

## 6 Implementação do iPH

O **iPH** foi desenvolvido utilizando a linguagem Visual C# da plataforma .NET<sup>4</sup> e possui duas versões: uma versão para dispositivos que executem o Windows XP e o .NET Framework, chamada **iPH - XP**; e outra para ser executada em dispositivos computacionais mais limitados como *palmtops* e *smartphones*, executando o Windows Mobile e .NET Compact Framework (.NET CF). Esta última versão é chamada **iPH - Mobile**. A Tabela 3 exibe os requisitos de execução das versões do iPH.

**Tabela 3 - Versões do Interactive Presenter for Handhelds - iPH**

Versão	Dispositivos Computacionais	Sistema Operacional <sup>5</sup>	Plataforma .NET
iPH - XP	<i>Desktops, notebooks e tablet pcs</i>	Windows XP	.NET Framework 2.0
iPH - Mobile	<i>Palmtops e smartphones</i>	Windows Mobile	.NET Compact Framework 2.0

Qualquer uma das versões pode ser executada localmente, sem necessidade de interface de rede e/ou conexão em uma sessão colaborativa. Funcionalidades como a visualização de um *deck* e a edição dos quadros dos mesmos estão disponíveis. Ao utilizar o iPH de maneira distribuída, conectado em uma sessão de colaboração, pode-se constatar suas demais funcionalidades como: o envio e recebimento de quadros e contribuições entre os participantes; a submissão de

---

<sup>4</sup> Para o desenvolvimento do iPH - XP e do iPH - Mobile foram utilizadas as versões 2.0 da .NET Framework e .NET Compact Framework.

<sup>5</sup> Apesar de não terem sido feitos testes utilizando outros sistemas operacionais, como Windows 2000 e Windows Vista para o iPH - XP e Windows CE 4 para iPH - Mobile, acredita-se que estas aplicações podem ser executadas normalmente, bastando que esteja instalado a plataforma .NET correspondente.

contribuições; o controle de sincronia entre os diferentes participantes; o gerenciamento de regras de contexto; entre outras.

Apesar das versões do iPH serem executadas em plataformas diferentes, elas compartilham praticamente o mesmo código-fonte, variando apenas em detalhes de interface com o usuário e acesso a arquivos de configuração. Isto é possível, pois, como mencionado, praticamente todo código desenvolvido para o .NET CF é compatível com o .NET Framework. Estas funcionalidades do iPH foram divididas em diversos pacotes que são utilizados por ambas as versões. Apesar de esta decisão ter aumentado a complexidade do código, visto que existem funcionalidades disponíveis apenas para o .NET Framework, ela agilizou o desenvolvimento do aplicativo, pois não houve necessidade de re-trabalho para desenvolver uma mesma funcionalidade para diferentes plataformas. Isto também facilitou o gerenciamento do desenvolvimento e a aplicação de testes.

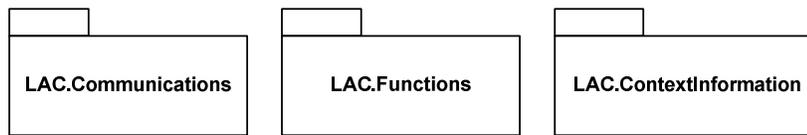
O desenvolvimento do iPH foi realizado em três etapas:

1. Desenvolvimento de funcionalidades básicas necessárias ao iPH e que pudessem ser re-aproveitadas por qualquer outro projeto .NET;
2. Desenvolvimento de funcionalidades específicas ao iPH e comum a ambas as plataformas .NET;
3. Desenvolvimento da interface do usuário e funcionalidades específicas de cada versão do iPH.

As funcionalidades mencionadas acima foram divididas em pacotes que foram implementados através de bibliotecas compatíveis com o .NET CF.

### **6.1.1. Funcionalidades Básicas - Pacote LAC**

Durante o desenvolvimento do iPH, foram identificadas várias funcionalidades básicas que poderiam ser reaproveitadas em quaisquer outros projetos. Foi criado o pacote **LAC**, que possui funcionalidades de comunicação, acesso a informações de contexto e funções gerais como: redimensionamento de imagem, formulários de entrada de dados e outras. O pacote LAC está dividido em três pacotes, como exibido na Figura 38.



