

**REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO**

Oficio N° 8592

Ref. Expediente N° NC2017/0003239

Señor(a)
HOME BOX OFFICE, INC
notificaciones@clarkemodet.com.co

Asunto: PATENTE DE INVENCION - PCT

Patente de Invención	PT	Patente de Modelo de Utilidad	
Vía Nacional			
Vía PCT (Fase Nacional)	PCT	Capítulo I	Capítulo II
No. Solicitud Internacional	PCT/US2015/048418		
Fecha de Presentación Internacional	3 de septiembre de 2015		

Como resultado del examen de fondo realizado, con fundamento en el Artículo 45 de la Decisión 486 de la Comisión de la Comunidad Andina, se comunica:

1. Documentos estudiados

Descripción	Radicado N° NC2017/0003239	3 de abril de 2017
Reivindicaciones:	Radicado N° NC2017/0003239	3 de abril de 2017
	Radicado N° NC2017/0003239	2 de junio de 2017
Dibujos:	Radicado N° NC2017/0003239	3 de abril de 2017
Prioridades:	País: US, ESTADOS UNIDOS DE AMÉRICA Número: 62/046,109 Fecha: 4 de septiembre de 2014	

**REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO**

Oficio N° 8592

Ref. Expediente N° NC2017/0003239

	País: US, ESTADOS UNIDOS DE AMÉRICA Número: 14/843,792 Fecha: 2 de septiembre de 2015
--	--

2. Análisis de la Prioridad

Se acepta la reivindicación de prioridad porque esta solicitud fue presentada dentro del plazo establecido y el contenido técnico de esta solicitud corresponde con el de la prioridad reivindicada.

3. Modificaciones presentadas (Art. 34 D 486 y Art. 19, 22 y 34 PCT)

Se aceptan las modificaciones presentadas comoquiera que se ajustan a las prescripciones contenidas en el reglamento del PCT y en el Art 34 de la Decisión 486.

4. Objeto de la invención

De acuerdo con lo establecido en el numeral 1.2.2.2 del Capítulo Primero del Título X de la Circular Única para efectos del examen de patentabilidad, se tendrá en cuenta la tasa pagada por el solicitante al momento de comenzar el examen para las reivindicaciones adicionales; esto es que cuando la solicitud contenga más de diez (10) reivindicaciones y no se haya pagado la tasa complementaria, sólo se estudiarán las cubiertas por la tasa pagada. Por lo tanto, se estudiarán las reivindicaciones 1 a 20 del capítulo reivindicatorio (Radicado N° NC2017/0003239 del 2 de junio de 2017).

Título de la solicitud: "MÉTODO Y SISTEMA QUE TIENE MODELOS POPULADOS CON DATOS ASINCRÓNICOS" (Radicado N° NC2017/0003239 del 2 de junio de 2017).

El problema planteado en esta solicitud consiste en la necesidad de optimizar el proceso de recuperación o recopilación de información desde una fuente de datos hacia una vista que requiere de dicha información.

Para resolver este problema técnico, la solicitud en estudio proporciona un sistema y métodos para recopilar información a través de modelos asíncronos que vinculan a la fuente de datos con la vista. Una vista solicita a un modelo de vista para devolver datos de vista especificados; a su vez, el modelo de vista hace una o más peticiones

**REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO**

Oficio N° 8592

Ref. Expediente N° NC2017/0003239

a un modelo de datos para recuperar la información correspondiente a los datos de vista especificados. El modelo de datos se comunica con una fuente de datos, tales como un servicio web o una memoria caché para devolver la información solicitada al modelo de vista.

El objeto de la presente solicitud se relaciona con la Clasificación Internacional de Patentes (CIP): G06F 9/44.

