



República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) BR 112014023828-6 B1



(22) Data do Depósito: 12/03/2013

(45) Data de Concessão: 12/07/2022

(54) Título: MÉTODO E SISTEMA PARA DINAMICAMENTE INSTALAR PLUGINS EM APLICATIVO DISTRIBUÍDO EM SISTEMAS REMOTOS

(51) Int.Cl.: G06F 9/44.

(30) Prioridade Unionista: 27/03/2012 US 13/431,131.

(73) Titular(es): MICROSOFT TECHNOLOGY LICENSING, LLC.

(72) Inventor(es): PHILLIP M. HOFF; ANTHONY LAVEL CRIDER; SUNG HON WU; LUBOMIR I. BIROV; JOHN A. CUNNINGHAM.

(86) Pedido PCT: PCT US2013030307 de 12/03/2013

(87) Publicação PCT: WO 2013/148144 de 03/10/2013

(85) Data do Início da Fase Nacional: 25/09/2014

(57) Resumo: MÉTODO E SISTEMA PARA DINAMICAMENTE INSTALAR PLUGINS EM APLICATIVO. São descritas técnicas neste documento que são capazes de dinamicamente instalar programa(s) de recursos adicionais ("plugins") para aplicativo(s). Um programa de recursos adicionais agente é levado a executar em uma implementação de um aplicativo especificado (por exemplo, através de várias máquinas em um ambiente em nuvem ou "nos locais"). O aplicativo especificado é empacotado para incluir o programa de recursos adicionais agente. O programa de recursos adicionais agente é utilizado para instalar programa(s) de recursos adicionais designado(s) dinamicamente baseado na informação de configuração com respeito ao aplicativo especificado. A informação de configuração indica que o(s) programa(s) de recursos especiais designados são para ser instalados em resposta ao aplicativo especificando sendo implementado.

Relatório Descritivo da Patente de Invenção para
**"MÉTODO E SISTEMA PARA DINAMICAMENTE INSTALAR
PLUGINS EM APLICATIVO DISTRIBUÍDO EM SISTEMAS
REMOTOS".**

ANTECEDENTES

[001] Um programa de recursos adicionais ("plugin") é um componente de software que adiciona uma ou mais capacidades designadas para um aplicativo de software. O programa de recursos adicionais pode ser utilizado para permitir a programadores terceirizados criarem habilidades que estendem o aplicativo, para suportar facilmente a adição de novos aspectos (por exemplo, definição de perfil, diagnósticos etc.) para o aplicativo, para reduzir o tamanho do aplicativo etc.. Programadores tradicionalmente incluem programas de recursos adicionais em um aplicativo antes da implementação do aplicativo. Uma limitação com respeito às técnicas convencionais para instalar programas de recursos adicionais é que o proprietário do aplicativo, o qual não é necessariamente o programador do aplicativo, tipicamente não está apto a adicionar, remover, e/ou atualizar o programa de recursos adicionais baseado na necessidade atual sem reconstruir e reimplementar o aplicativo. Reconstruir e reimplementar o aplicativo pode ser um processo demorado e/ou oneroso. Além disso, tais técnicas convencionais frequentemente proporcionam várias cópias do(s) mesmo(s) programa(s) de recursos adicionais empacotadas dentro de vários aplicativos diferentes, os quais podem impedir implementação difundida de versões atualizadas do(s) programa(s) de recursos adicionais. Por exemplo, se uma nova versão de um programa de recursos adicionais soluciona uma vulnerabilidade de segurança, cada aplicativo que foi empacotado com o respectivo programa de recursos adicionais tradicionalmente é novamente empacotado e

reimplementado; este novo empacotamento e reimplementação pode ocorrer relativamente lentamente. Por consequência, as técnicas convencionais podem resultar em aumento de tamanho do pacote, de custos de armazenamento, e/ou de custos de transmissão.

[002] Em algumas técnicas convencionais, programadores criaram suas próprias tarefas de inicialização que utilizam mecanismos de instalação existentes (por exemplo, Instalador de Plataforma Internet) para instalar pré-requisitos (isto é, dependências) do aplicativo. Um pré-requisito de um aplicativo é um componente de software que não é incluído no aplicativo e que o aplicativo necessita de modo a funcionar de modo apropriado. Por exemplo, se o aplicativo é baseado em JAVA®, o tempo de execução de JAVA® pode ser um pré-requisito do aplicativo. Em outro exemplo, se o aplicativo é um aplicativo PHP™, o tempo de execução de PHP™ pode ser um pré-requisito do aplicativo. Cada tarefa de inicialização frequentemente utiliza um script para chamar um utilitário de linha de comando para instalar um produto particular necessário pelo aplicativo. Enquanto utilizar scripts desta maneira pode não ser excessivamente incômodo para aplicativos individuais, os scripts tipicamente são escritos a mão e testados por cada programador, para cada aplicativo. Nenhum mecanismo padrão, geral e robusto é conhecido para instalar tais componentes. Adicionalmente, se o aplicativo requer um conjunto diferente de pré-requisitos (por exemplo, uma nova versão de um pré-requisito), pode ser necessário reconstruir e reimplementar o aplicativo de modo a instalar e executar o conjunto diferente de pré-requisitos. Novamente, nenhum mecanismo padrão, geral e robusto existe para instalar tais componentes após a implementação.

SUMÁRIO

[003] Várias abordagens são descritas neste documento para, dentre outras coisas, dinamicamente instalar programa(s) de recursos

adicionais para aplicativo(s). Um método é descrito, no qual a informação de configuração com respeito a um aplicativo especificado é identificada. A informação de configuração indica que o(s) programa(s) de recursos adicionais designado(s) selecionado(s) a partir de um ou mais programas de recursos adicionais, os quais são armazenados em uma localização que é acessível para aplicativos implementados, são para serem instalados em resposta ao aplicativo especificado sendo implementado. Um programa de recursos adicionais agente é levado a executar em uma implementação do aplicativo especificado. Por exemplo, a implementação pode ser executado através de várias máquinas em um ambiente em nuvem. Em outro exemplo, a implementação pode ser executada "nos locais" (isto é, em uma ou mais máquinas locais). O aplicativo especificado é empacotado para incluir o programa de recursos adicionais agente. O programa de recursos adicionais agente é utilizado para instalar o(s) programa(s) de recursos adicionais designados de modo dinâmico baseado na informação de configuração. Por consequência, tal(tais) programa(s) de recursos adicionais é(são) de modo intercambiável referido(s) neste documento como "programa(s) de recursos adicionais dinâmico(s)".

[004] É descrito um sistema que inclui um módulo de identificação, um módulo de causação, e um módulo de instalação. O módulo de instalação é configurado para identificar informação de configuração com respeito a um aplicativo especificado. A informação de configuração indica que o(s) programa(s) de recursos adicionais designado(s) selecionados a partir de um ou mais programas de recursos adicionais, os quais são armazenados em uma localização que é acessível aos aplicativos implementados, são para serem instalados em resposta ao aplicativo especificado sendo implementado. O módulo de causação é configurado para fazer com

que um programa de recursos adicionais agente execute em uma implementação do aplicativo especificado (por exemplo, através de várias máquinas em um ambiente em nuvem ou "nos locais"). O aplicativo especificado é empacotado para incluir o programa de recursos adicionais agente. O módulo de instalação é configurado para utilizar o programa de recursos adicionais para instalar o(s) programa(s) de recursos adicionais designado(s) de modo dinâmico baseado na informação de configuração.

[005] É descrito um produto de programa de computador que inclui um meio legível por computador possuindo lógica de programa de computador gravada no mesmo para permitir que um sistema baseado em processador instale de modo dinâmico programa(s) de recursos adicionais designado(s) com respeito a um aplicativo especificado. O produto de programa de computador inclui primeiro, segundo e terceiro módulos de lógica de programa. O primeiro módulo de lógica de programa é para permitir ao sistema baseado em processador identificar a informação de configuração com respeito ao aplicativo especificado. A informação de configuração indica que o(s) programa(s) de recursos adicionais designado(s) selecionado(s) a partir de um ou mais programas de recursos adicionais, os quais são armazenados em uma localização que é acessível para aplicativos implementados, são para serem instalados em resposta ao aplicativo especificado sendo implementado. O segundo módulo de lógica de programa é para permitir ao sistema baseado em processador fazer com que um programa de recursos adicionais agente execute em uma implementação do aplicativo especificado (por exemplo, através de várias máquinas em um ambiente em nuvem ou "nos locais"). O aplicativo especificado é empacotado para incluir o programa de recursos adicionais agente. O terceiro módulo de lógica de programa é para permitir ao sistema baseado em processador utilizar o programa

de recursos adicionais agente para instalar o(s) programa(s) de recursos adicionais designado(s) de modo dinâmico baseado na informação de configuração.

[006] Este sumário é proporcionado para introduzir uma seleção de conceitos de uma forma simplificada que, os quais são adicionalmente descritos abaixo na Descrição Detalhada. Este sumário não é pretendido para identificar aspectos chave ou aspectos essenciais do assunto reivindicado, nem é pretendido para ser utilizado para limitar o escopo do assunto reivindicado. Além disso, é observado que a invenção não é limitada às concretizações específicas descritas na Descrição Detalhada e/ou em outras seções deste documento. Tais concretizações são apresentadas neste documento apenas para propósitos de ilustração. Concretizações adicionais serão aparentes aos versados na técnica relevante baseadas nos ensinamentos contidos neste documento.

BREVE DESCRIÇÃO DOS DESENHOS/FIGURAS

[007] Os desenhos acompanhantes, os quais são incorporados neste documento e formam parte do relatório descritivo, ilustram concretizações da presente invenção e, juntos com a descrição, adicionalmente servem para explicar os princípios envolvidos e para possibilitar aos versados na técnica relevante realizar e utilizar as tecnologias reveladas.

[008] A FIG. 1 é um diagrama de blocos de um sistema de computador ilustrativo de acordo com uma concretização.

[009] A FIG. 2 é um diagrama de blocos de uma implementação ilustrativa de um módulo de instalação de programa de recursos adicionais dinâmico apresentado na FIG. 1 que ilustra submissão de um programa de recursos adicionais para utilização por um programador em um aplicativo de acordo com uma concretização.

[0010] A FIG. 3 é um diagrama de blocos de uma implementação

ilustrativa de um módulo de instalação de programa de recursos adicionais dinâmico apresentado na FIG. 1 que ilustra seleção de dependências de um aplicativo por um programador de acordo com uma concretização.

[0011] A FIG. 4 é um diagrama de blocos de uma implementação ilustrativa de um módulo de instalação de programa de recursos adicionais dinâmico apresentado na FIG. 1 que ilustra seleção dos programas de recursos adicionais para um aplicativo por um programador de acordo com uma concretização.

[0012] A FIG. 5 é um diagrama de blocos de uma implementação ilustrativa de um módulo de instalação de programa de recursos adicionais dinâmico apresentado na FIG. 1 que ilustra instalação e execução das dependências e dos programas de recursos adicionais que são selecionados como apresentado nas respectivas FIGS. 3 e 4 de acordo com uma concretização.

[0013] As FIGS. 6 até 9 descrevem fluxogramas de métodos ilustrativos para dinamicamente instalar programa(s) de recursos adicionais para aplicativo(s) de acordo com concretizações.

[0014] A FIG. 10 é um diagrama de blocos de outra implementação ilustrativa de um módulo de instalação de programa de recursos adicionais dinâmico apresentado na FIG. 1 de acordo com uma concretização.

[0015] A FIG. 11 descreve um computador ilustrativo no qual concretizações podem ser implementadas.

[0016] Os aspectos e vantagens das tecnologias reveladas irão se tornar mais aparentes a partir da descrição detalhada exposta abaixo quando considerada em conjunto com os desenhos, nos quais caracteres de referência iguais identificam elementos correspondentes em todas as partes. Nos desenhos, números de referência iguais geralmente indicam elementos idênticos, de função similar e/ou

estruturalmente similares. O desenho no qual um elemento aparece primeiro é indicado pelo(s) dígito(s) mais a esquerda no número de referência correspondente.

DESCRIÇÃO DETALHADA

I. Introdução

[0017] A descrição detalhada seguinte se refere aos desenhos acompanhantes que ilustram as concretizações ilustrativas da presente invenção. Entretanto, o escopo da presente invenção não é limitado a estas concretizações, mas, ao invés disso, definido pelas reivindicações anexas. Assim, concretizações além destas apresentadas nos desenhos acompanhantes, tais como versões modificadas das concretizações ilustradas, podem, no entanto, serem abrangidas pela presente invenção.

[0018] Referências no relatório descritivo a "uma concretização", "a concretização", "uma concretização ilustrativa", ou coisa parecida, indicam que a concretização descrita pode incluir um aspecto, estrutura ou característica particular, mas cada concretização pode não necessariamente incluir o aspecto, estrutura ou característica particular. Além disso, tais frases não estão necessariamente referindo-se à mesma concretização. Adicionalmente, quando um aspecto, estrutura ou característica particular é descrito em conexão com uma concretização, é sugerido que ele esteja dentro do conhecimento do(s) versado(s) na técnica relevante para implementar tal aspecto, estrutura ou característica em conexão com outras concretizações descritas explicitamente ou não.

II. Concretizações Ilustrativas

[0019] As concretizações ilustrativas descritas neste documento são capazes de dinamicamente instalar programa(s) de recursos adicionais para aplicativo(s). De acordo com as concretizações ilustrativas, um programa de recursos adicionais agente é levado a

executar em uma implementação de um aplicativo especificado. Por exemplo, a implementação pode ser executada através de várias máquinas em um ambiente em nuvem. Em outro exemplo, a implementação pode ser executada "nos locais" (isto é, em uma ou mais máquinas locais). O aplicativo especificado é empacotado para incluir o programa de recursos adicionais agente. O programa de recursos adicionais agente é utilizado para instalar o(s) programa(s) de recursos adicionais designado(s) de modo dinâmico baseado na informação de configuração com respeito ao aplicativo especificado. Por consequência, tal(tais) programa(s) de recursos adicionais designado(s) é(são) de modo intercambiável referido(s) neste documento "programa(s) de recursos adicionais dinâmico(s)". A informação de configuração indica que o(s) programa(s) de recursos adicionais é(são) para serem instalados em resposta ao aplicativo especificado sendo implementado.

