

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 972 147**

51 Int. Cl.:

G06F 9/451 (2008.01)

G06F 3/048 (2013.01)

G06Q 10/10 (2013.01)

G06Q 30/0601 (2013.01)

G06N 7/01 (2013.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **12.03.2019** **PCT/US2019/021874**

87 Fecha y número de publicación internacional: **19.09.2019** **WO19178115**

96 Fecha de presentación y número de la solicitud europea: **12.03.2019** **E 19767139 (9)**

97 Fecha y número de publicación de la concesión europea: **22.11.2023** **EP 3765992**

54 Título: **Sistema y procedimiento para generar GUIs basadas en predicción para mejorar los tiempos de respuesta de la GUI**

30 Prioridad:

13.03.2018 US 201815920319

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
11.06.2024

73 Titular/es:

CLOUDBLUE LLC (100.0%)
3351 Michelson Drive, Suite 100
Irvine, CA 92612, US

72 Inventor/es:

KORYAKIN, ROSTISLAV

74 Agente/Representante:

GONZÁLEZ PECES, Gustavo Adolfo

ES 2 972 147 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Sistema y procedimiento para generar GUIs basadas en predicción para mejorar los tiempos de respuesta de la GUI

Campo técnico

5 La presente divulgación se refiere a un sistema y procedimiento para la creación dinámica de interfaces de navegación de usuario en base a la predicción, para mejorar los tiempos de respuesta de la interfaz de navegación

Antecedentes

Una interfaz gráfica de usuario (GUI) o, alternativamente denominada una interfaz de navegación de usuario, es una característica común de los sistemas informáticos y una aplicación bien conocida por cualquiera que tenga una habilidad ordinaria en las técnicas. Las GUIs tradicionales presentan al usuario una interfaz de navegación frontal que se comunica con un back-end para extraer datos e información, o realizar acciones. En tales realizaciones tradicionales, se puede considerar que las GUIs proporcionan un contexto particular que incluye elementos de navegación o control tal como, un botón cliqueable, un enlace, un control deslizante, o similares. El propósito de tales elementos de navegación o control es permitir que un usuario opere la GUI para realizar una acción deseada. Al operar la GUI utilizando un elemento de navegación o control, la GUI genera una acción de respuesta la cual requiere que la GUI se comunique con el back-end. Por ejemplo, si el usuario desea recuperar información a través de una GUI, la GUI puede incluir un elemento de control tal como, un elemento de control "ABRIR" el cual, cuando se opera, inicia una acción de respuesta que requiere que la GUI recupere datos a partir del back-end conectado operacionalmente y visualice tales datos al usuario a través GUI. En tales realizaciones tradicionales, la acción de respuesta depende de la operación particular realizada por el usuario (es decir, si el usuario no opera el elemento de control "ABRIR", es posible que la GUI no tenga que recuperar datos del back-end como acción de respuesta).

Además de recuperar datos del back-end, también puede ser necesario cambiar el contexto GUI. Por ejemplo, es posible que sea necesario cambiar los elementos de control dentro GUI, por lo que los elementos de control GUI visualizados anteriormente ya no son relevantes. Continuando con el ejemplo anterior, después de que el usuario opere el elemento de control "ABRIR" y la GUI recupere los datos a partir del back-end y los visualice, es posible que el elemento de control "ABRIR" ya no sea relevante ya que la información ya se ha recuperado. En un tal ejemplo, el elemento de control "ABRIR" puede sustituirse por un elemento de control "CERRAR" que permita al usuario cerrar o finalizar de otro modo una operación dentro de la GUI.

En una tal realización tradicional, el estado futuro de la GUI (es decir, el cambio a un nuevo elemento de control) depende de las acciones del usuario en el estado actual de la GUI. Es decir, cuando se realizan acciones en la GUI en el estado actual, la GUI podrá transicionar a un estado nuevo/futuro el cual puede incluir la comunicación con el back-end para realizar la acción y, en consecuencia, presentar al usuario una respuesta. En tales realizaciones, la respuesta del back-end puede retrasarse o de otro modo incluir latencia, por lo que la transición de la GUI del estado actual al estado nuevo/futuro también puede retrasarse. En tales casos, la experiencia del usuario no es la ideal, ya que tiene que esperar mientras la GUI espera una respuesta del back-end. Los tiempos de espera o latencia excesivos son comunes, especialmente en situaciones en las que el back-end está sometido a una alta carga operativa (es decir, la GUI tarda más en responder a una acción del usuario ya que no ha recibido una respuesta del back-end y está esperando). El tiempo de espera adicional conduce inevitablemente a una mala experiencia de usuario, ya que ese último se retrasa en la realización de su operación o transacción.

La publicación EP2221719A1 divulga un procedimiento de predicción de la siguiente etapa para ayudar al usuario a operar el teléfono móvil, lo cual implica determinar la siguiente pantalla de la pantalla actualmente visualizada de acuerdo con las probabilidades de transferencia en el modelo de predicción del usuario instalado en dispositivos específicos.

La publicación US2016/360382A1 divulga sistemas y procedimientos para identificar proactivamente y mostrar contenido relevante en un dispositivo sensible al tacto.

45 Además, las consideraciones de seguridad añaden una sobrecarga adicional en la gestión de la transición de la GUI. Por ejemplo, un usuario que intenta operar una GUI también puede hacerlo dentro de los límites de su autorización (es decir, un usuario no puede realizar una acción para la que no está autorizado). Por lo tanto, la factorización en los controles de seguridad añade tiempo de procesamiento adicional por lo que la latencia se incrementa aún más.

50 En entornos de intermediario de servicios en la nube (CSB), la representación de GUI puede ser más complicada. El funcionamiento de los CSB se divulga en la Solicitud de Patente de Utilidad de los Estados Unidos número 15/142,256 para MÉTODO Y NORMA PARA INTEGRAR APLICACIONES EN UNA NUBE. Un estándar de empaquetado de aplicaciones (APS) es un procedimiento para que los proveedores independientes de software (ISV) distribuyan aplicaciones en la nube (servicios en la nube) a través de CSB y permitan que diferentes aplicaciones en

la nube colaboren a través de interfaces de programa de aplicación (API) estandarizadas de transferencia de estado representacional (RESTful). Ayuda a los ISVs a integrar sus aplicaciones con CSBs habilitados para APS creando un paquete de conector (realizado de acuerdo con un APS) para cada aplicación.

En algunas realizaciones tradicionales, el paquete de conector puede incluir un archivo que contenga metadatos, descripciones de procedimientos de control y archivos de contenido, los cuales son utilizables para declarar y definir los recursos de la aplicación, los servicios, los componentes de la interfaz de usuario, y la lógica de los procedimientos de control necesarios para gestionar los recursos (es decir, un esquema de recursos el cual puede crearse que es aplicable a cada aplicación individual). El paquete de conector incluye además información sobre la GUI de una aplicación, tal como, por ejemplo, el diseño de pantalla, los controles de navegación, los formularios, y similares. El paquete de conector puede incluir además archivos declarativos utilizados para definir los recursos de aplicación del ISV y del "conector frontal", es decir, la interfaz web que ven los clientes del CSB cuando acceden a la interfaz del CSB. En tales entornos CSB, los desarrolladores de paquetes de conector desarrollan una interfaz web utilizando infraestructuras específicas, las cuales pueden incluir widgets, una estructura de página predefinida, procedimientos de navegación y un mecanismo de inserción en pantalla. Por ejemplo, un conector frontal típico consiste en pantallas insertadas en vistas predefinidas, lógica personalizada para renderizar widgets y estructura de navegación personalizada. Los widgets pueden visualizar información que el desarrollador del conector considere valiosa para el cliente de la aplicación en nube correspondiente, por ejemplo, informes de uso, información de suscripción y servicio, instrucciones, etc.

Esta integración permite la gestión de aplicaciones integradas en la nube a través de un único panel web con un estilo común y una colección de controles (es decir, una GUI integrada). Por ejemplo, Dropbox® puede ser una de las aplicaciones disponibles de un ISV. Cuando un usuario accede a Dropbox®, la aplicación del conector de Dropbox® facilita la actividad del usuario presentando las pantallas/interfases GUI adecuadas (es decir, la aplicación del conector de Dropbox® gestiona la actividad GUI del usuario). Continuando con este ejemplo, si hay múltiples aplicaciones integradas, la GUI puede incluir numerosas interfaces específicamente adaptadas a cada una de las aplicaciones específicas. En este ejemplo, deben determinarse todas las pantallas posibles para que la integración entre las numerosas aplicaciones se produzca sin interrupciones (tal y como las ve el usuario). Sin embargo, preparar todas las combinaciones de pantallas requiere un amplio desarrollo, el cual aumenta a medida que se integran más y más aplicaciones. Por otra parte, la obtención de pantallas de GUI en tiempo real (es decir, cuando un usuario está utilizando la aplicación) genera lentitud ya que el entorno CSB puede tener que calcular la pantalla de GUI adecuada para renderizar, a partir de un gran número de posibles combinaciones de pantallas de GUI.

Por lo tanto, existe la necesidad de un sistema y procedimiento para generar GUIs en base a predicciones para mejorar los tiempos de respuesta de la GUI.

Sumario

La invención se define mediante el conjunto de reivindicaciones adjuntas.

La presente divulgación se refiere a un sistema y procedimiento para la creación dinámica de interfaces de navegación de usuario en base a la predicción, para mejorar los tiempos de respuesta de la interfaz de navegación. En al menos una realización, se proporciona un procedimiento para generar pantallas de interfaz gráfica de usuario (GUI) basadas en predicción para mejorar los tiempos de respuesta de la pantalla de GUI. El procedimiento incluye recibir en un punto final, una solicitud de vista para una pantalla de GUI, comprendiendo la pantalla de GUI una interfaz de usuario integrada construida a partir de una pluralidad de paquetes de conector, validar en un predictor, una validez de un modelo de transición de GUI, el modelo de transición de GUI correspondiente a cada uno de la pluralidad de paquetes de conector, en el que la validación del modelo de transición de GUI comprende comprobar una fecha de ámbito de seguridad modificad por última vez, construir en un constructor de modelo de comportamiento, un nuevo modelo de transición de GUI, comprendiendo el nuevo modelo de transición de GUI una pluralidad de probabilidades de transición para cada transición posible de la pantalla de GUI, la pluralidad de probabilidades de transición depende al menos en parte de los atributos de recursos disponibles para cada uno de la pluralidad de paquetes de conector, seleccionando en el constructor del modelo de comportamiento, una transición probable de la pluralidad de probabilidades de transición, siendo la transición probable una transición a una nueva pantalla de GUI que es más probable que sea solicitada, creando en un módulo de navegación, un árbol de navegación en base al menos en parte a la transición probable y regresando una respuesta al punto final, comprendiendo la respuesta el árbol de navegación, y comprendiendo además una respuesta de navegación para la nueva pantalla de GUI.

