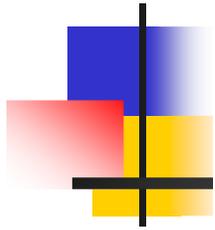


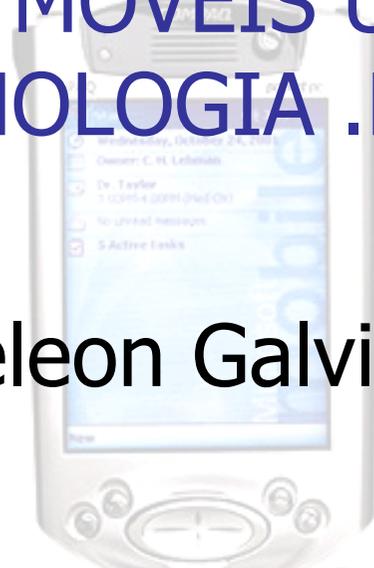
PROTÓTIPO DE SISTEMA CRM PARA DISPOSITIVOS MÓVEIS UTILIZANDO A TECNOLOGIA .NET



Deleon Galvin

ORIENTADOR

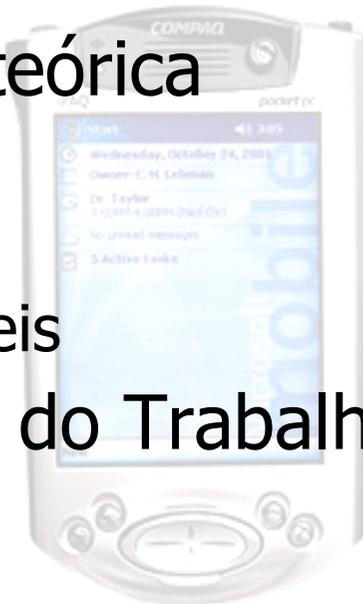
Prof. Francisco Adell Péricas

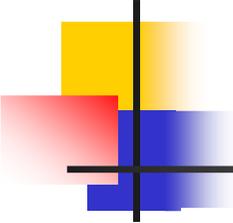


Roteiro

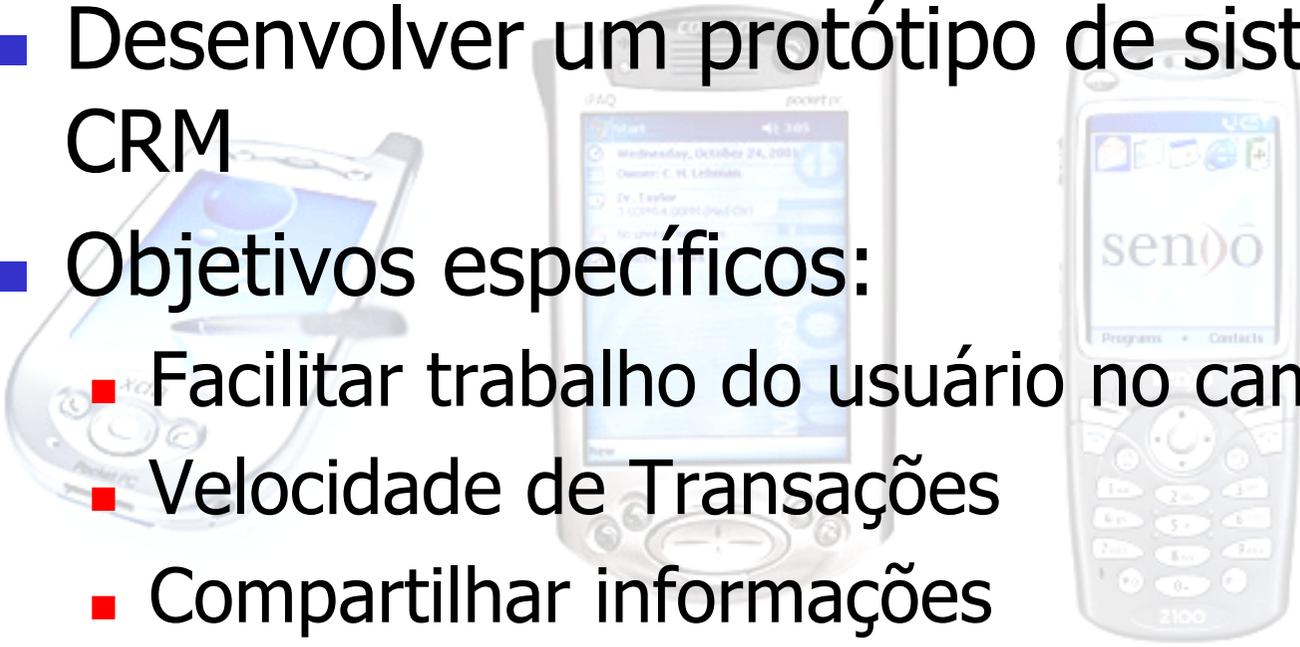
Introdução

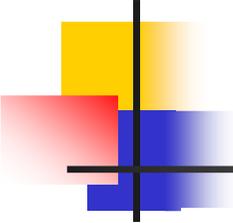
- Fundamentação teórica
 - CRM
 - .NET
 - Dispositivos Móveis
- Desenvolvimento do Trabalho
 - Requisitos
 - Especificação
 - Implementação
- Resultados, Discussões, Conclusão e Extensões



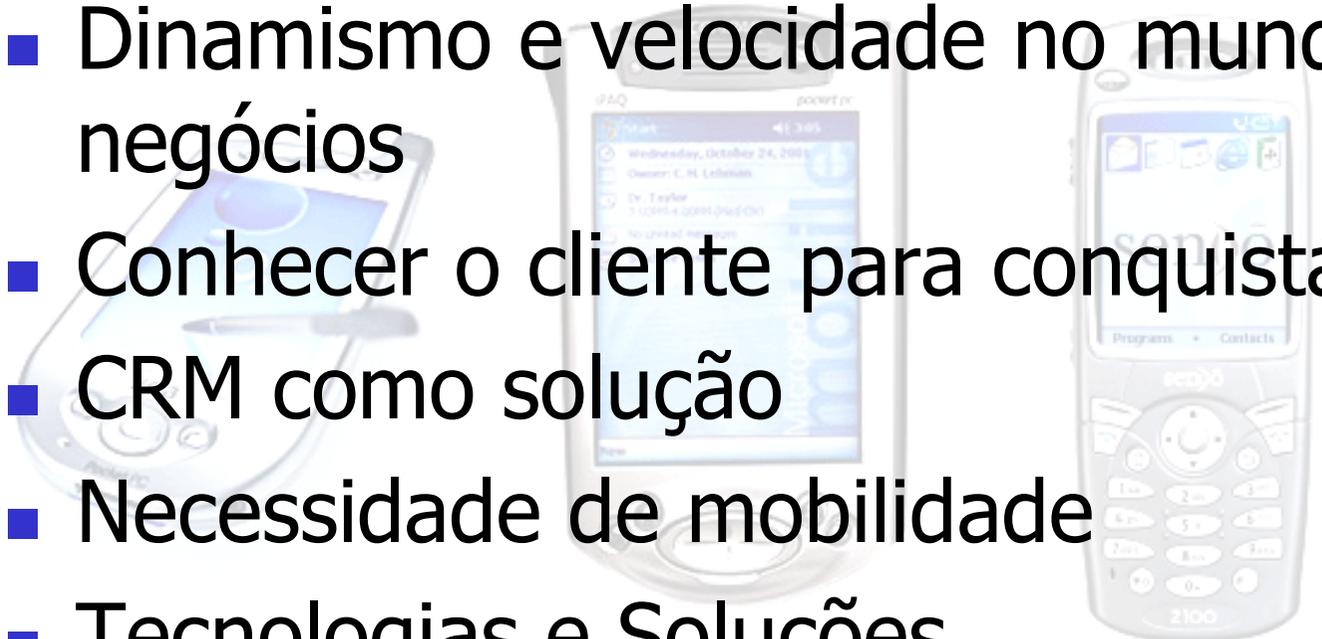


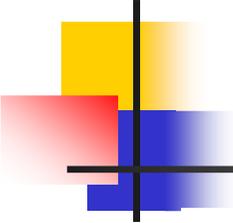
Objetivos

- Desenvolver um protótipo de sistema CRM
 - Objetivos específicos:
 - Facilitar trabalho do usuário no campo
 - Velocidade de Transações
 - Compartilhar informações
 - Mobilidade e Flexibilidade
- 