5. De la solicitud de patente

Reivindicaciones (Art. 30 D 486, Art. 22 PCT)

- El solicitante no utiliza la estructura correcta en las reivindicaciones ya que el preámbulo no indica cuál es el objeto de la invención, que suele coincidir con el título de la invención y las características técnicas conocidas en el estado de la técnica. Por ejemplo en las reivindicaciones 1 y 17 el preámbulo solo menciona “método” sin definir cuál es el objeto de este.
- Las expresiones de la Reivindicación 14 “(...) y más tarde devuelve (...)” y Reivindicación 20 “(...) más instrucciones ejecutables por máquina (...)”, son imprecisas, por lo cual no permiten distinguir la invención del estado de la técnica. Se sugiere sustituirlas por un término preciso o un rango concreto.

6. Determinación del Estado de la Técnica

El estado de la técnica se determinó con base en la fecha de prioridad: 4 de septiembre de 2014, y se encontraron los siguientes documentos que anticipan el objeto de esta solicitud:

Ítem	Documento	Título	Fecha de Publicación
D1	Artículo*	“Async MVVM Modern UI”	16/Agosto/2012
D2	Artículo**	“JavascriptPromise”	22/Abril/2013

* Juan Pablo G.C. ** Martin Fowler.

REPÚBLICA DE COLOMBIA SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

Oficio N° 8592

Ref. Expediente N° NC2017/0003239

El documento D1¹ divulga un ejemplo del uso de MVVM con Windows 8 Modern UI con enlace de datos, comandos y propiedades utilizando la metodología asincrónica y manteniendo el modo de diseño con plantillas de datos (Sección Introducción).

El documento D2² divulga características y algunos ejemplos de promesas implementadas en Javascript (Todo el documento).

7. Examen de Patentabilidad (Art. 14 D486)

Novedad (Art 16 D486)

La reivindicación 11 consiste en: “*Un sistema que comprende: (...)*” (Radicado N° NC2017/0003239 del 2 de junio de 2017).

TABLA 1. Comparación de las características técnicas esenciales, con las del estado de la técnica:

Características esenciales	D1
Un modelo de vista (102) que comprende una interfaz de modelo de vista (104) y un proveedor de modelo de vista (108), el modelo de vista configurado para ser acoplado a un objeto de vista (106) a través de la interfaz de modelo de vista.	El modelo de vista implementa una versión personalizada de <i>INotifyPropertyChanged</i> para ser asincrónico y actualiza la UI (Interfaz de Usuario) cuando es capaz (Sección ViewModel).
Un modelo de datos (110) que comprende una interfaz de modelo de datos (112) y un proveedor de modelo de datos (114), el modelo de vista configurado para ser acoplado al modelo de datos a través de la interfaz de modelo de datos.	El modelo es muy simple, por lo general prefiero agregar un presentador para la vista en lugar de agregar un convertidor, use la opción que desee. Tenga en cuenta que en <i>W8RP</i> el formato de fecha ha cambiado a <i>GetDateTimeFormats</i> (Sección Model).
El modelo de vista configurado para recibir una solicitud (802) de datos de vista especificados del objeto de vista.	Finalmente llegamos al XAML, he creado una página simple con los botones de comando, <i>GridView</i> y <i>TextBlock</i> para el elemento seleccionado. Cuida lo siguiente: - Creo una instancia de <i>ViewModel</i> en <i>Page.DataContext</i> .

1 <https://www.codeproject.com/Articles/441752/Async-MVVM-Modern-UI>

2 <https://martinfowler.com/bliki/JavascriptPromise.html>

REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

Oficio N° 8592

Ref. Expediente N° NC2017/0003239

	- El primer botón vincula el comando con <i>'GetFilesCommand'</i> y enlaza <i>IsEnabled</i> con <i>IsIdle (...)</i> (Sección View). (...) El comando <i>'GetFilesCommand'</i> para leer archivos (...) (Sección Properties and Fields).
El proveedor del modelo de vista configurado para solicitar (810) de forma asíncrona la información correspondiente a los datos de vista especificados desde el modelo de datos.	El modelo de vista implementa una versión personalizada de <i>INotifyPropertyChanged</i> para ser asincrónico y actualiza la UI (Interfaz de Usuario) cuando es capaz (Sección ViewModel).
El proveedor de modelo de datos configurado para obtener un conjunto de datos incluyendo la información de una fuente de datos (116) y devolver (1022) la información al modelo de vista, y el modelo de vista configurado además para devolver una respuesta que comprende los datos de vista especificados al objeto vista en base a la información.	El modelo es muy simple, por lo general prefiero agregar un presentador para la vista en lugar de agregar un convertidor, use la opción que desee. Tenga en cuenta que en W8RP el formato de fecha ha cambiado a <i>GetDateTimeFormats</i> (Sección Model). El modelo de vista implementa una versión personalizada de <i>INotifyPropertyChanged</i> para ser asincrónico y actualiza la UI (Interfaz de Usuario) cuando es capaz (Sección ViewModel).

