtualizacao = Map();
mapaAtualizacao.put("Product_Details",listaProduto);
info zoho.crm.updateRecord("Quotes", idOrcamento,mapaAtualizacao);
// info response;
```

12-Adicionando uma nota dentro de Orçamento no campo notas:

```
	// pegando os dados do campo relacionado de notas getRelatedRecords= pegando campos relacionados
	notas = zoho.crm.getRelatedRecords("Notes", "Quotes", idQuote);
	// info notas;

	// criando uma mapa para armazenar as  notas e faz callback
	note1 = Map();

	//criando as notas e dando um put
	note1.put("Note_Title", "Produto Criado");
	note1.put("Note_Content", "Produto adicionado no orçamento");

	// setando o ID do ID ARGUMENTS
	note1.put("Parent_Id", idQuote);

	//Setando o Modulo que estou pegando as notas no caso em orcamento"QUOTES";
	note1.put("se_module", "Quotes");

	info note1;
	info zoho.crm.createRecord("Notes", note1);
```

12-Adicionando uma nota dentro de Orçamento no campo notas:(DETALHANDO);

```
	// peagando o objeto e armazenando da variavel notas
	notas = zoho.crm.getRelatedRecords("Notes", "Quotes", idNota);

	//criando o Mapa para armazenrar o array de notas
	nota0 = Map();

	//adicionando os dados dentro de nota
	nota0.put("Note_Title","NEXT NEX NE N");
	nota0.put("Note_Content","Helioptile pelipper minun. Illumise oricorio bonsly samurott kabuto koffing turtwig quagsire aerodactyl. Jigglypuff bagon larvesta carnivine pineco mewtwo flareon blacephalon.");
	// setando onde esta o campo notas MODULO e ID
	nota0.put("se_module","Quotes");
	nota0.put("Parent_Id",idNota);

	// printando e salvando as notas
	info zoho.crm.createRecord("Notes", nota0);
```

13-Detalhando a função para adicionar elementos dentro de uma lista da página orçamentos:

```
//pegando dados do orcamento
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
```

```
// info orcamento;

//pegando dados de oe=rcamento>detalhes do produto
detalhesProdutos = orcamento.get("Product_Details");
// info detalhesProdutos;

// criando lista para adicionar um novo produto
listaProduto = list();

//prestar atenção nos niveis de cada elemento olhar o Jason antes
produtoNovo = {"product":{"id":"5284804000000387089"},"quantity":3,"list_price":57};

//criar um laço de repetição para adicionar os produtos na lista
for each produto in detalhesProdutos
{
	//criando um map que armazena e faz um callback para retornar um array;
	mapaProduto = Map();

	//adicionando elementos no nosso array

	mapaProduto.put("quantity",produto.get("quantity"));
	mapaProduto.put("list_price",produto.get("list_Price"));
	mapaIdProduto = {"id":produto.get("product").get("id")};
	mapaProduto.put("product",mapaIdProduto);

	//salvando esse mapa na Lista de Produto, fazemos isso pois no for each no fina os dados
sao pertidos na ultima passada
	listaProduto.add(mapaProduto);
}

//adicionando o produto novo na lista de produto
listaProduto.add(produtoNovo);

// info listaProduto;
mapaAtualizacao = Map();

mapaAtualizacao.put("Product_Details",listaProduto);

info zoho.crm.updateRecord("Quotes",idOrcamento,mapaAtualizacao);
// info response;
```

14-Ao cadastrar um novo produto e ele for cadastrado corretamente exibir uma mensagem de sucesso. Desafio

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
produtos = orcamento.get("Product_Details");
// info produtos;

listaProduto = List();
produtoNovo = {"product":{"id":"5284804000000387069"},"quantity":3,"list_price":57};
for each produto in produtos
{
```

```
        mapaProduto = Map();
        mapaProduto.put("quantity",produto.get("quantity"));
        mapaProduto.put("list_price", produto.get("list_price"));
        mapaIdProduto = {"id":produto.get("product").get("id")};
        mapaProduto.put("product", mapaIdProduto);

        listaProduto.add(mapaProduto);
}
listaProduto.add(produtoNovo);
// info listaProduto;
mapaAtualizacao = Map();
mapaAtualizacao.put("Product_Details",listaProduto);

// info mapaAtualizacao;

retornoProduto = zoho.crm.updateRecord("Quotes", idOrcamento,mapaAtualizacao);
info retornoProduto;


notas = zoho.crm.getRelatedRecords("Notes", "Quotes", idOrcamento);
note1 = Map();
note1.put("Note_Title", "Produto Criado");
note1.put("Note_Content", "Produto adicionado no orçamento");
note1.put("Parent_Id", idOrcamento);
note1.put("se_module", "Quotes");
info note1;
info zoho.crm.createRecord("Notes", note1);
```

15- Modificar os valores dos itens na tabela de itens cotados campos(preço e desconto), verificar se o desconto do produto passou de 20%, caso tenha passado criar uma nota mostrando os nomes dos produtos que passaram o desconto e exibir essas informações em tarefas também.