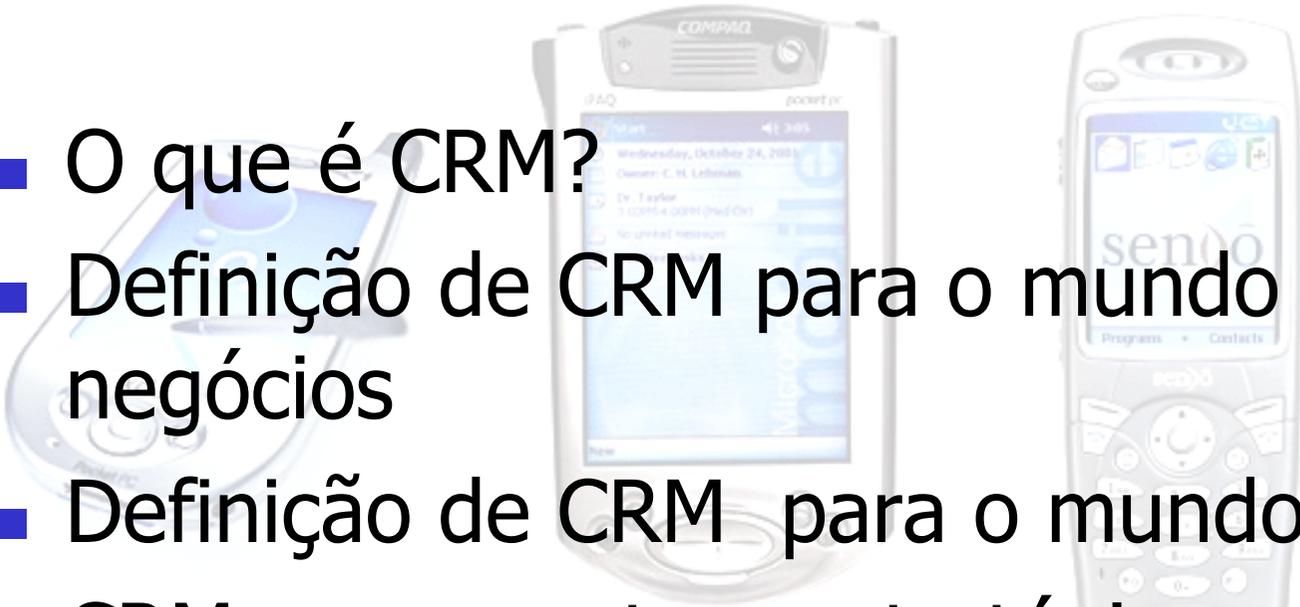


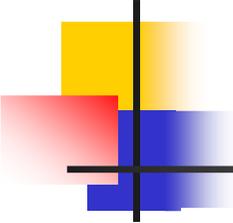
Introdução

- Dinamismo e velocidade no mundo dos negócios
 - Conhecer o cliente para conquistá-lo
 - CRM como solução
 - Necessidade de mobilidade
 - Tecnologias e Soluções
 - .NET como solução
- 



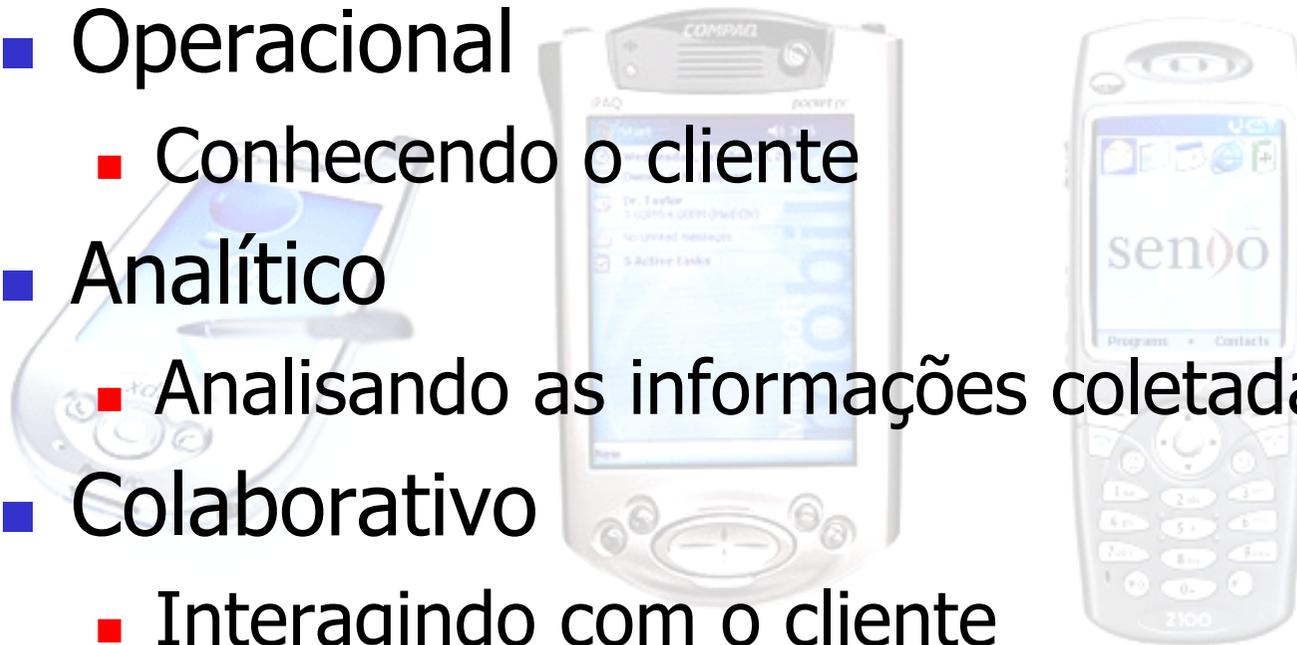
CRM

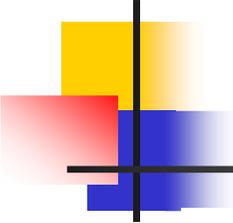
- 
- O que é CRM?
 - Definição de CRM para o mundo dos negócios
 - Definição de CRM para o mundo de TI
 - CRM como postura estratégica



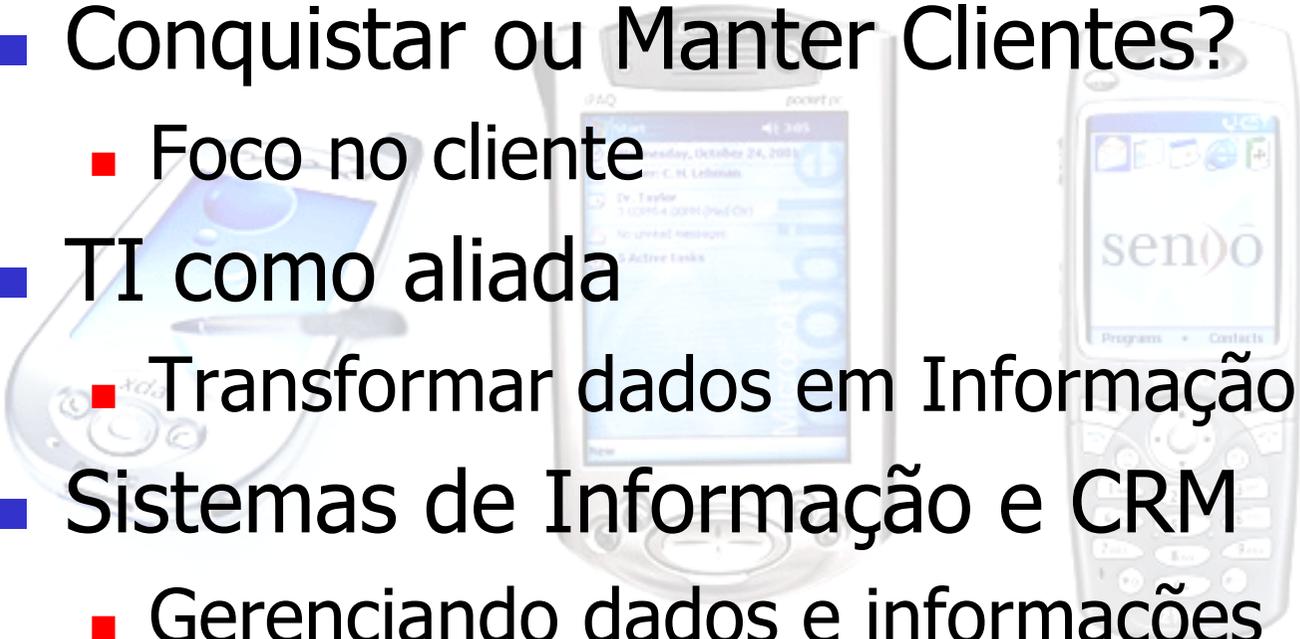
Divisões do CRM

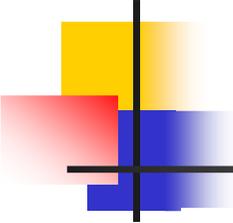
- Operacional
 - Conhecendo o cliente
- Analítico
 - Analisando as informações coletadas
- Colaborativo
 - Interagindo com o cliente