[0020] As técnicas ilustrativas descritas neste documento possuem uma variedade de benefícios à medida que comparadas às técnicas convencionais para instalação de programas de recursos adicionais. Por exemplo, as técnicas ilustrativas podem permitir a um usuário (por exemplo, um programador de aplicativo, um proprietário de aplicativo etc.) adicionar, remover, e/ou atualizar programas de recursos adicionais e/ou dependências de um aplicativo com respeito a uma implementação do aplicativo baseado na necessidade atual sem requerer uma reconstrução e/ou reimplementação do aplicativo. Por exemplo, as técnicas ilustrativas podem proporcionar um mecanismo padrão, geral e robusto para instalar os programas de recursos adicionais e/ou as dependências (por exemplo, após o aplicativo ter sido implementado). Por consequência, as técnicas ilustrativas podem consumir menos tempo e/ou menos recursos do que as técnicas convencionais. Por exemplo, utilizar as técnicas descritas neste

documento pode resultar em um tamanho de pacote menor, custos de armazenamento menores, e/ou menores custos de transmissão à medida que comparados às técnicas convencionais.

[0021] A FIG. 1 é um diagrama de blocos de um sistema de computador ilustrativo 100 de acordo com uma concretização. Em geral, o sistema de computador 100 opera para proporcionar informação para usuários (por exemplo, programadores de aplicativo) em resposta a solicitações (por exemplo, solicitações de protocolo de transferência de hipertexto (HTTP)) que são recebidas a partir dos usuários. A informação pode incluir documentos (por exemplo, páginas da Rede, imagens, arquivos de vídeo etc.), saída de executáveis, e/ou qualquer outro tipo adequado de informação. De acordo com concretizações ilustrativas descritas neste documento, o sistema de computador 100 de modo dinâmico instala programa(s) de recursos adicionais para aplicativo(s). Por exemplo, se um programador de aplicativo programa um aplicativo utilizando o sistema de computador 100, o sistema de computador 100 pode fazer com que o programa de recursos adicionais agente execute em uma implementação do aplicativo e então utilizar o programa de recursos adicionais agente para de modo dinâmico instalar o(s) programa(s) de recursos adicionais selecionados para o aplicativo. Detalhes com respeito às técnicas para de modo dinâmico instalar programa(s) de recursos adicionais para aplicativo(s) são proporcionados na discussão seguinte.

[0022] Como apresentado na FIG. 1, o sistema de computador 100 inclui vários sistemas do usuário 102A até 102M, uma rede 104, vários servidores 106A até 106N, e fonte(s) de programa de recursos adicionais 110. Comunicação entre os sistemas do usuário 102A até 102M, servidores 106A até 106N e fonte(s) de programa de recursos adicionais 110 é realizada através da rede 104 utilizando protocolos de

comunicação de rede bem conhecidos. A rede 104 pode ser uma rede de longa distância (por exemplo, a Internet), uma rede de área local (LAN), outro tipo de rede, ou uma combinação das mesmas.

[0023] A(s) fonte(s) de programa de recursos adicionais 110 são sistemas de processamento que são capazes de se comunicar com os servidores 106A até 106N. Um exemplo de um sistema de processamento é um sistema que inclui pelo menos um processador que é capaz de manipular dados de acordo com um conjunto de instruções. Por exemplo, um sistema de processamento pode ser um computador, um assistente pessoal digital etc. A(s) fonte(s) de programa de recursos adicionais é(são) configurada(s) para proporcionar programa(s) de recursos adicionais 112 para processamento adicional por um módulo de instalação de programa de recursos adicionais dinâmico 108, o qual é discutido em maiores detalhes abaixo. Por exemplo, fonte(s) de programa de recursos adicionais 110 podem proporcionar o(s) programa(s) de recursos adicionais 112 em resposta a receber solicitações a partir do(s) programa(s) de recursos adicionais 112 a partir do módulo de instalação de programa de recursos adicionais dinâmico 108.

[0024] Os sistemas do usuário 102A até 102M são sistemas de processamento que são capazes de se comunicar com os servidores 106A até 106N. Os sistemas do usuário 102A até 102M são configurados para proporcionar solicitações para os servidores 106A até 106N para solicitar informação armazenada nos (ou de outro modo acessível via) servidores 106A até 106N. Por exemplo, um usuário pode iniciar uma solicitação para executar um programa de computador utilizando um cliente (por exemplo, um navegador da Rede, um rastreador da Rede, ou outro tipo de cliente) implementado em um sistema do usuário 102 que é de propriedade ou de outro modo acessível pelo usuário. De acordo com algumas concretizações

ilustrativas, os sistemas do usuário 102A até 102M são capazes de acessar domínios (por exemplos, sites da Rede) hospedados pelos servidores 104A até 104N, de modo que os sistemas do usuário 102A até 102M possam acessar informação que é disponível via os domínios. Tal domínio pode incluir páginas da Rede, as quais podem ser proporcionadas como documentos de linguagem de marcação de hipertexto (HTML) e objetos (por exemplo, arquivos) que são ligados neste local, por exemplo.

[0025] Será reconhecido que qualquer um ou mais sistemas do usuário 102A até 102M podem se comunicar com qualquer um ou mais servidores 106A até 106N. Apesar dos sistemas do usuário 102A até 102M serem descritos como computadores de mesa na FIG. 1, os versados na técnica relevante irão apreciar que os sistemas do usuário 102A até 102M podem incluir qualquer sistema ou dispositivo adaptado para uso com o cliente, incluindo, mas não limitado a um computador de mesa, um computador laptop, um computador tablet, um assistente pessoal digital, um telefone celular, ou coisa parecida.

[0026] Os servidores 106A até 106N são sistemas de processamento que são capazes de se comunicar com os sistemas do usuário 102A até 102M e com fonte(s) de programa de recursos adicionais 110. Os servidores 106A até 106N são configurados para executar programas de computador que proporcionam informação para usuários em resposta a receber solicitações a partir dos usuários. Por exemplo, a informação pode incluir documentos (por exemplo, páginas da Rede, imagens, arquivos de vídeo etc.), saída de executáveis, ou qualquer outro tipo adequado de informação. De acordo com algumas concretizações ilustrativas, os servidores 106A até 106N são configurados para hospedar os respectivos sites da Rede, de modo que os sites da Rede sejam acessíveis aos usuários do sistema de computador 100.

[0027] O(s) primeiro(s) servidor(servidores) 106 é apresentado como incluindo o módulo de instalação de programa de recursos adicionais dinâmico 108. O módulo de instalação de programa de recursos adicionais 110 dinâmico 108 é configurado para encaminhar o(s) programa(s) de recursos adicionais 112, os quais são recebidos a partir da(s) fonte(s) de programa de recursos adicionais 110, para uma localização que é acessível para aplicativos implementados. Por exemplo, o módulo de instalação de programa de recursos adicionais dinâmico 108 pode receber o(s) programa(s) de recursos adicionais 112 em resposta a proporcionar solicitações para o(s) programa(s) de recursos adicionais 112 para a(s) fonte(s) de programa de recursos adicionais 110. O módulo de instalação de programa de recursos adicionais dinâmico 108 é adicionalmente configurado para identificar a informação de configuração com respeito a um aplicativo especificado. A informação de configuração indica que o(s) programa(s) de recursos adicionais designado(s) selecionado(s) a partir do(s) programa(s) de recursos adicionais 112, os quais são armazenados na localização que é acessível para aplicativos implementados, são para serem instalados em resposta ao aplicativo especificado sendo implementado. O módulo de instalação de programa de recursos adicionais dinâmico 108 é adicionalmente configurado para fazer com que o programa de recursos adicionais agente execute em uma implementação do aplicativo especificado (por exemplo, através de várias máquinas em um ambiente em nuvem ou "nos locais"). O aplicativo especificado é empacotado para incluir o programa de recursos adicionais agente. O módulo de instalação de programa de recursos adicionais dinâmico 108 é adicionalmente configurado para utilizar o programa de recursos adicionais agente para instalar o(s) programa(s) de recursos adicionais designado(s) baseado na informação de configuração. Algumas técnicas ilustrativas

para dinamicamente instalar programa(s) de recursos adicionais para aplicativo(s) são discutidas em maiores detalhes abaixo com referência às FIGS. 2 até 10.

[0028] Deve ser observado que uma instância de módulo de instalação de programa de recursos adicionais dinâmico 108 é incluída em cada servidor que compreende o aplicativo especificado. Cada instância de módulo de instalação de programa de recursos adicionais dinâmico 108 faz com que o programa de recursos adicionais agente instale e execute programa(s) de recursos adicionais dinâmico(s) na máquina na qual o módulo de instalação de programa de recursos adicionais dinâmico 108 está incluído.

[0029] O módulo de instalação de programa de recursos adicionais dinâmico 108 é apresentado como sendo incorporado no(s) primeiro(s) servidor(servidores) 106A para propósito de ilustração e não é pretendido para ser limitativo. Será reconhecido que a instância(s) do módulo de instalação de programa de recursos adicionais dinâmico 108 podem ser incorporada em qualquer um ou mais dos servidores 106A até 106N.

[0030] O módulo de instalação de programa de recursos adicionais dinâmico 108 pode ser implementado de vários modos para dinamicamente instalar programa(s) de recursos adicionais para aplicativo(s), incluindo sendo implementado em hardware, software, firmware, ou qualquer combinação dos mesmos. Por exemplo, o módulo de instalação de programa de recursos adicionais dinâmico 108 pode ser implementado como código de programa de computador configurado para ser executado em um ou mais processadores. Em outro exemplo, o módulo de instalação de programa de recursos adicionais dinâmico 108 pode ser implementado como lógica de hardware/sistema de circuitos elétricos. Em uma concretização, o módulo de instalação de programa de recursos adicionais dinâmico 108

pode ser implementado em um sistema em chip (SoC). Cada SoC pode incluir um chip de circuito integrado que inclui um ou mais dentre um processador (por exemplo, microcontrolador, microprocessador, processador de sinal digital (DSP) etc.), memória, uma ou mais interfaces de comunicação, e/ou circuitos adicionais e/ou firmware incorporado para executar suas funções.

[0031] As FIGS. 2 até 5 descrevem módulos de instalação de programa de recursos adicionais dinâmicos 200, 300, 400, e 500, os quais são implementações ilustrativas do módulo de instalação de programa de recursos adicionais dinâmico 108 apresentado na FIG. 1, de acordo com concretizações. A FIG. 2 ilustra a submissão de um programa de recursos adicionais para utilização por um programador em um aplicativo de acordo com uma concretização. Como apresentado na FIG. 2, o módulo de instalação de programa de recursos adicionais dinâmico 200 inclui um portal de agente dinâmico 202, um armazenamento de blob 204, e uma ferramenta de empacotamento 206. O portal de agente dinâmico 202 interpreta solicitações (por exemplo, instruções) de programadores para adicionar programas de recursos adicionais para o armazenamento de blob 204, atualizar programas de recursos adicionais que são armazenados no armazenamento de blob 204, e/ou remover programas de recursos adicionais a partir do armazenamento de blob 204 baseado nas solicitações. Por exemplo, o portal de agente dinâmico 202 pode adicionar um programa de recursos adicionais designado para o armazenamento de blob 204 de acordo com uma solicitação de carregamento a partir de um primeiro programador. O portal de agente dinâmico 202 pode atualizar o programa de recursos adicionais designado de acordo com a(s) solicitação(solicitações) de atualização a partir do primeiro programador e/ou de outro(s) programador(programadores). O portal de agente dinâmico 202 pode

remover o programa de recursos adicionais designado a partir do armazenamento de blob 204 de acordo com a solicitação de remoção a partir do primeiro programador ou de outro programador. Cada um dos programas de recursos adicionais pode possuir qualquer funcionalidade adequada. Por exemplo, um ou mais dos programas de recursos adicionais podem executar operações de diagnóstico (por exemplo, monitoramento de problema(s), identificar fonte(s) de problema(s), depuração, definição de perfil, controle etc.) com respeito a um aplicativo.

[0032] Um programa de recursos adicionais pode incluir qualquer número adequado de componentes (por exemplo, arquivos). Por exemplo, um programa de recursos adicionais pode incluir um ou mais binários executáveis, uma ou mais bibliotecas de suporte, um manifesto de programa de recursos adicionais etc. Um manifesto de programa de recursos adicionais é um arquivo (por exemplo, um arquivo de linguagem de marcação extensiva (XML)) que descreve configurações (por exemplo, parâmetros) de um programa de recursos adicionais. Por exemplo, o manifesto de programa de recursos adicionais pode indicar quais valores de configuração são necessários para o programa de recursos adicionais. O portal de agente dinâmico 202 pode proporcionar uma interface com o usuário que permite a um usuário especificar os valores de configuração que correspondem às respectivas configurações do programa de recursos adicionais.

[0033] Como apresentado na FIG. 2, um programa de recursos adicionais 212 rotulado "Plugin.cspluginx" inclui um primeiro arquivo 208 rotulado "PluginAgent.exe" e um segundo arquivo 210 rotulado "Plugin.csplugin" para propósitos de ilustração. A ferramenta de empacotamento 206 é capaz de combinar os arquivos 208 e 210 para criar o programa de recursos adicionais 212, como apresentado na FIG. 2. Entretanto, será reconhecido que o programador pode

combinar os arquivos 208 e 210 para criar o programa de recursos adicionais 212.

[0034] Não obstante, o programa de recursos adicionais 212 é carregado via o portal de agente dinâmico 202. Por exemplo, o programador pode carregar o programa de recursos adicionais 212 utilizando um cliente implementado em um sistema do usuário (por exemplo, um dos sistemas do usuário 102A até 102M). O programa de recursos adicionais 212 pode ser configurado para possuir qualquer formato adequado (por exemplo, um formato que é baseado nas Convenções de Empacotamento Aberto). A ferramenta de empacotamento 206 pode empacotar o programa de recursos adicionais 212 no formato mencionado acima antes do programa de recursos adicionais 212 ser carregado via o portal de agente dinâmico 202. Em uma concretização ilustrativa, a ferramenta de empacotamento 206 proporciona uma forma online para o programador para conclusão antes do programa de recursos adicionais 212 ser carregado via o portal de agente dinâmico 202. Em outra concretização ilustrativa, o portal de agente dinâmico 202 proporciona a forma online para o programador.