**Figura 38 - O pacote LAC**

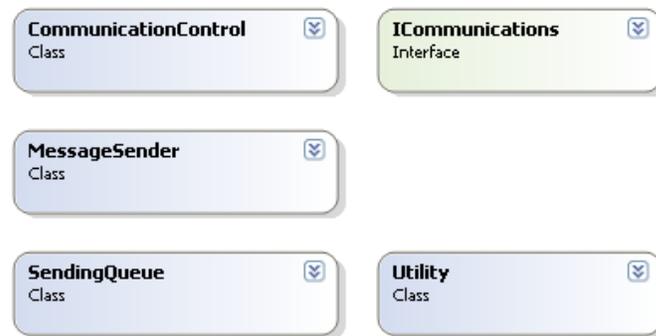
Esses três pacotes são utilizados e distribuídos de maneira independente, sendo que cada um deles foi implementado como uma biblioteca do .NET CF. A Tabela 4 resume os pacotes que são detalhados a seguir.

**Tabela 4 - Descrição do pacote LAC**

Pacote	Descrição
<b>LAC.Communications</b>	Contém classes que fazem a comunicação entre aplicações distribuídas através do CompactConferenceXP. É através destas classes que são enviadas e recebidas as diversas mensagens durante uma sessão de colaboração do iPH.
<b>LAC.ContextInformation</b>	O pacote contém classes que encapsulam chamadas ao MoCA/WS, facilitando o acesso e o uso de informações de contexto.
<b>LAC.Functions</b>	Controles e classes que não são específicos da aplicação foram desenvolvidos neste pacote. Funcionalidades como formulários de entrada de dados, funções de redimensionamento de imagens, controles visuais inexistentes no .NET CF, entre outras.

#### 6.1.1.1. LAC.Communications

O pacote LAC.Communications contém classes que fazem a comunicação entre as aplicações durante uma sessão de colaboração, encapsulando todas as chamadas ao CompactConferenceXP. A Figura 39 exibe as classes e a interface que pertencem a este pacote.



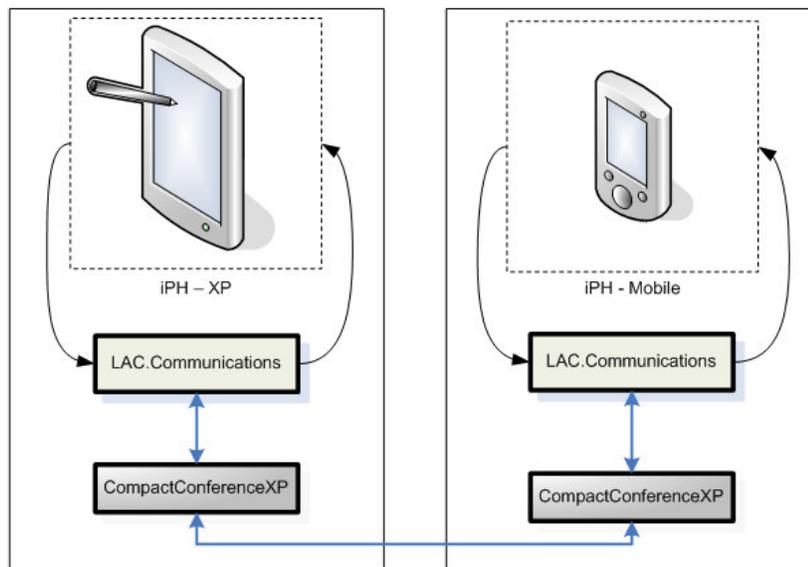
**Figura 39 - O pacote LAC.Communications**

Para que uma aplicação possa enviar e receber mensagens durante uma sessão colaborativa, basta que ela possua um atributo da classe *CommunicationControl*, e passe como argumento do método construtor deste, um objeto de uma classe que implemente a interface *ICommunications*. Esta interface possui um só método a ser implementado, chamado *ReceiveObject*, que é utilizado para processar as mensagens recebidas.

