

Create

Interação com os dados via C#



```
public static Guid Create()
{
    var registro = new Entity("vnb_registro_exemplo");
    registro["vnb_int_chave_alternativa"] = 0;
    registro["vnb_st_texto"] = "Texto";
    registro["vnb_pl_conjunto_de_opcoes"] = new OptionSetValue(1);
    registro["vnb_mpl_conjunto_de_opcoes_multipla_escolha"] = new OptionSetValueCollection() {
        new OptionSetValue(1),
        new OptionSetValue(3)
    };
    registro["vnb_bo_duas_opcoes"] = false;
    registro["vnb_int_numero_inteiro"] = 100;
    registro["vnb_fl_numero_de_ponto_flutuante"] = 100.1d;
    registro["vnb_dc_numero_decimal"] = 100.2m;
    registro["vnb_mn_moeda"] = new Money(100.3m);
    registro["vnb_dt_data_hora"] = DateTime.Now;
    registro["vnb_lp_contatoid"] = new EntityReference("contact", new Guid("8bcbdd6b-66e7-ec11-bb3c-002248de5579"));
    registro["vnb_lp_clienteid"] = new EntityReference("contact", new Guid("8bcbdd6b-66e7-ec11-bb3c-002248de5579"));
    return d365.ServiceClient.Create(registro);
}
```

Update

Interação com os dados via C#



```
public static void UpdateComId(Guid id)
{
    var registro = new Entity("vnb_registro_exemplo", id);
    registro["vnb_st_texto"] = "Atualizado via Id";
    d365.ServiceClient.Update(registro);
}
```

1 referência

```
public static void UpdateComChaveAlternativa(int chaveAlternativa)
{
    var registro = new Entity("vnb_registro_exemplo", "vnb_int_chave_alternativa", chaveAlternativa);
    registro["vnb_st_texto"] = "Atualizado via Chave Alternativa";
    d365.ServiceClient.Update(registro);
}
```

Upsert

Interação com os dados via C#



```
public static void Upsert(int chaveAlternativa)
{
    var registro = new Entity("vnb_registro_exemplo", "vnb_int_chave_alternativa", chaveAlternativa);
    registro["vnb_st_texto"] = "Criado/Atualizado via Upsert";

    var requisicao = new UpsertRequest()
    {
        Target = registro
    };

    var resposta = (UpsertResponse)d365.ServiceClient.Execute(requisicao);
}
```

Retrieve

Interação com os dados via C#



```
public static void Retrieve(Guid id)
{
    Entity registro = d365.ServiceClient.Retrieve("vnb_registro_exemplo",
        id,
        new ColumnSet(
            "vnb_st_texto",
            "vnb_pl_conjunto_de_opcoes",
            "vnb_mn_moeda",
            "vnb_lp_contatoid")
        );

    var texto = registro.GetAttributeValue<string>("vnb_st_texto");
    var conjuntoDeOpcoes = registro.GetAttributeValue<OptionSetValue>("vnb_pl_conjunto_de_opcoes");
    var moeda = registro.GetAttributeValue<Money>("vnb_mn_moeda");
    var contato = registro.GetAttributeValue<EntityReference>("vnb_lp_contatoid");

    System.Console.WriteLine($"Texto: {texto}");
    if (conjuntoDeOpcoes != null)
        System.Console.WriteLine($"Conjunto de Opções: {conjuntoDeOpcoes.Value}");
    if (moeda != null)
        System.Console.WriteLine($"Moeda: {moeda.Value}");
    if (contato != null)
        System.Console.WriteLine($"Tabela: {contato.LogicalName}, Id: {contato.Id}, Nome: {contato.Name}");
    System.Console.ReadLine();
}
```



Evite utilizar...
new columnSet(true);