[0035] O portal de agente dinâmico 202 gera uma alimentação 214, rotulada "PluginFeed.xml", que inclui o programa de recursos adicionais 212, a informação de localização com respeito a uma localização do programa de recursos adicionais 212, e a informação de instalação que inclui instruções para instalar o programa de recursos adicionais 212. A alimentação pode ser uma alimentação de instalador de plataforma da rede (WebPI), por exemplo, apesar de que será reconhecido que a alimentação pode ser qualquer tipo adequado de alimentação. Uma alimentação PI da Rede é um arquivo de configuração que permite a um instalador de plataforma da Rede estar apto a localizar e instalar programas de recursos adicionais e/ou

dependências que o instalador de plataforma da Rede é instruído para instalar. Em um exemplo, o programa de recursos adicionais 212 pode ser definido como um produto virtual dentro da alimentação 214. De acordo com este exemplo, o produto virtual identifica dependências do programa de recursos adicionais 212. A informação de instalação pode incluir instruções para descompactação do programa de recursos adicionais 212. A alimentação 214 pode ser compatível com o instalador de plataforma da Rede, apesar de que o escopo das concretizações ilustrativas não é limitado neste respeito. Um exemplo não limitativo de um trecho de código que define uma alimentação é proporcionado na seção III.A abaixo para propósitos de ilustração. O portal de agente dinâmico 202 proporciona o programa de recursos adicionais 212 e a alimentação 214 para o armazenamento de blob 204.

[0036] O armazenamento de blob 204 é um depósito público e acessível por rede (por exemplo, acessível pela Rede). O armazenamento de blob 204 armazena o programa de recursos adicionais 212 e a alimentação 214, junto com vários outros programas de recursos adicionais que são carregados via portal de agente dinâmico 202 e suas alimentações correspondentes. O armazenamento de blob 204 pode ser qualquer tipo adequado de depósito. Um tipo de depósito é uma base de dados. Por exemplo, o armazenamento de blob 204 pode ser uma base de dados relacional, uma base de dados de relação de entidade, uma base de dados de objeto, uma base de dados relacional de objeto, uma base de dados de linguagem de marcação extensível (XML) etc..

[0037] A FIG. 3 ilustra seleção de dependências de um aplicativo por um programador de acordo com uma concretização. Como apresentado na FIG. 3, o módulo de instalação de programa de recursos adicionais dinâmico 300 inclui portal de agente dinâmico 202,

armazenamento de blob 204, e uma interface 320. O portal de agente dinâmico 202 e o armazenamento de blob 204 são capazes de operar como descrito acima com referência à FIG. 2. O portal de agente dinâmico 202 interpreta solicitações (por exemplo, instruções) a partir dos programadores para instalar dependências com respeito ao aplicativo.

[0038] A interface 320 proporciona uma representação visual de dependências que são disponíveis para instalação com respeito ao aplicativo. Por exemplo, a interface 320 apresenta representações visuais 322A, 322B, e 322C das respectivas dependências A, B, e C. As caixas de seleção 324A até 324C correspondem às respectivas dependências A, B e C. O usuário pode selecionar a caixa de seleção 324A para fazer com que a dependência A seja instalada com respeito ao aplicativo; o usuário pode selecionar a caixa de seleção 324B para fazer com que a dependência B seja instalada com respeito ao aplicativo, e assim por diante.

[0039] Como apresentado na FIG. 3, as caixas de seleção 324A e 324C são selecionadas, e a caixa de seleção 324B não é selecionada. Por consequência, o portal de agente dinâmico 202 gera uma alimentação de produto 316, rotulada "ProductFeed.xml", para incluir dependências A e C não incluir dependência B. O portal de agente dinâmico 202 gera uma alimentação agregada 318, rotulada "AggregatedFeed.xml", incluindo alimentações 214 e 316. As alimentações 316 e 318 podem ser compatíveis com instalador de plataforma da Rede, apesar do escopo das concretizações ilustrativas não ser limitado com este respeito. O portal de agente dinâmico 202 proporciona as alimentações 316 e 318 para o armazenamento de blob 204.

[0040] O armazenamento de blob 204 armazena alimentações 316 e 318, junto com o programa de recursos adicionais 212 e alimentação

214. Cada uma das alimentações armazenadas pelo armazenamento de blob 204 pode corresponder a uma dependência conhecida ou uma dependência desconhecida. Uma dependência conhecida é uma dependência que é conhecida em um tempo no qual o aplicativo é desenvolvido. Por exemplo, uma estrutura de execução, tal como uma estrutura de execução Java®, é um tipo ilustrativo de dependência conhecida. Uma dependência desconhecida é uma dependência que não é conhecida no tempo no qual o aplicativo é desenvolvido. Por exemplo, um programa de recursos adicionais de medição de performance pode constituir uma dependência desconhecida. Deve ser observado que uma dependência desconhecida pode ser selecionada para instalação com respeito ao aplicativo após a implementação do aplicativo ser inicializada.

[0041] A FIG. 4 ilustra a seleção de programas de recursos adicionais para um aplicativo. Por um programador de acordo com uma concretização. Como apresentado na FIG. 4, um módulo de instalação de programa de recursos adicionais dinâmico 400 inclui o portal de agente dinâmico 202, armazenamento de blob 204, e uma interface 420. O portal de agente dinâmico 202 e o armazenamento de blob 204 são capazes de operar como descrito acima com referência às FIGS. 2 e 3. O portal de agente dinâmico 202 interpreta solicitações (por exemplo, instruções) a partir dos programadores para instalar programas de recursos adicionais com respeito ao aplicativo.

[0042] A interface 420 proporciona uma representação visual de programas de recursos adicionais que estão disponíveis para instalação com respeito ao aplicativo. Por exemplo, a interface 420 apresenta as representações visuais 422A, 422B, e 422C dos respectivos programas de recursos adicionais A, B, e C. As caixas de seleção 424A até 424C correspondem aos respectivos programas de recursos adicionais A, B, e C. O usuário pode selecionar a caixa de seleção 424A para fazer com

que o programa de recursos adicionais A seja instalado com respeito ao aplicativo; o usuário pode selecionar a caixa de seleção 424B para fazer com que o programa de recursos adicionais B seja instalado com respeito ao aplicativo, e assim por diante. O usuário pode igualmente configurar os programas de recursos adicionais (por exemplo, especificar parâmetros dos programas de recursos adicionais).

[0043] Como apresentado na FIG. 4, as caixas de seleção 424A e 424C são selecionadas e a caixa de seleção 424B não é selecionada. Por consequência, o portal de agente dinâmico 202 modifica a alimentação de produto 316 para incluir alimentações de programa de recursos adicionais para os programas de recursos adicionais A e C. Observe que a alimentação de programa de recursos adicionais para o programa de recursos adicionais B não é adicionada à alimentação de produto 316 devido ao fato de que a caixa de seleção 424B não é selecionada. O portal de agente dinâmico 202 gera os parâmetros 426, rotulado "Settings.xml", com respeito aos programas de recursos adicionais A e C em resposta às caixas de seleção 424A e 424C sendo selecionadas e/ou em resposta a receber instruções de configuração a partir do usuário. O portal de agente dinâmico 202 proporciona os parâmetros 426 para o armazenamento de blob 204 ser armazenado neste local.

[0044] O portal de agente dinâmico 202 pode não armazenar os parâmetros 426 em uma localização explicitamente pública. Por exemplo, os parâmetros 426 podem incluir dados confidenciais (por exemplo, cadeias de caracteres de conexão de base de dados). Em uma técnica ilustrativa para manter segurança de tais dados confidenciais, um URI baseado em identificador globalmente único (baseado em GUID) é utilizado em um container que não pode ser navegado para armazenar os parâmetros 426. Apesar de o URI tecnicamente ser público, acessar o URI por métodos de força bruta é

relativamente irrealizável. Será reconhecido que esta é meramente uma técnica ilustrativa, e outra(s) técnica(s) pode(m) ser implementadas em adição ou ao invés desta técnica ilustrativa.

[0045] Um exemplo não limitativo de um trecho de código que define uma alimentação gerada pela seleção de dois programas de recursos adicionais é proporcionado na seção III.B abaixo para propósitos de ilustração. Um exemplo não limitativo de um trecho de código que define um arquivo de configuração gerado pela seleção de dois programas de recursos adicionais é proporcionado na seção III.C abaixo para propósitos de ilustração.

[0046] A FIG. 5 ilustra instalação e execução das dependências e dos programas de recursos adicionais que são selecionados como apresentado nas respectivas FIGS. 3 e 4 de acordo com uma concretização. Como apresentado na FIG. 5, o módulo de instalação de programa de recursos adicionais dinâmico 500 inclui o portal de agente dinâmico 202 e o armazenamento de blob 204, os quais são capazes de operar como descrito acima com referências às FIGS. 2 até 4. Um aplicativo 528 é apresentado na FIG. 5 incluindo um papel 530 e um programa de recursos adicionais agente 532. O papel 530 define operação do aplicativo 528. O programa de recursos adicionais agente 532 age como um agente para o aplicativo 528. Por exemplo, o programa de recursos adicionais agente 532 proporciona uma capacidade para dinamicamente compreender uma topologia de uma rede na qual o aplicativo 528 está instalado. Devido às máquinas (por exemplo, máquinas físicas e/ou máquina virtual) em um ambiente em nuvem frequentemente estarem ocultas por uma barreira de proteção ("firewall"), a topologia do aplicativo 528 pode ser muito difícil de determinar diretamente. Por consequência, o programa de recursos adicionais agente 532 é responsável por determinar o que inclui as máquinas virtuais constituintes que estão associadas com o aplicativo

528. Uma vez que a topologia do aplicativo 528 seja determinada, o programa de recursos adicionais agente 532 pode implementar outros programas de recursos adicionais, dependências destes programas de recursos adicionais, e dependências do aplicativo junto às localizações relevantes e então inicializar e executar estes programas de recursos adicionais e dependências.

[0047] O programa de recursos adicionais agente 532 inclui um agente WebPI 534 e um agente dinâmico 536 para propósitos ilustrativos e não é pretendido para ser limitativo. O agente WebPI 534 é configurado para instalar as dependências e os programas de recursos adicionais que são selecionados como apresentado nas respectivas FIGS. 3 e 4. O agente dinâmico 536 é configurado para executar os programas de recursos adicionais que são selecionados como apresentado nas respectivas FIGS. 3 e 4.

[0048] O agente WebPI 534 utiliza bibliotecas e uma interface de programação de aplicativo (API) do instalador de plataforma da Rede para ler a alimentação agregada 318 para determinar quais dependências e programas de recursos adicionais baixar e instalar. Por exemplo, o agente WebPI 534 proporciona uma consulta (por exemplo, consulta 538) para o portal de agente dinâmico 202. A consulta inclui uma solicitação para a alimentação agregada 318 e um identificador de implementação (ID) que é associado com a implementação do aplicativo 528. O portal de agente dinâmico 202 redireciona a consulta para uma localização de armazenamento no armazenamento de blob 204 que corresponde um identificar de recurso uniforme (URI), tal como um localizador de recurso uniforme (URL) ou um nome de recurso uniforme (URN), que é baseado no ID de implementação na consulta. O agente WebPI 534 baixa a alimentação agregada 318 a partir da localização de armazenamento que corresponde ao URI para instalação das dependências e os

programas de recursos adicionais que são selecionados como apresentado nas respectivas FIGS. 3 e 4.

[0049] Em um exemplo, o agente WebPI 534 pode ser configurado dentro do aplicativo 528 para utilizar uma alimentação predeterminada para instalar um conjunto predeterminado de produtos (por exemplo, uma ou mais dependências e/ou um ou mais programas de recursos adicionais) a partir desta alimentação ou um produto "padrão" se nenhum produto é especificado pela alimentação predeterminada. O produto padrão pode ser indicado por uma associação de uma palavra chave "padrão" para o produto dentro da alimentação predeterminada.

[0050] Em outro exemplo, se o agente WebPI 534 não é especificamente configurado, o agente WebPI 534 pode solicitar uma alimentação a partir de um aplicativo da Rede. O agente WPI 534 pode proporcionar o ID de implementação que é associado com a implementação do aplicativo 528, fazendo com que o aplicativo da rede redirecione o agente WPI 534 para a alimentação associada com a implementação. Se nenhuma alimentação for especificada ou nenhuma alimentação for associada com a implementação do aplicativo 528, o agente WebPI 534 pode não instalar as dependências e/ou programas de recursos adicionais que são selecionados como apresentado nas respectivas FIGS. 3 e 4.

[0051] O agente dinâmico 536 proporciona uma consulta (por exemplo, consulta 538) para o portal de agente dinâmico 202. A consulta inclui uma solicitação dos parâmetros 426 e um identificador de implementação (ID) que é associado com a implementação do aplicativo 528. O portal de agente dinâmico 202 redireciona a consulta para a localização de armazenamento no armazenamento de blob 204 que corresponde ao URI que é baseado no ID de implementação na consulta. O agente dinâmico 536 baixa os parâmetros 426 a partir da localização de armazenamento que corresponde ao URI para

execução dos programas de recursos adicionais que são selecionados como apresentado na FIG. 4. Os parâmetros 426 podem (ou não) incluir parâmetros com respeito a dependências.

[0052] Em um exemplo, o agente dinâmico 536 pode ser configurado dentro do aplicativo 528 para utilizar um arquivo de configuração predeterminado que identifica os programas de recursos adicionais e os parâmetros de configuração destes programas de recursos adicionais. Em outro exemplo, se o agente dinâmico 536 não for especificamente configurado, o agente dinâmico 536 pode solicitar um arquivo de configuração a partir de um aplicativo da Rede. O agente dinâmico 536 pode proporcionar o ID de implementação que é associado com a implementação do aplicativo 528, fazendo com que o aplicativo da Rede redirecione o agente dinâmico 536 para o arquivo de configuração associado com a implementação. Se nenhum arquivo de configuração for especificado ou nenhum arquivo de configuração for associado com a implementação do aplicativo 528, o agente dinâmico 536 pode não executar os programas de recursos adicionais que são selecionados como apresentado nas respectivas FIGS. 3 e 4.

[0053] Em uma concretização ilustrativa, para cada programa de recursos adicionais configurado, o agente dinâmico 536 localiza o manifesto correspondente de programa de recursos adicionais (por exemplo, na máquina local). O agente dinâmico 536 então executa cada um dos binários indicados dentro do manifesto. Se existirem parâmetros de configuração associados com o programa de recursos adicionais, o agente dinâmico 536 passa os parâmetros de configuração para o programa de recursos adicionais via variáveis de ambiente. Se uma instrução para aguardar pelo programa de recursos adicionais para descontinuar execução for especificada no manifesto de programa de recursos adicionais, o agente dinâmico 536 aguarda por este programa de recursos adicionais para descontinuar execução.

De outro modo, o agente dinâmico 536 muda para o próximo programa de recursos adicionais. Quando todos os programas de recursos adicionais configurados tiverem sido executados, o agente dinâmico 536 descontinua a execução.