A classe *CommunicationControl* oferece métodos para conexão e desconexão, e possui um atributo da classe *MessageSender* utilizado para envio de mensagens pela rede através do método *SendObject*, que insere a mensagem em uma fila de mensagens a serem enviadas pertencentes à classe *SendingQueue*. Esta classe possui uma *thread* que retira as mensagens da fila e as envia através de chamadas a API do CompactConferenceXP. Quando a fila está vazia, esta *thread* permanece em modo de espera, até a próxima mensagem ser inserida na fila.

A classe *Utility* possui um método estático que serializa as mensagens utilizando o CompactFormatter antes de inserir na fila de envio. Quando a mensagem é recebida, esta é deserializada pelo objeto *CommunicationControl*.

A Figura 40 exibe a comunicação entre diferentes versões do iPH. É válido lembrar que este pacote é independente, e logo, qualquer aplicação pode fazer uso do mesmo para comunicação através do CompactConferenceXP.

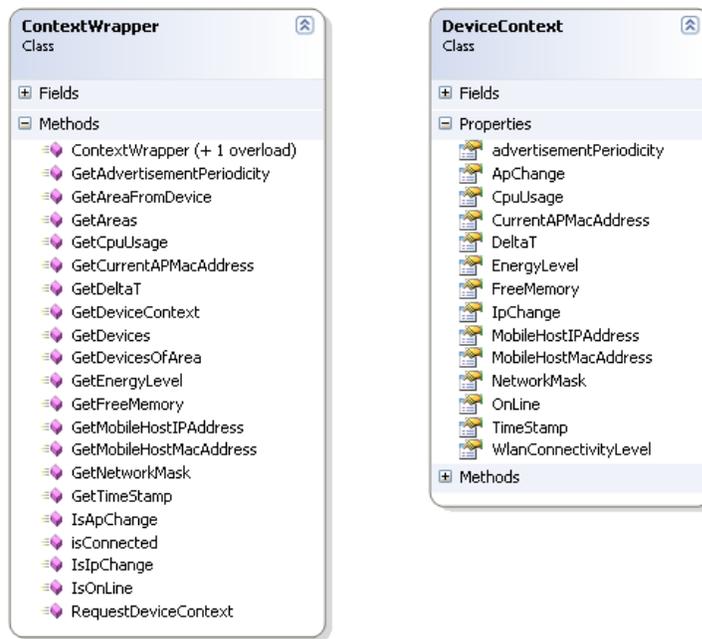


**Figura 40 - Comunicação através do LAC.Communications**

O pacote LAC.Communications foi baseado na implementação da comunicação do aplicativo ClassroomPresenter, que possui uma maneira semelhante de enviar e receber mensagens utilizando o ConferenceXP.

#### **6.1.1.2. LAC.ContextInformation**

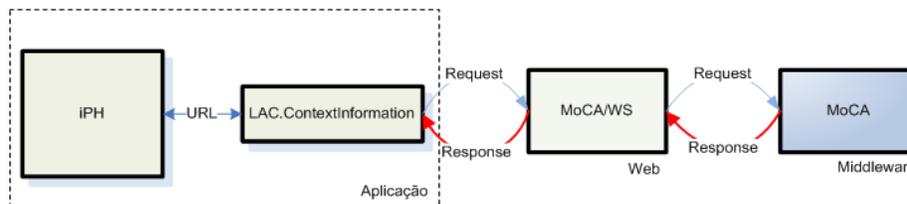
Este pacote foi criado com o objetivo de facilitar o acesso a informações de contexto por qualquer aplicação .NET. Além da própria referência ao MoCA/WS (vide seção 4.2.2), existem duas classes neste pacote: a classe *ContextWrapper*, que contém os métodos para o acesso a informações de contexto; e a classe *DeviceContext*, que encapsula informações de contexto de um determinado dispositivo. A Figura 41 exibe as classes mencionadas e os métodos utilizados para acessar as informações de contexto fornecidas pelo MoCA.



**Figura 41 - As classes do pacote LAC.ContextInformation**

Para fazer uso do LAC.ContextInformation, uma aplicação deve possuir um atributo do tipo *ContextWrapper*, e passar como argumento do seu método construtor o endereço do MoCA/WS. Com isto, a aplicação pode fazer requisições a respeito de informações de contexto como: energia disponível; qualidade do enlace; localização simbólica de um dispositivo; dispositivos presentes em uma localização simbólica; e outras.