RetrieveMultiple

Interação com os dados via C#



```
public static void RetrieveMultiple(int opcao)
{
    var consulta = @"<fetch version='1.0' output-format='xml-platform' mapping='logical' distinct='false'>
        <entity name='vnb_registro_exemplo'>
            <attribute name='vnb_registro_exemploid' />
            <attribute name='vnb_st_texto' />
            <attribute name='vnb_pl_conjunto_de_opcoes' />
            <attribute name='vnb_mn_moeda' />
            <attribute name='vnb_lp_contatoid' />
            <attribute name='createdby' />
            <filter type='and'>
                <condition attribute='vnb_pl_conjunto_de_opcoes' operator='eq' value='{opcao}' />
            </filter>
            <link-entity name='systemuser' from='systemuserid' to='createdby' visible='false' link-type='outer' alias='criado_por'>
                <attribute name='businessunitid' alias='unidade_de_negocios' />
            </link-entity>
        </entity>
    </fetch>";

    var resultados = d365.ServiceClient.RetrieveMultiple(new FetchExpression(consulta));

    System.Console.WriteLine($"{resultados.Entities.Count} encontrados para a opção {opcao}");

    foreach (var registro_ in resultados.Entities)
    {
        var id = registro_.Id;
        var texto = registro_.GetAttributeValue<string>("vnb_st_texto");
        var conjuntoDeOpcoes = registro_.GetAttributeValue<OptionSetValue>("vnb_pl_conjunto_de_opcoes");
        var moeda = registro_.GetAttributeValue<Money>("vnb_mn_moeda");
        var contato = registro_.GetAttributeValue<EntityReference>("vnb_lp_contatoid");
        var unidadeDeNegociosCriadoPor = new EntityReference();
        if (registro_.Contains("createdby"))
            unidadeDeNegociosCriadoPor = (EntityReference)registro_.GetAttributeValue<AliasedValue>("unidade_de_negocios").Value;
    }
}
```



Evite utilizar...
<all-attributes/>

Paginação = 5000

Permite Agregação

Linq

Interação com os dados via C#



```
public static void Linq()
{
    using (var contexto = new OrganizationServiceContext(d365.ServiceClient))
    {
        var registros = (from m_ in contexto.CreateQuery("vnb_registro_exemplo")
                        where (Int32)m_["vnb_int_numero_inteiro"] > 500
                        select new
                        {
                            Id = m_.Id,
                            Texto = m_["vnb_st_texto"],
                            Inteiro = m_["vnb_int_numero_inteiro"]
                        }).ToList();

        System.Console.WriteLine($"{registros.Count} encontrados para o valor maior que 500");

        foreach (var registro_ in registros)
        {
            // ....
        }
    }
}
```

Lambda

Interação com os dados via C#



```
public static void Lambda()
{
    using (var contexto = new OrganizationServiceContext(d365.ServiceClient))
    {
        var registros = contexto.CreateQuery("vnb_registro_exemplo")
            .Where(w => (Int32)w["vnb_int_numero_inteiro"] < 500
                && (OptionSetValue)w["vnb_pl_conjunto_de_opcoes"] != null
                && (OptionSetValue)w["vnb_pl_conjunto_de_opcoes"] == new OptionSetValue(1))
            .Select(s => new
            {
                Id = s.Id,
                Texto = s["vnb_st_texto"],
                Inteiro = s["vnb_int_numero_inteiro"]
            }).ToList();

        System.Console.WriteLine($"{registros.Count} encontrados com valores menores que 500 e Opção A");

        foreach (var registro_ in registros)
        {
            // ....
        }
    }
}
```

QueryExpression

Interação com os dados via C#



```
public static EntityCollection QueryExpression(D365 d365, int status = 0)
{
    QueryExpression consulta = new QueryExpression("vnb_registro_exemplo");
    consulta.ColumnSet.AddColumn("vnb_st_texto", "vnb_int_numero_inteiro");

    FilterExpression filtro = new FilterExpression();

    ConditionExpression condicao = new ConditionExpression();
    condicao.AttributeName = "statecode";
    condicao.Operator = ConditionOperator.Equal;
    condicao.Values.Add(0);
    filtro.Conditions.Add(condicao);

    consulta.Criteria.AddFilter(filtro);

    EntityCollection resultados = d365.ServiceClient.RetrieveMultiple(consulta);

    System.Console.WriteLine($"{resultados.Entities.Count} encontrados com status Ativo");

    return resultados;
}
```