De acuerdo con la tabla anterior, las características esenciales del sistema de la reivindicación 11 habían sido divulgadas previamente en el documento D1, por lo tanto, la materia reivindicada no es nueva.

Asimismo, el documento D1 divulga las características de las reivindicaciones dependientes así:

- Reivindicación 12: Divido *'GetFilesAsync'* para mostrar que simplemente está llamando a un método dentro de *Task.Run*. Ese método necesita tener *async* como prefijo para saber que lo quieres en el 'segundo plano'. *'addfile'* es la acción que agrega archivos mientras *_cancel* no está activado, como puede ver, no necesito ninguna lógica especial para saber el valor de cancelar (Sección The heart).
- Reivindicación 13: El modelo de vista implementa una versión personalizada de *INotifyPropertyChanged* para ser asincrónico y actualiza la UI (Interfaz de Usuario) cuando es capaz (Sección ViewModel). El modelo es muy simple, por

REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

Oficio N° **8592**

Ref. Expediente N° NC2017/0003239

lo general prefiero agregar un presentador para la vista en lugar de agregar un convertidor, use la opción que desee. Tenga en cuenta que en W8RP el formato de fecha ha cambiado a *GetDateTimeFormats* (Sección Model).

Las reivindicaciones dependientes 12 a 13 no mencionan alguna característica que haga nuevo al sistema de la reivindicación 11, por lo cual se considera que no son nuevas.

Nivel Inventivo (Art. 18 D486)

La reivindicación 1 consiste en: “*Un método realizado al menos en parte en al menos una máquina, que comprende: (...)*” (Radicado N° NC2017/0003239 del 2 de junio de 2017).

TABLA 2. Comparación de las características técnicas esenciales, con las del estado de la técnica:

Características esenciales	D1	D2
Recibir (802), en un modelo de vista (102), una petición de una vista (106) para datos de vista especificados.	Finalmente llegamos al XAML, he creado una página simple con los botones de comando, <i>GridView</i> y <i>TextBlock</i> para el elemento seleccionado. Cuida lo siguiente: - Creo una instancia de <i>ViewModel</i> en <i>Page.DataContext</i>. - El primer botón vincula el comando con '<i>GetFilesCommand</i>' y enlaza <i>IsEnabled</i> con <i>IsIdle (...)</i> (Sección View). (...) El comando ' <i>GetFilesCommand</i> ' para leer archivos (...) (Sección Properties and Fields).	No se menciona.
En el modelo de vista (102), hacer una o más peticiones asincrónicas a un modelo de datos para la información correspondiente a los datos de vista especificados.	El modelo de vista implementa una versión personalizada de <i>INotifyPropertyChanged</i> para ser asincrónico y actualiza la UI (Interfaz de Usuario) cuando es capaz (Sección ViewModel). Divido ' <i>GetFilesAsync</i> ' para mostrar que simplemente está llamando a un método dentro de <i>Task.Run</i> . Ese método necesita tener <i>async</i> como prefijo para saber que lo	No se menciona.



**REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO**

Oficio N° 8592

Ref. Expediente N° NC2017/0003239

	quieres en el 'segundo plano'. 'addfile' es la acción que agrega archivos mientras <u>cancel</u> no está activado, como puede ver, no necesito ninguna lógica especial para saber el valor de cancelar (Sección The heart).	
Devolver (804) una promesa del modelo de vista de la vista en respuesta a la solicitud de los datos de vista especificados.	No se menciona.	En Javascript, las promesas son objetos que representan el resultado pendiente de una operación asíncrona. Puede usarlos para programar más actividades después de que la operación asíncrona haya finalizado proporcionando una devolución de llamada (Párrafo 1).
Obtener un conjunto de datos (1010, 1018) que incluye la información en el modelo de datos.	El comando 'GetFilesCommand' para leer archivos (...) (Sección Properties and Fields).	No se menciona.
Proveer (1022) la información del modelo de datos para el modelo de vista en respuesta a una o más solicitudes asíncronas.	El modelo de vista implementa una versión personalizada de <i>INotifyPropertyChanged</i> para ser asíncrono y actualiza la UI (Interfaz de Usuario) cuando es capaz (Sección ViewModel).	No se menciona.
Devolver (904) los datos de vista especificados, basándose en la información devuelta al modelo de vista, desde el modelo de vista a la vista para cumplir la promesa.	Para actualizar la UI cuando usa asíncrono, debe usar el despachador de ventana actual. Como puede ver, creo un método asíncrono que actualiza la UI llamada 'UiThreadAction' y varios métodos como 'PropertyChangedAsync' que llama 'PropertyChanged' dentro de la acción del despachador (Sección Implementing INotifyPropertyChanged).	No se menciona.

El documento D1 se considera el estado de la técnica más cercano a la invención definida en la reivindicación 1 porque divulga un ejemplo del uso de MVVM con Windows 8 Modern UI con enlace de datos, comandos y propiedades utilizando la

**REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO**

Oficio N° 8592

Ref. Expediente N° NC2017/0003239

metodología asincrónica y manteniendo el modo de diseño con plantillas de datos (Sección Introducción).

La diferencia entre la invención, definida en la reivindicación 1 y el método de D1 consiste en que en el método de la reivindicación 1 incluye el uso de promesas en el modelo de vista.

El efecto que logra el uso de promesas es gestionar las respuestas de las solicitudes de datos especificados.

Por lo tanto, el problema técnico objetivo que pretende resolver esta invención se puede formular así: “¿cómo modificar el método conocido en D1 para gestionar las respuestas de las solicitudes de datos especificados?”.

Sin embargo, el uso de promesas para gestionar las respuestas de las solicitudes de datos especificados ya está divulgado en el documento D2 que se refiere a que en Javascript, las promesas son objetos que representan el resultado pendiente de una operación asincrónica. Puede usarlos para programar más actividades después de que la operación asíncrona haya finalizado proporcionando una devolución de llamada (Párrafo 1).

En consecuencia, la persona del oficio normalmente versada en la materia incluiría el uso de promesas para gestionar las respuestas de las solicitudes de datos especificados mencionado en documento D2 en el método divulgado en el documento D1 para llegar así al objeto de la reivindicación 1 de la solicitud. Por lo cual se considera obvia.

Asimismo, los documentos D1 y D2 divulgan las características de las reivindicaciones dependientes así:

- Reivindicación 3: En Javascript, las promesas son objetos que representan el resultado pendiente de una operación asincrónica. Puede usarlos para programar más actividades después de que la operación asíncrona haya finalizado proporcionando una devolución de llamada (D2, Párrafo 1).
- Reivindicación 9: En Javascript, las promesas son objetos que representan el resultado pendiente de una operación asincrónica. Puede usarlos para



**REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO**

Oficio N° 8592

Ref. Expediente N° NC2017/0003239

programar más actividades después de que la operación asíncrona haya finalizado proporcionando una devolución de llamada (D2, Párrafo 1).

- Reivindicación 10: (...) Creo una instancia de ViewModel en Page.DataContext (...) (Sección View). (...) Aquí inicializo los Comandos, el despachador y la colección (...) (D1, Sección Constructor).

Las reivindicaciones dependientes 3 y 9 a 10 no mencionan alguna característica que, junto con las características de la reivindicación independiente, aporte nivel inventivo al método de la reivindicación 1, por lo cual se considera que no tienen nivel inventivo.

La reivindicación 17 consiste en: “*Un método que comprende: (...)*” (Radicado N° NC2017/0003239 del 2 de junio de 2017).