A Figura 42 exibe o acesso a informações de contexto através do LAC.ContextInformation, que encapsula a referência ao MoCA/WS e suas chamadas.



**Figura 42 – Utilização do LAC.Communications**

### 6.1.1.3. LAC.Functions

O pacote LAC.Functions é bastante simples e objetiva oferecer funcionalidades básicas que não possuem uma categoria específica como comunicação ou informação de contexto. Este pacote é subdividido nos pacotes: *Drawing*, *Forms*, *Networks* e *Objects*.

O sub-pacote *Drawing* contém: uma classe chamada *DrawingFunctions*, que oferece métodos estáticos relacionados ao tratamento de imagens; o método *CreateImageFromImage*, que retorna uma imagem redimensionada a partir de outra imagem, passando como argumentos o tamanho da nova imagem e o tamanho da imagem original na nova imagem, utilizado pelo iPH para o desenho dos quadros de uma apresentação; e o método *GetRelativeSize*, que retorna a altura e largura relativas a uma relação original passada como parâmetro. Este último é utilizado no desenho dos quadros de uma apresentação para desenhar os e mesmos sempre em uma mesma proporção.

Dentro do pacote *Forms* encontram-se três tipos de formulários: *AboutBox*, utilizado para prover informações sobre a aplicação que está sendo executada; *InputBox*, que é utilizado para requisitar ao usuário a entrada de texto; e *InputTrackBarForm*, utilizado para requisitar ao usuário que escolha um valor inteiro dentro de uma margem pré-estabelecida. Somente os dois primeiros formulários foram utilizados para o iPH, onde o *AboutBox* é o formulário utilizado para fornecer as informações da aplicação, e o *InputBox* é utilizado durante vários momentos como a configuração do nome de um quadro adicionado e a adição de um texto em um quadro. A Figura 43 exibe o formulário *InputBox* em ambas as plataformas .NET.

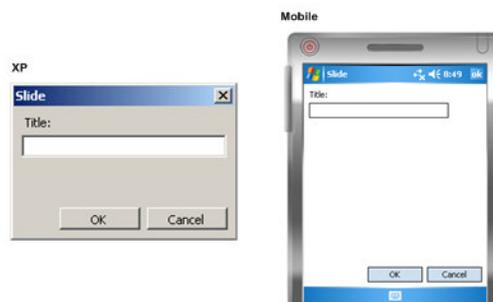


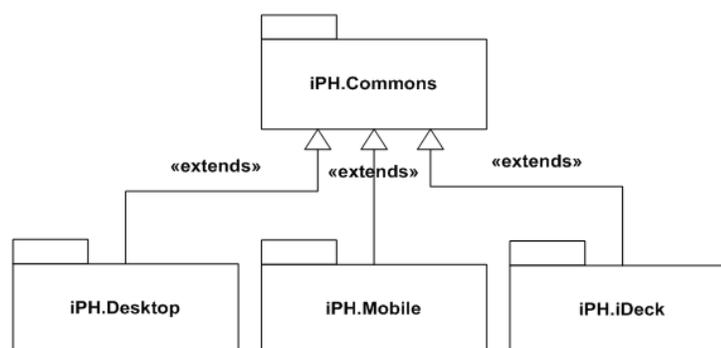
Figura 43 - *InputBox* sendo visualizado em ambas as plataformas

O pacote *Networks* contém uma única classe chamada *PhysicalAddress*, que possui um método estático que retorna o endereço MAC do dispositivo computacional que está executando a aplicação. Este endereço é necessário para fazer as chamadas ao MoCA/WS, e assim recuperar as informações de contexto do dispositivo utilizado. Por último, o pacote *Controls* contém uma única classe chamada *ImageButton*, que representa um botão associado a uma imagem e que foi criada para a versão iPH – Mobile.

O objetivo do pacote LAC.Functions é oferecer funcionalidades comuns a aplicações desenvolvidas para plataforma .NET, e à medida que este pacote seja utilizado por outros projetos, outras funcionalidades serão necessárias e inseridas no pacote, tornando-o cada vez mais útil.