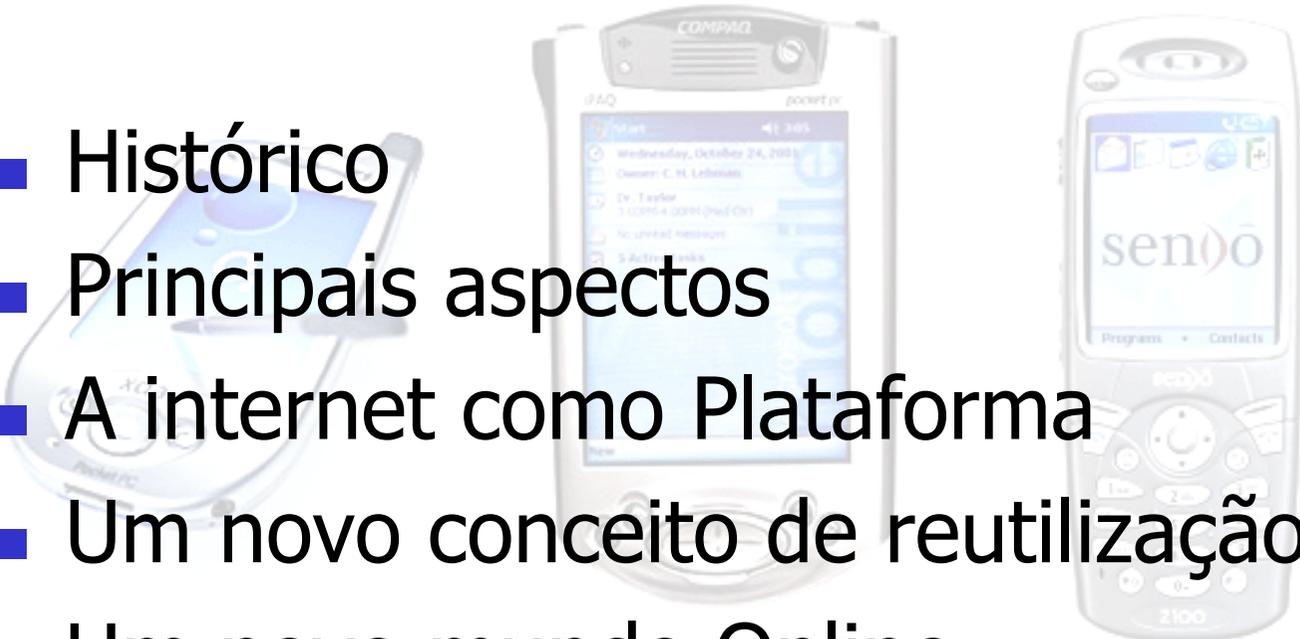


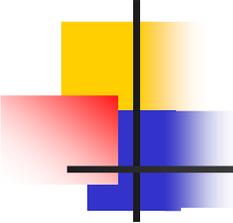
CRM e Estratégias

- Conquistar ou Manter Clientes?
 - Foco no cliente
 - TI como aliada
 - Transformar dados em Informação
 - Sistemas de Informação e CRM
 - Gerenciando dados e informações
- 

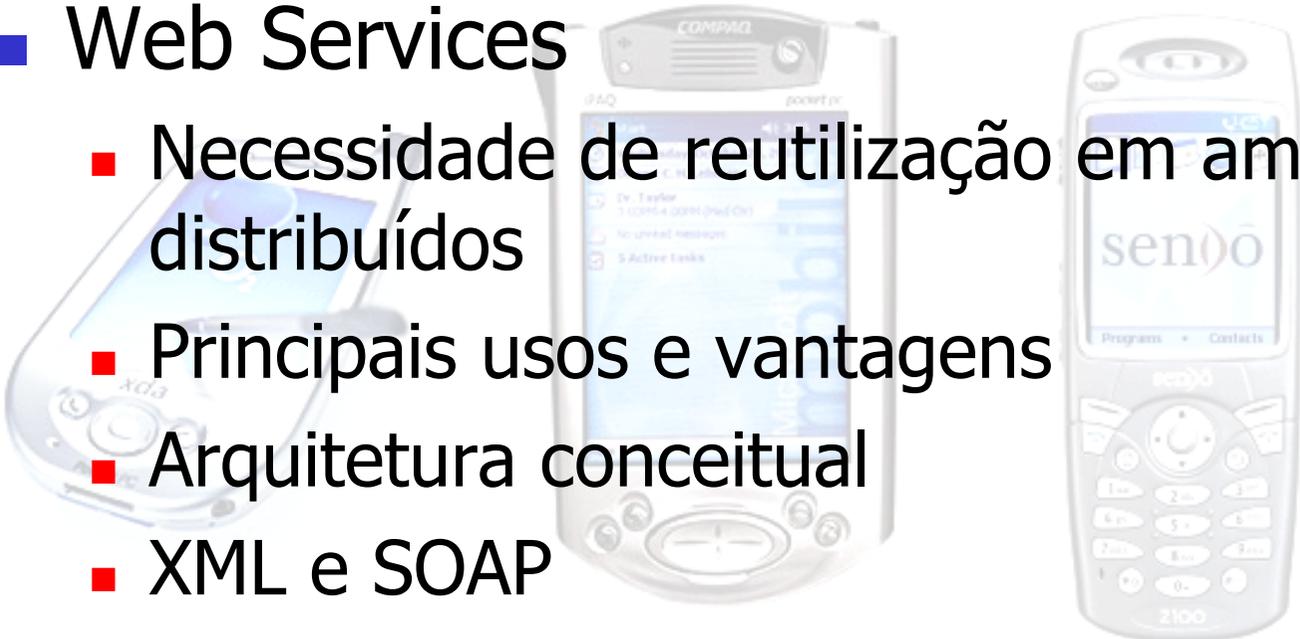


Microsoft .NET

- Histórico
 - Principais aspectos
 - A internet como Plataforma
 - Um novo conceito de reutilização
 - Um novo mundo Online
- 

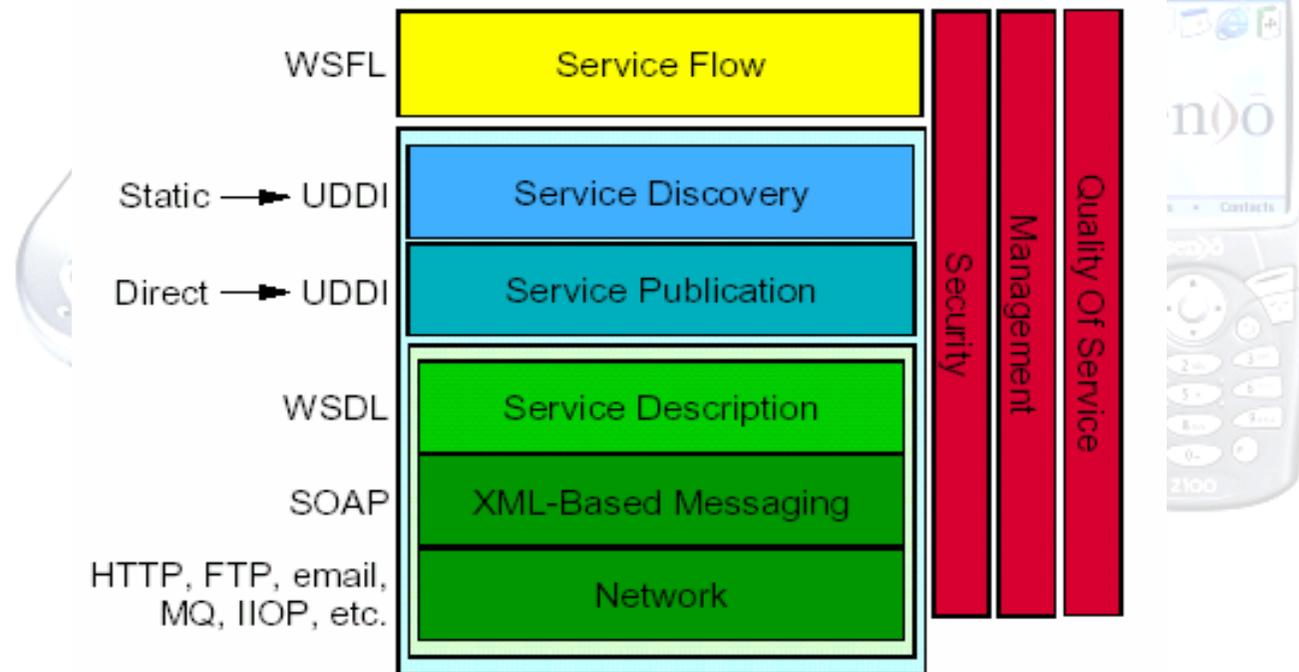


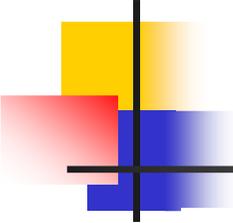
Microsoft .NET

- Web Services
 - Necessidade de reutilização em ambientes distribuídos
 - Principais usos e vantagens
 - Arquitetura conceitual
 - XML e SOAP
- 