Delete

Interação com os dados via C#



```
public static void Delete(EntityReference refRegistro)
{
    d365.ServiceClient.Delete(refRegistro.LogicalName, refRegistro.Id);
}
```

ExecuteTransactionRequest

Interação com os dados via C#



```
public void Transacao()
{
    var transacao = new ExecuteTransactionRequest()
    {
        Requests = new OrganizationRequestCollection()
    };

    var registro1 = new Entity("vnb_registro_exemplo");
    registro1["vnb_int_chave_alternativa"] = 999;
    var acao1 = new CreateRequest()
    {
        Target = registro1
    };
    transacao.Requests.Add(acao1);

    var registro2 = new Entity("vnb_registro_exemplo");
    registro2["vnb_int_chave_alternativa"] = 998;
    var acao2 = new CreateRequest()
    {
        Target = registro2
    };
    transacao.Requests.Add(acao2);

    var registro3 = new Entity("vnb_registro_exemplo");
    registro3["vnb_int_chave_alternativa"] = "997";
    var acao3 = new CreateRequest()
    {
        Target = registro3
    };
    transacao.Requests.Add(acao3);

    d365.ServiceClient.Execute(transacao);
}
```

ExecuteMultipleRequest

Interação com os dados via C#



```
public void MultiplasRequisicoes()
{
    var requisicao = new ExecuteMultipleRequest()
    {
        Settings = new ExecuteMultipleSettings()
        {
            ContinueOnError = true,
            ReturnResponses = true,
        },
        Requests = new OrganizationRequestCollection()
    };

    for (int i = 9000; i < 9050; i++)
    {
        var registro = new Entity("vnb_registro_exemplo");
        registro["vnb_int_chave_alternativa"] = i;
        var criar = new CreateRequest()
        {
            Target = registro
        };
        requisicao.Requests.Add(criar);

        if (requisicao.Requests.Count >= 25)
        {
            d365.ServiceClient.Execute(requisicao);
            requisicao.Requests.Clear();
        }
    }
}
```

Create com Earlybound

Interação com os dados via C#



```
public static Guid CreateEarlybound()
{
    var registro = new vnb_registro_exemplo();
    registro.vnb_int_chave_alternativa = 0;
    registro.vnb_st_texto = "Texto";
    registro.vnb_pl_conjunto_de_opcoes = vnb_registro_exemplo_vnb_pl_conjunto_de_opcoes.OpcaoA;
    registro.vnb_mpl_conjunto_de_opcoes_multipla_escolha =
        new List<vnb_registro_exemplo_vnb_mpl_conjunto_de_opcoes_multipla_escolha>() {
            vnb_registro_exemplo_vnb_mpl_conjunto_de_opcoes_multipla_escolha.Opcao1,
            vnb_registro_exemplo_vnb_mpl_conjunto_de_opcoes_multipla_escolha.Opcao2
        };
    registro.vnb_bo_duas_opcoes = false;
    registro.vnb_int_numero_inteiro = 100;
    registro.vnb_fl_numero_de_ponto_flutuante = 100.1d;
    registro.vnb_dc_numero_decimal = 100.2m;
    registro.vnb_mn_moeda = new Money(100.3m);
    registro.vnb_dt_data_hora = DateTime.Now;
    registro.vnb_lp_contatoid = new EntityReference(Contact.EntityLogicalName, new Guid("8bcbdd6b-66e7-ec11-bb3c-002248de5579"));
    registro.vnb_lp_clienteid = new EntityReference(Contact.EntityLogicalName, new Guid("8bcbdd6b-66e7-ec11-bb3c-002248de5579"));
    return d365.ServiceClient.Create(registro);
}
```