[0054] Em outra concretização ilustrativa, a operação do agente dinâmico 536 imita o comportamento do processo inicializador da plataforma que executa os programas de recursos adicionais empacotados com o aplicativo 528. Desse modo, o agente dinâmico 536 pode reduzir (por exemplo, minimizar) diferenças arquiteturais entre os programas de recursos adicionais executando como parte do aplicativo 528 e programas de recursos adicionais executando como parte deste cenário dinâmico. Alguns programas de recursos adicionais podem ser capazes de execução em ambos ambientes sem alterações para os programas de recursos adicionais.

[0055] Deve ser observado que a topologia da rede na qual o aplicativo 528 é instalado pode buscar informação a partir do portal de agente dinâmico 202 para então configurar ela própria baseada nos parâmetros que estão no portal de agente dinâmico 202. Isto é diferente a partir das técnicas de envio que tradicionalmente são empregadas para tentar descobrir topologia de rede.

[0056] As FIGS. 6 até 9 descrevem fluxogramas 600, 700, 800, e 900 de métodos ilustrativos para dinamicamente instalar programa(s) de recursos adicionais para aplicativo(s) de acordo com concretizações. Os fluxogramas 600, 700, 800, e 900 podem ser executados pelo módulo de instalação de programa de recursos adicionais dinâmico 108 do sistema 100 apresentado na FIG. 1, por exemplo. Para propósitos de ilustração, os fluxogramas 600, 700, 800, e 900 são descritos com respeito ao módulo de instalação de programa de recursos adicionais dinâmico 1000 apresentado na FIG. 10, o qual é outro exemplo de módulo de instalação de programa de

recursos adicionais dinâmico 108, de acordo com uma concretização. Como apresentado na FIG. 10, o módulo de instalação de programa de recursos adicionais dinâmico 1000 inclui um módulo de identificação 1002, um módulo de causação 1004, um módulo de instalação 1006, um módulo de execução 1008, um módulo de interface 1010, um módulo de seleção 1012, um módulo de geração 1014, um módulo de determinação 1016, e um armazenamento 1018. Adicionalmente, concretizações estruturais e operacionais irão se tornar aparentes ao(s) versado(s) na técnica relevante baseados na discussão com respeito aos fluxogramas 600, 700, 800, e 900.

[0057] Como apresentado na FIG. 6, o método do fluxograma 600 inicia na etapa 602. Na etapa 602, a informação de configuração com respeito a um aplicativo especificado é identificada. A informação de configuração indica que o(s) programa(s) de recursos adicionais selecionado(s) a partir de um ou mais programas de recursos adicionais, os quais são armazenados em uma localização que é acessível para aplicativos implementados, são para serem instalados em resposta ao aplicativo especificado sendo implementado. A informação de configuração pode ser informação de configuração específica do aplicativo que é específica para o aplicativo especificado. Por exemplo, a informação de configuração pode definir o conteúdo do aplicativo especificado em termos de seu arquivo de instalação e/ou uma descrição do aplicativo especificado para consumo humano. Em uma implementação ilustrativa, o módulo de identificação 1002 identifica a informação de configuração 1026. A informação de configuração 1026 indica que o(s) programa(s) de recursos adicionais 1034 selecionado(s) a partir do(s) programas de recursos adicionais 1022, os quais são armazenados no armazenamento 1018, são para serem instalados em resposta ao aplicativo especificado 1030 sendo implementado. Será reconhecido que, em algumas concretizações

ilustrativas, a informação de configuração indica que nenhum programa de recursos adicionais designado é para ser instalado e/ou executado.

[0058] Na etapa 604, um programa de recursos adicionais agente é levado a executar em uma primeira implementação do aplicativo especificado (por exemplo, através de máquinas em um ambiente em nuvem ou "nos locais"). O aplicativo especificado é empacotado para incluir o programa de recursos adicionais agente. Em uma implementação ilustrativa, o módulo de causação 1004 faz com que o programa de recursos adicionais agente 1032 seja executado em uma primeira implementação do aplicativo especificado 1030. De acordo com esta implementação, o aplicativo especificado 1030 é empacotado para incluir o programa de recursos adicionais agente 1032.

[0059] Na etapa 606, o programa de recursos adicionais agente é utilizado para instalar o(s) programa(s) de recursos adicionais dinamicamente (por exemplo, dinamicamente enquanto o aplicativo especificado está executando) baseado na informação de configuração. Em uma implementação ilustrativa, o módulo de instalação utiliza o programa de recursos adicionais agente 1032 para instalar o(s) programa(s) de recursos adicionais designado 1034 dinamicamente baseado na informação de configuração 1026.

[0060] Em uma concretização ilustrativa, a etapa 602 inclui identificar uma alimentação que inclui o(s) programa(s) de recursos adicionais designado(s), informação de localização com respeito a uma localização do(s) programa(s) de recursos adicionais designado, e informação de instalação que inclui instruções para instalar o(s) programa(s) de recursos adicionais. Por exemplo, o módulo de identificação 1002 pode identificar a alimentação. De acordo com esta concretização, o programa de recursos adicionais agente é utilizado

para instalar o(s) programa(s) de recursos adicionais agente dinamicamente na etapa 606 baseado na informação de localização e na informação de instalação. Por exemplo, o módulo de instalação 1006 pode utilizar o programa de recursos adicionais agente 1032 para instalar o(s) programa(s) de recursos adicionais designado(s) 1034 dinamicamente baseado na informação de localização e na informação de instalação. Será reconhecido que a alimentação pode adicionalmente incluir uma ou mais dependências do aplicativo especificado. Por exemplo, a uma ou mais dependências do aplicativo especificado pode ser um superconjunto dos programa de recursos adicionais designados.

[0061] Em algumas concretizações ilustrativas, uma ou mais etapas 602, 604, e/ou 606 do fluxograma 600 podem não ser executadas. Além disso, etapas em adição ou no lugar das etapas 602, 604, e/ou 606 podem ser executadas. Por exemplo, em uma concretização ilustrativa, a informação de configuração adicionalmente indica uma ou mais dependências do aplicativo especificado. Será reconhecido que qualquer uma dentre a uma ou mais dependências pode possuir uma ou mais dependências, qualquer uma das mesmas podendo possuir uma ou mais dependências, e assim por diante. Por consequência, cada dependência pode possuir sua própria uma ou mais dependências. De acordo com esta concretização, o método do fluxograma 600 adicionalmente inclui utilizar o programa de recursos adicionais agente para instalar a uma ou mais dependências baseado na informação de configuração. Em uma implementação ilustrativa, o módulo de instalação 1006 utiliza o programa de recursos adicionais agente 1032 para instalar a uma ou mais dependências baseado na informação de configuração 1026.

[0062] Em outra concretização ilustrativa, a informação de configuração adicionalmente indica uma ou mais dependências do(s)

programa(s) de recursos adicionais designado(s). De acordo com esta concretização, o método do fluxograma 600 adicionalmente inclui utilizar o programa de recursos adicionais agente para instalar a uma ou mais dependências do(s) programa(s) de recursos adicionais designado(s) baseado na informação de configuração. Por exemplo, a uma ou mais dependências podem ser armazenadas no armazenamento 1018.

[0063] Ainda em outra concretização ilustrativa, cada um dos programas de recursos adicionais designados inclui uma respectiva coleção de binários executáveis. Um binário executável é um binário que é executável. De acordo com esta concretização, a informação de configuração indica para cada um dos programas de recursos adicionais designados um respectivo subconjunto da respectiva coleção de binários executáveis. Cada subconjunto de uma respectiva coleção de binários executáveis pode incluir qualquer um ou mais dentre os binários executáveis na respectiva coleção (por exemplo, menos do que todos os binários executáveis, todos os binários executáveis, nenhum dos binários executáveis etc.). Adicionalmente, de acordo com esta concretização, o método do fluxograma 600 adicionalmente inclui utilizar o programa de recursos adicionais agente para executar cada subconjunto baseado na informação de configuração. Em uma implementação ilustrativa, o módulo de execução 1008 utiliza o programa de recursos adicionais agente 1032 para executar cada subconjunto baseado na informação de configuração 1026. Em um exemplo, o módulo de instalação 1006 pode gerar um indicador de instalação 1020 em resposta ao(s) programa(s) de recursos adicionais designado(s) 1034 sendo instalado(s). Por exemplo, o indicador de instalação 1020 pode indicar que o(s) programa(s) de recursos adicionais designado(s) 1034 foram instalados. De acordo com este exemplo, o módulo de execução 1008

pode utilizar o programa de recursos adicionais agente 1032 para executar cada subconjunto em resposta a receber o indicador de instalação 1020 a partir do módulo de instalação 1006.

[0064] É observado que qualquer um ou mais dos programas de recursos adicionais designados podem incluir um único binário executável. É adicionalmente observado que qualquer um ou mais dos programas de recursos adicionais designados podem incluir um ou mais binários suplementares (por exemplo, biblioteca) em adição ao(s) binário(s) executável (executáveis).

[0065] Em ainda outra concretização ilustrativa, a etapa 602 inclui identificar um arquivo de configuração predeterminado. O arquivo de configuração predeterminado pode ser incluído na informação de configuração, apesar do escopo da concretização ilustrativa não ser limitado a este respeito. Por exemplo, o módulo de identificação 1002 pode identificar o arquivo de configuração predeterminado (por exemplo, arquivo de configuração 10036), o qual pode ser incluído na informação de configuração 1026. De acordo com esta concretização, o arquivo de configuração predeterminado indica pelo menos um parâmetro de configuração para cada um dos programas de recursos adicionais. Por exemplo, a informação de configuração 1026 pode indicar parâmetro(s) de configuração para cada um dos programas de recursos adicionais designados 1034. Em um exemplo, um parâmetro de configuração pode especificar uma frequência na qual o aplicativo especificado é para ser acessado para propósitos de monitoramento. De acordo com este exemplo, o parâmetro de configuração pode especificar que o aplicativo é para ser acessada a cada dez segundos, ou a cada 5 microssegundos, ou a cada 30 minutos etc. Adicionalmente, de acordo com esta concretização, o método do fluxograma 600 adicionalmente inclui utilizar o programa de recursos adicionais agente para executar o(s) programa(s) de recursos

adicionais designado(s) baseado no arquivo de configuração predeterminado em resposta a utilizar o programa de recursos adicionais agente para instalar o(s) programa(s) de recursos adicionais designado(s). Em uma implementação ilustrativa, o módulo de execução 1008 utiliza o programa de recursos adicionais agente 1032 para executar o(s) programa(s) de recursos adicionais designado(s) 1034 baseado no arquivo de configuração predeterminado. Em um exemplo, o arquivo de configuração predeterminado pode ser diferente de outro arquivo que indica que pelo menos um programa de recursos adicionais designado é para ser instalado em resposta ao aplicativo especificado sendo implementado. Em outro exemplo, a informação de configuração que indica que pelo menos um programa de recursos adicionais designado é para ser instalado em resposta ao aplicativo especificado sendo implementado pode ser incluída no arquivo de configuração predeterminado.

[0066] Ainda em outra concretização ilustrativa, a etapa 602 inclui solicitar um arquivo de configuração que corresponde à primeira implementação do aplicativo especificado a partir do aplicativo da Rede através do qual o arquivo de configuração é gerado. Por exemplo, o módulo de identificação 1002 pode proporcionar uma solicitação de arquivo 1028, a qual solicita um arquivo de configuração 1036 que corresponde à primeira implementação do aplicativo especificado 1030 a partir do aplicativo da Rede. De acordo com esta concretização, a etapa 602 adicionalmente inclui receber o arquivo de configuração a partir do aplicativo da Rede. O arquivo de configuração pode ser incluído na informação de configuração, apesar do escopo das concretizações ilustrativas não estar limitado a este respeito. O arquivo de configuração indica pelo menos um parâmetro de configuração para cada um dentre os programas de recursos adicionais designados. Por exemplo, o módulo de execução 1008

pode receber o arquivo de configuração 1036 a partir do aplicativo da Rede. Adicionalmente, de acordo com esta concretização, o método do fluxograma 600 adicionalmente inclui utilizar o programa de recursos adicionais agente para executar o(s) programa(s) de recursos adicionais designado(s) baseado no arquivo de configuração em resposta a utilizar o programa de recursos adicionais agente para instalar o(s) programa(s) de recursos adicionais designado(s). Em uma implementação ilustrativa, o módulo de execução 1008 utiliza o programa de recursos adicionais agente 1032 para executar o(s) programa(s) de recursos adicionais designado(s) 1034 baseado no arquivo de configuração 1036. Em um exemplo, o arquivo de configuração pode ser diferente de outro arquivo que indica que o pelo menos um programa de recursos adicionais designado é para ser instalado em resposta ao aplicativo especificado sendo implementado. Em outro exemplo, a informação de configuração que indica que o pelo menos um programa de recursos adicionais designado é para ser instalado em resposta ao aplicativo especificado sendo implementado pode ser incluída no arquivo de configuração.

[0067] Em uma concretização ilustrativa, o método do fluxograma 600 inclui uma ou mais das etapas apresentadas na FIG. 7. Como apresentado na FIG. 7, o método do fluxograma 700 inicia na etapa 702. Na etapa 702, uma interface é proporcionada através da qual o um ou mais programas de recursos adicionais são carregados pelo(s) usuário(s). Em uma implementação ilustrativa, o módulo de interface 1010 proporciona interface 1038. Por exemplo, a interface 1038 pode ter sido utilizada pelo(s) usuário(s) para carregar o(s) programa(s) de recursos especiais 1022.

[0068] Na etapa 704, o(s) programa(s) de recursos adicionais designado(s) é(são) selecionado(s) a partir do um ou mais programas de recursos adicionais baseado em instruções iniciadas pelo usuário

que especificam o(s) programa(s) de recursos adicionais. Em uma implementação ilustrativa, o módulo de seleção 1012 seleciona o(s) programa(s) de recursos adicionais designado(s) 1034 a partir do(s) programa(s) de recursos adicionais designado(s) 1022 baseado nas primeiras instruções iniciadas pelo usuário. Por exemplo, o módulo de seleção 1012 pode gerar uma solicitação de seleção 1040 para especificar o(s) programas de recursos adicionais designados 1034 baseado nas primeiras instruções iniciadas pelo usuário. De acordo com este exemplo, o módulo de seleção 1012 pode proporcionar a solicitação de seleção 1040 para o armazenamento 1018 para selecionar o(s) programa(s) de recursos adicionais designados 1034 a partir do(s) programas de recursos adicionais 1022.