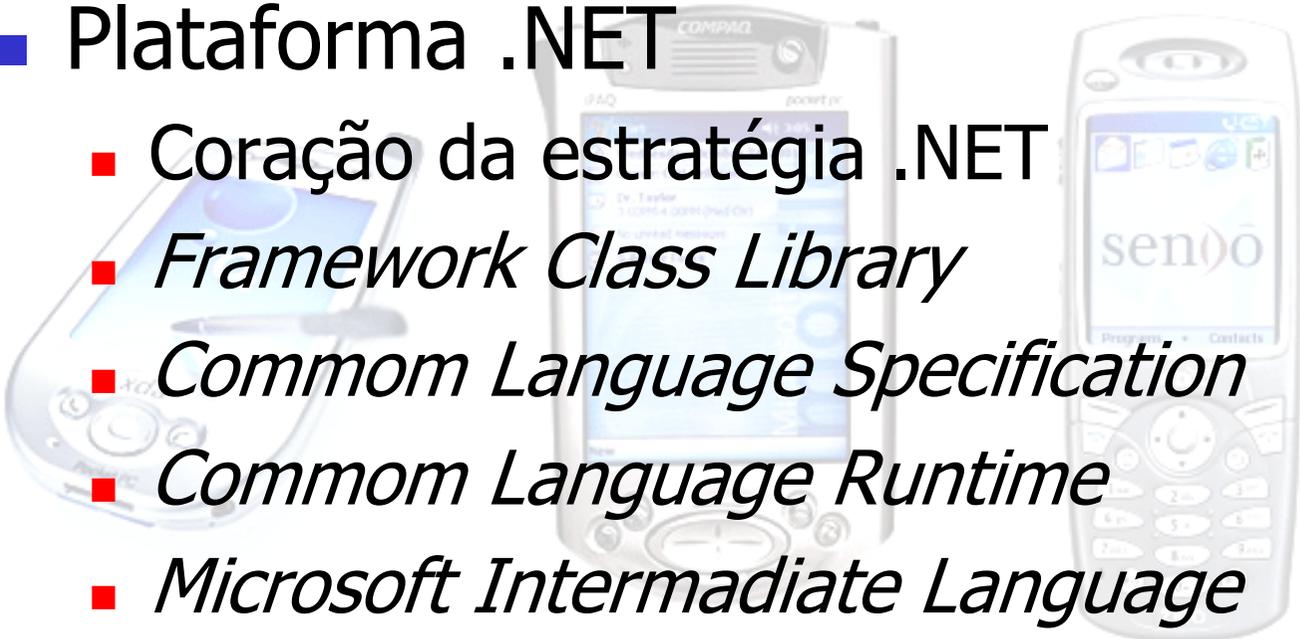
Microsoft .NET

- Arquitetura do Web Service



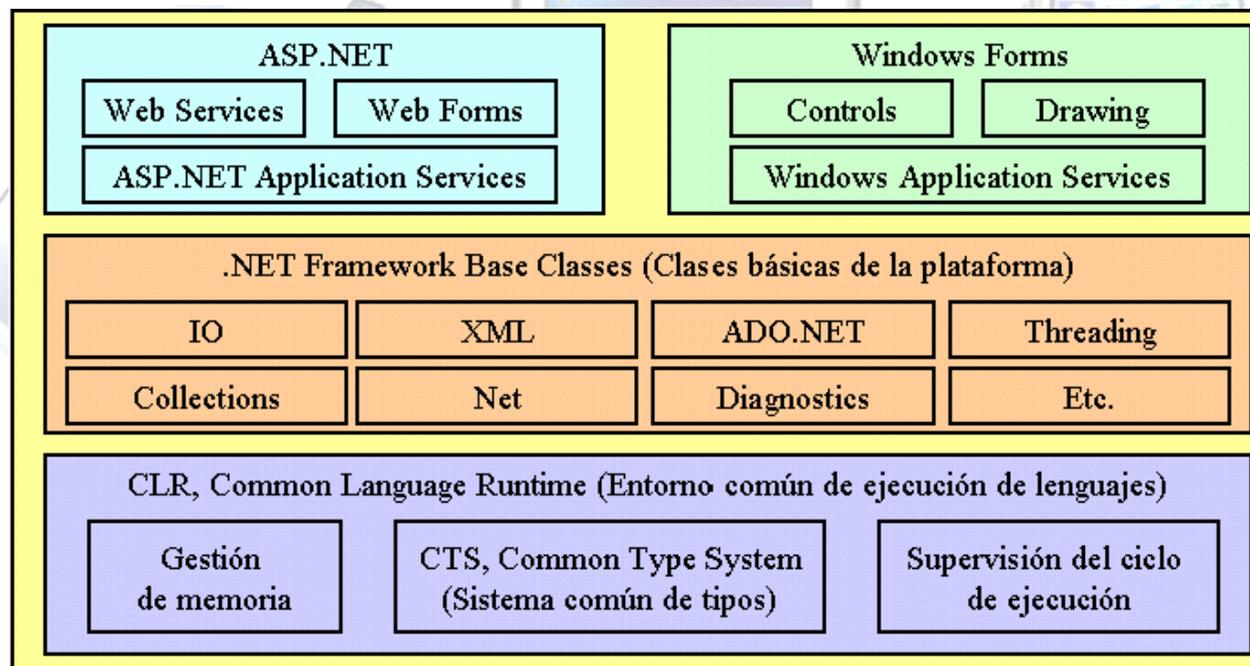


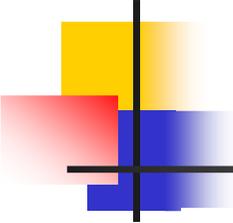
Microsoft .NET

- Plataforma .NET
 - Coração da estratégia .NET
 - *Framework Class Library*
 - *Common Language Specification*
 - *Common Language Runtime*
 - *Microsoft Intermediate Language*
- 

Microsoft .NET

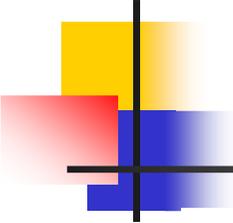
■ *Framework .NET*





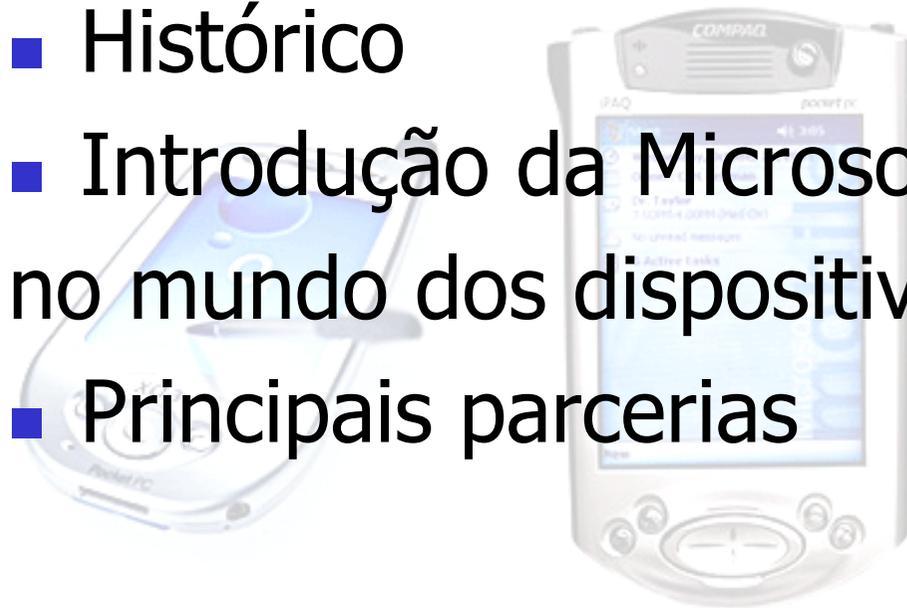
Microsoft .NET

- *Compact Framework .NET*
 - Desenvolvimento *Smart Devices*
 - Características
 - Linguagens
 - Utilização dos recursos de *hardware*
- 



Dispositivos Móveis

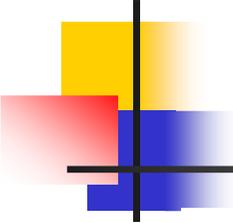
- Histórico
- Introdução da Microsoft no mundo dos dispositivos
- Principais parcerias



Dispositivos Móveis

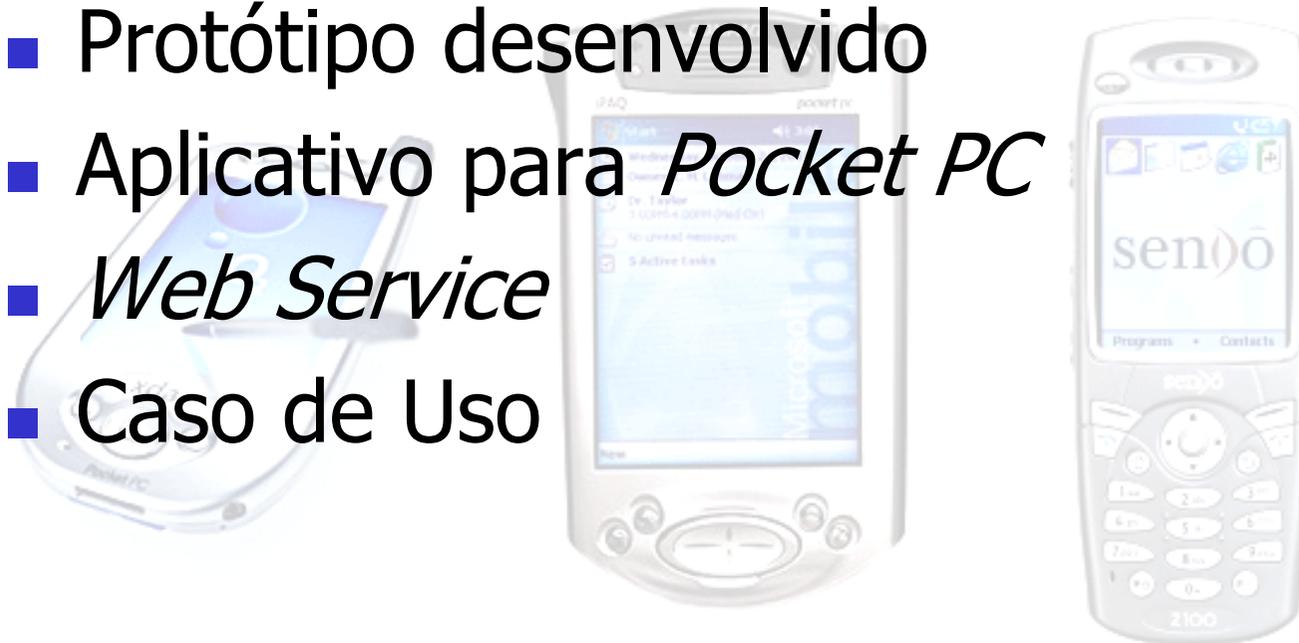
- Suporte ao *Windows CE .NET*
- Vantagens dos Dispositivos Móveis