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
// info orcamento;
detalhesProdutos =  orcamento.get("Product_Details");
// info detalhesProdutos;

// listaProduto = list();
// novoProduto =
{"product":{"id":"5284804000000387069"},"quantity":7,"list_price":700,"Discount":700};

for each  produto in detalhesProdutos
{
        mapaProduto = Map();
//      mapaProduto.put("quantity",produto.get("quantity"));
//      mapaProduto.put("list_price",produto.get("list_price"));
//      mapaProduto.put("Discount",produto.get("Discount"));
        mapaIdProduto = {"id":produto.get("product").get("id")};
```

```
        montante = produto.get("total");
        info montante;

//      mapaProduto.put("product", mapaIdProduto);
//      listaProduto.add(mapaProduto);
}



// listaProduto.add(novoProduto);
// mapaAtualizacao = Map();
// mapaAtualizacao.put("Product_Details",listaProduto);
// info mapaAtualizacao;
// info  zoho.crm.updateRecord("Quotes", idOrcamento,mapaAtualizacao);

// montante = orcamento.get("total");
//  montante = detalhesProdutos.get("total");

—----------------------------------------------------------------------
```

v2

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
// info orcamento;
detalhesProdutos =  orcamento.get("Product_Details");
// info detalhesProdutos;

// listaProduto = list();
// novoProduto =
{"product":{"id":"5284804000000387069"},"quantity":7,"list_price":700,"Discount":700};

for each  produto in detalhesProdutos
{
        mapaProduto = Map();
//      mapaProduto.put("quantity",produto.get("quantity"));
//      mapaProduto.put("list_price",produto.get("list_price"));
//      mapaProduto.put("Discount",produto.get("Discount"));
//      mapaIdProduto = {"id":produto.get("product").get("id")};

        codigoProduto = produto.get("Product_Code"); //perguntar como pegar o id de um
obejeto acimma
        montante = produto.get("total");
        tickteDesconto = produto.get("Discount");
   vintePorcento= 0.2;
        totalDesconto = montante*vintePorcento;

//      info produto;
```

```
        if ( tickteDesconto >= totalDesconto )
    {
                info "_____Reprovado_____";
                info produto;
                info "Ticket de desconto dado:" +tickteDesconto;
             info  "Valor 20% maximo de desconto: "+totalDesconto;

    }else
    {
            info "_____Aprovado_____";
            info "Desconto aprovado! para o prroduto: ";
            info produto;

    }

//          mapaProduto.put("product", mapaIdProduto);
//          listaProduto.add(mapaProduto);
}


// listaProduto.add(novoProduto);
// mapaAtualizacao = Map();
// mapaAtualizacao.put("Product_Details",listaProduto);
// info mapaAtualizacao;
// info  zoho.crm.updateRecord("Quotes", idOrcamento,mapaAtualizacao);

// montante = orcamento.get("total");
//  montante = detalhesProdutos.get("total");


// mapaAtualizacao = Map();
// mapaAtualizacao.put("Product_Details",listaProduto);
// info mapaAtualizacao;
// info  zoho.crm.updateRecord("Quotes", idOrcamento,mapaAtualizacao);

// montante = orcamento.get("total");
//  montante = detalhesProdutos.get("total");
```

V3 criou a nota dentro de produto Reprovado

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
// info orcamento;
detalhesProdutos =  orcamento.get("Product_Details");
// info detalhesProdutos;

// listaProduto = list();
```

```
// novoProduto =
{"product":{"id":"5284804000000387069"},"quantity":7,"list_price":700,"Discount":700};

for each  produto in detalhesProdutos
{
	mapaProduto = Map();
//	mapaProduto.put("quantity",produto.get("quantity"));
//	mapaProduto.put("list_price",produto.get("list_price"));
//	mapaProduto.put("Discount",produto.get("Discount"));
//	mapaIdProduto = {"id":produto.get("product").get("id")};

	codigoProduto = produto.get("Product_Code"); //perguntar como pegar o id de um
obejeto acimma
	montante = produto.get("total");
	tickteDesconto = produto.get("Discount");
   vintePorcento= 0.2;
	totalDesconto = montante*vintePorcento;

//	info produto;



	if ( tickteDesconto >= totalDesconto )
   {
		info "_____Reprovado_____";
		info produto;
		info "Ticket de desconto dado:" +tickteDesconto;
	    info  "Valor 20% maximo de desconto: "+totalDesconto;


		notas = zoho.crm.getRelatedRecords("Notes", "Quotes", idOrcamento);
		note1 = Map();
		note1.put("Note_Title", "Produto Reprovado no desconto");
		note1.put("Note_Content", "Produto Reprovado no desconto");
		note1.put("Parent_Id", idOrcamento);
		note1.put("se_module", "Quotes");
		info note1;
	   zoho.crm.createRecord("Notes", note1);


   }else
   {
	   info "_____Aprovado_____";
	   info "Desconto aprovado! para o prroduto: ";
	   info produto;

   }

//	mapaProduto.put("product", mapaIdProduto);
//	listaProduto.add(mapaProduto);
}
```