[0069] Na etapa 706, a informação de configuração é gerada para incluir parâmetros de configuração do(s) programa(s) de recursos designados. Os parâmetros de configuração são especificados pelas segundas instruções iniciadas pelo usuário. Em uma implementação ilustrativa, o módulo de geração 1014 gera a informação de configuração 1026 incluindo os parâmetros de configuração do(s) programas de recursos adicionais designado(s) 1034, os quais são especificados pelas segundas instruções iniciadas pelo usuário.

[0070] Na etapa 708, uma ou mais dependências do aplicativo especificado são selecionadas baseadas nas terceiras instruções iniciadas pelo usuário que especificam a um ou mais dependências. Em uma implementação ilustrativa, o módulo de seleção 1012 seleciona uma ou mais dependências do aplicativo especificado 1030 baseado nas terceiras instruções iniciadas pelo usuário.

[0071] Em outra concretização ilustrativa, o método do fluxograma 600 inclui uma ou mais das etapas apresentadas na FIG. 8. Como apresentado na FIG. 8, o método do fluxograma 800 inicia na etapa 802. Na etapa 802, é realizada uma determinação de que a

informação de configuração é alterada enquanto o aplicativo especificado está executando para proporcionar informação de configuração alterada. A informação de configuração alterada indica que o(s) segundo(s) programa(s) de recursos adicionais é (são) para ser instalado. A informação de configuração antes de ser alterada não indica que o(s) segundo(s) programa(s) de recursos adicionais designado(s) é (são) para ser instalado(s). Em uma implementação ilustrativa, o módulo de determinação 10016 determina que a informação de configuração 1026 é alterada enquanto o aplicativo especificado 1030 está executando para proporcionar a informação de configuração alterada.

[0072] Na etapa 804, o programa de recursos adicionais agente é utilizado para instalar o segundo programa(s) de recursos adicionais designado(s) dinamicamente enquanto o aplicativo especificado está executando baseado na informação de configuração alterada. Em uma implementação ilustrativa, o módulo de instalação 1006 utiliza o programa de recursos adicionais agente 1032 para instalar o segundo programa(s) de recursos adicionais designado dinamicamente enquanto o aplicativo especificado 1030 está executando.

[0073] Ainda em outra concretização ilustrativa, o método do fluxograma 600 inclui uma ou mais das etapas apresentadas na FIG. 9. Como apresentado na FIG. 9, o método do fluxograma 900 inicia na etapa 902. Na etapa 902, o programa de recursos adicionais agente é utilizado para executar uma primeira execução do(s) programa(s) de recursos adicionais designado(s). Por exemplo, a etapa 902 pode ser executada em resposta a utilizar o programa de recursos adicionais agente para instalar o(s) programa(s) de recursos adicionais na etapa 606 do fluxograma 600. Em uma implementação ilustrativa, o módulo de execução 1008 utiliza o programa de recursos adicionais agente 1032 para executar uma primeira execução do(s) programa(s) de

recursos adicionais 1034.

[0074] Na etapa 904, um estado da primeira execução é identificado. Por exemplo, o estado pode ser associado com alguns parâmetros de configuração do(s) programa(s) de recursos adicionais. Em uma implementação ilustrativa, o módulo de identificação 1002 identifica o estado da primeira execução.

[0075] Na etapa 906, é realizada uma determinação de que o aplicativo especificado é para ser reciclado. Em uma implementação ilustrativa, o módulo de determinação 1016 determina que o aplicativo especificado é para ser reciclado. Será reconhecido que "para ser reciclado" significa "para ser reiniciado". Por exemplo, a determinação de que o aplicativo especificado é para ser reciclado pode ser baseada em um ou mais fatores, incluindo, mas não limitado à determinação que uma máquina (por exemplo, uma máquina física ou uma máquina virtual) na qual o aplicativo especificado é implementado é para ser reiniciada (ou foi reiniciada), detecção de uma mau funcionamento de hardware com respeito à máquina na qual o aplicativo especificado é implementado, detecção de um mau funcionamento com respeito ao aplicativo especificado, uma determinação de que um sistema operacional associado com o aplicativo especificado é para ser atualizado, uma determinação de que uma plataforma na qual o aplicativo especificado está executando solicitou que o aplicativo especificado seja reciclado etc..

[0076] Na etapa 908, o programa de recursos adicionais agente é levado a executar em uma implementação subsequente do aplicativo especificado. A implementação subsequente é iniciada após o início da primeira implementação. Em uma implementação ilustrativa, o módulo de causação 1004 causa que o programa de recursos adicionais agente 1032 execute em uma implementação subsequente do aplicativo especificado 1030.

[0077] Na etapa 910, o programa de recursos adicionais agente é utilizado para reinstalar o(s) programa(s) de recursos adicionais designado(s) dinamicamente baseado na informação de configuração. Em uma implementação ilustrativa, o módulo de instalação 1006 utiliza o programa de recursos adicionais agente 1032 para reinstalar o(s) programa(s) de recursos adicionais designado(s) 1034 dinamicamente baseado na informação de configuração 1026.

[0078] Na etapa 912, o programa de recursos adicionais agente é utilizado para iniciar uma segunda execução do(s) programa(s) de recursos adicionais designado(s) no estado da primeira execução. Por exemplo, o programa de recursos adicionais agente pode ser utilizado para iniciar a segunda execução utilizando algum parâmetro de configuração do(s) programas de recursos adicionais designados que estão associados com o primeiro estado da primeira execução. Em uma implementação ilustrativa, o módulo de execução 1008 utiliza o programa de recursos adicionais agente para iniciar uma segunda execução do(s) programa(s) de recursos adicionais designado(s) 1034 no estado da primeira execução.

[0079] Será reconhecido que o módulo de instalação de programa de recursos adicionais dinâmico 1000 pode não incluir um ou mais dentre o módulo de identificação 1002, o módulo de causação 1004, o módulo de instalação 1006, o módulo de execução 1008, o módulo de interface 1010, o módulo de seleção 1012, o módulo de geração 1014, o módulo de determinação 1016, e/ou o armazenamento 1018. Adicionalmente, o módulo de instalação de programa de recursos adicionais dinâmico 1000 pode incluir módulos em adição ou em lugar do módulo de identificação 1002, do módulo de causação 1004, do módulo de instalação 1006, do módulo de execução 1008, do módulo de interface 1010, do módulo de seleção 1012, do módulo de geração 1014, do módulo de determinação 1016, e/ou do armazenamento

1018.

[0080] O módulo de instalação de programa de recursos adicionais dinâmico 108, o portal de agente dinâmico 202, o módulo de identificação 1002, o módulo de causa 1004, o módulo de instalação 1006, o módulo de execução 1008, o módulo de interface 1010, o módulo de seleção 1012, o módulo de geração 1014, o módulo de determinação 1016, o fluxograma 600, o fluxograma 700, o fluxograma 800, e o fluxograma 900 podem ser implementados em hardware, software, firmware, ou em qualquer combinação dos mesmos.

[0081] Por exemplo, o módulo de instalação de programa de recursos adicionais dinâmico 108, o portal de agente dinâmico 202, o módulo de identificação 1002, o módulo de causação 1004, o módulo de instalação 1006, o módulo de execução 1008, o módulo de interface 1010, o módulo de seleção 1012, o módulo de geração 1014, o módulo de determinação 1016, o fluxograma 600, o fluxograma 700, o fluxograma 800, e/ou o fluxograma 900 podem ser implementados como código de programa de computador configurado para ser executado em um ou mais processadores.

[0082] Em outro exemplo, o módulo de instalação de programa de recursos adicionais dinâmico 108, o portal de agente dinâmico 202, o módulo de identificação 1002, o módulo de causação 1004, o módulo de instalação 1006, o módulo de execução 1008, o módulo de interface 1010, o módulo de seleção 1012, o módulo de geração 1014, o módulo de determinação 1016, o fluxograma 600, o fluxograma 700, o fluxograma 800, e/ou o fluxograma 900 podem ser implementados como sistema de lógica/elétrico de hardware.

[0083] Por exemplo, em uma concretização, um ou mais dentre o módulo de instalação de programa de recursos adicionais dinâmico 108, o portal de agente dinâmico 202, o módulo de identificação 1002, o módulo de causação 1004, o módulo de instalação 1006, o módulo

de execução 1008, o módulo de interface 1010, o módulo de seleção 1012, o módulo de geração 1014, o módulo de determinação 1016, o fluxograma 600, o fluxograma 700, o fluxograma 800, e/ou o fluxograma 900 podem ser implementados em um sistema em chip (SoC). O SoC pode incluir um chip de circuito integrado que inclui um ou mais dentre um processador (por exemplo, um microcontrolador, um microprocessador, um processador de sinal digital (DSP) etc.), memória, uma ou mais interfaces de comunicação, e/ou circuitos adicionais e/ou firmware incorporado para executar suas funções.

[0084] A FIG. 11 descreve um computador ilustrativo 1100 no qual concretizações podem ser implementadas. Qualquer um dentre o um ou mais clientes 102A até 102M, qualquer uma dentre a(s) fonte(s) de programa de recursos adicionais 110, ou qualquer um dentre o um ou mais dos servidores 106A Até 106N apresentados na FIG. 1 (ou qualquer um do um ou mais subcomponentes dos mesmos apresentados nas FIGS. 2 até 5 e 10) pode ser implementado utilizando computador 1100, incluindo um ou mais aspectos do computador 1100 e/ou aspectos alternativos. O computador 1100 pode ser um dispositivo de computação de propósito geral na forma de um computador pessoal convencional, um computador móvel, ou uma estação de trabalho, por exemplo, ou o computador 1100 pode ser um dispositivo de computação de propósito especial. A descrição do computador 1100 é proporcionada neste documento para propósitos de ilustração, e não é pretendida para ser limitativa. As concretizações podem ser implementadas em tipos adicionais de sistemas de computador, como seria conhecido pelos versados na técnica relevante.

[0085] Como apresentado na FIG. 11, o computador 1100 inclui uma unidade de processamento 1102, uma memória do sistema 1104, e um barramento 1106 que acopla vários componentes do sistema

incluindo a memória do sistema 1104 com a unidade de processamento 1102. O barramento 1106 representa um ou mais dentre os vários tipos de estruturas de barramento, incluindo um barramento de memória ou controlador de memória, um barramento periférico, uma porta de gráfico acelerado, e um processador ou barramento local utilizando qualquer uma dentre uma variedade de arquiteturas de barramento. A memória do sistema 1104 inclui memória somente leitura (ROM) 1108 e memória de acesso aleatório (RAM) 1110. Um sistema básico de entrada/saída 1112 (BIOS) é armazenado na ROM 1108.

[0086] O computador 1100 também possui um dentre a uma ou mais unidades seguintes: uma unidade de disco rígido 1114 para ler e gravar em um disco rígido, uma unidade de disco magnético 1116 para ler ou gravar em um disco magnético removível 1118, e uma unidade de disco ótico 1120 para ler e gravar em um disco ótico removível 1122, tal como um CD ROM, DVD ROM, ou outro meio ótico. A unidade de disco rígido 1114, a unidade de disco magnético 1116, e unidade de disco ótico 1120 são conectadas com o barramento 1106 por uma interface de unidade de disco rígido 1124, uma interface de unidade de disco magnético 1126, e uma interface de unidade ótica 1128, respectivamente. As unidades e seus meios de armazenamento legíveis por computador associados proporcionam armazenamento não volátil de instruções legíveis por computador, estruturas de dados, módulos de programa, e outros dados para o computador. Apesar de um disco rígido, um disco magnético removível e um disco ótico removível serem descritos, outros tipos de meio de armazenamento legível por computador pode ser utilizado para armazenar dados, tal como cartões de memória flash, discos de vídeo digital, memórias de acesso aleatório (RAMs), memórias somente para leitura (ROM), e assim por diante.

[0087] Vários módulos de programa podem ser armazenados no disco rígido, disco magnético, disco ótico, ROM, ou RAM. Estes programas incluem um sistema operacional 1130, um ou mais programas aplicativos 1132, outros módulos de programa 1134, e dados de programa 1136. Os programas aplicativos 1132 ou módulos de programa 1134 podem incluir, por exemplo, lógica de programa de computador para implementar o módulo de instalação de programa de recursos adicionais dinâmico 108, portal de agente dinâmico 202, módulo de identificação 1002, módulo de causação 1004, módulo de instalação 1006, módulo de execução 1008, módulo de interface 1010, módulo de seleção 1012, módulo de geração 1014, módulo de determinação 1016, fluxograma 600 (incluindo qualquer etapa do fluxograma 600), fluxograma 700 (incluindo qualquer etapa do fluxograma 700), fluxograma 800 (incluindo qualquer etapa do fluxograma 800), e/ou fluxograma 900 (incluindo qualquer etapa do fluxograma 900), como descrito neste documento.

[0088] Um usuário pode entrar comandos e informação no computador 1100 através de dispositivos de entrada tal como teclado 1138 e dispositivo apontador 1140. Outros dispositivos de entrada (não apresentados) podem incluir um microfone, joystick, console de jogo, antena de satélite, scanner, tela sensível ao toque, câmera, acelerômetro, giroscópio, ou coisa parecida. Estes e outros dispositivos de entrada são frequentemente conectadas com a unidade de processamento 1102 através de uma interface de porta serial 1142 que é acoplado ao barramento 1106, mas podem ser conectados por outras interfaces, tal como uma porta paralela, uma porta de jogo, ou um barramento serial universal (USB).

[0089] Um dispositivo de vídeo 1144 (por exemplo, um monitor) também é conectado ao barramento 1106 via uma interface, tal como um adaptador de vídeo 1146. Em adição ao dispositivo de vídeo 1144,

o computador 1100 pode incluir outros dispositivos periféricos de saída (não apresentados), tais como alto-falantes e impressoras.

[0090] O computador 1100 é conectado com a rede 1148 (por exemplo, a Internet) através de uma interface ou adaptador de rede 1150, um modem 1152, ou outro dispositivo para estabelecer comunicações através da rede. O modem 1152, o qual pode ser interno ou externo, é conectado ao barramento 1106 via interface de porta serial 1142.