### 6.1.2. Funcionalidades Específicas – Pacote iPH

O pacote **iPH** contém as classes e interfaces específicas ao aplicativo desenvolvido, como as classes que representam uma apresentação e seus quadros, as classes das mensagens transmitidas, as classes de formulários de interface com o usuário, entre outras, além de fazer uso das funcionalidades oferecidas pelo pacote LAC. Este pacote está dividido em três pacotes que se relacionam como na Figura 44.



**Figura 44 - O pacote iPH**

O pacote **iPH.Commons** contém todas as classes que são utilizadas por todas as versões, enquanto que os pacotes **iPH.Desktop** e **iPH.Mobile** possuem classes específicas a versão do iPH utilizada. O pacote **iPH.iDeck** contém as classes relacionadas ao aplicativo **iDeck**, utilizado para criação de arquivos de

*decks*, e é descrito no anexo 10.1. A Tabela 5 contém um breve resumo desses pacotes, que são detalhados a seguir.

**Tabela 5 - Descrição do pacote iPH**

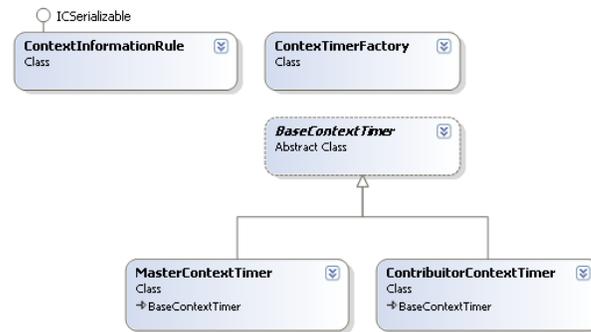
<b>Pacote</b>	<b>Descrição</b>	<b>Plataforma</b>
<b>iPH.Commons</b>	Contém diversas classes e interfaces necessárias para o desenvolvimento do iPH como: classes que representam os quadros de uma apresentação, classes de mensagens; formulários utilizados em ambas as versões, classes de controles visuais específicos, gerenciadores de participantes, regras de contexto, entre outras.	.NET CF
<b>iPH.Desktop</b>	Contém as classes que implementam a interface visual do iPH – XP, e algumas funcionalidades específicas para a plataforma .NET Framework.	.NET Framework
<b>iPH.Mobile</b>	Contém as classes que implementam a interface visual do iPH – Mobile, e algumas funcionalidades específicas para a plataforma .NET CF.	.NET CF
<b>iPH.iDeck</b>	Contém as classes pertencentes ao aplicativo iDeck, utilizado para criação de <i>decks</i> . Este aplicativo é descrito no Anexo 10.1.	.NET Framework

#### **6.1.2.1. iPH.Commons**

O pacote iPH.Commons é o núcleo do iPH, pois nele estão contidas as classes e interfaces que são utilizadas por ambas as versões da aplicação. Para executar funcionalidades específicas, essas classes disponibilizam métodos virtuais e abstratos que são implementados por cada uma das versões. Este pacote está subdividido em diversos outros pacotes que relacionam diversas áreas do iPH.

O pacote **Configuration** contém a classe *InteractiveConfiguration*, que define a configuração da aplicação com atributos como: o endereço do MoCA/WS utilizado; as cores para quadros públicos e particulares; a utilização de regras de contexto para adaptação; e o período em que estas regras são aplicadas. Esta classe possui valores padrão, porém seus atributos podem ser configurados por arquivos de configuração ou durante a execução da aplicação.

O pacote **Context** contém a classe que representa uma regra de contexto, chamada *ContextInformationRule*, e as classes que aplicam estas regras durante uma apresentação, dependendo do papel que está sendo exercido pelo participante. Para isto foi utilizado o padrão *Factory*<sup>6</sup>. A Figura 45 exibe as classes deste sub-pacote.