```
// listaProduto.add(novoProduto);
// mapaAtualizacao = Map();
// mapaAtualizacao.put("Product_Details",listaProduto);
// info mapaAtualizacao;
// info  zoho.crm.updateRecord("Quotes", idOrcamento,mapaAtualizacao);

// montante = orcamento.get("total");
//  montante = detalhesProdutos.get("total");
```

v4 Clean

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
// info orcamento;
detalhesProdutos =  orcamento.get("Product_Details");
// info detalhesProdutos;

for each  produto in detalhesProdutos
{

        codigoProduto = produto.get("Product_Code"); //perguntar como pegar o id de um
obejeto acimma

        montante = produto.get("total");
        tickteDesconto = produto.get("Discount");
   vintePorcento= 0.2;
        totalDesconto = montante*vintePorcento;


        if ( tickteDesconto >= totalDesconto )
   {
                info "_________________Reprovado_____________________";

                info "Ticket de desconto dado:" +tickteDesconto;
             info  "Valor 20% maximo de desconto: "+totalDesconto;
                notas = zoho.crm.getRelatedRecords("Notes", "Quotes", idOrcamento);
                note1 = Map();
                note1.put("Note_Title", "Produto Reprovado no desconto");
                note1.put("Note_Content", "Produto Reprovado no desconto");
                note1.put("Parent_Id", idOrcamento);
                note1.put("se_module", "Quotes");
                info note1;
                zoho.crm.createRecord("Notes", note1);
                info produto;

   }else
   {
           info "_____________Aprovado_____________________";
           info "Desconto aprovado! para o prroduto: ";
           info "Ticket de desconto dado:" +tickteDesconto;
           info  "Valor 20% maximo de desconto: "+totalDesconto;
```

```
        info produto;

    }

}
```

16-Criando uma tarefa no módulo cliente potências (“Lead”), leadId

```
leadDetails = zoho.crm.getRecordById("Leads",input.leadId);
reminderTime = leadDetails.get("Created_Time").addDay(1).toTime();
//info reminderTime;
mp=map();
mp.put("Subject","Follow up");
mp.put("$se_module","Leads");
mp.put("What_Id",input.leadId);
mp.put("Owner",leadDetails.get("Owner").get("id"));
mp.put("Due_Date",leadDetails.get("Created_Time").toDate().addDay(2));
mp.put("Remind_At",{"ALARM":"FREQ=NONE;ACTION=EMAIL;TRIGGER=DATE-TIME:" +
reminderTime.toString("yyyy-MM-dd'T'HH:mm:ss'+05:30'")});
mp.put("Status","Not Started");
mp.put("Send_Notification_Email",true);
createTask = zoho.crm.createRecord("Tasks",mp);
info mp;
info createTask;
```

17- Clean criando as notas para os produtos que tiveram desconto acima do permetido.

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
idNegocio = orcamento.get("Deal_Name").get("id");
info idNegocio;
// info orcamento;
detalhesProdutos =  orcamento.get("Product_Details");
// info detalhesProdutos;

listaProduto = list();
for each  produto in detalhesProdutos
{
        codigoProduto = produto.get("Product_Code"); //perguntar como pegar o id de um
obejeto acimma
        montante = produto.get("total");
        tickteDesconto = produto.get("Discount");
   vintePorcento= 0.2;
        totalDesconto = montante*vintePorcento;
        if ( tickteDesconto >= totalDesconto )
   {
                listaProduto.add(produto.get("product").get("name"));

        }
```

```
}

info listaProduto;
```

18-Adicionando uma tarefa e uma nota ao ter itens com desconto acima do permitido

```
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
idNegocio = orcamento.get("Deal_Name").get("id");
// info idNegocio;
// info orcamento;
detalhesProdutos =  orcamento.get("Product_Details");
// info detalhesProdutos;

listaProduto = list();
for each  produto in detalhesProdutos
{
        codigoProduto = produto.get("Product_Code"); //perguntar como pegar o id de um
obejeto acimma
        montante = produto.get("total");
        tickteDesconto = produto.get("Discount");
   vintePorcento= 0.2;
        totalDesconto = montante*vintePorcento;
        if ( tickteDesconto >= totalDesconto )
   {
//              info produto;
                listaProduto.add(produto.get("product").get("name"));//pegando os produtos
        }
}
// Criando Notas e a notificação de que o desconto não foi aprovado
// info "Esses foram os produtos que não tem desconto aprovado:";
// info listaProduto;
        notas = zoho.crm.getRelatedRecords("Notes", "Quotes", idOrcamento);
        note1 = Map();
        note1.put("Note_Title", "Produto Reprovado no desconto");
        note1.put("Note_Content", listaProduto);
        note1.put("Parent_Id", idOrcamento);
        note1.put("se_module", "Quotes");
//      info note1;
//      zoho.crm.createRecord("Notes", note1);
// Criando Notas e a notificação de que o desconto não foi aprovado