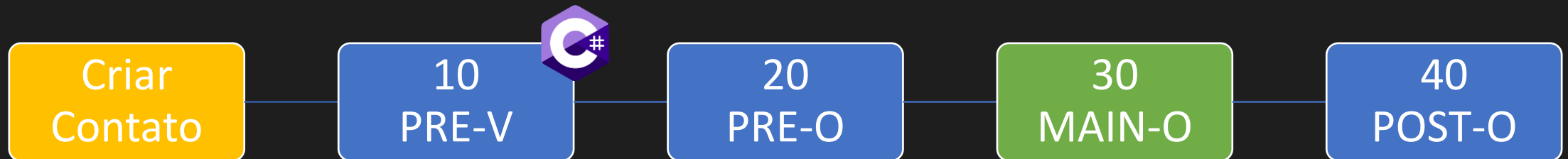
Pre-Validation

```
namespace Done.Extends.Plugins.Contato
{
    2 referências
    public class PreValidation : PluginBase
    {
        0 referências
        public PreValidation() : base(typeof(PreValidation)) { }

        2 referências
        protected override void ExecuteCrmPlugin(LocalPluginContext localContext)
        {
            var contact = localContext.GetTarget<Entity>();

            var task = new Entity("task");
            task["subject"] = "Criado via Plugin PRE-VALIDATION";
            task["ownerid"] = contact["ownerid"];
            localContext.OrganizationService.Create(task);

            throw new InvalidPluginExecutionException("Exceção na thread PRE-VALIDATION");
        }
    }
}
```



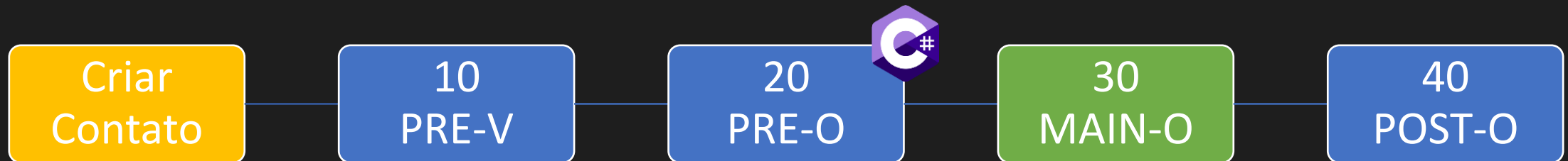
Pre-Operation

```
namespace Done.Extends.Plugins.Contato
{
    2 referências
    public class PreOperation : PluginBase
    {
        0 referências
        public PreOperation() : base(typeof(PreOperation)) { }

        2 referências
        protected override void ExecuteCrmPlugin(LocalPluginContext localContext)
        {
            var contact = localContext.GetTarget<Entity>();

            var task = new Entity("task");
            task["subject"] = "Criado via Plugin PRE-OPERATION";
            task["ownerid"] = contact["ownerid"];
            localContext.OrganizationService.Create(task);

            throw new InvalidPluginExecutionException("Exceção na thread PRE-OPERATION");
        }
    }
}
```



Post-Operation

```
namespace Done.Extends.Plugins.Contato
{
    2 referências
    public class PostOperation : PluginBase
    {
        0 referências
        public PostOperation() : base(typeof(PostOperation)) { }

        2 referências
        protected override void ExecuteCrmPlugin(LocalPluginContext localContext)
        {
            var contact = localContext.GetTarget<Entity>();

            var task = new Entity("task");
            task["subject"] = "Criado via Plugin POST-OPERATION";
            task["regardingobjectid"] = contact.ToEntityReference();
            task["ownerid"] = contact["ownerid"];
            localContext.OrganizationService.Create(task);

            throw new InvalidPluginExecutionException("Exceção na thread POST-OPERATION");
        }
    }
}
```

Criar
Contato

10
PRE-V

20
PRE-O

30
MAIN-O

40
POST-O