**Figura 45 - As classes do sub-pacote iPH.Commons.Context**

Em **Forms** estão contidas todas as classes de formulários utilizados por ambas as versões do iPH, que são: formulário de configuração, utilizado para configurar a aplicação; formulário de sessão, utilizado para ingressar em uma sessão de colaboração; formulário de informações de contexto do participante; formulário de listagem e adição de regras de contexto; formulário de listagem de participantes de uma sessão colaborativa; e o formulário principal da aplicação iPH, o *InteractivePresentationForm*. Esta última classe contém as principais funcionalidades do iPH, é abstrata e deve ser implementada pelas versões específicas de cada plataforma.

O pacote **Functions** contém as classes que representam as funções disponíveis no iPH, e são utilizadas para definir o que cada tipo de participante poderá executar e visualizar na aplicação. Essas classes também são utilizadas na definição de regras de contexto, pois uma regra é definida por uma condição e a função que desabilitará se esta condição for verdadeira.

---

<sup>6</sup> *Factory* é um padrão de projeto que permite a criação de objetos ou famílias de objetos relacionados ou dependentes, através de uma única interface e sem que a classe concreta seja especificada [Gamma, 1995].

O pacote **Ink** possui uma única classe contendo um único método estático utilizado para definir a largura da *ink*. Dependendo do sistema operacional que está sendo utilizado, Windows Mobile ou Windows XP, a largura das linhas desenhadas difere, visto que para um *handheld* esta largura deve ser menor que a largura de uma linha desenhada em um *notebook*.

No pacote **Manager** estão todos os gerenciadores utilizados pela aplicação. Esses gerenciadores controlam as regras de contexto existentes, a lista de participantes conectados em uma sessão de colaboração, as contribuições de cada quadro de uma apresentação colaborativa, entre outros.

As classes de mensagens que são transmitidas entre as aplicações do iPH estão contidas no pacote **Messages**. A classe base é chamada de *BaseMessage* e possui como atributos a chave da sessão de colaboração, um *timestamp* da criação desta mensagem, e o participante remetente. As outras classes derivam desta classe base como a classe *DeckMessage*, que representa um *deck*, e a classe *SlideAddMessage*, que contém um quadro a ser inserido pelos participantes que recebem esta mensagem. Todas as classes deste pacote implementam os métodos definidos pela classe *ISerializable* do CompactFormatter para serialização e deserialização. A Figura 46 exibe as classes do pacote Messages:

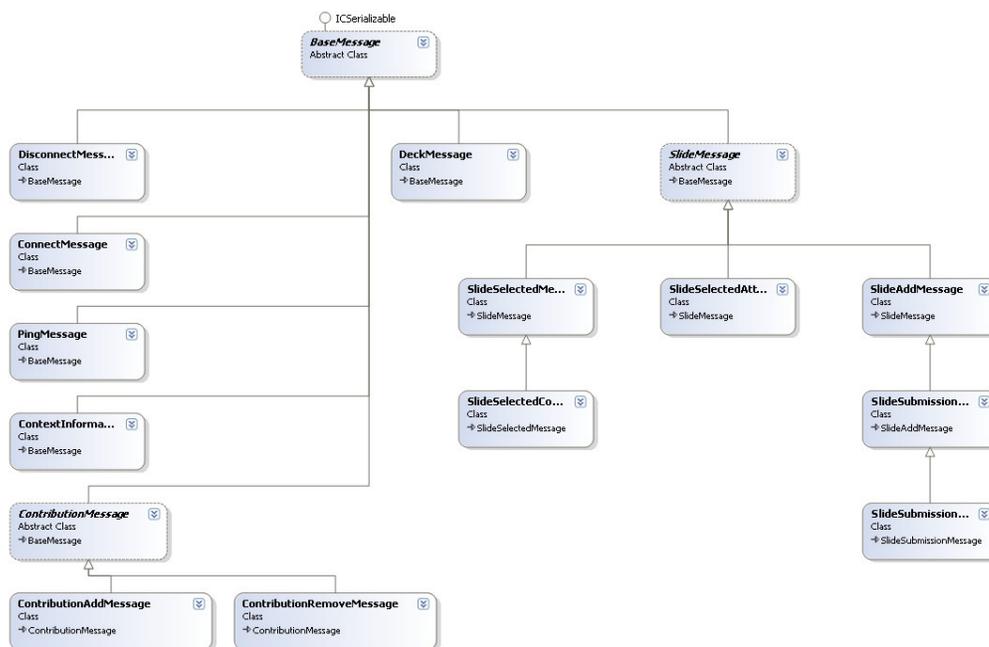
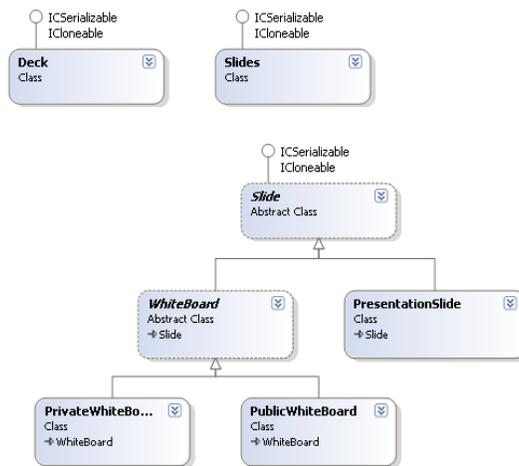