En al menos una realización de la presente divulgación, el procedimiento incluye modelos de transición seleccionados a partir de un grupo que consiste en un modelo Ngram, un modelo Markov oculto, y un modelo Markov de máxima entropía.

En al menos una realización de la presente divulgación, el procedimiento incluye los modelos de transición que incluyen además clasificadores potenciados seleccionados a partir de un grupo que consiste en una máquina de vectores de soporte (SVM), árbol de decisión, bosque aleatorio, y vecinos K más cercanos.

5 En al menos una realización de la presente divulgación, el procedimiento incluye una solicitud para la pantalla de GUI que se recibe a partir de un cliente, y un modelo de transición de GUI que incluye una fecha de modificación del ámbito de seguridad.

10 En al menos una realización de la presente divulgación, el procedimiento incluye comprobar en el predictor, si hay datos suficientes para volver a calcular la pluralidad de probabilidades de transición para cada transición posible de la pantalla de GUI, volver a calcular en el predictor, la pluralidad de probabilidades de transición para cada transición posible de la pantalla de GUI, y crear un nuevo modelo de transición de GUI, en base al menos en parte a los datos, estableciendo en el predictor, una nueva fecha de creación del modelo de transición de GUI, y operando el rellenedor de caché, el rellenedor de caché configurado para sustituir el modelo de transición de GUI original en una caché con el nuevo modelo de transición de GUI.

15 En al menos una realización de la presente divulgación, los datos incluyen transiciones de pantalla de GUI y clasificadores de refuerzo. Las probabilidades de transición incluyen además el cálculo de un conjunto de probabilidades, en base al menos en parte a los clasificadores de refuerzo.

20 En al menos una realización de la presente divulgación, el procedimiento incluye recibir en un punto final REST, una solicitud de navegación para una primera pantalla de GUI, comprobar en un predictor, una fecha de modificación del ámbito de seguridad, comprendiendo la fecha de modificación del ámbito de seguridad un conjunto de recursos disponibles para un usuario para el cual se está solicitando la solicitud de navegación para la primera pantalla de GUI, recuperar en el predictor, un modelo de transición de GUI, identificando el modelo de transición de GUI una nueva pantalla de GUI para la transición desde la primera pantalla de GUI, comprobando en el predictor, si el modelo de probabilidad de transición es más reciente que la fecha de modificación del ámbito de seguridad, y recuperando en el predictor, una respuesta, comprendiendo la respuesta un árbol de navegación con un enlace al código de fuente de la nueva pantalla de GUI y el estado actual de los recursos de un usuario necesarios para ejecutar el código de fuente de pantalla de GUI.

25 En al menos una realización de la presente divulgación, el árbol de navegación es un gráfico de navegación que muestra transiciones GUI. En al menos una realización de la presente divulgación, el procedimiento incluye comprobar el predictor incluido en la etapa de limpiar el modelo de transición de GUI y solicitar la creación de un nuevo modelo de transición de GUI, si el modelo de probabilidad de transición es más antiguo.

30 En al menos una realización de la presente divulgación, el sistema incluye una plataforma de intermediario de servicios en la nube que incluye un predictor, en la que el predictor está configurado para validar el modelo de transición de GUI comprobando una última fecha del ámbito de seguridad modificada, una base de datos, un constructor de modelo de comportamiento configurado para crear un modelo de transición de GUI, un rellenedor de caché configurado para recuperar las vistas GUI más probables que serán visitadas por un usuario, en base al menos en parte al modelo de transición de GUI, un módulo de navegación configurado para crear respuestas de navegación, un limpiador de caché configurado para limpiar una caché de respuesta, y un punto final REST configurado para simular un punto final de servicio de navegación GUI.

35 En al menos una realización de la presente divulgación, el sistema incluye un cliente conectado operativamente a la plataforma de intermediario de servicios en la nube, y configurado para solicitar una nueva pantalla de GUI y recibir una respuesta de navegación para la nueva pantalla de GUI.

40 En al menos una realización de la presente divulgación, el controlador del intermediario de servicios en la nube comprende una pluralidad de aplicaciones de conector, cada una de las cuales está configurada para cada aplicación para su integración a través de la plataforma del intermediario de servicios en la nube y en la que los paquetes de conector incluyen módulos de interfaz de usuario para construir elementos de navegación, y un ámbito de seguridad para cada uno de los elementos de navegación.

Breve descripción de los dibujos

45 Las realizaciones y otras características, ventajas, y divulgaciones contenidas en la presente memoria, así como la forma de obtenerlas, se harán evidentes y la presente divulgación se entenderá mejor mediante referencia a la siguiente descripción de diversas realizaciones ejemplares de la presente divulgación tomadas en conjunto con los dibujos adjuntos, en los que:

La Figura 1A visualiza un diagrama de estado ejemplar de un trayecto de navegación con todas las transiciones de navegación posibles.

La Figura 1B visualiza un diagrama de estado ejemplar de un trayecto de navegación simple.

La Figura 1C visualiza un árbol de navegación ejemplar ensamblado mediante marcadores de posición (técnica anterior).

5 La Figura 2 visualiza un dibujo esquemático de un sistema para generar GUIs basadas en predicción para mejorar los tiempos de respuesta de la GUI.

La Figura 3 visualiza un dibujo esquemático de un sistema para generar GUIs basadas en predicción para mejorar los tiempos de respuesta de la GUI.

La Figura 4 visualiza un dibujo esquemático de un procedimiento para generar GUIs basadas en predicción para mejorar los tiempos de respuesta de la GUI.

10 La Figura 5 visualiza un dibujo esquemático de un procedimiento para generar GUIs basadas en predicción para mejorar los tiempos de respuesta de las GUIs.

La Figura 6 visualiza un dibujo esquemático de un procedimiento para generar GUIs basadas en predicción para mejorar los tiempos de respuesta de la GUI.

Descripción detallada

15 Con el propósito de promover la comprensión de los principios de la presente divulgación, se hará referencia ahora a las realizaciones ilustradas en los dibujos, y se utilizará un lenguaje específico para describirlas. No obstante, se entenderá que no se pretende limitar el ámbito de la presente divulgación.

La presente descripción detallada se presenta en términos de programas, estructuras de datos o procedimientos ejecutados en un ordenador o red de ordenadores. Los programas de software implementados por el sistema pueden
20 estar escritos en cualquier lenguaje de programación, ya sea interpretado, compilado, o de otro tipo. Estos lenguajes pueden incluir, pero no limitarse a, Xcode, iOS, cocoa, cocoa touch, MacRuby, PHP, ASP.net, HTML, HTML5, Ruby, Perl, Java, Python, C++, C#, JavaScript, y/o el lenguaje de programación Go. Debe apreciarse, por supuesto, que un experto en la técnica apreciará que otros lenguajes pueden ser utilizados en lugar de, o en combinación con lo anterior y que la web y / o infraestructuras de aplicaciones móviles también pueden ser utilizadas, tales como, por ejemplo,
25 Ruby on Rails, Sistema.js, Zend, Symfony, Revel, Django, Struts, Spring, Play, Jo, Twitter Bootstrap y otros. Asimismo, se debe tener en cuenta que los sistemas y procedimientos descritos en la presente memoria pueden estar incorporados en un software como servicio disponible a través de una red informática, tal como, por ejemplo, Internet. Además, la presente divulgación puede permitir servicios web, interfaces de programación de aplicaciones y/o arquitectura orientada a servicios a través de una o más interfaces de programación de aplicaciones o de otro modo.

30 De nuevo con referencia a la Figura 1A, se muestra un diagrama de estados ejemplar de un trayecto de navegación con todas las posibles transiciones de navegación, generalmente indicado en 100. El diagrama 100 de estado incluye una pluralidad de estados GUI (es decir, 102, 104, 106, 108, 110, y 112). En al menos una realización de la presente divulgación, cada estado GUI es indicativo del estado GUI actual. Por ejemplo, cuando un usuario utiliza la GUI por primera vez, la GUI puede inicializarse en el estado 102 GUI (Inicio) y la GUI puede presentar información de
35 acompañamiento o elementos de control en el mismo, apropiados para el estado 102 GUI. Para cada uno de los estados de la GUI, hay al menos una transición a otro estado GUI. A modo de ejemplos, el estado 102 GUI (Home) tiene sólo una transición 102A al estado 104 GUI (Lista de Usuario); el estado 112 GUI (Asignar Servicio) incluye la transición 112A al estado 102 GUI, y la transición 112B al estado 104 GUI. Se apreciará que una transición de estado GUI es indicativa de una acción u operación (por ejemplo, realizada por un usuario mediante la operación de un elemento de control GUI), que transiciona la GUI desde el estado actual, al estado siguiente/futuro. Continuando con el ejemplo anterior, si una GUI se inicializa en el estado 102 GUI, la(s) operación(es) disponible(s) dentro del estado 102 GUI invocará(n) una transición 102A de modo que la GUI transiciona al estado 104 GUI. De manera similar, si el estado GUI se encuentra actualmente en el estado 112 GUI, se le pueden presentar al usuario funciones de GUI que
40 pueden seleccionarse operativamente para permitir que la GUI pase al estado 102 GUI (a través de la transición 112A), o al estado 104 GUI (a través de la transición 112B). Se apreciará además que el diagrama de estado visualiza todas las posibles transiciones de navegación dado un estado actual GUI. Se apreciará que cada uno de los estados de la GUI puede ser creado independientemente por un ISV para sus respectivas aplicaciones.

Con referencia ahora a la Figura 1B, se muestra un diagrama 150 de estado ejemplar, de un trayecto de navegación simple, de acuerdo con al menos una realización de la presente divulgación. En el diagrama 150 de estado, el estado
50 102 GUI es el estado inicial (es decir, un usuario se inicializa para ver la GUI tal y como se presenta en el estado 102 GUI). A modo de ejemplo, tras la operación de la GUI, la GUI transiciona al estado 104 GUI a través de la transición 102A; en el estado 104 GUI, la GUI transiciona al estado 110 GUI a través de la transición 104A; en el estado 110 GUI, la GUI transiciona al estado 112 GUI a través de la transición 11A; en el estado 112 GUI, la GUI transiciona

mediante la operación al estado 108 GUI, a través de la transición 112B; en el estado 108 GUI, la GUI transiciona mediante la operación al estado 106 GUI, a través de la transición 108A; y finalmente, en el estado 106 GUI, la GUI transiciona al estado 104 GUI, a través de la transición 106A.