// Criando a notificação da  tarefa; >>>>>>>>>>
        tarefas = zoho.crm.getRelatedRecords("Tasks", "Quotes", idOrcamento);
        info tarefas;
//      mp=map();
//      mp.put("Subject","Gustavo teste");
//      mp.put("$se_module","Leads");
//      mp.put("What_Id",input.idOrcamento);
//      mp.put("Owner",leadDetails.get("Owner").get("id"));
//      mp.put("Due_Date",tarefas.get("Created_Time").toDate().addDay(2));
```

```
//
mp.put("Remind_At",{"ALARM":"FREQ=NONE;ACTION=EMAIL;TRIGGER=DATE-TIME:" +
reminderTime.toString("yyyy-MM-dd'T'HH:mm:ss'+05:30'")});
//          mp.put("Status","Not Started");
//          mp.put("Send_Notification_Email",true);
//          createTask = zoho.crm.createRecord("Tasks",mp);
//          info mp;
//          info createTask;

// Criando a notificação da  tarefa; <<<<<<<<<<<<
```

19-Adicionando uma tarefa e uma nota ao ter itens com desconto acima do permitido:

```
//a tarefa é para ser criado em negócios
// id do negocio referente a tarefa
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);
idNegocio = orcamento.get("Deal_Name").get("id");
// info idNegocio;
// info orcamento;
detalhesProdutos =  orcamento.get("Product_Details");
// info detalhesProdutos;

listaProduto = list();
for each  produto in detalhesProdutos
{
          codigoProduto = produto.get("Product_Code"); //perguntar como pegar
o id de um obejeto acimma
          montante = produto.get("total");
          tickteDesconto = produto.get("Discount");
    vintePorcento= 0.2;
          totalDesconto = montante*vintePorcento;

//        if{(ticketDesconto == "")
//                  info "Campo vazio preencha com algum valor valido!";
//        }
          if( tickteDesconto >= totalDesconto )
    {
//                  info produto;
                    listaProduto.add(produto.get("product").get("name"));//pegando
os produtos
          }
```

```
}
// Criando Notas e a notificação de que o desconto não foi aprovado
// info "Esses foram os produtos que não tem desconto aprovado:";
// info listaProduto;

        notas = zoho.crm.getRelatedRecords("Notes", "Quotes", idOrcamento);
        note1 = Map();
        note1.put("Note_Title", "Produto Reprovado no desconto");
        note1.put("Note_Content", listaProduto);
        note1.put("Parent_Id", idOrcamento);
        note1.put("se_module", "Quotes");
        info note1;

//      zoho.crm.createRecord("Notes", note1);
// Criando Notas e a notificação de que o desconto não foi aprovado

// Criando a notificação da  tarefa; >>>>>>>>>>


        mp=map();
        mp.put("Subject","Gustavo teste");
        mp.put("$se_module","Deals");
        mp.put("What_Id",idNegocio);
        mp.put("Owner",orcamento.get("Owner").get("id"));
        mp.put("Status","Not Started");
        mp.put("Send_Notification_Email",true);
        createTask = zoho.crm.createRecord("Tasks",mp);
        info mp;
        info createTask;

// // Criando a notificação da  tarefa; <<<<<<<<<<<


Teste Teste

// id do negocio referente a tarefa

// a tarefa esta em negocios
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);

idNegocio = orcamento.get("Deal_Name").get("id");
// info idNegocio;
// info orcamento;
detalhesProdutos =  orcamento.get("Product_Details");
```

```
// info detalhesProdutos;
listaProduto = list();
for each  produto in detalhesProdutos
{
        codigoProduto = produto.get("Product_Code"); //perguntar como pegar
o id de um obejeto acimma
        montante = produto.get("total");
        tickteDesconto = produto.get("Discount");
   vintePorcento= 0.2;
        totalDesconto = montante*vintePorcento;

        contarProduto = produto.length();

        // verificando se o campo nao está vazio

        if ( contarProduto == 0 )
    {
                info "O campo esta vazio!";
    }
        else

   {
      //info "O campo foi aceito!";

                if( tickteDesconto >= totalDesconto )

                {info produto;
                listaProduto.add(produto.get("product").get("name"));//pegando
os produtos
           }

   }
}
//Criando Notas e a notificação de que o desconto não foi aprovado
info "Esses foram os produtos que não tem desconto aprovado:";

//criando a lista de produto
info listaProduto;

        notas = zoho.crm.getRelatedRecords("Notes", "Quotes", idOrcamento);
        note1 = Map();
        note1.put("Note_Title", "Produto Reprovado no desconto");
        note1.put("Note_Content", listaProduto);
        note1.put("Parent_Id", idOrcamento);
```

```
        note1.put("se_module", "Quotes");
        info note1;
        zoho.crm.createRecord("Notes", note1);

//Criando Notas e a notificação de que o desconto não foi aprovado
// Criando a notificação da  tarefa; >>>>>>>>>>

        //criando o mapa das tarefas
        mp=map();
        mp.put("Subject","Gustavo teste teste teste ");
        //definindo o modulo que o campo esta atribuido Deals: Orcamento
        mp.put("$se_module","Deals");
        //id negocio setado la no começo
        mp.put("What_Id",idNegocio);
        mp.put("Owner",orcamento.get("Owner").get("id"));
        mp.put("Status","Not Started");
        mp.put("Send_Notification_Email",true);
        createTask = zoho.crm.createRecord("Tasks",mp);
        info mp;
        info createTask;