[0091] Como utilizado neste documento, os termos "meio de programa de computador" e "meio legível por computador" são geralmente utilizados para se referir a meio não temporário tal como disco rígido associado com unidade de disco rígido 1114, disco magnético removível 1118, disco ótico removível 1122, bem como outro meio não temporários, tal como cartões de memória flash, discos de vídeo digital, memórias de acesso aleatório (RAMs), memórias somente para leitura (ROM), e assim por diante. Tais meios de armazenamento legíveis por computador são distintos e não se sobrepõem com meios de comunicação. Os meios de comunicação tipicamente incorporam instruções legíveis por computador, estruturas de dados, módulos de programa ou outros dados em um sinal de dados modulado tal como uma onda portadora. O termo "sinal de dados modulado" significa um sinal que possui uma ou mais de suas características estabelecidas ou alteradas de modo a codificar informação no sinal. A título de ilustração, e não de limitação, os meios de comunicação incluem meios sem uso de fios, tal como meios acústicos, RF, infravermelho ou outros meios sem o uso de fios. As concretizações ilustrativas também são direcionadas para tais meios de comunicação.

[0092] Como observado acima, programas de computador e módulos (incluindo programas aplicativos 1132 e outros módulos de

programa 1134) podem ser armazenados no disco rígido, disco magnético, disco ótico, ROM, ou RAM. Tais programas de computador também podem ser recebidos via interface da rede 1150 ou interface de porta serial 1142. Tais programas de computador, quando executados ou carregados por um aplicativo, permitem ao computador 1100 implementar aspectos das concretizações discutidas neste documento. Por consequência, tais programas de computador representam controladores do computador 1100.

[0093] As concretizações ilustrativas também são direcionadas para produtos de programa de computador compreendendo software (por exemplo, instruções legíveis por computador) armazenado em qualquer meio utilizável de computador. Tal software, quando executado em um ou mais dispositivos de processamento, faz com que o(s) dispositivo(s) de processamento de dados opere como descrito neste documento. Concretizações podem empregar qualquer meio utilizável por computador ou legível por computador, conhecido agora ou no futuro. Exemplos de meios legíveis por computador incluem, mas não estão limitados a dispositivos de armazenamento, tal como RAM, discos rígidos, disquetes, CD ROMs, DVD ROMs, discos zip, fitas, dispositivos de armazenamento magnético, dispositivos de armazenamento ótico, dispositivos de armazenamento baseado em MEMS, dispositivos de armazenamento baseados em nanotecnologia, e assim por diante.

III. Trechos de Código Ilustrativos

[0094] A seguir estão três trechos de código ilustrativos, rotulados "Trecho de Código Ilustrativo No. 1", "Trecho de Código Ilustrativo No.2, e "Trecho de Código Ilustrativo No.3", respectivamente.

A. Trecho de Código Ilustrativo No. 1

[0095] O trecho de código abaixo demonstra uma alimentação ilustrativa para um programa de recursos adicionais:

```

    <?xml version="1.0" encoding="utf-8"?>
    <feed xmlns="http://www.w3.org/2005/Atom">
      <webpiFile version="2.0.1.3" />
      <keywords />
      <languages>
        <language default="true">
          <languageId>en</languageId>
          <name resourceName="Lang_en_Name">English</name>
        </language>
      </languages>
      <entry>
        <productId>Microsoft.WindowsAzure.Plugins.TestAgent</productId>
        <title>Test Agent</title>
        <version>1.0.0.0</version>
        <summary />
        <installers>
          <installer>
            <id>1</id>
            <languageId>en</languageId>
            <installerFile>
              <fileSize>20</fileSize>
              <installerURL>https://dynamicagent.blob.core.windows.net/dynamic-agent-
                plugins/microsoft.windowsazure.plugins.testagent/1.0.0.0/plugin.cspluginx</installerURI
              >
              <sha1>493960AE4DB0A6EC0914AAC3FDC07D1F78D9331A</sha1>
            </installerFile>
            <installCommands>
              <unzip>
                <zipFile>%InstallerFile%</zipFile>

```

```

<targetDirectory>%WebPlatformInstallerInstallationCachePath%\Microsoft.WindowsAz
re.Plugins.TestAgent\1.0.0.0</targetDirectory>
  </unzip>
  </installCommands>
</installer>
</installers>
<keywords />
</entry>
</feed>

```

B. Trecho de Código Ilustrativo No. 2

[0096] O seguinte trecho de código demonstra uma alimentação ilustrativa que é gerada pela seleção de dois programas de recursos adicionais:

```

<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <webpiFile version="2.0.1.3" />
  <keywords>
    <keyword id="Default" resourceName="Keyword_Default">Default</keyword>
  </keywords>
  <languages>
    <language default="true">
      <languageId>en</languageId>
      <name resourceName="Lang_en_Name">English</name>
      <name resourceName="Lang_en_Name">English</name>
    </language>
  </languages>
  <entry>
    <productId>6c85078a-ffb5-4ef0-a373-3635d36fe85c</productId>
    <title />
    <version />
    <summary />
    <dependency>
      <and>
        <dependency>
          <productId>IntelliTrace</productId>
        </dependency>
        <dependency>

```

```

    <productId>Microsoft.WindowsAzure.Plugins.TestAgent</productId>
  </dependency>
</and>
</dependency>
<keywords>
  <keywordId>Default</keywordId>
</keywords>
</entry>
</feed>

```

C. Trecho de Código Ilustrativo No. 3

[0097] O trecho de código seguinte demonstra um arquivo de configuração ilustrativo que é gerado pela seleção de dois programas de recursos adicionais:

```

<?xml version="1.0" encoding="utf-8"?>
<Settings id="3911d90d-c6cc-4908-a8cf-d5a43c0b8823" name="Test Application"
xmlns="http://schemas.microsoft.com/DynamicAgent/2011/07/Settings">
  <Plugins>
    <Plugin name="IntelliTrace" version="1.5.0.0">
      <ConfigurationSettings>
        <Setting name="IntelliTrace.IntelliTraceConnectionString"
value="BaseEndpoint=core.windows.net;Protocol=http;AccountName=<name>;Account
Key=<key>" />
      </ConfigurationSettings>
    </Plugin>
    <Plugin name="Microsoft.WindowsAzure.Plugins.TestAgent" version="1.0.0.0">
      <ConfigurationSettings>
        <Setting name="Microsoft.WindowsAzure.Plugins.TestAgent.TestSetting"
value="Howdy!" />
      </ConfigurationSettings>
    </Plugin>
  </Plugins>
</Settings>

```

IV. Conclusão

[0098] Apesar de várias concretizações terem sido descritas acima, deve ser entendido que elas foram apresentadas somente a título de ilustração, e não de limitação. Será aparente aos versados na técnica relevante que várias alterações na forma e detalhes podem ser

realizadas neste documento sem afastamento a partir do espírito e escopo da invenção. Assim, a amplitude e escopo da presente invenção não devem ser limitados por qualquer uma das concretizações ilustrativas descritas acima, mas devem ser definidos apenas de acordo com as reivindicações seguintes e seus equivalentes.

REIVINDICAÇÕES

1. Método para módulo de instalação de plugin dinâmico (108, 200, 300, 400, 500), em que uma instância do módulo de instalação de plugin dinâmico (108, 200, 300, 400, 500) é incluída em cada servidor que compreende um aplicativo especificado (528, 1030), o método compreende as etapas de:

identificar (602) uma informação de configuração (1026) com relação a um aplicativo especificado (528, 1030), a informação de configuração (1026) indicando que pelo menos um plugin designado (1034) selecionado a partir de um ou mais plugins (1022), os quais são armazenados em uma localização que é acessível para aplicativos implementados, é para ser instalado em resposta ao aplicativo especificado (528, 1030) sendo implementado ao longo de múltiplas máquinas (106A, 106B, 106N) em um ambiente de nuvem, e identificar uma informação de localização com relação à localização do pelo menos um plugin designado (1034);

fazer com que (604) um plugin agente (532, 1032) execute em uma primeira implementação do aplicativo especificado (528, 1030), o aplicativo especificado (528, 1030) sendo empacotado para incluir o plugin agente (532, 1032), **caracterizado pelo fato de que** o plugin agente (532, 1032) determina que máquinas (106A, 106B, 106N) constituintes que estão associadas com o aplicativo especificado (528, 1030) são incluídas, em que o plugin agente (532, 1032) fornece uma capacidade de entender de maneira dinâmica uma topologia de uma rede em que o aplicativo especificado (528, 1030) é instalado; e

utilizar (606) o plugin agente (532, 1032) para instalar o pelo menos um plugin designado (1034) dinamicamente para localizações relevantes baseado na informação de configuração (1026) em resposta para fazer com que o plugin agente (532, 1032)

execute na primeira implementação do aplicativo especificado (528, 1030), em que o plugin agente (532, 1032) implementa outros plugins, dependências destes plugins, e dependências do aplicativo especificado (528, 1030) para as localizações relevantes.

2. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** cada um do pelo menos um plugin designado (1034) inclui uma respectiva pluralidade de binários executáveis;

em que a informação de configuração (1026) indica, para cada um do pelo menos um plugin designado (1034), um respectivo subconjunto da respectiva pluralidade de binários executáveis; e

em que o método ainda compreende:

utilizar o plugin agente (532, 1032) para executar cada subconjunto com base na informação de configuração (1026).

3. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** identificar a informação de configuração (1026) compreende:

identificar um arquivo de configuração predeterminado que indica pelo menos uma definição de configuração para cada um dos pelo menos um plugin designado (1034); e

em que o método ainda compreende:

utilizar o plugin agente (532, 1032) para executar o pelo menos um plugin designado (1034) com base no arquivo de configuração predeterminado em resposta a utilizar o plugin agente (532, 1032) para instalar pelo menos um plugin designado (1034).

4. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** identificar a informação de configuração (1026) compreende:

solicitar um arquivo de configuração que corresponde à

primeira implementação do aplicativo especificado (528, 1030) a partir de um aplicativo da web através do qual o arquivo de configuração é gerado; e

receber o arquivo de configuração a partir do aplicativo da web, o arquivo de configuração indicando pelo menos um parâmetro de configuração para cada um dos pelo menos um plugin designado (1034); e

em que o método ainda compreende:

utilizar o plugin agente (532, 1032) para executar pelo menos um plugin designado (1034) baseado no arquivo de configuração recebido em resposta a utilizar o plugin agente (532, 1032) para instalar o pelo menos um plugin designado (1034).

5. Sistema compreendendo um módulo de instalação de plugin dinâmico (108, 200, 300, 400, 500), em que uma instância do módulo de instalação de plugin dinâmico (108, 200, 300, 400, 500) é incluída em cada servidor que compreende um aplicativo especificado (528, 1030), o módulo de instalação de plugin dinâmico (108, 200, 300, 400, 500) compreendendo:

um módulo de identificação (1002) configurado para identificar uma informação de configuração (1026) com relação a um aplicativo especificado (528, 1030), a informação de configuração (1026) indicando que o pelo menos um plugin designado (1034) selecionado a partir de um ou mais plugins (1022), os quais são armazenados em uma localização que é acessível para aplicativos implementados, deve ser instalado em resposta ao aplicativo especificado (528, 1030) sendo implementado ao longo de múltiplas máquinas (106A, 106B, 106N) em um ambiente de nuvem, em que o módulo de instalação de plugin dinâmico (108, 200, 300, 400, 500) é configurado para identificar uma informação de localização com

relação à localização do pelo menos um plugin designado (1034);

um módulo de causação (1004), configurado para fazer com que um plugin agente (532, 1032) execute em uma primeira implementação do aplicativo especificado (528, 1030), o aplicativo especificado (528, 1030) sendo empacotado para incluir o plugin agente (532, 1032), **caracterizado pelo fato de que** o plugin agente (532, 1032) determina que máquinas (106A, 106B, 106N) constituintes que estão associadas com o aplicativo especificado (528, 1030) são incluídas, em que o plugin agente (532, 1032) fornece uma capacidade de entender de maneira dinâmica uma topologia de uma rede em que o aplicativo especificado (528, 1030) é instalado; e

um módulo de instalação (1006) configurado para utilizar o plugin agente (532, 1032) para instalar o pelo menos um plugin designado (1034) dinamicamente para localizações relevantes com base nas informações de configuração em resposta ao plugin agente (532, 1032) sendo levado a executar na primeira implantação do aplicativo especificado (528, 1030), em que o plugin agente (532, 1032) é configurado para implementar outros plugins, dependências destes plugins, e dependências do aplicativo especificado (528, 1030) para as localizações relevantes.

6. Sistema, de acordo com a reivindicação 5, **caracterizado pelo fato de que** a informação de configuração (1026) indica ainda uma ou mais dependências do aplicativo especificado (528, 1030); e

em que o módulo de instalação (1006) é ainda configurado para usar o plugin agente (532, 1032) para instalar as uma ou mais dependências com base na informação de configuração (1026).

7. Sistema, de acordo com a reivindicação 6, **caracterizado pelo fato de que** a informação de configuração (1026) indica ainda pelo menos uma dependência de pelo menos um plugin designado (1034); e

em que o módulo de instalação (1006) é ainda configurado para usar o plugin agente (532, 1032) para instalar a pelo menos uma dependência do pelo menos um plugin designado (1034) com base na informação de configuração (1026).

8. Sistema, de acordo com a reivindicação 5, **caracterizado pelo fato de que** o módulo de identificação (1002) é configurado para identificar uma alimentação que inclui pelo menos um plugin designado (1034), informações de localização em relação a uma localização de pelo menos um plugin designado (1034), e informações de instalação que incluem instruções para instalar pelo menos um plugin designado (1034); e

em que o módulo de instalação (1006) é configurado para usar o plugin agente (532, 1032) para instalar pelo menos um plugin designado (1034) dinamicamente com base nas informações de localização e nas informações de instalação.