Continuando con la presente realización ejemplar, se apreciará que el diagrama 150 de estado muestra un trayecto de navegación por el que un usuario opera a lo largo de este trayecto de navegación para lograr una tarea específica. Por ejemplo, con el fin de aprovisionar a un usuario en una plataforma de intermediario de servicios en la nube (CSB) (como se divulga además en la presente memoria), se inicializará a un administrador para que comience en una pantalla de Inicio (estado 102 GUI), cuya pantalla de Inicio puede ser el punto de partida disponible en un panel de control de la plataforma CSB. En la pantalla de Inicio (estado 102 GUI), se puede presentar al administrador un menú que incluya al menos un elemento de control, tal como un enlace a una lista de usuario. Al operar el elemento de control (es decir, al hacer clic en el enlace a la lista de usuario), la GUI transiciona a la pantalla Lista de usuario (estado 104 GUI). Se apreciará que en cada pantalla de GUI (es decir, estado GUI), se puede proporcionar una lista de elementos de control disponibles. Se apreciará que los elementos de control permiten al administrador navegar de una pantalla de GUI a la siguiente. Se apreciará además que hay dos tipos de transiciones de navegación: dinámicas y estáticas. En al menos una realización de la presente divulgación, una transición estática es una transición la cual se declara en la descripción de navegación de la pantalla de GUI. Es decir, las transiciones estáticas están predefinidas para transicionar de una pantalla de GUI a la siguiente pantalla de GUI. Se apreciará que esto crea una relación "rígida" entre las dos pantallas de GUI.

Continuando con el ejemplo anterior, en la pantalla de Lista de Usuario (estado 104 GUI), el administrador puede ser proporcionado nuevamente con un menú que incluye elementos de control, cuyos elementos de control pueden incluir un enlace para poder agregar un nuevo usuario. Al operar el enlace para agregar un nuevo usuario, la GUI transiciona a la pantalla de Nuevo Usuario (estado 110 GUI) mediante la cual se permite al administrador agregar un nuevo usuario a un sistema. En la pantalla de Nuevo Usuario (estado 110 GUI) el administrador puede operar la GUI para agregar un nuevo usuario, y también se le puede proporcionar un menú que incluya elementos de control para transicionar a la siguiente pantalla GUI (es decir, Asignar Servicio (estado 112 GUI)). Continuando con el ejemplo anterior, el administrador puede operar el elemento de control para permitir la asignación de servicios de la plataforma CSB al usuario en la pantalla Asignar Servicio (estado 112 GUI). Una vez completada esa asignación, el administrador puede operar la GUI para transicionar a la pantalla del nuevo servidor privado virtual (VPS) (cuyo servicio VPS se asigna en la etapa 112) (estado 108 GUI), tras lo cual el administrador puede transicionar a la Vista de Usuario (estado 106 GUI), y finalmente regresar a la pantalla de Lista de usuario (estado 104 GUI). Se apreciará que las transiciones de navegación anteriores son una realización ejemplar del desempeño de una acción por parte de un administrador en la plataforma CSB.

Con referencia ahora a la Figura 1C, se muestra un árbol de navegación ejemplar ensamblado a través de marcadores de posición, generalmente indicado en 175. En al menos una realización de la presente divulgación, el árbol 175 de navegación incluye la aplicación 180 y la aplicación 182. Se apreciará que las aplicaciones 180 y 182 pueden ser cualquiera de las aplicaciones disponibles a través de la plataforma CSB. La aplicación 180 incluye además la raíz 184 de navegación y el elemento 186 de menú. El elemento 186 de menú incluye además la vista 188 y la vista 190 derecha. En al menos una realización de la presente divulgación, se utilizan marcadores de posición para integrar la GUI de la aplicación 180 y la aplicación 182. Por ejemplo, la aplicación 180 y la aplicación 182 pueden incluir cada una un paquete de conector asociado con una aplicación de conector. Para que la aplicación 182 de conector pueda conectarse operativamente a la aplicación 180, las pantallas de GUI de las aplicaciones respectivas deben estar conectadas operativamente a través de marcadores de posición. En al menos una realización de la presente divulgación, la aplicación 180 está configurada para incorporar la vista 190 derecha en el marcador 194 de posición de la aplicación 192. Se apreciará que durante la generación de pantallas de GUI para la operación de una aplicación en particular (es decir, la aplicación 180 o la aplicación 182), se construyen conexiones estáticas entre los árboles de navegación de la aplicación 180 y la aplicación 182, de modo que la vista 190 derecha tiene dos nodos principales (es decir, la vista 188 y el marcador 194 de posición 194) y de manera contraria, los dos nodos principales (la vista 188 y el marcador 194 de posición) tienen un nodo secundario común (es decir, la vista 190 derecha). Se apreciará además que tal uso de marcadores de posición puede convertir un árbol de navegación en un gráfico de navegación. También se apreciará que la vista 190 derecha también sirve como punto de integración entre la aplicación 180 y la aplicación 182. Por ejemplo, la vista 190 derecha es el punto en el cual un usuario puede transicionar de la aplicación 180 a la aplicación 182 (o viceversa, si procede). De manera similar, la vista 190 derecha también puede considerarse una bifurcación para insertar un subárbol (es decir, el de la aplicación 182) en el subárbol de la aplicación 180.

Con referencia ahora a la Figura 2, se muestra una realización ejemplar de un sistema para generar GUIs basadas en predicción para mejorar los tiempos de respuesta GUI, indicada generalmente con 200. En al menos una realización de la presente divulgación, el sistema 200 incluye un cliente 202, una plataforma 204 CSB, un predictor 206, un módulo 208 de navegación, un controlador 210 CSB, y una red 216. En al menos una realización de la presente divulgación, el sistema 200 puede incluir uno o más ordenadores de servidor, dispositivos informáticos, o sistemas de un tipo

conocido en la técnica. El sistema 200 incluye además tal software, hardware, y componentes que se le podrían ocurrir a un experto en la técnica, tal como, por ejemplo, microprocesadores, sistemas de memoria, dispositivos de entrada/salida, adaptadores de bus de anfitrión, canal de fibra, conectores de interfaz de sistemas informáticos pequeños, buses de interfaz paralelos de alto rendimiento, dispositivos de almacenamiento (por ejemplo, disco duro, unidad de estado sólido, unidades de memoria flash), controladores de dispositivos, sistemas de visualización, y similares. El sistema 200 puede incluir uno de los muchos servidores conocidos, tales como, por ejemplo, el Servidor AS/400® de IBM®, el Servidor AIX UNIX® de IBM®, o el Servidor WINDOWS NT® de MICROSOFT®.

En al menos una realización de la presente divulgación, el cliente 202 puede incluir una combinación de infraestructura de GUI y módulos específicos de aplicación de conector con navegador web para permitir al cliente 202 descargar archivos JavaScript con código específico de conector y código de Plataforma CSB para una GUI a partir del controlador 210 CSB. Se apreciará que un usuario puede utilizar el cliente 202 para visitar páginas GUI de la plataforma 204 CSB integradas con la GUI específica de la aplicación asociada con las aplicaciones 212 de conector. El cliente 202 también le da manejo a la lógica de negocio GUI, realiza solicitudes al controlador 210 CSB para obtener, eliminar o modificar cualquier recurso, y realiza solicitudes de navegación para cambiar la vista activa.

En al menos una realización de la presente divulgación, cada cliente 202 incluye una aplicación web de página única que consiste en una infraestructura JavaScript y módulos JavaScript de aplicaciones integradas. Se apreciará que el cliente 202 incluye un navegador web con una infraestructura y módulos que se ejecutan en su interior. En al menos una realización de la presente divulgación, el cliente 202 comprende módulos de UI de aplicación de conector los cuales son almacenamiento de sesión para módulos de UI específicos de conector (código GUI) de aplicaciones en la nube integradas las cuales son utilizadas por el usuario de este cliente 202 (por ejemplo, el usuario tiene suscripciones para estas aplicaciones, utiliza algunos recursos correspondientes con estas aplicaciones, etc., cualquier conexión entre el usuario y las aplicaciones en la nube que pueda requerir visualizar las vistas relacionadas con estas aplicaciones). Inicialmente, estos módulos (código GUI) específicos para aplicaciones en la nube se colocan en paquetes de conector en forma de archivos con código fuente o módulos de UI. Los metadatos de estos paquetes de conector describen la jerarquía (conexiones directas), la conexión y las variables necesarias (condiciones de seguridad) para visualizar las vistas referidas a estas piezas de código fuente.

En al menos una realización de la presente divulgación, la infraestructura del cliente 202 está conectada operativamente al controlador 210 CSB y al predictor 206. En al menos una realización de la presente divulgación, el controlador 210 CSB es llamado para realizar operaciones RESTfull con recursos, a la vez que el predictor 206 es llamado para moverse a otra vista en una pantalla de GUI. A modo de ejemplo, una infraestructura es responsable de descargar los módulos de UI específicos de la aplicación integrada a partir de la plataforma 204 CSB a través del controlador 210 CSB. Los módulos de UI específicos de la aplicación son archivos de código fuente necesarios para visualizar vistas específicas de la aplicación. Una vista puede consistir en varios módulos de UI conectados entre sí. Normalmente hay un módulo principal que representa una vista particular. Cuando el módulo de UI principal no puede operar por sí mismo, declara módulos de UI dependientes en la sección de dependencias, en base a la cual una infraestructura puede descargar módulos de UI dependientes para visualizar una vista.

En al menos una realización de la presente divulgación, la red 216 puede incluir uno de los diferentes tipos de redes, tales como, por ejemplo, Internet, intranet, red de área local (LAN), red de área amplia (WAN), una red de área metropolitana (MAN), una red telefónica (tal como la Red Telefónica Pública Conmutada), Internet, una red basada en fibra óptica (o fibra óptica), una red de televisión por cable, una red de televisión por satélite, o una combinación de redes, y similares. La red 216 puede ser una red dedicada o una red compartida. La red compartida representa una asociación de los diferentes tipos de redes que utilizan una variedad de protocolos, por ejemplo, el Protocolo de Transferencia de Hipertexto (HTTP), el Protocolo de Control de Transmisión/Protocolo de Internet (TCP/IP), el Protocolo de Aplicación Inalámbrica (WAP), y similares, para comunicarse entre sí. Se apreciará además que la red 216 puede incluir uno o más dispositivos de procesamiento de datos y/o transferencia de datos, incluyendo enrutadores, puentes, servidores, dispositivos informáticos, dispositivos de almacenamiento, un módem, un conmutador, un cortafuegos, una tarjeta de interfaz de red (NIC), un concentrador, un puente, un servidor proxy, un multiplexor óptico de adición-descarga (OADM), o algún otro tipo de dispositivo que procese y/o transfiera datos, como sería bien conocido por los expertos en la técnica. Se debe apreciar que, en diversas otras realizaciones, son posibles otras diversas configuraciones. Otras redes informáticas, tales como las redes Ethernet, las redes basadas en cable, y las redes de comunicaciones por satélite, bien conocidas por los expertos en la técnica, y/o cualquier combinación de redes se contemplan dentro del ámbito de la divulgación.