Figura 46 - As classes do sub-pacote iPH.Commons.Messages

O pacote **Parameters** contém uma classe que disponibiliza os parâmetros utilizados durante uma sessão colaborativa, como a definição de que um participante pode ou não enviar contribuições, o sincronismo entre os diferentes participantes, entre outros.

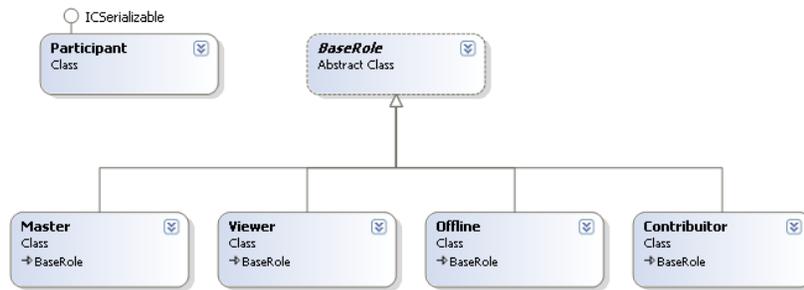
As classes relacionadas aos quadros de uma apresentação, como um *deck*, um quadro público ou privado, e os controles utilizados para visualizar objetos dessas classes estão contidos no pacote **Presentation**. Essas classes definem todo o modelo da apresentação colaborativa utilizada, e são exibidas na Figura 47.



**Figura 47 - As classes de uma apresentação colaborativa**

O pacote **Session** contém classes relacionadas ao controle da sessão de colaboração, como a classe que representa a sessão de colaboração utilizada, e a classe relacionada a chave da sessão de colaboração.

O último pacote, chamado **User**, contém informações a respeito dos participantes e seus papéis em uma sessão de colaboração. As classes que representam seus papéis contêm métodos que definem quais mensagens cada tipo de participante pode enviar ou receber. Essas classes são exibidas na Figura 48.



**Figura 48 - Classes do pacote iPH.Commons.User**

### 6.1.2.2. iPH.Desktop

Neste pacote estão contidas todas as classes específicas para a versão iPH – XP. A classe do formulário principal, chamado *MainForm*, estende o formulário *InteractivePresentationForm* e implementa seus métodos virtuais e abstratos. Basicamente, esta classe fornece apenas a interface com o usuário e o acesso ao arquivo de configuração da aplicação, e a partir das interações do usuário com esta interface, os métodos da classe pai são chamados. É também estendido o formulário de configuração da aplicação, pois para configurar as cores dos quadros públicos, são utilizados diálogos específicos para a plataforma utilizada.

### 6.1.2.3. iPH.Mobile

Este pacote é o correspondente do pacote iPH.Desktop para a versão iPH – Mobile. A classe *MobileMainForm* estende a classe *InteractivePresentationForm*, assim como o formulário de configuração, visto que os diálogos para a configuração de cor são específicos para a plataforma utilizada. Além destas duas classes, existem outras duas classes de formulários que são utilizadas para a formatação da fonte do texto a ser inserido em um quadro, e visualização das informações de um quadro de uma apresentação colaborativa.