//Criando a notificação da  tarefa; <<<<<<<<<<<<




// id do negocio referente a tarefa

// a tarefa esta em negocios
orcamento = zoho.crm.getRecordById("Quotes", idOrcamento);

idNegocio = orcamento.get("Deal_Name").get("id");
// info idNegocio;
// info orcamento;
detalhesProdutos =  orcamento.get("Product_Details");
// info detalhesProdutos;
listaProduto = list();
for each  produto in detalhesProdutos
{
        codigoProduto = produto.get("Product_Code"); //perguntar como pegar
o id de um obejeto acimma
        montante = produto.get("total");
        tickteDesconto = produto.get("Discount");
   vintePorcento= 0.2;
```

```
        totalDesconto = montante*vintePorcento;

        contarProduto = produto.length();

        // verificando se o campo nao está vazio

        if ( contarProduto == 0 )
    {
                info "O campo esta vazio!";
    }
        else

   {
      //info "O campo foi aceito!";

                if( tickteDesconto >= totalDesconto )

                {info produto;
                listaProduto.add(produto.get("product").get("name"));//pegando
os produtos
            }

   }
}
//Criando Notas e a notificação de que o desconto não foi aprovado
info "Esses foram os produtos que não tem desconto aprovado:";

//criando a lista de produto
info listaProduto;

        notas = zoho.crm.getRelatedRecords("Notes", "Quotes", idOrcamento);
        note1 = Map();
        note1.put("Note_Title", "Produto Reprovado no desconto");
        note1.put("Note_Content", listaProduto);
        note1.put("Parent_Id", idOrcamento);
        note1.put("se_module", "Quotes");
        info note1;
        zoho.crm.createRecord("Notes", note1);

//Criando Notas e a notificação de que o desconto não foi aprovado
// Criando a notificação da  tarefa; >>>>>>>>>>

        //criando o mapa das tarefas
        mp=map();
```

```
        mp.put("Subject","Gustavo teste teste teste ");
        //definindo o modulo que o campo esta atribuido Deals: Orcamento
        mp.put("$se_module","Deals");
        //id negocio setado la no começo
        mp.put("What_Id",idNegocio);
        mp.put("Owner",orcamento.get("Owner").get("id"));
        mp.put("Status","Not Started");
        mp.put("Send_Notification_Email",true);
        createTask = zoho.crm.createRecord("Tasks",mp);
        info mp;
        info createTask;

//Criando a notificação da  tarefa; <<<<<<<<<<<<
```

19-Verificando se a porcentagem de desconto dada é superior a 20% se isso acontecer emitir uma nota e uma tarefa notificando.

```
orcamento = zoho.crm.getRecordById("Quotes",idOrcamento);
idNegocio = orcamento.get("Deal_Name").get("id");
detalhesProdutos =  orcamento.get("Product_Details");

listaProduto = list();
for each  produto in detalhesProdutos
{
        codigoProduto = produto.get("Product_Code"); //perguntar como pegar
o id de um obejeto acimma
        montante = produto.get("total");
        tickteDesconto = produto.get("Discount");
   vintePorcento= 0.2;
        totalDesconto = montante*vintePorcento;
    //contarProduto = produto.length();
   //info produto;
if ( tickteDesconto >= totalDesconto )

   {
      info "O campo foi aceito!";
      info produto;
              listaProduto.add(produto.get("product").get("name"));//pegando
os produtos
```

```
   }
}
if (listaProduto.length() == 0){
        info "a lista esta vazia";
        }
else{
//Criando Notas e a notificação de que o desconto não foi aprovado
info "Esses foram os produtos que não tem desconto aprovado:";
//criando a lista de produto
// info listaProduto;
   notas = zoho.crm.getRelatedRecords("Notes", "Quotes", idOrcamento);
        note1 = Map();
        note1.put("Note_Title", "Produto Reprovado no desconto");
        note1.put("Note_Content", listaProduto);
        note1.put("Parent_Id", idOrcamento);
        note1.put("se_module", "Quotes");
        info note1;
        zoho.crm.createRecord("Notes", note1);
   //-----------------------------------------
        mp=map();
        mp.put("Subject","Gustavo teste teste teste ");
        //definindo o modulo que o campo esta atribuido Deals: Orcamento
        mp.put("$se_module","Deals");
        //id negocio setado la no começo
        mp.put("What_Id",idNegocio);
        mp.put("Owner",orcamento.get("Owner").get("id"));
        mp.put("Status","Not Started");
        mp.put("Send_Notification_Email",true);
        createTask = zoho.crm.createRecord("Tasks",mp);
        info mp;
        info createTask;

}
```

20- Criar uma verificação no campo de CPF no módulo Contas, verificações:

-Se o campo está vazio
-Se tem quantidade de dígitos mínimos