En al menos una realización de la presente divulgación, la plataforma 204 CSB incluye además un predictor 206. El predictor 206 incluye además tal software, hardware, y componentes que se le ocurrirían a un experto en la técnica, tales como, por ejemplo, microprocesadores, sistemas de memoria, dispositivos de entrada/salida, controladores de dispositivos, sistemas de visualización, y similares, los cuales colectivamente son operables para realizar las funciones asignadas al predictor 206 de acuerdo con la presente divulgación. Para propósitos de claridad, el predictor 206 se

muestra como un componente de la plataforma 204 CSB. Sin embargo, está dentro del ámbito de la presente divulgación, y será apreciado por aquellos expertos en la técnica, que el predictor 206 puede ser dispar y remoto de la plataforma 204 CSB. Se apreciará además que el servidor remoto o dispositivo informático en el cual reside el predictor 206, está conectado electrónicamente a la plataforma 204 CSB, al módulo 208 de navegación, al controlador 210 CSB, y a la red 216 de tal manera que el predictor 206 sea capaz de una transferencia de datos bidireccional continua con cada uno de los componentes del sistema 200.

En al menos una realización de la presente divulgación, la plataforma 204 CSB incluye además un controlador 210 CSB. El controlador 210 CSB incluye al menos una aplicación 212 de conector. Se apreciará que el controlador 210 CSB puede incluir una pluralidad de aplicaciones de conector.

En al menos una realización de la presente divulgación, la plataforma 204 CSB también incluye una base 214 de datos. La base 214 de datos contiene datos estáticos y dinámicos. Los datos estáticos son los metadatos del paquete de conector, tales como el nombre del paquete, la versión del paquete, el esquema de tipos de recursos, el esquema de navegación y el código fuente para todos los módulos de la UI del conector. Los datos dinámicos son un estado de todas las instancias de recursos en la plataforma 204 CSB como usuarios, dominios, direcciones de protocolo de Internet (IP) y cualquier instancia de recurso específica del conector (por ejemplo, servidor VPS para el paquete de conector VPS). Se apreciará por aquellos expertos en la técnica que la base 214 de datos puede comprender una pluralidad de bases de datos conectadas por sistemas de software de un tipo bien conocido en la técnica, los cuales colectivamente son operables para realizar las funciones delegadas a la base 214 de datos de acuerdo con la presente divulgación. La base 214 de datos también puede formar parte de una arquitectura de datos distribuidos, tales como, por ejemplo, la arquitectura Hadoop, para servicios de grandes datos. La base 214 de datos puede comprender una arquitectura de base de datos relacional, noSQL, OLAP, u otra arquitectura de base de datos de un tipo conocido en la técnica de las bases de datos. La base 214 de datos puede comprender uno de los muchos sistemas de gestión de bases de datos conocidos, tales como, por ejemplo, bases de datos SQL, bases de datos NoSQL, Servidor SQL de MICROSOFT, ACCESS de MICROSOFT, MongoDB, Redis. El Hadoop, o los sistemas de gestión de bases de datos DB2 de IBM's, o los sistemas de gestión de bases de datos disponibles de ORACLE o SYBASE, y cualquiera de los otros servicios necesarios para la aplicación que serían conocidos por un experto en las técnicas. La base 214 de datos almacena de manera recuperable información que se comunica a la base 214 de datos desde dentro de la plataforma 204 CSB.

En al menos una realización de la presente divulgación, el sistema 200 incluye un cliente 202. El cliente 202 está a distancia de la plataforma 204 CSB, y conectado a la plataforma 204 CSB a través de la red 216. El cliente 202 puede estar configurado para transmitir información e interactuar en general con la plataforma 204 CSB, a través de la red 216. El cliente 202 puede incluir un navegador web, una aplicación móvil, un socket o túnel, u otro software conectado a la red de tal manera que la comunicación con la plataforma 204 CSB sea posible a través de la red 216.

Se apreciará que el cliente 202 incluye uno o más ordenadores, teléfonos inteligentes, tabletas, tecnología portable, dispositivos informáticos, o sistemas de un tipo bien conocido en la técnica, tales como un ordenador central, una estación de trabajo, un ordenador personal, un ordenador portátil, un ordenador de mano, un teléfono móvil, un reproductor MP3, o un asistente digital personal. El cliente 202 comprende tal software, hardware, y componentes que se le ocurrirían a un experto en la técnica, tales como, por ejemplo, uno o más microprocesadores, sistemas de memoria, dispositivos de entrada/salida, controladores de dispositivos, y similares. El cliente 202 también comprende uno o más medios de entrada de datos (no se muestran en la Figura 1) operable por los usuarios del cliente 202 para la introducción de datos, tales como, por ejemplo, control por voz o audio, un dispositivo señalador (tal como un ratón), teclado, pantalla táctil, micrófono, reconocimiento de voz, y/u otros medios de introducción de datos conocidos en la técnica. El cliente 202 también comprende medios de visualización (no se muestran en la Figura 1) los cuales pueden incluir diversos tipos de pantallas conocidas, tales como pantallas de diodos de cristal líquido, pantallas de diodos emisores de luz, y similares, en las cuales la información puede visualizarse de manera perceptible para el usuario. Se apreciará que el cliente 202 puede comprender además tal software, hardware, y componente que se le ocurrirían a un experto en la técnica, para realizar operativamente las funciones asignadas al cliente 202 de acuerdo con la presente divulgación. Se apreciará además que el sistema 200, aunque se refiere a un único cliente, también puede incluir una pluralidad de clientes.

En al menos una realización de la presente divulgación, el cliente 202 incluye una combinación de infraestructura de GUI y módulos específicos de aplicación, bien conocidos por quien tenga conocimientos ordinarios en las técnicas. La infraestructura de GUI le da manejo a la lógica de negocio GUI, y realiza solicitudes al controlador 210 CSB, el cual está configurado para realizar acciones tales como crear, leer/obtener, eliminar, o modificar recursos. Se apreciará además que la GUI del cliente 202 está configurada operativamente para realizar solicitudes de navegación para cambiar la presente vista GUI.

Con referencia ahora a la Figura 3, se muestra una vista alternativa del predictor 206, en al menos una realización de la presente divulgación. El predictor 206 incluye un constructor 302 del modelo de comportamiento, un rellenedor 304 de caché, un limpiador 306 de caché, una base 308 de datos, y un punto 310 final REST. En al menos una realización de la presente divulgación, el constructor 302 del modelo de comportamiento está configurado para reconstruir un modelo de predicción con nuevos datos a partir del historial, como se divulga además en la presente memoria. El constructor 302 del modelo de comportamiento construye el modelo de predicción con probabilidades de transición y está configurado para almacenar tales modelos de predicción en la base 308 de datos.

En al menos una realización de la presente divulgación, el rellenedor 304 de caché está configurado para recuperar respuestas de navegación para las vistas más probables que van a ser visitadas por el usuario operando una GUI, como se divulga además en la presente memoria. Las respuestas de navegación pueden incluir respuestas en Notación de objetos JavaScript (JSON), que contienen información de subgráficos sobre la vista de destino, el contexto de los recursos de disponibilidad, el trayecto hacia atrás, y los elementos principales aplicables. Se apreciará que el rellenedor 304 de caché no recuperará respuestas de navegación para vistas más probables que la cantidad predefinida cada vez para no desbordar la caché 206 de predicción. El rellenedor 304 de caché si además está configurado para guardar respuestas de navegación (por ejemplo, un árbol de navegación) recibidas a partir del módulo 208 de navegación, en una caché de solicitud-respuesta (almacenada dentro de la base 308 de datos). En al menos una realización de la presente divulgación, la base 308 de datos almacena el historial de vistas de GUI visitadas, modelos de predicción, identificadores de vistas probables, respuestas de navegación de pantalla de GUI, y similares, como se divulga además en la presente memoria.

A modo de ejemplo, la continuación es una lista ejemplar de posibles transacciones de navegación:

Número	Solicitud	Respuesta
1	Vista: "Inicio"	Vista: "Inicio"
		Fuente: "Inicio js"
		Elementos del Menú: ["Inicio", "Lista de Usuario"]
2	Vista: "Lista de Usuario", Contexto: ["Inicio"]	Vista: "Lista de Usuario",
		Fuente: "lista de usuario js",
		Elementos del Menú: ["Inicio", "Lista de Usuario"] Contexto: ["Inicio"]
3	Vista: "Nuevo Usuario", Contexto: ["Inicio", "Lista de Usuario"]	Vista: "Nuevo Usuario",
		Fuente: "nuevo usuario js"
		Elementos del Menú: ["Inicio", "Lista de usuario"] Contexto: ["Inicio", "Lista de Usuario"]
4	Vista: "Asignar Servicio", fuente_id: 126, Contexto: ["Inicio", "Lista de Usuario", "Nuevo Usuario"]	Vista: "Asignar Servicio",
		Fuente: "asignar servicio js",
		Elementos del Menú: ["Inicio", "Lista de Usuario"] Contexto: ["Inicio", "Lista de Usuario", "Nuevo Usuario"] Vars: ["Suscripción": {<suscripción_objeto>}]
5	Vista: "Nuevo VPS", fuente_id: 126, Contexto: ["Inicio", "Lista de Usuario", "Nuevo Usuario", "Asigna Servicio"]	Vista: "Nuevo VPS",
		Fuente: "configura vps.js",
		Elementos del Menú: ["Inicio", "Lista de Usuario"] Contexto: ["Inicio", "Lista de Usuario", "Nuevo Usuario", "Asigna Servicio"] Vars: ["Suscripción": {<suscripción_objeto>}]
6	Vista: "Vista de Usuario", Fuente_id: 126	Vista: "Vista de Usuario",
		Fuente: "vista de usuario.js",
		Elementos del Menú: ["Inicio", "Lista de Usuario"] Vars: ["Usuario": {<usuario_objeto>}] Contexto: []

En al menos una realización de la presente divulgación, el limpiador 306 de caché está configurado para solicitar al controlador 210 CSB la última marca de tiempo de modificación del ámbito de seguridad, como se divulga en la presente memoria. Por ejemplo, pueden existir numerosas condiciones de seguridad que pueden ser aplicables a la transición de GUI de un usuario. Estas condiciones pueden afectar a los recursos disponibles para tal usuario (y, en consecuencia, a las vistas disponibles relacionadas con los paquetes de conector cuyos metadatos establecen los recursos operados por la referida aplicación de conectores), y cuyas condiciones (o recursos) pueden cambiar con el tiempo. Continuando con este ejemplo, si un elemento de navegación solicitado requiere recursos prohibidos o inexistentes para resolver las condiciones de seguridad, el elemento de navegación no se incluirá en la respuesta de navegación y, por lo tanto, la vista no estará disponible para el usuario. (Es decir, es posible que un usuario no pueda acceder a todos los recursos disponibles ya que no tiene acceso a tales recursos) A modo de otro ejemplo, se supone que un usuario es suscriptor de un plan de servicios del intermediario de servicios en la nube, que cubre la venta de suscripciones únicamente a Dropbox® y Office 365®, al usuario no se le deberían mostrar vistas relacionadas con otras aplicaciones integradas en la plataforma 204 CSB y asociadas con recursos referidos (por ejemplo, "servidor VPS"). A modo de otro ejemplo, restringir las transiciones GUI o sustituir una vista por otra son condiciones de seguridad que se expresan a través del mecanismo de variables. Los metadatos de navegación incluyen un lenguaje específico del dominio para describir las condiciones de seguridad en base a los recursos disponibles en el ámbito de seguridad del usuario. Los ejemplos de condiciones no limitantes incluyen: ambos recursos están disponibles, ninguno de los recursos está disponible, el recurso está disponible y contiene un atributo con un valor particular, y similares. En una realización de la presente divulgación, este lenguaje específico de dominio en base al Lenguaje de Consulta de Recursos (RQL).