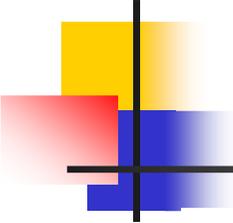




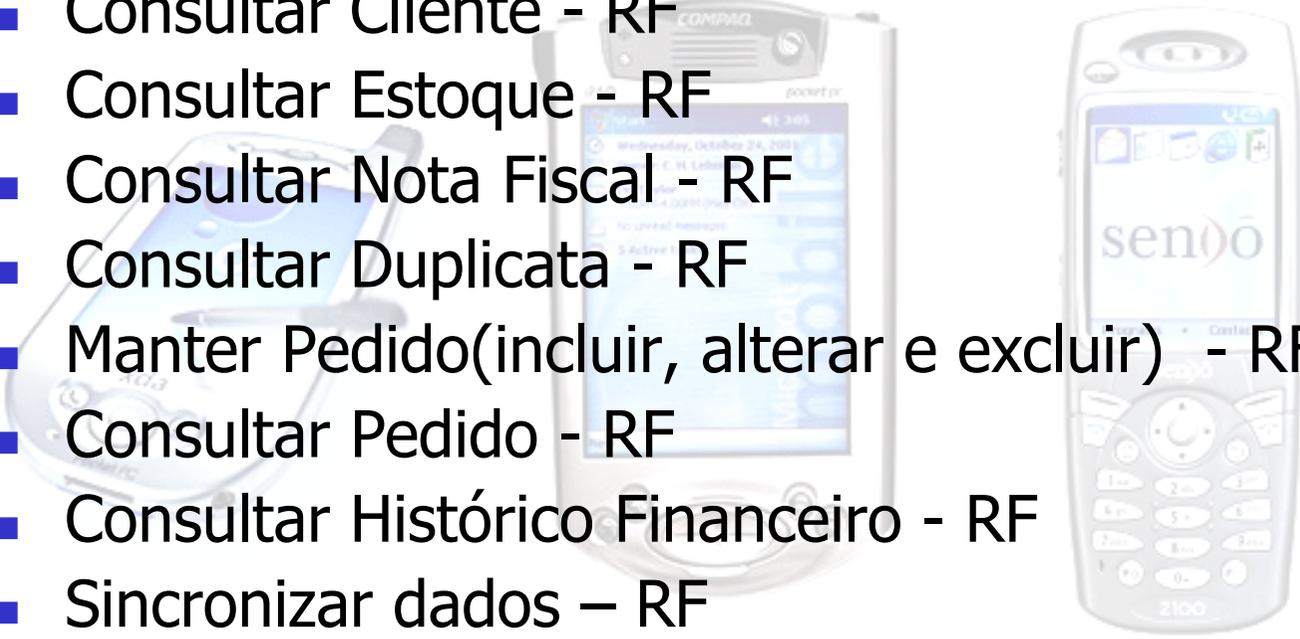
Desenvolvimento

- Protótipo desenvolvido
- Aplicativo para *Pocket PC*
- *Web Service*
- Caso de Uso



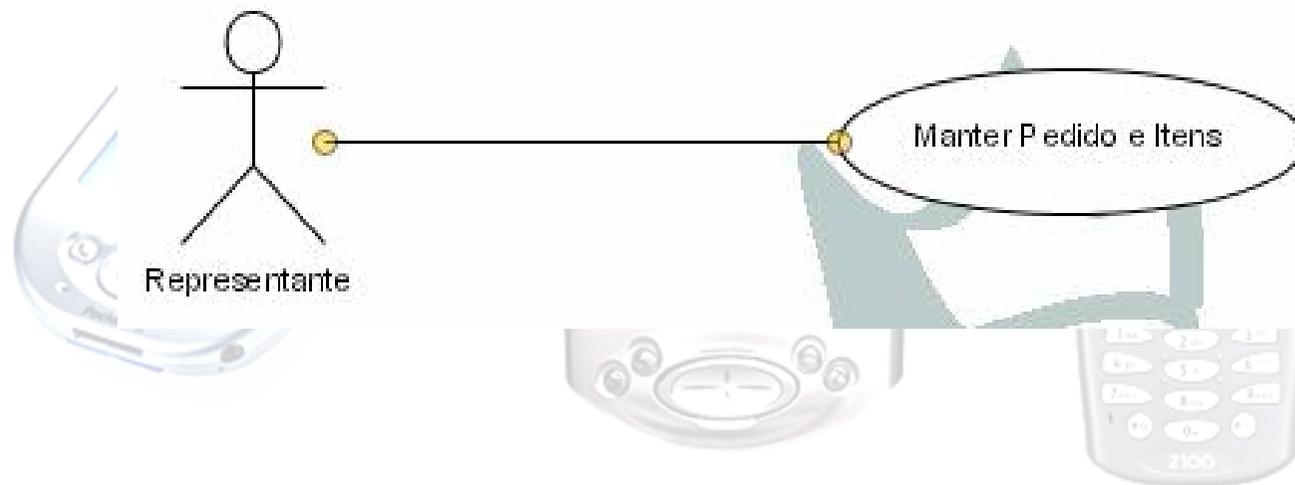


Desenvolvimento – Requisitos

- Consultar Cliente - RF
 - Consultar Estoque - RF
 - Consultar Nota Fiscal - RF
 - Consultar Duplicata - RF
 - Manter Pedido(incluir, alterar e excluir) - RF
 - Consultar Pedido - RF
 - Consultar Histórico Financeiro - RF
 - Sincronizar dados – RF
 - Interface Reduzida e amigável – RNF
 - Base de dados Stand Alone - RNF
- 