```
-fazer o cálculo para ver se o dígito verificador é válido do cpf e do cnpj
-

v1 —----------------------------------------------------------


   contatos = zoho.crm.getRecordById("Contacts", idContatos);
        pegandoCpf = contatos.get("Cpf_Cnpj");
   info "CPF inicial: "+pegandoCpf;

        if (pegandoCpf == null )
   {
                info "campo vazio preencher!";
        }else {
        contando = pegandoCpf.length();
        info "Quantidade caracteres cpf inicial: "+contando;

        info"_____________________________________________________";

        VarCpfFormatted = pegandoCpf.replaceAll("[a-zA-z!@#$%^&*.-]","");
        info "CPF formatado: "+VarCpfFormatted;
        contando2 = VarCpfFormatted.length();
        info "Quantidade de carcteres pós formatacao: "+contando2;

        if(contando2>11){
                info "quantidade de caracteres acima do permetido";
        }

                if(contando2 != 11 || VarCpfFormatted == "00000000000" ||
VarCpfFormatted == "11111111111" ||VarCpfFormatted ==
"22222222222"||VarCpfFormatted == "33333333333" ||VarCpfFormatted ==
"44444444444" ||VarCpfFormatted == "55555555555" ||VarCpfFormatted ==
"66666666666" ||VarCpfFormatted == "77777777777" ||VarCpfFormatted ==
"88888888888" ||VarCpfFormatted == "99999999999") {

                info "Digite um CPF Válido com 11 digitos válidos!";
 }
  if ( contando2 == 11 ) {
          info "tem números válidos";
          info VarCpfFormatted;


num1=VarCpfFormatted.subString(0,1);num2=VarCpfFormatted.subString(1,2);
num3=VarCpfFormatted.subString(2,3);
```

```
num4=VarCpfFormatted.subString(3,4);num5=VarCpfFormatted.subString(4,5);
num6=VarCpfFormatted.subString(5,6);

num7=VarCpfFormatted.subString(6,7);num8=VarCpfFormatted.subString(7,8);
num9=VarCpfFormatted.subString(8,9);
num10=VarCpfFormatted.subString(9,10);num11=VarCpfFormatted.subString(1
0,11);

    num1Int = num1.toNumber(); num2Int = num2.toNumber();num3Int =
num3.toNumber();num4Int = num4.toNumber();
        num5Int = num5.toNumber();num6Int = num6.toNumber();num7Int =
num7.toNumber();num8Int = num8.toNumber();
        num9Int = num9.toNumber();num10Int = num10.toNumber();num11Int =
num11.toNumber();

        calc1= num1Int*10;calc2= num2Int*9;calc3= num3Int*8;calc4=
num4Int*7;calc5= num5Int*6;
        calc6= num6Int*5;calc7= num7Int*4;calc8= num8Int*3;calc9= num9Int*2;

        somandoDigitos01 =
calc1+calc2+calc3+calc4+calc5+calc6+calc7+calc8+calc9;
        info "Somatoria: "+somandoDigitos01;

        divisao01 = (somandoDigitos01*10)/11;
        divisaoResto = (somandoDigitos01*10)%11;

        info "Divisão: " + divisao01;
        info "Resto da divisão:"+divisaoResto;
        info "_____";

        // Verificando o primeiro digito do CPF

        if ( divisaoResto == 10 || divisaoResto ==11)
   {
                info "1º Digito verificador = 0";

       }else if ( divisaoResto==num10Int) {
       info "1º Digito verificador = "+num10Int;

       }if(divisaoResto == 10 || divisaoResto ==11 || divisaoResto==num10Int){

               info "uma das validaçoes foi aceita" ;
```

```
                info "_____";
            info "Calculando o segundo Digito!";

                calc1= num1Int*11;
                calc2= num2Int*10;
                calc3= num3Int*9;
                calc4= num4Int*8;
                calc5= num5Int*7;
            calc6= num6Int*6;
            calc7= num7Int*5;
            calc8= num8Int*4;
            calc9= num9Int*3;
                calc10 = num10Int*2;

//                  info calc1;info calc2;info calc3;info calc4;info calc5;info
calc6;info calc7;info calc8;info calc9;info calc10;
                    somandoDigitos2 =
calc1+calc2+calc3+calc4+calc5+calc6+calc7+calc8+calc9+calc10;

          info "Somatoria dos digitos:"+somandoDigitos2;
          divisao2 = (somandoDigitos2*10)/11;
          divisaoResto2 = (somandoDigitos2*10)%11;
          info "Valor da divisao:"+ divisao2;
          info  "Resto da divisão:"+divisaoResto2;
           y = num11.toNumber();

           if(divisaoResto2 == num11Int){
                    info "2º Digito válido!";
                    info "CPF ACEITO PARABENS!!";
                    }

                    if(divisaoResto2 == 10 || divisaoResto2 == 11){
                    info "Último digito 0";
                   }
        }

          } // Barra de quando o CPF TEM 11 digitos
}// else depois de verificar se o campo é nulo
```

21-Criar uma verificação no campo de Cnpj no módulo Contas, implementar o máximo de verificações possíveis:

```
contatos = zoho.crm.getRecordById("Contacts", idContatos);
pegandoCnpj = contatos.get("Cnpj");

        pegandoCnpj = contatos.get("Cnpj");
    info "CNPJ inicial: "+pegandoCnpj;

        if (pegandoCnpj == null )
    {
                info "campo vazio preencher!";
        }else {
        contando = pegandoCnpj.length();
        info "Quantidade caracteres cnpj inicial: "+contando;

        info"_____";

        VarCpfFormatted = pegandoCnpj.replaceAll("[a-zA-z!@#$%^&*./-]","");
        info "CPF formatado: "+VarCpfFormatted;
        contando2 = VarCpfFormatted.length();
        info "Quantidade de carcteres pós formatacao: "+contando2;

        if(contando2>14){
                info "quantidade de caracteres acima do permetido";
        }

                if(contando2 != 14 || VarCpfFormatted == "00000000000" ||
VarCpfFormatted == "11111111111" ||VarCpfFormatted ==
"22222222222"||VarCpfFormatted == "33333333333" ||VarCpfFormatted ==
"44444444444" ||VarCpfFormatted == "55555555555" ||VarCpfFormatted ==
"66666666666" ||VarCpfFormatted == "77777777777" ||VarCpfFormatted ==
"88888888888" ||VarCpfFormatted == "99999999999") {

                info "Digite um CNPJ Válido com 14 digitos válidos!";
 }
  if ( contando2 == 14 ) {
          info "tem números válidos";
          info VarCpfFormatted;
```

```
num1=VarCpfFormatted.subString(0,1);num2=VarCpfFormatted.subString(1,2);
num3=VarCpfFormatted.subString(2,3);

num4=VarCpfFormatted.subString(3,4);num5=VarCpfFormatted.subString(4,5);
num6=VarCpfFormatted.subString(5,6);

num7=VarCpfFormatted.subString(6,7);num8=VarCpfFormatted.subString(7,8);
num9=VarCpfFormatted.subString(8,9);
num10=VarCpfFormatted.subString(9,10);num11=VarCpfFormatted.subString(1
0,11);

    num1Int = num1.toNumber(); num2Int = num2.toNumber();num3Int =
num3.toNumber();num4Int = num4.toNumber();
        num5Int = num5.toNumber();num6Int = num6.toNumber();num7Int =
num7.toNumber();num8Int = num8.toNumber();
        num9Int = num9.toNumber();num10Int = num10.toNumber();num11Int =
num11.toNumber();

        calc1= num1Int*10;calc2= num2Int*9;calc3= num3Int*8;calc4=
num4Int*7;calc5= num5Int*6;
        calc6= num6Int*5;calc7= num7Int*4;calc8= num8Int*3;calc9= num9Int*2;

        somandoDigitos01 =
calc1+calc2+calc3+calc4+calc5+calc6+calc7+calc8+calc9;
        info "Somatoria: "+somandoDigitos01;

        divisao01 = (somandoDigitos01*10)/11;
        divisaoResto = (somandoDigitos01*10)%11;

        info "Divisão: " + divisao01;
        info "Resto da divisão:"+divisaoResto;
        info "_____";

        // Verificando o primeiro digito do CPF

        if ( divisaoResto == 10 || divisaoResto ==11)
   {
                info "1º Digito verificador = 0";

       }else if ( divisaoResto==num10Int) {
       info "1º Digito verificador = "+num10Int;

       }if(divisaoResto == 10 || divisaoResto ==11 || divisaoResto==num10Int){
```

```
                info "uma das validaçoes foi aceita" ;

                info "_____";
            info "Calculando o segundo Digito!";

                calc1= num1Int*11;
                calc2= num2Int*10;
                calc3= num3Int*9;
                calc4= num4Int*8;
                calc5= num5Int*7;
             calc6= num6Int*6;
             calc7= num7Int*5;
             calc8= num8Int*4;
             calc9= num9Int*3;
                calc10 = num10Int*2;

//              info calc1;info calc2;info calc3;info calc4;info calc5;info
calc6;info calc7;info calc8;info calc9;info calc10;
                somandoDigitos2 =
calc1+calc2+calc3+calc4+calc5+calc6+calc7+calc8+calc9+calc10;

          info "Somatoria dos digitos:"+somandoDigitos2;
          divisao2 = (somandoDigitos2*10)/11;
          divisaoResto2 = (somandoDigitos2*10)%11;
          info "Valor da divisao:"+ divisao2;
          info  "Resto da divisão:"+divisaoResto2;
           y = num11.toNumber();

           if(divisaoResto2 == num11Int){
                info "2º Digito válido!";
                info "CPF ACEITO PARABENS!!";
                }

                if(divisaoResto2 == 10 || divisaoResto2 == 11){
                info "Último digito 0";
               }
        }

         } // Barra de quando o CPF TEM 11 digitos
}// else depois de verificar se o campo é nulo
```

21 v2