Se apreciará que la última marca de tiempo de modificación determina la última fecha de cualquier modificación/provisión o eliminación de recursos en el ámbito del modelo de predicción actual. Si el ámbito ha sido modificado después de la creación de la caché, el limpiador 306 de caché limpia la caché de solicitud-respuesta (la caché puede estar almacenada en la base 308 de datos). En al menos una realización de la presente divulgación, cada usuario está asociado con un token que se utiliza para llevar a cabo todas las transacciones dentro del sistema. Por ejemplo, se asocia un token al ámbito de seguridad del usuario, y qué ámbito de seguridad es recuperable (por ejemplo, a partir de la base 308 de datos) en base al token. La base 308 de datos está configurada para almacenar respuestas de vista asociadas para cada token de usuario, presentando así una respuesta de vista personalizada aplicable a cada usuario, y basada al menos en parte en el ámbito de seguridad del usuario.

En al menos una realización de la presente divulgación, el limpiador 306 de caché está configurado para acceder a cada uno de los tokens de usuario y periódicamente solicita cambios en el ámbito de seguridad asociado con el token, a partir del controlador 210 CSB. Se apreciará que, si se cambia el ámbito de seguridad asociado con un token, el limpiador 306 de caché "limpia" (es decir, modifica o elimina) las respuestas de vista lista en la base 308 de datos, asociadas con tal token.

En al menos una realización de la presente divulgación, el módulo 208 de navegación está configurado para construir gráficos de navegación y respuesta utilizados para la construcción de pantallas de GUI, como se divulga en la presente memoria. En al menos una realización de la presente divulgación, el módulo 208 de navegación está configurado para integrarse con cada una de la pluralidad de aplicaciones 212 de conector a través del controlador 210 CSB (cuyas aplicaciones 212 de conector pueden ser proporcionadas por ISVs o desarrolladores de conectores de terceros). El módulo 208 de navegación está configurado para generar un árbol de navegación (por ejemplo, como se muestra en la Figura 1C), como se describe además en la presente memoria.

En al menos una realización de la presente divulgación, el punto 310 final REST simula un punto final de navegación, y está configurado para recibir solicitudes y proporcionar respuestas de navegación válidas, como se divulga además en la presente memoria. Por ejemplo, durante una operación GUI que es una solicitud de adición, se agrega un elemento de operación a una lista del historial de navegación. El punto 310 final REST está configurado además para calcular el recuento de nuevos elementos y periódicamente (por ejemplo, para cada solicitud número 100) llamar al constructor 302 del modelo de comportamiento para reconstruir el modelo con nuevos datos en base a la lista del historial de navegación. Se apreciará que en cada solicitud el punto 310 final REST está configurado para comunicarse con la caché almacenada en la base 308 de datos para solicitar una respuesta precalculada si la hubiera. Si no hay respuesta, el punto 310 final REST está configurado para consultar directamente al servicio de navegación y regresar una respuesta. Se apreciará además que en cada llamada de solicitud por el punto 310 final REST, el historial de caché se rota para evitar el consumo excesivo de espacio del disco y la memoria. También se apreciará que el limpiador 306 de caché opera para invalidar la caché si es necesario.

En al menos una realización de la presente divulgación, el punto 310 final REST está configurado para almacenar un historial de vistas visitadas en la base 308 de datos, como se divulga además en la presente memoria. Por ejemplo,

si un usuario opera una pantalla de GUI, el historial de vistas visitadas de la pantalla de GUI se almacena en la base 308 de datos. En al menos una realización de la presente divulgación, el punto 310 final REST está configurado para recuperar una respuesta de vista particular (respuesta de navegación para la vista particular) de una pantalla de GUI de la base 308 de datos. Si no se recibe una respuesta (es decir, la respuesta de vista particular no está disponible), el punto 310 final REST está configurado para solicitar una respuesta de vista a partir de la navegación 208. Se apreciará que cuando el predictor 206 recopila información sobre la actividad del usuario, tal información de actividad se almacena en la base 308 de datos. Se apreciará además que el constructor 302 del modelo de comportamiento puede utilizar esta información en la base 308 de datos para desarrollar modelos de predicción de transición de pantalla de GUI, como se divulga además en la presente memoria.

En al menos una realización de la presente divulgación, las vistas de transición de GUI, las etapas GUI, la experiencia del usuario, pueden recuperarse a partir de bases de datos de terceros (no se muestran), tal como sistemas de inteligencia empresarial, sistemas de gestión de relaciones con los clientes, y similares. Tal información puede almacenarse dentro la base 308 de datos.

Con referencia ahora a la Figura 4, se muestra un procedimiento para generar GUIs basadas en predicción para mejorar los tiempos de respuesta GUI, generalmente indicado en 400. En al menos una realización de la presente divulgación, el procedimiento 400 incluye un inicio por temporizador en la etapa 402, la carga de secuencias de vistas en la etapa 404, la comprobación de datos en la etapa 406, el recálculo de probabilidades en la etapa 408, el establecimiento de una fecha de creación del modelo en la etapa 410, y la solicitud de relleno de caché en la etapa 412.

En al menos una realización de la presente divulgación, el procedimiento 400 describe la secuencia de gestión de la pantalla de GUI y las transiciones. Se apreciará que, en cualquier pantalla de GUI, una transición del estado actual a un estado futuro (una nueva pantalla de GUI) incluye una determinada probabilidad de que se produzca una tal actividad. Por ejemplo, si la pantalla de GUI es una pantalla de inicio de sesión en una aplicación (por ejemplo, la aplicación 182), el estado actual de tal pantalla de GUI puede incluir áreas para ingresar credenciales de inicio de sesión y enviar dichas credenciales. En una tal realización ejemplar, el estado futuro (es decir, la siguiente pantalla de GUI) puede ser probablemente: 1) un inicio de sesión con éxito; o 2) un inicio de sesión sin éxito. Continuando con este ejemplo entonces, la probabilidad probable de un inicio de sesión con éxito (la cual depende de las restricciones de seguridad) o el inicio de sesión sin éxito, es un cálculo de probabilidad ejemplar de la transición de GUI. Es decir, la transición a partir de la pantalla de GUI de inicio de sesión puede limitarse esencialmente a dos opciones: una pantalla de GUI de inicio de sesión con éxito, o una pantalla de GUI de inicio de sesión sin éxito.

Se apreciará que un usuario que realiza las etapas de GUI está esencialmente accediendo secuencialmente a una serie de pantallas (o vistas) de GUI. En al menos una realización de la presente divulgación, la pantalla de GUI es un conjunto predefinido de vistas que pueden configurarse como parte de una aplicación (por ejemplo, la aplicación 182). Se apreciará que cuando un usuario utiliza una aplicación (por ejemplo, la aplicación 182), las posibles pantallas de GUI son limitadas. Por ejemplo, con referencia a la Figura 1, es posible ir al estado 110 GUI (Nuevo Usuario) a partir del estado 102 GUI (Inicio), pero es poco probable que este escenario sea implementado si no fluye lógicamente y si no hay un caso de negocio que requiera tal soporte de funcionalidad. Sin embargo, otras transiciones de GUI y conexiones de pantallas se utilizarán casi exclusivamente en un orden predefinido. Por ejemplo, con referencia a la Figura 1B, agregar un nuevo usuario implica pasar la pantalla del estado 106 GUI (Vista del usuario), porque esta conexión entre vistas es estática y se desarrolla como el trayecto lógico principal. Se apreciará entonces que un modelo de navegación creado en base al diagrama 150 de estado asume que a menudo hay estados GUI próximos o anteriores (o pantallas de GUI) más probables que son de acceso o vistos en una secuencia serie específica (es decir, la actividad GUI de un usuario es lineal). En al menos una realización de la presente divulgación, es posible utilizar algoritmos de aprendizaje automático para encontrar patrones y predecir la siguiente pantalla de GUI más probable (o estado GUI) en la cadena de actividad del usuario.

En al menos una realización de la presente divulgación, se puede construir un árbol de navegación utilizando un gran modelo estático, por lo tanto, los principales escenarios de uso para la mayoría de las pantallas de GUI utilizables o aplicación se gestionan. A modo de ejemplo, casi todos los usuarios rellenan los datos de contacto al actualizar su perfil; cada administrador abre una lista de usuarios para gestionar; cada administrador se redirige a la lista de usuarios después de crear un nuevo usuario. Se apreciará que estos comportamientos predefinidos pueden predecirse en base al estado actual de la pantalla de GUI. Se apreciará además que el árbol de navegación para un usuario puede desarrollarse en base al comportamiento previo del usuario. Con referencia al diagrama 150 de estado como ejemplo, el administrador de usuario accede al estado 120 GUI (Inicio) y puede entonces comprobar primero las notificaciones. El administrador también puede realizar otras acciones tales como proporcionar acceso, bloquear/desbloquear usuarios, y similares. Continuando con este ejemplo, se puede construir un modelo predictivo en base a las secuencias de navegación más probables.

Se apreciará que se pueden utilizar varios mecanismos de predicción de secuencia, tales como el modelo Ngram, el modelo de Markov Oculto, el modelo de Markov de Máxima Entropía (combinación de un clasificador basado en la entropía y un Modelo de Markov Oculto), clasificadores ocultos que incluyen, pero no se limita a, SVM (máquina de vector de soporte), clasificadores de árbol de decisión, clasificadores de bosque aleatorio, K vecinos más cercanos, y cualquier Red Neuronal capaz de realizar la predicción de secuencia, por nombrar algunos ejemplos no limitantes.