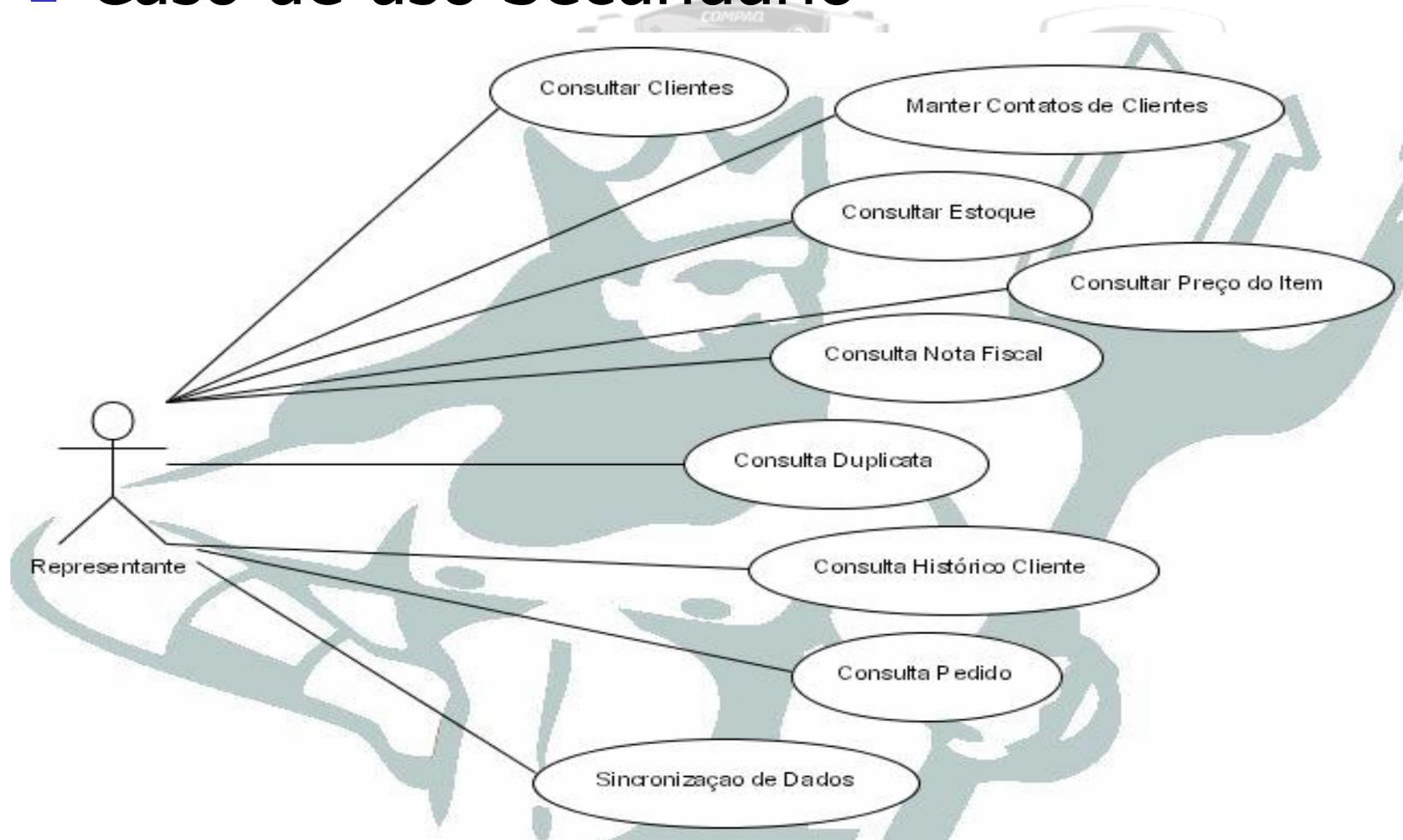
Desenvolvimento Especificação

- Caso de uso Primário



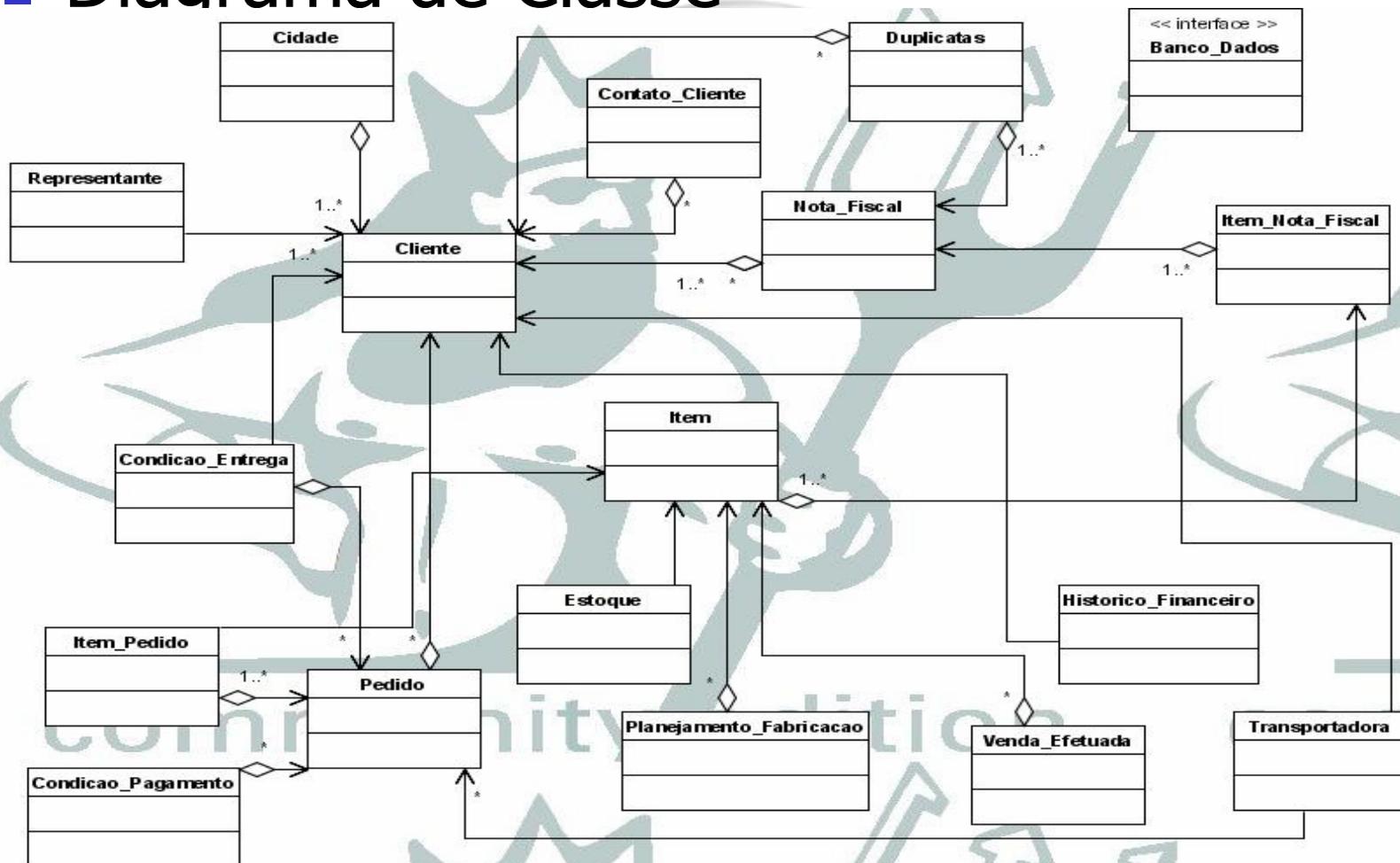
Desenvolvimento Especificação

■ Caso de uso Secundário



Desenvolvimento Especificação

■ Diagrama de Classe



Desenvolvimento Implementação

- Ferramentas utilizadas
 - Visual Studio .NET
 - SQL Server 2000
 - Poseidon UML



Desenvolvimento Implementação

- Visual Studio .NET
 - Compatibilidade e Vantagens
- Linguagens suportadas pelo ambiente
 - C#, C++, J#, VB.NET, ASP.NET
- Linguagem Utilizada
 - Visual Basic .NET

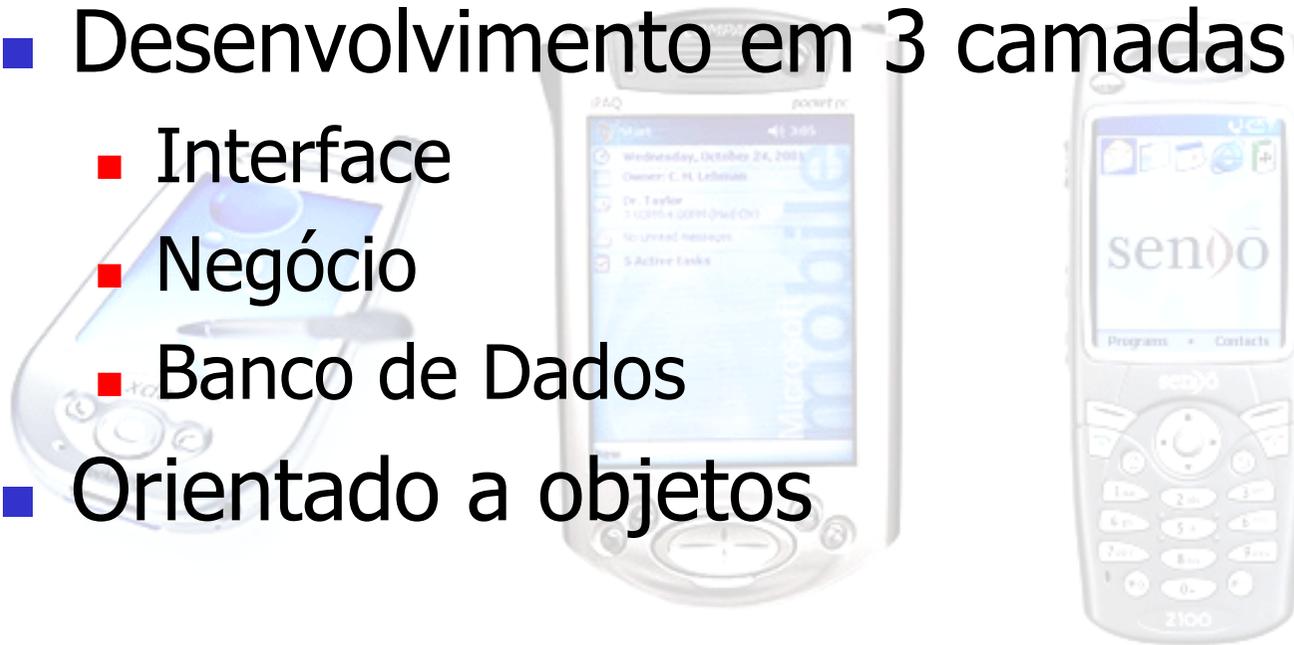
Desenvolvimento Implementação

- SQL Server 2000
 - Vantagens
 - Versões
- SQL Server 2000 Compact Edition
 - Vantagens
 - Funcionalidades



Desenvolvimento Implementação

- Desenvolvimento em 3 camadas
 - Interface
 - Negócio
 - Banco de Dados
- Orientado a objetos



Desenvolvimento Implementação

■ Exemplo de código de interface

```
Private Sub btnPesquisarPedido_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles btnPesquisarPedido.Click
    Dim dt As New DataTable
    Try
        If TxtboxCdCliente.Text = "" Then
            MessageBox.Show("Favor informar um cliente válido", "Atenção")
            TxtboxCdCliente.Focus()
            Exit Sub
        End If
        If TxtboxPedido.Text <> "" Then
            dt = negPed.buscar_Pedido(TxtboxPedido.Text, CInt(TxtboxCdCliente.Text))
        Else
            dt = negPed.buscar_Pedido(TxtboxDtInicial.Text, TxtboxDtFinal.Text, CInt(TxtboxCdCliente.Text))
        End If
        DtgrPedido.DataSource = dt
    Catch ex As Exception
        MessageBox.Show("Erro ao tentar carregar os Pedidos" & ex.Message, "Atencao", MessageBoxButtons.OK, _
            MessageBoxIcon.Exclamation, MessageBoxDefaultButton.Button1)
    End Try
End Sub
```