```
contatos = zoho.crm.getRecordById("Contacts", idContatos);
pegandoCnpj = contatos.get("Cnpj");

        pegandoCnpj = contatos.get("Cnpj");
    info "CNPJ inicial: "+pegandoCnpj;

        if (pegandoCnpj == null )
    {
                info "campo vazio preencher!";
        }else {
        contando = pegandoCnpj.length();
        info "Quantidade caracteres cnpj inicial: "+contando;

        info"_____";

        VarCpfFormatted = pegandoCnpj.replaceAll("[a-zA-z!@#$%^&*./-]","");
        info "CPF formatado: "+VarCpfFormatted;
        contando2 = VarCpfFormatted.length();
        info "Quantidade de carcteres pós formatacao: "+contando2;

        if(contando2>14){
                info "quantidade de caracteres acima do permetido";
        }

                if(contando2 != 14 || VarCpfFormatted == "00000000000" ||
VarCpfFormatted == "11111111111" ||VarCpfFormatted ==
"22222222222"||VarCpfFormatted == "33333333333" ||VarCpfFormatted ==
"44444444444" ||VarCpfFormatted == "55555555555" ||VarCpfFormatted ==
"66666666666" ||VarCpfFormatted == "77777777777" ||VarCpfFormatted ==
"88888888888" ||VarCpfFormatted == "99999999999") {

                info "Digite um CNPJ Válido com 14 digitos válidos!";
 }
  if ( contando2 == 14 ) {
          info "tem números válidos";
          info VarCpfFormatted;
```

```
num1=VarCpfFormatted.subString(0,1);num2=VarCpfFormatted.subString(1,2);
num3=VarCpfFormatted.subString(2,3);

num4=VarCpfFormatted.subString(3,4);num5=VarCpfFormatted.subString(4,5);
num6=VarCpfFormatted.subString(5,6);

num7=VarCpfFormatted.subString(6,7);num8=VarCpfFormatted.subString(7,8);
num9=VarCpfFormatted.subString(8,9);
num10=VarCpfFormatted.subString(9,10);num11=VarCpfFormatted.subString(1
0,11);num12=VarCpfFormatted.subString(11,12);

num13=VarCpfFormatted.subString(12,13);num14=VarCpfFormatted.subString(
13,14);

    num1Int = num1.toNumber(); num2Int = num2.toNumber();num3Int =
num3.toNumber();num4Int = num4.toNumber();
        num5Int = num5.toNumber();num6Int = num6.toNumber();num7Int =
num7.toNumber();num8Int = num8.toNumber();
        num9Int = num9.toNumber();num10Int = num10.toNumber();num11Int =
num11.toNumber();num12Int = num12.toNumber();
        num13Int = num13.toNumber();num14Int = num14.toNumber();

        calc1= num1Int*5;calc2= num2Int*4;calc3= num3Int*3;calc4=
num4Int*2;calc5= num5Int*9;
        calc6= num6Int*8;calc7= num7Int*7;calc8= num8Int*6;calc9=
num9Int*5;calc10=num10Int*4;calc11=num11Int*3;
        calc12 = num12Int*2;

        somandoDigitos01 =
calc1+calc2+calc3+calc4+calc5+calc6+calc7+calc8+calc9;
        info "Somatoria: "+somandoDigitos01;

        divisao01 = (somandoDigitos01*10)/11;
        divisaoResto = (somandoDigitos01*10)%11;

        info "Divisão: " + divisao01;
        info "Resto da divisão:"+divisaoResto;
        info "_____";

        // Verificando o primeiro digito do CPF

        if ( divisaoResto == 10 || divisaoResto ==11)
    {
```

```
            info "1º Digito verificador = 0";

        }else if ( divisaoResto==num10Int) {
        info "1º Digito verificador = "+num10Int;

        }if(divisaoResto == 10 || divisaoResto ==11 || divisaoResto==num10Int){

            info "uma das validaçoes foi aceita" ;

            info "_____";
          info "Calculando o segundo Digito!";

            calc1= num1Int*11;
            calc2= num2Int*10;
            calc3= num3Int*9;
            calc4= num4Int*8;
            calc5= num5Int*7;
         calc6= num6Int*6;
         calc7= num7Int*5;
         calc8= num8Int*4;
         calc9= num9Int*3;
            calc10 = num10Int*2;

//              info calc1;info calc2;info calc3;info calc4;info calc5;info
calc6;info calc7;info calc8;info calc9;info calc10;
            somandoDigitos2 =
calc1+calc2+calc3+calc4+calc5+calc6+calc7+calc8+calc9+calc10;

        info "Somatoria dos digitos:"+somandoDigitos2;
        divisao2 = (somandoDigitos2*10)/11;
        divisaoResto2 = (somandoDigitos2*10)%11;
        info "Valor da divisao:"+ divisao2;
        info  "Resto da divisão:"+divisaoResto2;
         y = num11.toNumber();

         if(divisaoResto2 == num11Int){
            info "2º Digito válido!";
            info "CPF ACEITO PARABENS!!";
            }

            if(divisaoResto2 == 10 || divisaoResto2 == 11){
            info "Último digito 0";
           }
        }
```

```
            } // Barra de quando o CPF TEM 11 digitos
        }// else depois de verificar se o campo é nulo
```