De nuevo con referencia al procedimiento 400, el procedimiento comienza por temporizador en la etapa 402. En al menos una realización de la presente divulgación, se utiliza un temporizador para programar los cálculos de las probabilidades de las transiciones GUI y los modelos de comportamiento. El predictor 206 verifica si hay suficientes datos (es decir, en la caché en la base 308 de datos) para calcular probabilidades de transición en base a las secuencias de vistas guardadas. Si no hay datos suficientes, el procedimiento 400 regresa a la etapa 402, pero se procesa en la etapa 408 en caso contrario.

En la etapa 408, se calculan las probabilidades de transición y se crea un modelo de probabilidades de transición y se establece la fecha de creación del modelo en la etapa 410. Por ejemplo, una entrada de un clasificador es un conjunto de características, y la salida es un conjunto de probabilidades de que, dado el conjunto de características, se produzca un resultado particular (es decir, un evento de GUI). Por ejemplo, el constructor 302 del modelo de comportamiento calcula las probabilidades de las transiciones más probables GUI. Se apreciará que las probabilidades calculadas son indicativas de la probabilidad de que un usuario transiciona de una vista GUI actual a una vista GUI futura. A modo de ejemplo, una pantalla de GUI puede visualizar controles para la administración de usuario (por ejemplo, crear/actualizar un rol de usuario). Un usuario administrador del sistema que opere la pantalla de GUI de administración de usuario para crear un "nuevo usuario" empezará primero en la pantalla principal de administración de usuario. La siguiente pantalla (es decir, la siguiente etapa) es probablemente una pantalla de formulario para rellenar información demográfica sobre el nuevo usuario, tal como, por ejemplo, el nombre completo del usuario, la información de contacto, y similares. Continuando con este ejemplo entonces, la vista futura más probable para el usuario administrador del sistema que se encuentra actualmente en la pantalla de creación de nuevo usuario será probablemente la pantalla de formulario. La menor probabilidad es que el usuario administrador del sistema regrese a la pantalla anterior, sin acceder a la pantalla del formulario.

En al menos una realización de la presente divulgación, el rellenador 304 de caché es entonces llamado para guardar el modelo de probabilidades de transición en la caché y en la base 308 de datos.

Con referencia ahora a la Figura 5, se muestra un procedimiento para generar GUIs basadas en predicción para mejorar los tiempos de respuesta GUI, generalmente indicado en 500. En al menos una realización de la presente divulgación, el procedimiento 500 gestiona el manejo de solicitudes comprobando si se recibe una solicitud del constructor 302 del modelo de comportamiento en la etapa 502, obteniendo IDs de vistas con una probabilidad más alta que el umbral para ser solicitadas en base a predicciones del modelo de comportamiento en la etapa 504, solicitando navegación 208 en la etapa 506, y almacenando las respuestas de vistas recibidas (es decir, respuestas de navegación) en la etapa 508.

En al menos una realización de la presente divulgación, cada solicitud recibida para una pantalla de GUI (o vista) se guarda en la base 308 de datos como una secuencia de solicitud.

En al menos una realización de la presente divulgación, los IDs de vista GUI con probabilidades más altas que el umbral se recuperan en la etapa 504. A modo de ejemplo, los umbrales pueden ser diferentes y dependen del tipo de modelo de probabilidad utilizado: un Modelo de Markov Oculto (HMM) tendrá un umbral diferente en comparación con un Clasificador de Bosque Aleatorio. En al menos una realización de la presente divulgación, los datos utilizados para calcular tales probabilidades son los datos acumulados para transiciones de usuario anteriores (y almacenados en la base 308 de datos). Se apreciará que los datos pueden incluir cadenas de transición de GUI. Continuando con el ejemplo anterior, en el modelado del Clasificador de Bosque Aleatorio (RFC), el modelo RFC se ajusta con los datos y genera un conjunto de probabilidades. Teniendo en cuenta la distribución de probabilidades, el recuento de usuarios que trabajan simultáneamente y la limitación de memoria, se puede definir un umbral óptimo el cual satisfaga todos los requisitos: datos máximos en caché, tamaño de caché limitado. Se apreciará que, aunque pueden calcularse todas las probabilidades de las transiciones, se calculan las probabilidades de transición más probables para garantizar un uso eficiente de la memoria/almacenamiento caché limitados para almacenar tales modelos.

Se apreciará además que la presente divulgación proporciona el servicio de predicción unificado para recopilar datos sobre los clientes, los cuales pueden ser útiles para fines de prueba o análisis del comportamiento del usuario que pueden utilizarse además para optimizar las vistas y la navegación GUI.

Si el modelo contiene próximos IDs de vistas GUI con probabilidades de transición más altas que el umbral, las respuestas de navegación para esas vistas GUI se preparan como posibles respuestas, en la etapa 506. Las respuestas se almacenan en la base 308 de datos.

Con referencia ahora a la Figura 6, se muestra un procedimiento para generar GUIs basadas en predicción para mejorar los tiempos de respuesta GUI, generalmente indicado en 600. El procedimiento 600 incluye comprobar si se recibe una solicitud de vista GUI en la etapa 602, obtener la última fecha de modificación del ámbito de seguridad en la etapa 604, comprobar si el modelo de probabilidad de transición es más reciente que la fecha de modificación del ámbito de seguridad en la etapa 606, y en caso afirmativo, entonces proceder a la etapa 608, en caso contrario, proceder a la etapa 612.

En al menos una realización de la presente divulgación, el procedimiento 600 incluye además comprobar si el modelo contiene respuestas de navegación preparadas para la vista GUI solicitada en la etapa 608, y obtener la respuesta de vista GUI (respuesta de navegación) a partir de la base 308 de datos en la etapa 610. En al menos una realización de la presente divulgación, el procedimiento 600 incluye además la etapa 614 de solicitar una navegación en la etapa 614, almacenar la respuesta de la vista GUI en la base 308 de datos en la etapa 616, y regresar la respuesta en la etapa 618.

REIVINDICACIONES

1. Un procedimiento para generar pantallas de interfaz gráfica de usuario, GUI, basadas en predicción para mejorar los tiempos de respuesta de las pantallas de GUI, comprendiendo el procedimiento las etapas:

5 recibir en un punto final, una solicitud de visualización para una pantalla de GUI, comprendiendo la pantalla de GUI una interfaz de usuario integrada construida a partir de una pluralidad de paquetes de conector; validar en un predictor (206), una validez de un modelo de transición de GUI, el modelo de transición de GUI correspondiente a cada uno de la pluralidad de paquetes de conector, en el que la validación del modelo de transición de GUI comprende la comprobación de una fecha de ámbito de seguridad modificada por última vez;
10 construir en un constructor (302) de modelo de comportamiento, un nuevo modelo de transición de GUI, comprendiendo el nuevo modelo de transición de GUI una pluralidad de probabilidades de transición para cada posible transición a partir de la pantalla de GUI, dependiendo la pluralidad de probabilidades de transición al menos en parte de atributos de recursos disponibles para cada uno de la pluralidad de paquetes de conector;
15 seleccionar en el constructor (302) de modelo de comportamiento, una transición probable de la pluralidad de probabilidades de transición, siendo la transición probable una transición a una nueva pantalla de GUI que es más probable que sea solicitada;
20 crear, en un módulo (208) de navegación, un árbol de navegación basado, al menos en parte, en la transición probable; y regresar una respuesta al punto final, comprendiendo la respuesta el árbol de navegación, y comprendiendo además una respuesta de navegación para la nueva pantalla de GUI.

2. El procedimiento de la reivindicación 1, en el que cada uno de la pluralidad de modelos de transición se selecciona a partir de un grupo que consiste en un modelo Ngram, un modelo de Markov oculto, y un modelo de Markov de máxima entropía.

3. El procedimiento de la reivindicación 2, en el que cada uno de la pluralidad de modelos de transición comprende además clasificadores potenciados seleccionados a partir de un grupo que consiste en una máquina de vectores de soporte (SVM), un árbol de decisión, un bosque aleatorio, y K vecinos más cercanos.

4. El procedimiento de la reivindicación 1, en el que la solicitud de la pantalla de GUI se recibe a partir de un cliente.

5. El procedimiento de la reivindicación 1, en el que el modelo de transición de GUI comprende una fecha de modificación del ámbito de seguridad.

6. El procedimiento de acuerdo con una cualquiera de las reivindicaciones anteriores, en el que la última fecha del ámbito de seguridad modificada comprende una fecha en la que se modificó por última vez el modelo de transición de GUI.

7. El procedimiento de la reivindicación 1, que comprende además la etapa de almacenamiento de la respuesta en una base (308) de datos.

8. El procedimiento de la reivindicación 7, que comprende además la etapa de almacenamiento del nuevo modelo de transición de GUI.

9. El procedimiento de la reivindicación 8, que comprende además almacenar una fecha de creación del modelo para el nuevo modelo de transición de GUI.

10. Un sistema (200) para generar pantallas de interfaz gráfica de usuario, GUI, basadas en predicción para mejorar los tiempos de respuesta de la pantalla de GUI, comprendiendo el sistema (200): una plataforma (204) de intermediario de servicios en la nube que comprende:

un predictor (206), en el que el predictor (206) está configurado para validar el modelo de transición de GUI comprobando una última fecha del ámbito de seguridad modificada;
una base (308) de datos;
un constructor (302) de modelo de comportamiento configurado para crear un modelo de transición de GUI, siendo el modelo de transición de GUI una probabilidad de una transición de GUI a partir de una primera pantalla de GUI a una nueva pantalla de GUI, en el que la nueva pantalla de GUI comprende una interfaz de usuario integrada construida a partir de una pluralidad de paquetes de conector;

- un relleno (304) de caché configurado para recuperar las vistas GUI más probables que serán visitadas por un usuario, en base al menos en parte al modelo de transición de GUI;
- un módulo (208) de navegación configurado para crear respuestas de navegación para la nueva pantalla de GUI en base, al menos en parte, a al menos un estado de recurso de un ámbito de seguridad aplicable a un usuario que solicita la nueva pantalla de GUI;
- un limpiador (306) de caché configurado para limpiar una caché de respuesta en base, al menos en parte, a si la última fecha del ámbito de seguridad modificada es más reciente que el modelo de transición de GUI; y
- un punto(310) final REST configurado para simular un punto final de servicio de navegación GUI.
- 10 11. El sistema (200) de la reivindicación 10 que comprende además un cliente operablemente conectado a la plataforma (204) de intermediario de servicio en la nube, y configurado para solicitar una nueva pantalla de GUI y recibir una respuesta de navegación para la nueva pantalla de GUI.
- 15 12. El sistema (200) de la reivindicación 10, en el que la plataforma (204) de intermediario de servicio en la nube comprende además un controlador (210) de intermediario de servicio en la nube.
- 20 13. El sistema (200) de la reivindicación 12, en el que el controlador (210) de intermediario de servicio en la nube comprende una pluralidad de aplicaciones de conector, cada una de la pluralidad de aplicaciones (212) de conector está configurada para cada aplicación para su integración a través de la plataforma (204) de intermediario de servicio en la nube.
14. El sistema (200) de la reivindicación 10, en el que la pluralidad de paquetes de conector comprende módulos de interfaz de usuario para construir elementos de navegación.
- 25 15. El sistema (200) de la reivindicación 14, en el que la pluralidad de paquetes de conector comprende además un ámbito de seguridad para cada uno de los elementos de navegación.