TABLA 3. Comparación de las características técnicas esenciales, con las del estado de la técnica:

Características esenciales	D1	D2
El retorno (804) de una primera promesa del modelo de vista a la vista.	No se menciona.	En Javascript, las promesas son objetos que representan el resultado pendiente de una operación asíncrona. Puede usarlos para programar más actividades después de que la operación asíncrona haya finalizado proporcionando una devolución de llamada (Párrafo 1).
Hacer (810) una solicitud asíncrona a un modelo de datos para información correspondiente a los datos de vista especificados.	El modelo de vista implementa una versión personalizada de <i>INotifyPropertyChanged</i> para ser asíncrono y actualiza la UI (Interfaz de Usuario) cuando es capaz (Sección ViewModel). Divido 'GetFilesAsync' para mostrar que simplemente está llamando a un método dentro de <i>Task.Run</i> . Ese método necesita tener <i>async</i> como prefijo para saber que lo quieres en el 'segundo plano'. 'addfile' es la acción que agrega archivos mientras <i>_cancel</i> no está activado, como puede ver,	No se menciona.

REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

Oficio N° 8592

Ref. Expediente N° NC2017/0003239

	no necesito ninguna lógica especial para saber el valor de cancelar (Sección The heart).	
Recibir (812) una segunda promesa del modelo de datos para el modelo de vista.	No se menciona.	En Javascript, las promesas son objetos que representan el resultado pendiente de una operación asíncrona. Puede usarlos para programar más actividades después de que la operación asíncrona haya finalizado proporcionando una devolución de llamada (Párrafo 1).
Hacer (1010) una petición asíncrona desde el modelo de datos a una fuente de datos para un conjunto de datos correspondientes a la información.	Divido 'GetFilesAsync' para mostrar que simplemente está llamando a un método dentro de <i>Task.Run</i> . Ese método necesita tener <i>async</i> como prefijo para saber que lo quieres en el 'segundo plano'. 'addfile' es la acción que agrega archivos mientras <i>_cancel</i> no está activado, como puede ver, no necesito ninguna lógica especial para saber el valor de cancelar (Sección The heart).	No se menciona.
Obtener (1018) el conjunto de datos en respuesta a la solicitud a la fuente de datos en el modelo de datos, el conjunto de datos que incluye la información.	El comando 'GetFilesCommand' para leer archivos (...) (Sección Properties and Fields).	No se menciona.
Devolver (1022) la información del modelo de datos al modelo de vista para cumplir con la segunda promesa.	El modelo de vista implementa una versión personalizada de <i>INotifyPropertyChanged</i> para ser asíncrono y actualiza la UI (Interfaz de Usuario) cuando es capaz (Sección ViewModel).	No se menciona.
Devolver (904) los datos de vista especificados, basándose en la información devuelta al modelo de vista, desde	Para actualizar la UI cuando usa asíncrono, debe usar el despachador de ventana actual. Como puede ver, creo un método asíncrono que actualiza la UI llamada 'UiThreadAction' y varios métodos como 'PropertyChangedAsync' que llama	No se menciona.

**REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO**

Oficio N° 8592

Ref. Expediente N° NC2017/0003239

el modelo de vista a la vista para cumplir la primera promesa.	'PropertyChanged' dentro de la acción del despachador (Sección Implementing INotifyPropertyChanged).	
--	--	--

El documento D1 se considera el estado de la técnica más cercano a la invención definida en la reivindicación 17 porque divulga un ejemplo del uso de MVVM con Windows 8 Modern UI con enlace de datos, comandos y propiedades utilizando la metodología asincrónica y manteniendo el modo de diseño con plantillas de datos (Sección Introducción).

La diferencia entre la invención, definida en la reivindicación 17 y el método de D1 consiste en que en el método de la reivindicación 17 incluye el uso de promesas en el modelo de vista.

El efecto que logra el uso de promesas es gestionar las respuestas de las solicitudes de datos especificados.

Por lo tanto, el problema técnico objetivo que pretende resolver esta invención se puede formular así: “¿cómo modificar el método conocido en D1 para gestionar las respuestas de las solicitudes de datos especificados?”.