Desenvolvimento Implementação

■ Exemplo de código de negócio

```
Friend Function gera_CodigoPedido() As String
    Dim dt As New DataTable
    Dim cdPedido As String

    Try
        dt = bd.busca_SequenciaPedido(1)
        cdPedido = dt.Rows(0).Item("CD_SEQ_SEQUENCIA")
        cdPedido = CStr(CInt(cdPedido) + 1)
        bd.grava_SequenciaPedido(1, cdPedido)
        Return cdPedido
    Catch ex As Exception
        MessageBox.Show("Erro ao gerar o codigo do pedido " & ex.Message, "Atenção", MessageBoxButtons.OK, _
            MessageBoxIcon.Asterisk, MessageBoxDefaultButton.Button1)
    End Try
End Function
```

Desenvolvimento Implementação

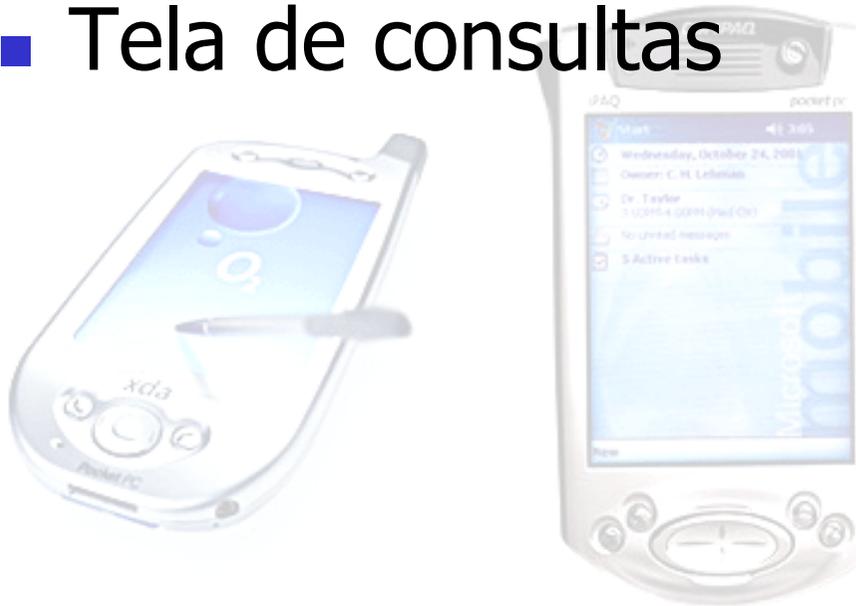
■ Exemplo de código de banco de dados

```
Protected Friend Function busca_Pedido(ByVal cdPedido As String, ByVal cdCliente As Integer) As DataTable
    Dim dt As New DataTable
    Dim daCliente As New SqlDataAdapter("SELECT CD_PEDIDO, CD_CONDICAO_ENTREGA, CD_CONDICAO_PAGTO," & _
        "CD_CLIENTE, CD_TRANSPORTADORA, CD_REPRESENTANTE, DS_PEDIDO_CLIENTE, DT_EMISSAO, DT_ENTREGA," & _
        "TP_FRETE, DS_OBSERVACAO_PEDIDO, VL_TOTAL, PR_DESCONTO_CAPA FROM PEDIDO WHERE CD_PEDIDO = " & _
        cdPedido & " AND CD_CLIENTE = " & cdCliente, frmAcesso.conString)

    Try
        daCliente.Fill(dt)
        Return dt
    Catch ex As SqlCeException
        MessageBox.Show("Erro ao buscar PEDIDOS em BD " & ex.Message, "Atencao", MessageBoxButtons.OK, _
            MessageBoxIcon.Exclamation, MessageBoxDefaultButton.Button1)
    End Try
End Function
```

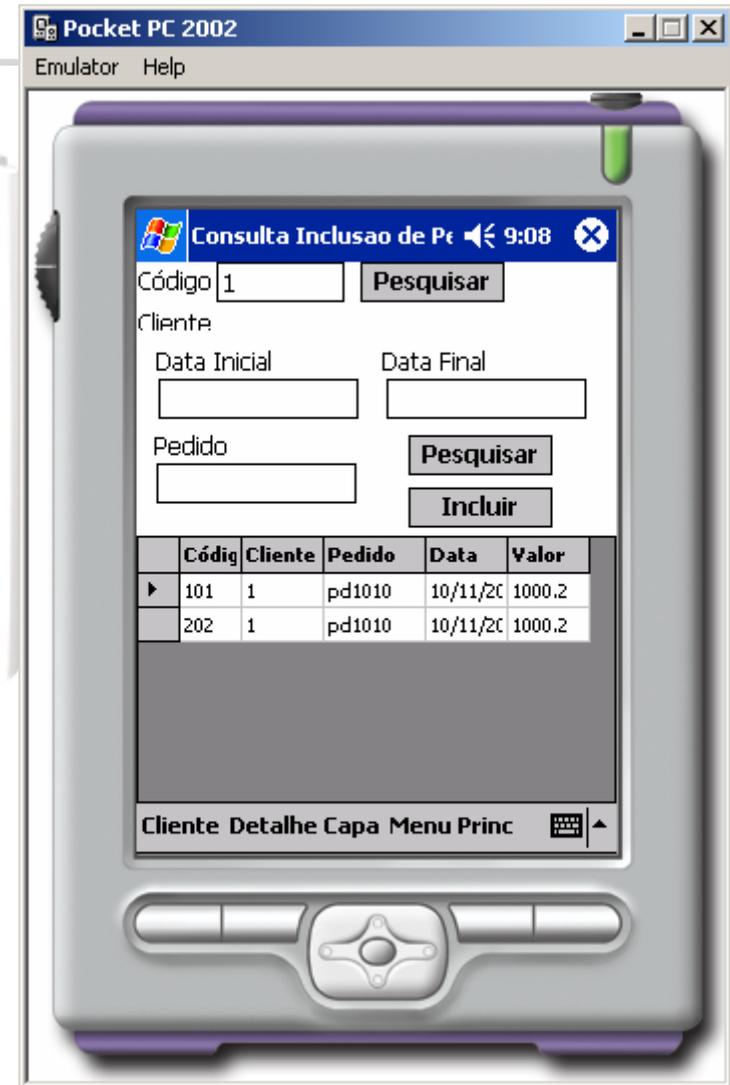
Desenvolvimento Operacionalidade

- Tela de consultas



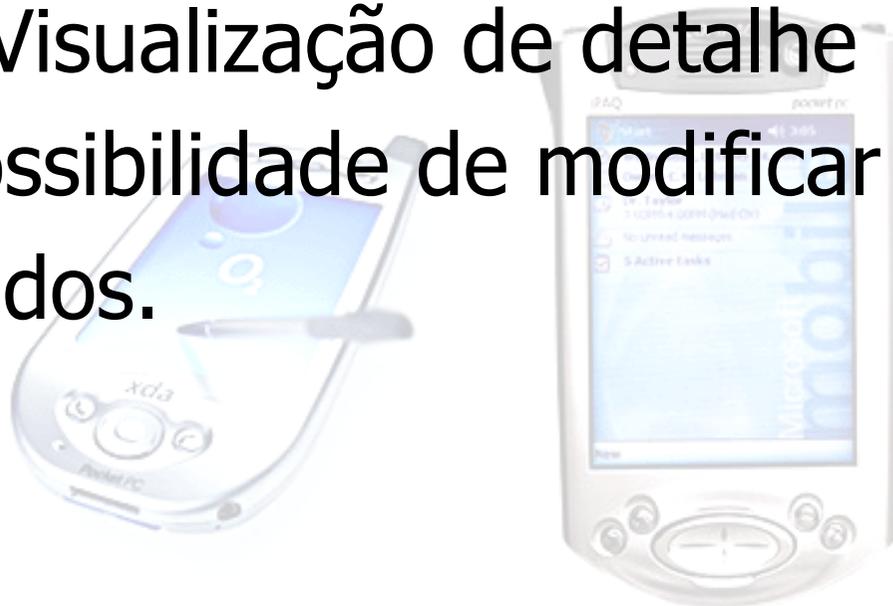
Desenvolvimento Operacionalidade

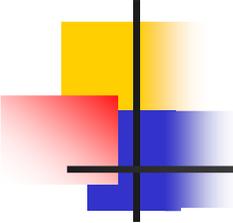
- Tela de consulta com detalhamento e inclusão de dados