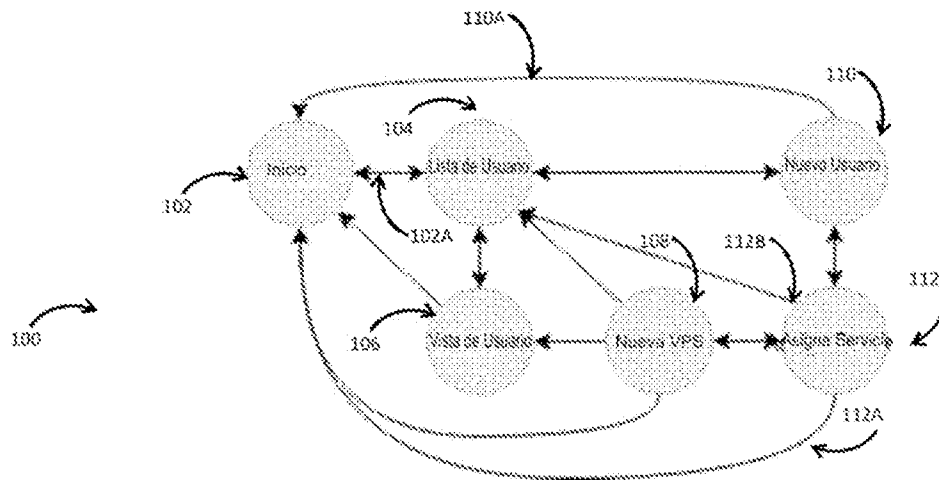


FIG. 1A

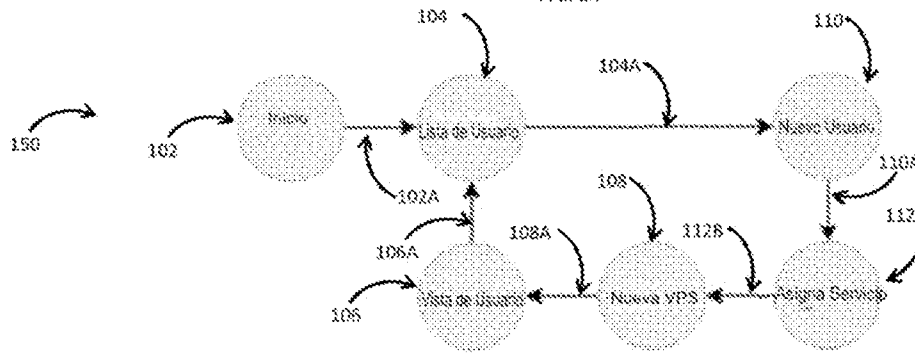


FIG. 1B

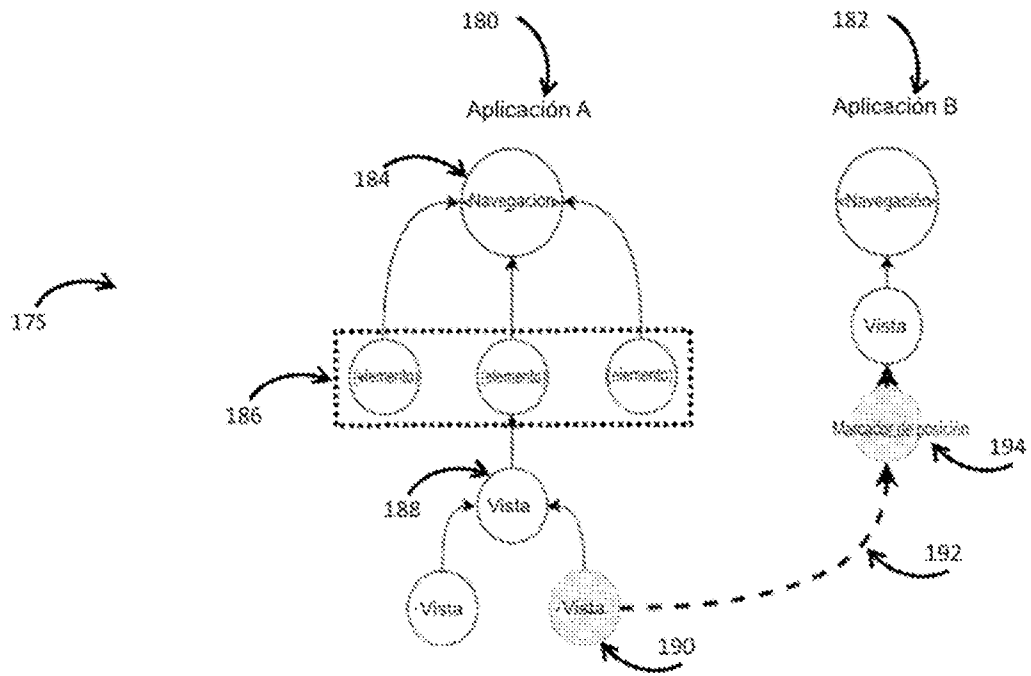


FIG. 1C

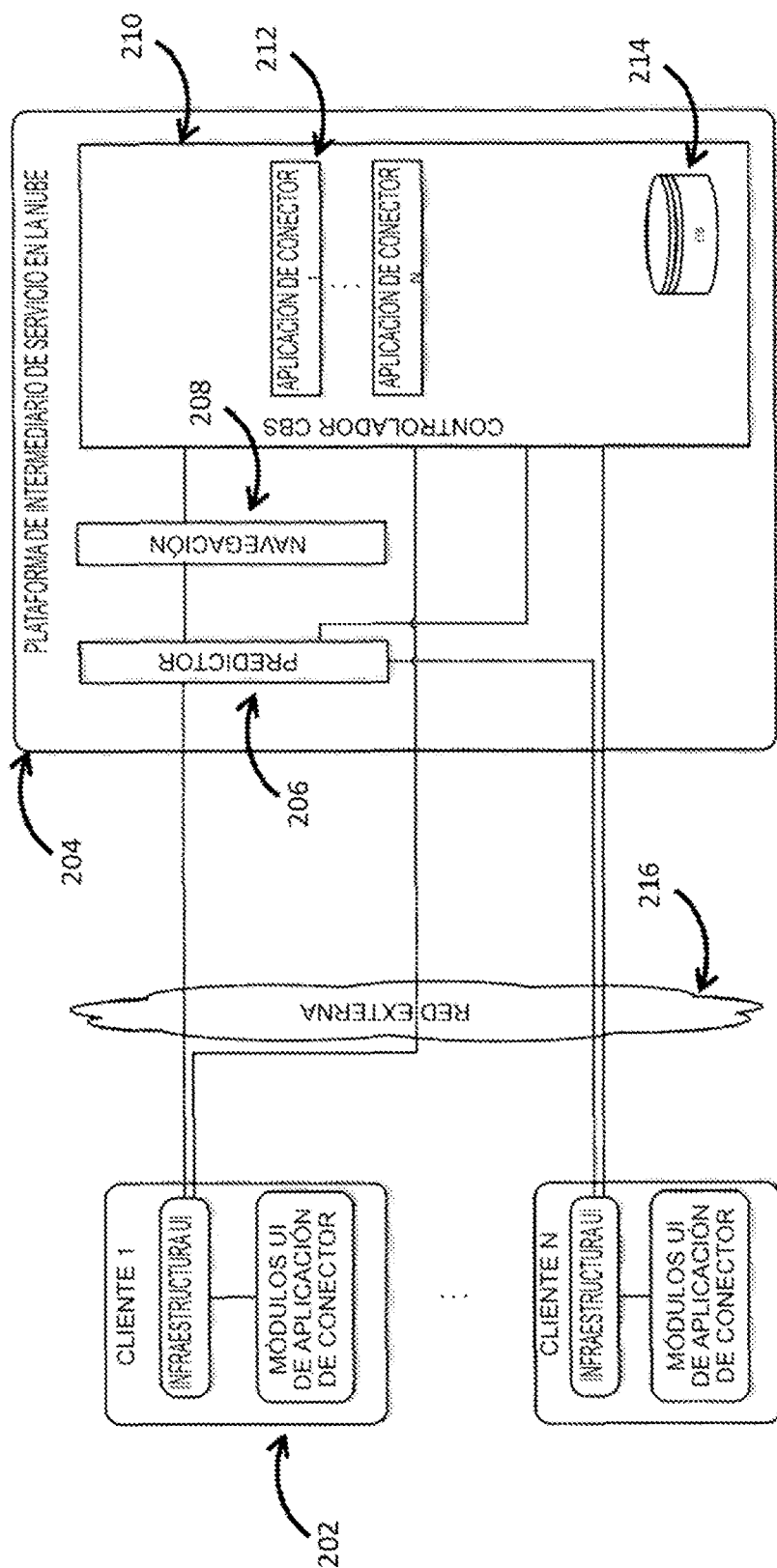
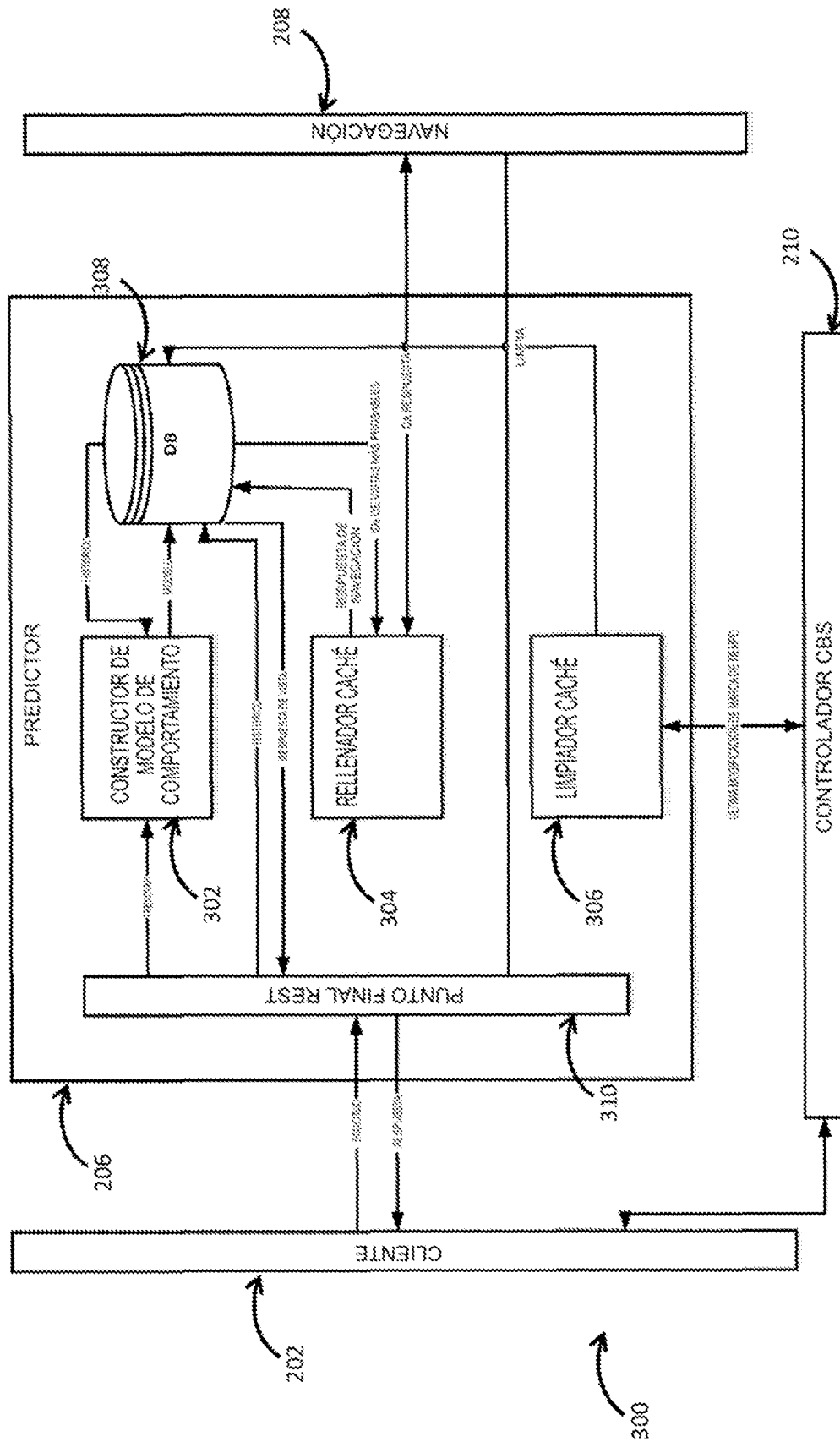