Sin embargo, el uso de promesas para gestionar las respuestas de las solicitudes de datos especificados ya está divulgado en el documento D2 que se refiere a que en Javascript, las promesas son objetos que representan el resultado pendiente de una operación asincrónica. Puede usarlos para programar más actividades después de que la operación asíncrona haya finalizado proporcionando una devolución de llamada (Párrafo 1).

En consecuencia, la persona del oficio normalmente versada en la materia incluiría el uso de promesas para gestionar las respuestas de las solicitudes de datos especificados mencionado en documento D2 en el método divulgado en el documento D1 para llegar así al objeto de la reivindicación 17 de la solicitud. Por lo cual se considera obvia.

Asimismo, los documentos D1 y D2 divulgan las características de las reivindicaciones dependientes así:

**REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO**

Oficio N° 8592

Ref. Expediente N° NC2017/0003239

- Reivindicación 18: El comando '*GetFilesCommand*' para leer archivos (...) (Sección Properties and Fields). En Javascript, las promesas son objetos que representan el resultado pendiente de una operación asincrónica. Puede usarlos para programar más actividades después de que la operación asincrónica haya finalizado proporcionando una devolución de llamada (Párrafo 1).
- Reivindicación 19: El modelo de vista implementa una versión personalizada de *INotifyPropertyChanged* para ser asincrónico y actualiza la UI (Interfaz de Usuario) cuando es capaz (D1, Sección ViewModel).

La reivindicación dependiente 19 no mencionan alguna característica que, junto con las características de la reivindicación independiente, aporte nivel inventivo al método de la reivindicación 17, por lo cual se considera que no tiene nivel inventivo.

8. Conclusión

Los requisitos de Patentabilidad de la presente solicitud están afectados así:

Ítem	Documento	Reivindicaciones afectadas	Requisito que afecta
D1	Artículo	11 a 13	Novedad
		1, 10 y 17 a 19	Nivel Inventivo
D2	Artículo	1, 3, 9 y 17 a 18	Nivel Inventivo

En cumplimiento del artículo 45, de la Decisión 486 de la Comisión de la Comunidad Andina, se le indica que debe dar respuesta dentro del término de sesenta (60) días contados a partir de la fecha de notificación de la presente comunicación. En caso de no dar respuesta al presente requerimiento, o si a pesar de la respuesta subsistieran los impedimentos para la concesión, se denegará la patente.

Si como respuesta a este requerimiento el solicitante llegara a modificar: i) el capítulo reivindicatorio, ii) la descripción de la invención a proteger, iii) o presente un nuevo capítulo reivindicatorio, y desee que se realice un nuevo examen de patentabilidad a esta modificación, no deberá pagar la tasa de modificación voluntaria pero sí la

Página 12 de 13

REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

Oficio N° 8592

Ref. Expediente N° NC2017/0003239

correspondiente a la tasa de examen de patentabilidad. En todo caso, el número de requerimientos por no patentabilidad estará supeditado a que el término del estudio de patentabilidad no sea superior a 18 meses, contados desde la publicación de la solicitud de patente de invención. Adicionalmente, se le recuerda al solicitante que si como respuesta a este requerimiento llegara a modificar el capítulo reivindicatorio y éste contenga más de diez (10) reivindicaciones deberá pagar la tasa por reivindicación adicional.

Se invita al solicitante se sirva indicar, si así lo considera, su renuncia al término faltante para los sesenta (60) días.

Notifíquese el presente oficio conforme a lo dispuesto en el Título I, Actuaciones ante la Superintendencia de Industria y Comercio; Capítulo sexto, publicación, comunicación, notificación y ejecutoria de las decisiones de la Superintendencia de Industria y Comercio, informándoles que contra el mismo no procede el recurso de la vía gubernativa.

Atentamente,



OLGA ROSA GONZÁLEZ FONSECA
DIRECTORA DE NUEVAS CREACIONES (E)

Elaborado: CARLOS EDUARDO GUEVARA NICHÓY

Revisado: SONIA ANDREA DÍAZ PIRAZÁN