**Implementando la vista**

**Visión de conjunto**

Este documento describe cómo implementar los componentes de vista de una aplicación en términos genéricos. No describe cómo implementar características de IU específicas. Para obtener esta información, consulte los temas individuales en el Capítulo 4: Implementación de características de IU específicas.

**Contenido**

* [Diseñando la interfaz de usuario](https://translate.googleusercontent.com/translate_f#design)
* [Páginas](https://translate.googleusercontent.com/translate_f#page)
* [Componentes reutilizables](https://translate.googleusercontent.com/translate_f#region)
* [Conjuntos de atributos](https://translate.googleusercontent.com/translate_f#attr)
* [Parámetros de URL: Tokens, Encriptación, Codificación](https://translate.googleusercontent.com/translate_f#urlparams)
* [Hojas de estilo](https://translate.googleusercontent.com/translate_f#css)
* [Accesibilidad](https://translate.googleusercontent.com/translate_f#access)
* [Internacionalización](https://translate.googleusercontent.com/translate_f#internationalization)
* [Interacción del modelo](https://translate.googleusercontent.com/translate_f#model)
* [Menús y seguridad de página](https://translate.googleusercontent.com/translate_f#menu)

**Requisito de lectura**

Este documento asume que ha leído lo siguiente en la Guía del Desarrollador de OA Framework:

* [Edificio "¡Hola, mundo!"](https://translate.googleusercontent.com/gs/gs_hello.htm)
* [JSP Application Primer](https://translate.googleusercontent.com/ess/ess_jspprimer.htm)
* [Anatomía de una página de Marco OA](https://translate.googleusercontent.com/ess/ess_anatomy.htm)
* [OA Framework State Management](https://translate.googleusercontent.com/ess/ess_state.htm)
* [Implementando el Modelo](https://translate.googleusercontent.com/build_model.htm)

**Diseñando la interfaz de usuario**

Todas las aplicaciones de OA Framework deben ser diseñados de acuerdo con el [Oracle Browser Look-and-feel (BLAF) Directrices de interfaz de usuario](https://translate.google.com/translate?hl=es&prev=_t&sl=en&tl=es&u=http://uistandards/blaf/) [ [Versión OTN](https://translate.google.com/translate?hl=es&prev=_t&sl=en&tl=es&u=http://otn.oracle.com/tech/blaf/) ] (tenga en cuenta que se puede encontrar información acerca de cómo hacer el diseño de interfaz de usuario, incluyendo instrucciones sobre cómo utilizar UIX como la creación de prototipos herramienta, en este sitio). El cumplimiento de estas pautas produce varios beneficios:

* Los usuarios interactúan con una interfaz consistente, predecible y atractiva mientras navegan de una aplicación a la siguiente.
* Si está diseñado adecuadamente para su público objetivo, es probable que sus aplicaciones sean intuitivas y utilizables, ya que las propias directrices de la interfaz de usuario se han probado con facilidad de uso. Además, los resultados de las pruebas de usabilidad del equipo del producto se consideran y abordan de forma continua.
* Los beans UIX subyacentes que se extienden en OA Framework implementan las pautas de UI. Si su aplicación cumple con las pautas de UI, normalmente puede usar los beans web de OA "de fábrica" ​​sin un extenso esfuerzo programático.

Para las aplicaciones de Oracle E-Business Suite, cualquier desviación de estas pautas debe ser aprobada por el equipo corporativo de Diseño y Usabilidad de UI. En general, las desviaciones válidas son impulsadas por requisitos de productos únicos (que a menudo introducen diseños que eventualmente se incorporan a los estándares generales en beneficio de todos los equipos de productos). No es aceptable ignorar los estándares individuales simplemente porque no está de acuerdo con ellos.

**Diseño de aplicaciones móviles**

OA Framework admite la implementación de aplicaciones especialmente diseñadas en plataformas inalámbricas (inicialmente, PDA handheld devices como Palm Pilot). Para obtener información sobre el diseño, construcción, pruebas y despliegue de aplicaciones móviles, consulte la [aplicaciones Mobile](https://translate.googleusercontent.com/feat/feat_mobile.htm) documento en el Capítulo 4. Tenga en cuenta que usted debe seguir leyendo sobre lo básico de desarrollo de OA Framework ya que esta información se aplica al desarrollo móvil.

**Páginas**

Los pasos básicos para la creación de páginas, las regiones y los elementos se describen en [el Capítulo 2: Construcción, "Hola, mundo!"](https://translate.googleusercontent.com/gs/gs_hello.htm) Y en la ayuda JDeveloper OA extensión. Para obtener información sobre la implementación de regiones y elementos específicos de características, consulte el Capítulo 4.