FIG. 2



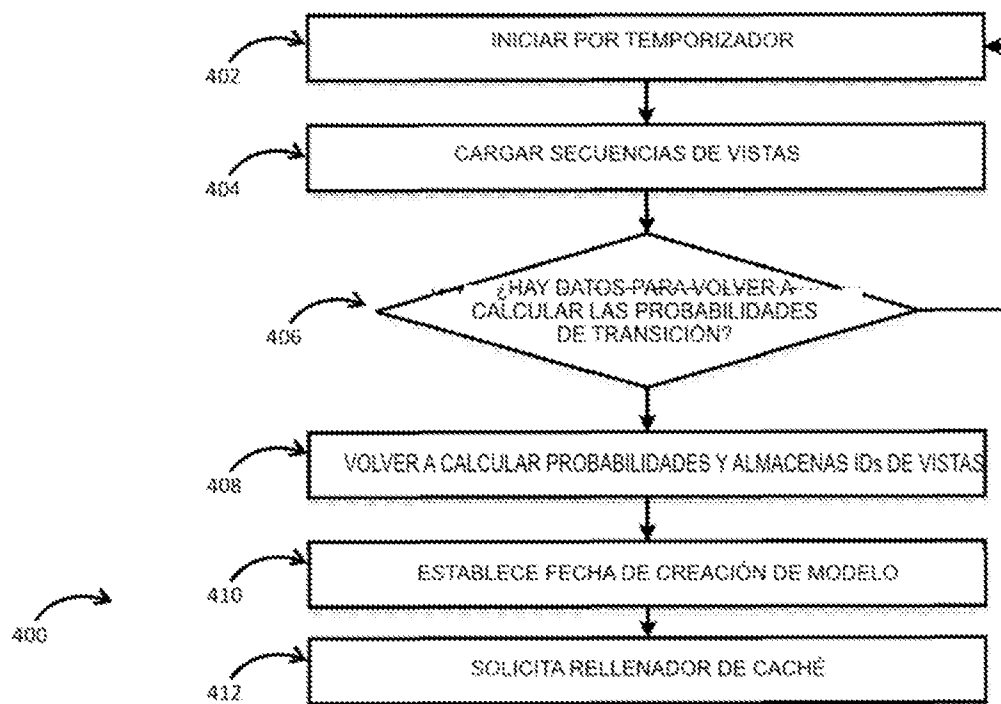


FIG. 4

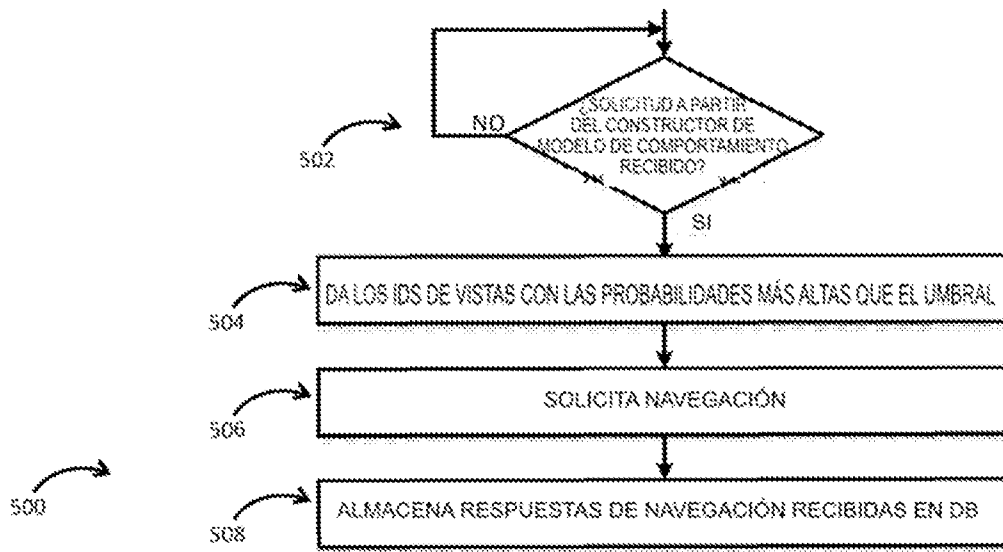


FIG. 5

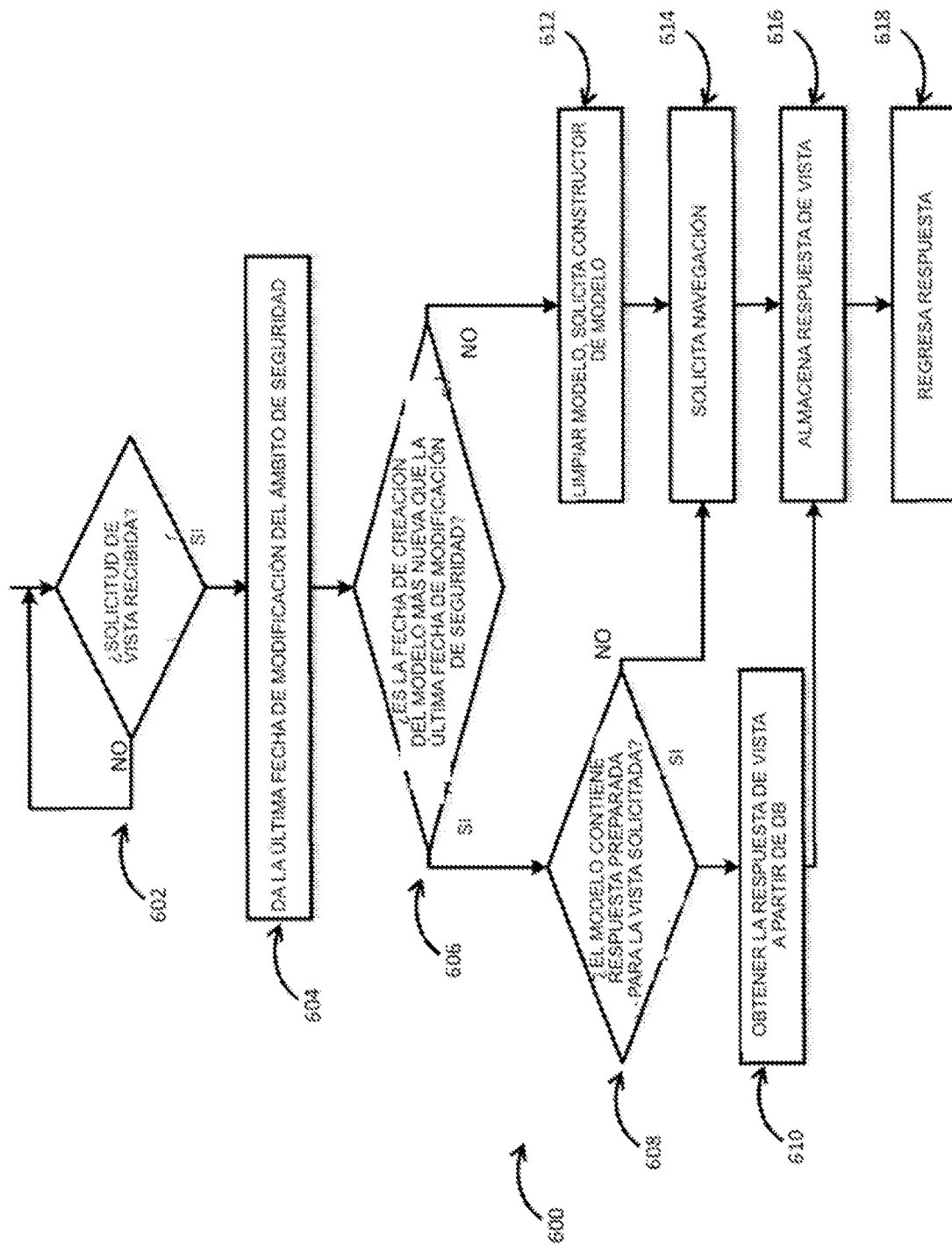


FIG. 6