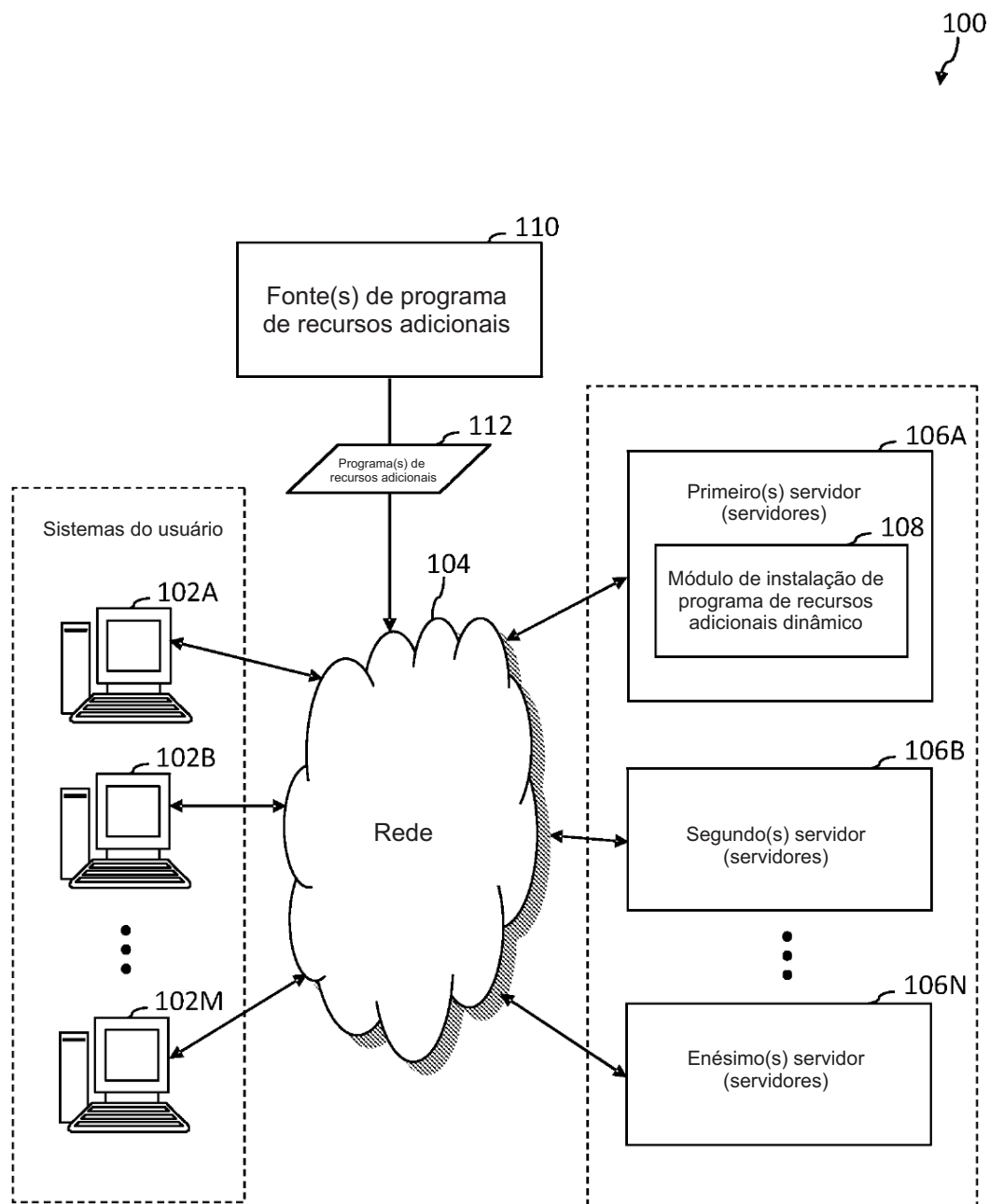


FIG. 1

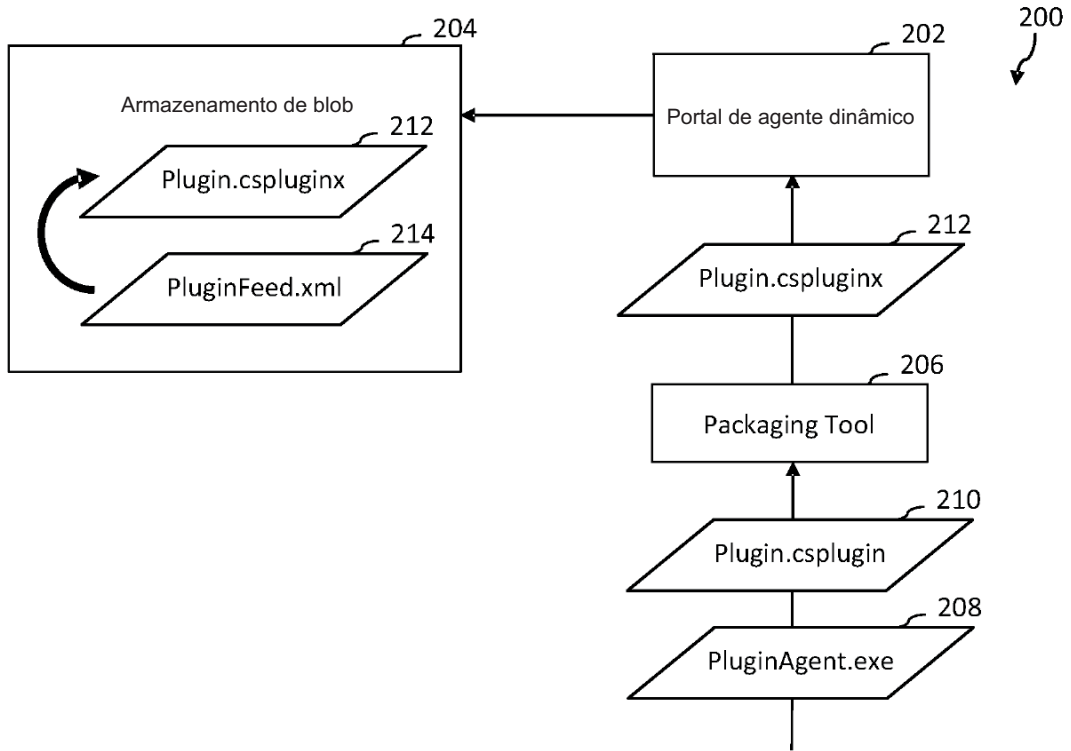


FIG. 2

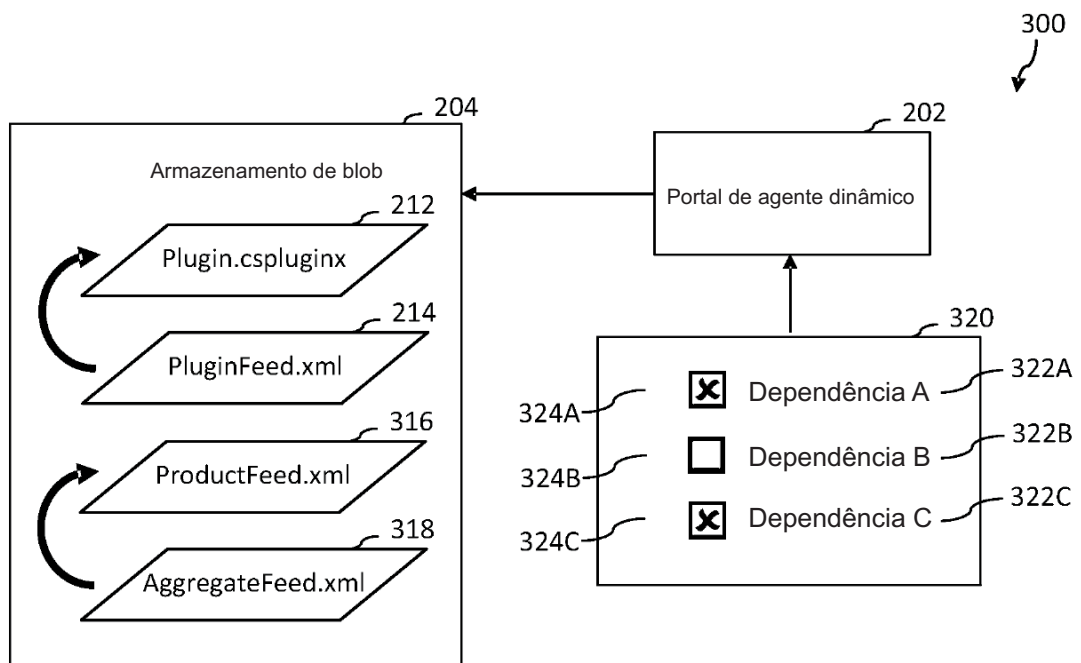


FIG. 3

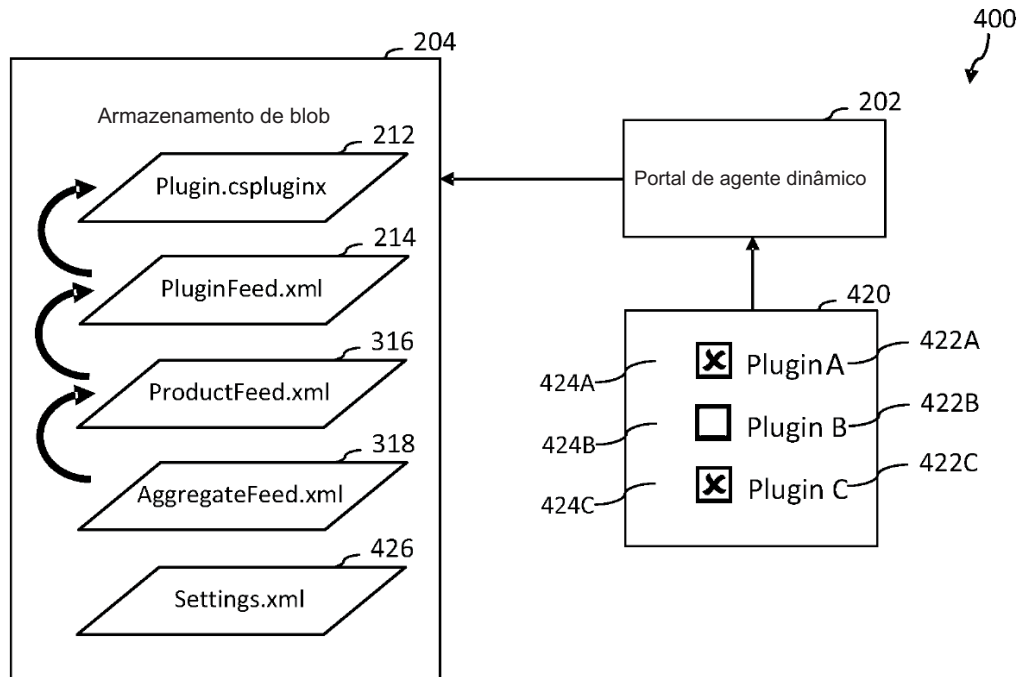


FIG. 4

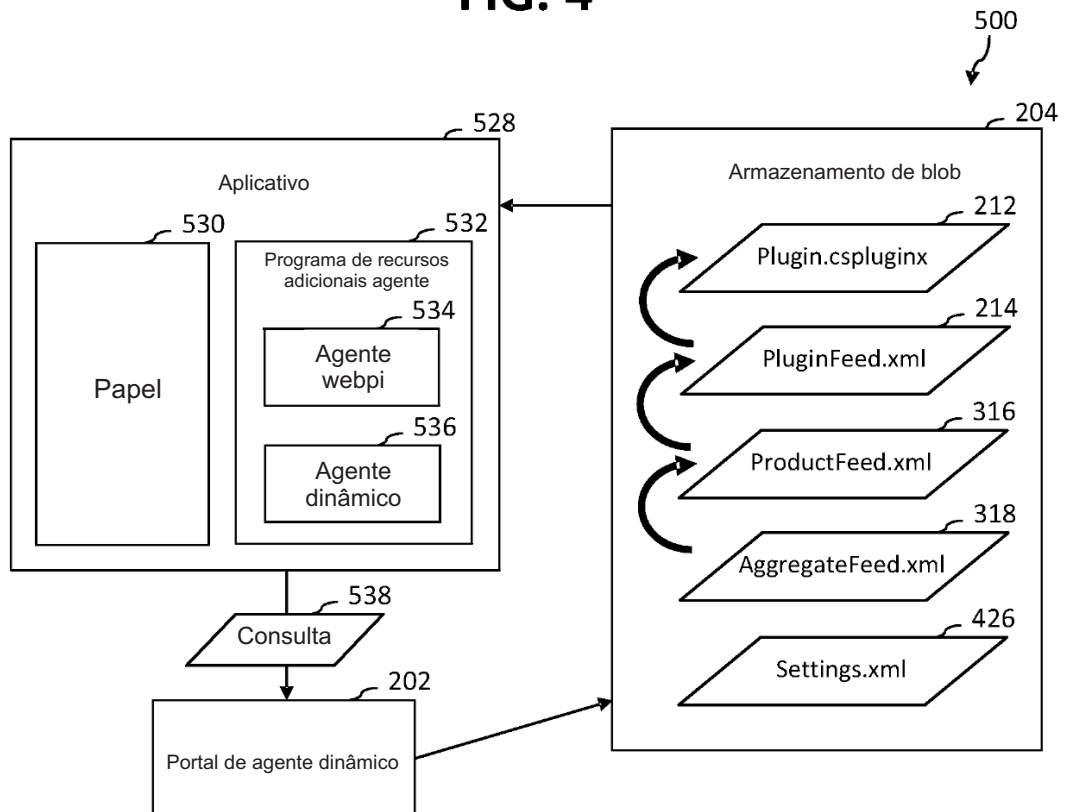


FIG. 5

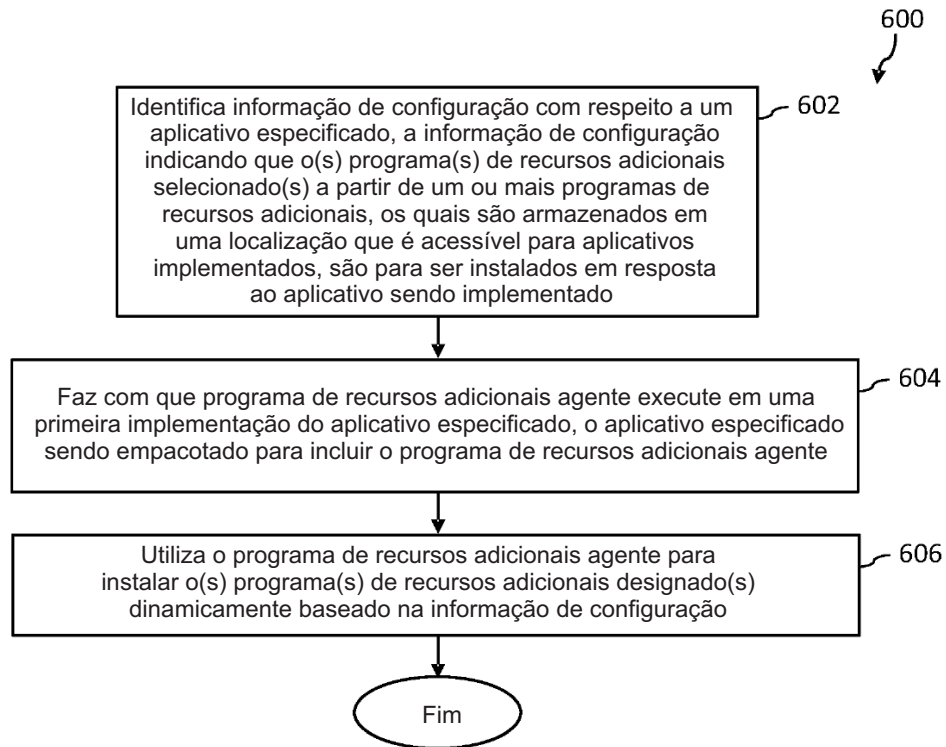