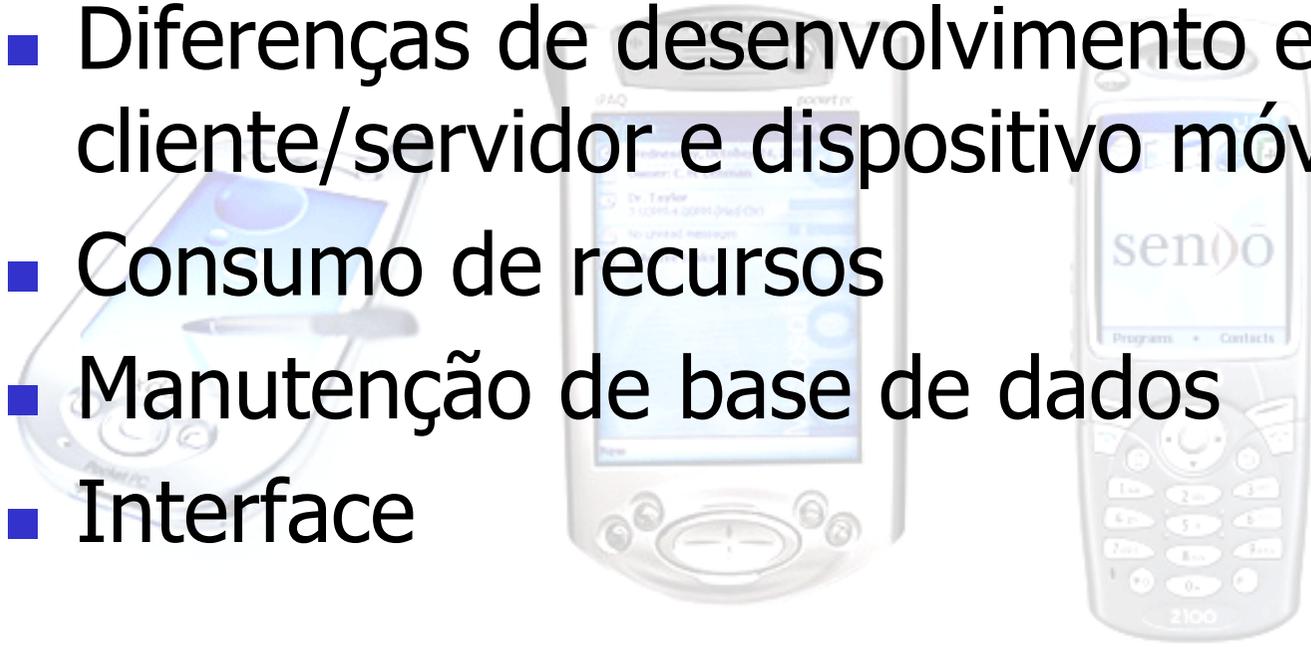
Desenvolvimento Operacionalidade

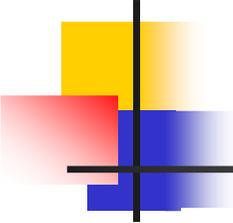
- Visualização de detalhe possibilidade de modificar dados.



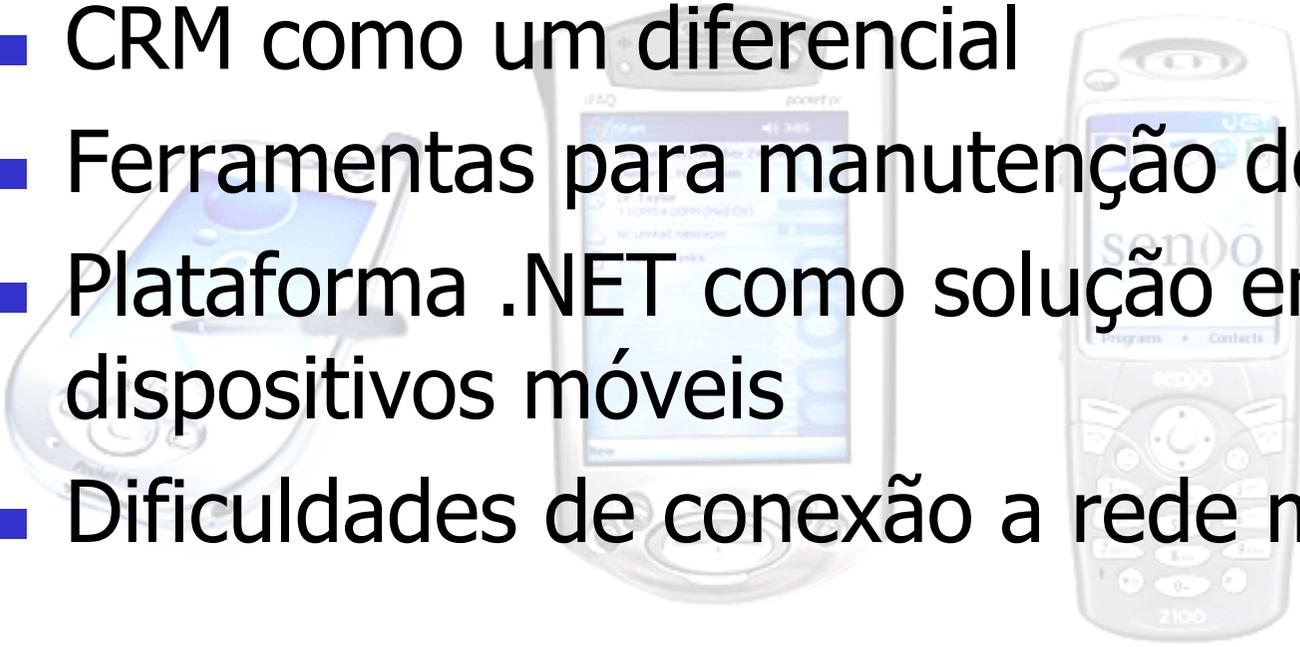


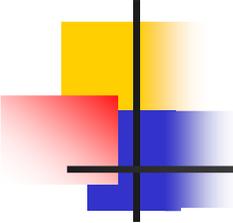
Resultados e Discussões

- Diferenças de desenvolvimento entre cliente/servidor e dispositivo móvel
 - Consumo de recursos
 - Manutenção de base de dados
 - Interface
- 

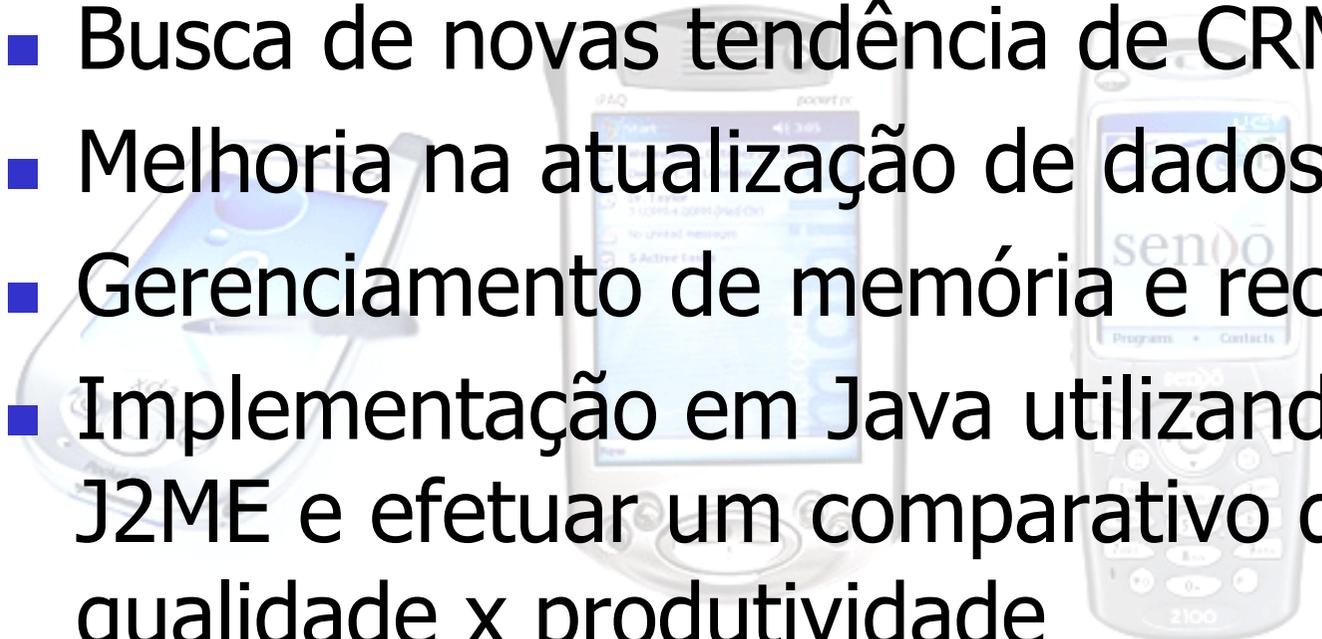


Conclusões

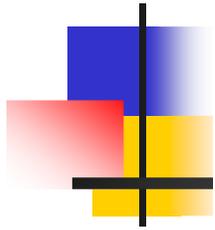
- CRM como um diferencial
 - Ferramentas para manutenção do CRM
 - Plataforma .NET como solução em dispositivos móveis
 - Dificuldades de conexão a rede mundial
- 



Extensões

- Busca de novas tendências de CRM
 - Melhoria na atualização de dados
 - Gerenciamento de memória e recursos
 - Implementação em Java utilizando J2ME e efetuar um comparativo de qualidade x produtividade
- 

Perguntas?



Apresentação do Protótipo