FIG. 6

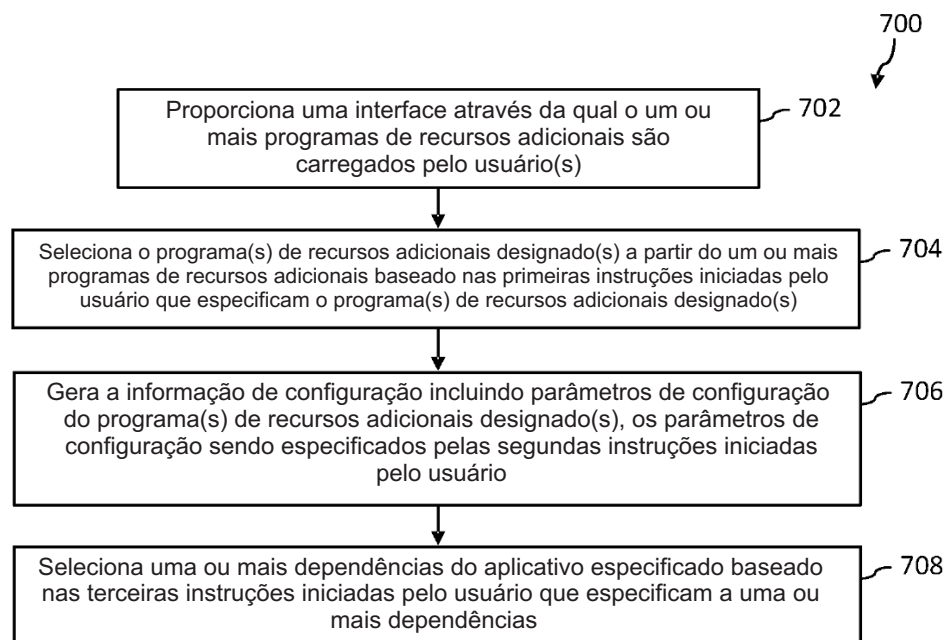
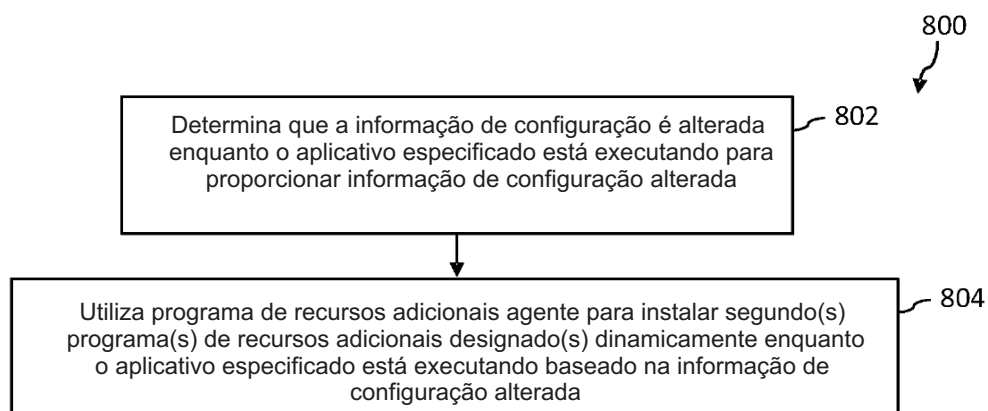
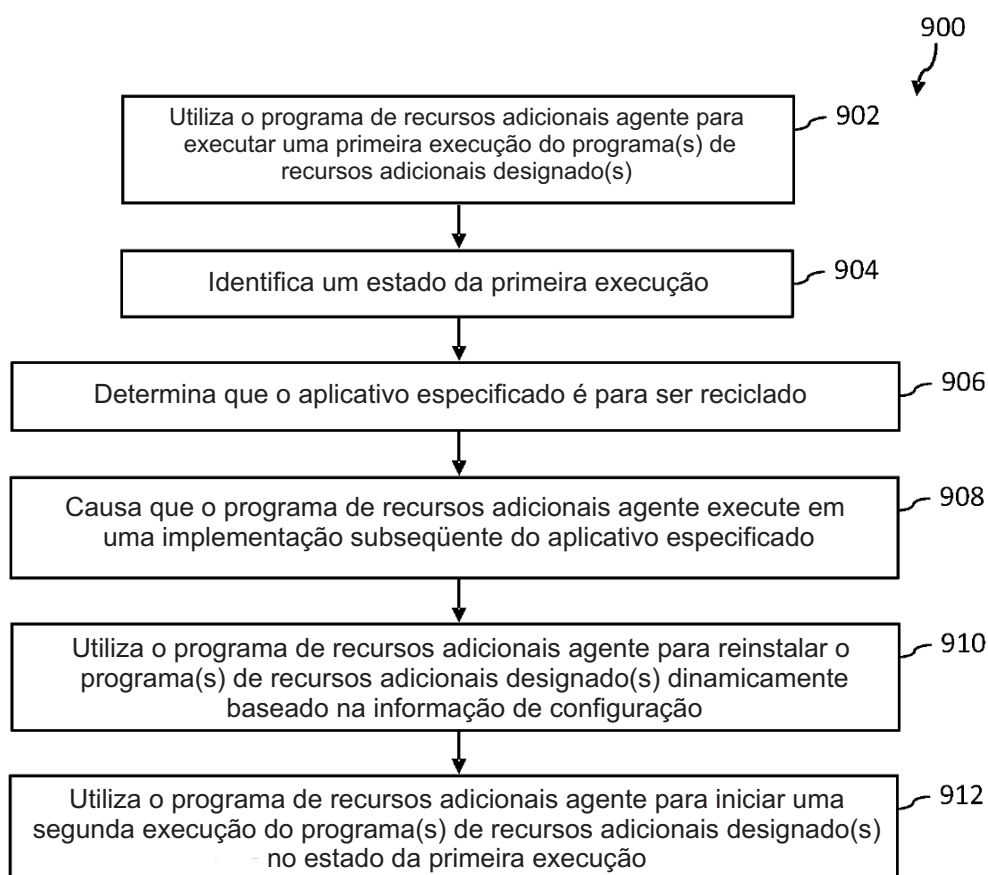


FIG. 7

**FIG. 8****FIG. 9**

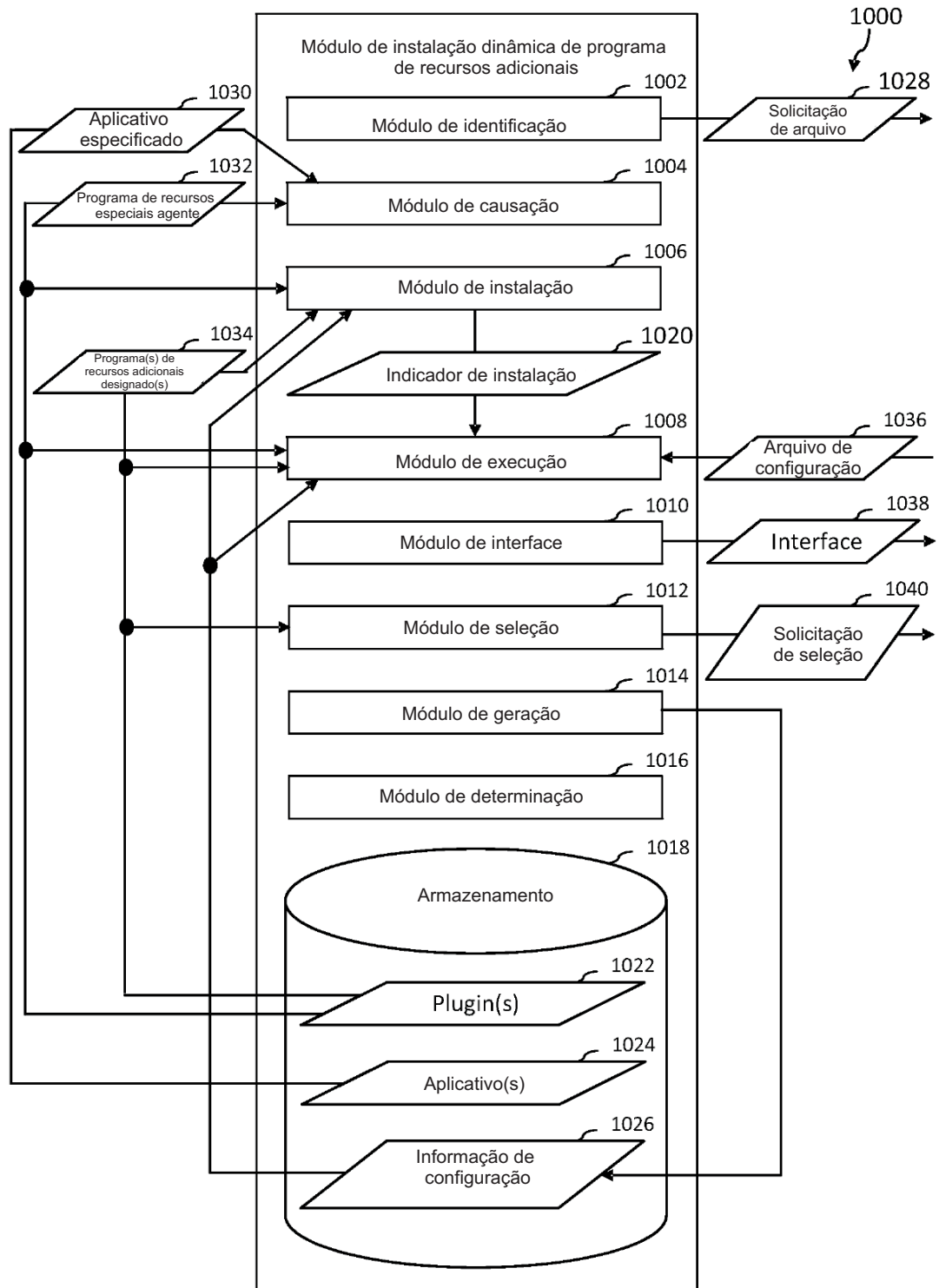


FIG. 10

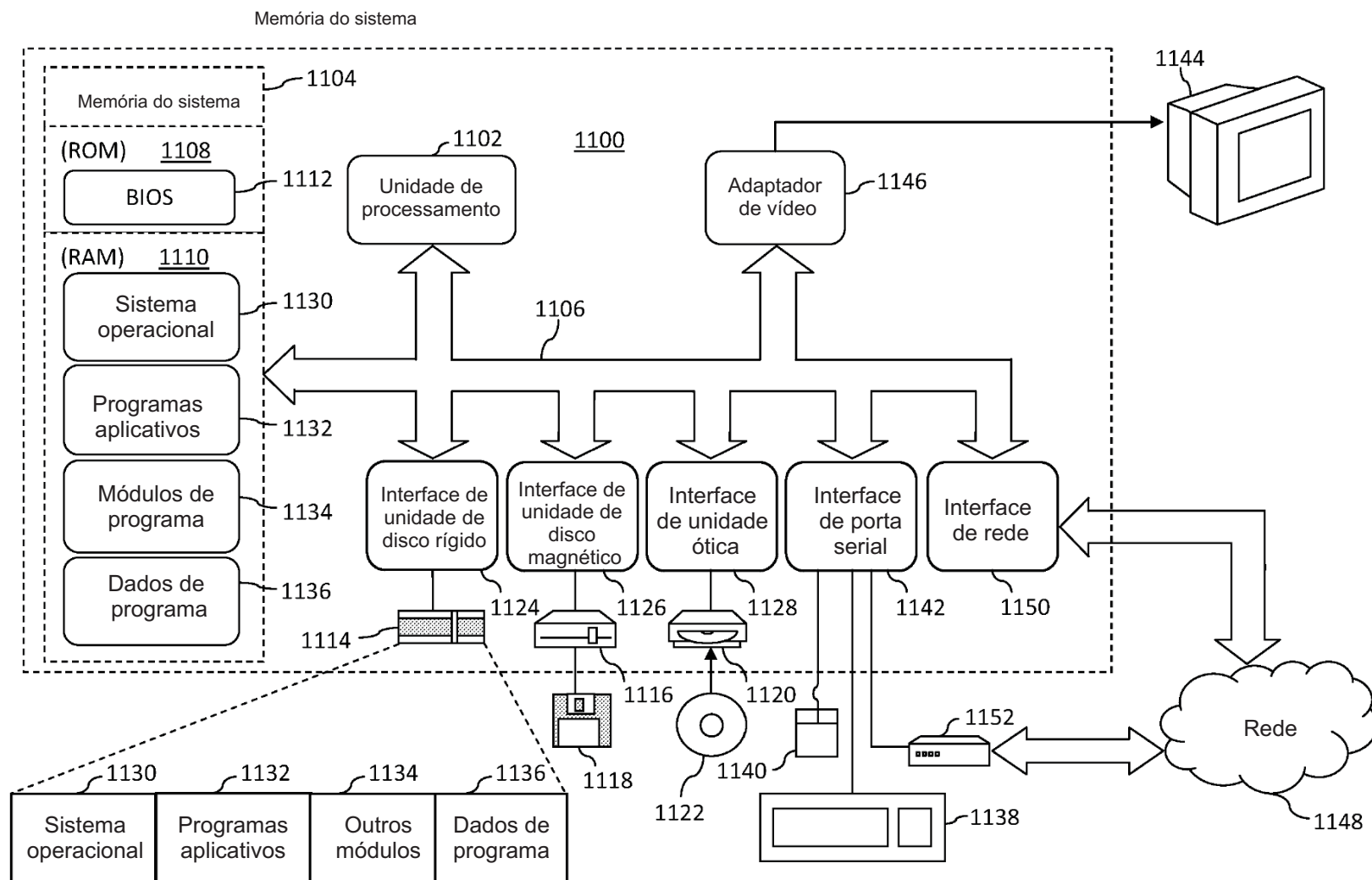


FIG. 11