**Codificación de cumplimiento de normas**

Antes de crear cualquier componente OA, debe leer cuidadosamente los siguientes documentos:

* [Capítulo 8: Naming Framework / File / Package / Directory Structure Standards](https://translate.googleusercontent.com/cs/cs_filedir.htm)
* [Capítulo 8: OA marco Ver estándares de codificación](https://translate.googleusercontent.com/cs/cs_viewgeneral.htm) (esto incluye normas para los componentes que se crean de forma declarativa)

**Propiedades de la región del diseño de página clave**

Siempre que se crea una región **pageLayout** , preste especial atención a las siguientes propiedades:

* AutoFooter - siempre se establece en **true** por lo que la privacidad y derechos de autor enlaces Aplicaciones-estándar hacen en su página.
* Ayudar a Target - si desea mostrar texto de ayuda de esta página cuando el usuario selecciona el botón de ayuda mundial en su página, debe especificar el objetivo de la ayuda (a menudo el nombre del archivo de ayuda) para mostrar aquí. Ver [Botones globales](https://translate.googleusercontent.com/feat/feat_buttonsglobal.htm) para información adicional sobre esta.
* Definición AM - aquí es donde se especifica el módulo de aplicación de la raíz para su página. Tenga en cuenta que debe especificar el nombre completo, tales como: **oracle.apps.fnd.framework.toolbox.tutorial.server.SearchAM**
* Nombre de la función - siempre especificar la función de garantizar a la página (ver la [Menús](https://translate.googleusercontent.com/translate_f#menu) sección de abajo para más información).
* Título de la ventana - este valor se muestra el texto en el título de la ventana del navegador.
* Título - este valor de texto hace que el encabezado de la página. También se usa como el nombre de la página si se muestran las migajas de pan (véase el [Capítulo 4: Pan rallado](https://translate.googleusercontent.com/feat/feat_breadcrumbs.htm) para obtener información detallada acerca de esta característica).
* Formulario - Siempre ponga esto en **cierto** para una región **pageLayout** (esta es la configuración por defecto), y nunca agregue los granos de forma adicional por hijos a su página. El Marco de OA admite solo 1 formulario por página.
* pageLayout Componentes - como se menciona en [Anatomía de una OA marco de páginas](https://translate.googleusercontent.com/ess/ess_anatomy.htm) , páginas incluyen especial "con nombre" componentes (también llamados niños nombrados), uno de los cuales es la imagen de marca . Para asociar una imagen de marca con su página, seleccione la región o el nodo pageLayout pageLayout componentes en el panel Estructura y utilice el botón derecho del ratón para seleccionar Nuevo> productBranding en el menú contextual. JDeveloper crea automáticamente un elemento de imagen cuya imagen URI propiedad debe estar establecido en <imageName> .gif.

**Propiedades clave del elemento**

Dado que cada tipo de elemento tiene un conjunto distinto de propiedades (que a menudo incluyen propiedades que son exclusivas del elemento), es imposible introducir todas y cada una de las propiedades relevantes. En cambio, esta sección presenta algunas de las propiedades comunes que debe comprender, ya que las establecerá con frecuencia.

* Se extiende - Para los artículos (y regiones, así), esto indica que el nuevo elemento se extiende un elemento existente. Esto es discutido con más detalle abajo.
* Conjunto de atributos - Un conjunto de propiedades con nombre aplicado para configurar rápidamente un elemento. Esto es discutido con más detalle abajo.
* URI de destino - Para los artículos que admiten la navegación, este es el destino de navegación. Por ejemplo:? OA.jsp page = / oracle / aplicaciones / enc / marco / caja de herramientas / tutorial / webui / PoDetailsPG y retainAM = Y.
* (Acción de clientes) Tipo de acción - Indica si el elemento es capaz de enviar el formulario, o causar una representación parcial de la página (PPR) evento. Véase el Capítulo 4 de [la interfaz de usuario dinámica](https://translate.googleusercontent.com/feat/feat_ppr.htm) y [declarativa Enviar formulario](https://translate.googleusercontent.com/feat/feat_submitform.htm) para obtener información adicional acerca de estas características.
* Clase CSS - Indica qué clase de hoja de estilo en cascada para aplicar al elemento (para muchos artículos, UIX establece este valor de forma automática a cumplir con las directrices de interfaz de usuario BLAF). Esto es discutido con más detalle abajo.
* Rendido - Indica si el objeto correspondiente está incluido en la jerarquía de Web Bean, y el código HTML que UIX envía al navegador para la representación. Para la mayoría de los elementos, esto indica si un elemento se muestra o no, pero para algunos elementos que en realidad nunca se muestran (como un campo de desarrollador oculto), esto indica si el objeto existe en la página.
* Ver Instancia - Para elementos que se unen a una vista de objeto subyacente para la lectura y la escritura (si es necesario) de datos, esto identifica la instancia del objeto de vista (en el contexto de un módulo de aplicación que contiene) al que se une el elemento.
* Ver Atributo - Este es un atributo de la instancia de vista al que se une el elemento.
* Personalización admin - Indica si la propiedad es el administrador del sistema personalizable. Ver la [Guía de Personalización OA marco](https://translate.googleusercontent.com/persguide/persguide.htm) para obtener información adicional sobre la personalización.
* Personalización de usuario - Indica si la propiedad es personalizable usuario. Ver la [Guía de Personalización OA marco](https://translate.googleusercontent.com/persguide/persguide.htm) para obtener información adicional sobre la personalización.
* Valor inicial - El valor por defecto para el elemento (tenga en cuenta que esto puede ser personalizado por los clientes). Ver el [Incumplimiento](https://translate.googleusercontent.com/translate_f#default) de los siguientes temas para obtener información adicional.

**El lenguaje de expresión posible más simple (SPEL)**

Para las propiedades seleccionadas, OA Framework admite el uso de expresiones SPEL para vincular rápidamente la propiedad a una fuente de datos subyacente que proporciona el valor de la propiedad. Por ejemplo, se puede enlazar la propiedad rendida de un botón para un atributo de objeto de vista para determinar si se debe ocultar o mostrar cuando la página se representa. La sintaxis de SPEL para esta propiedad se ve así:

$ {OA. *<ViewInstanceName>.* *<ViewAttributeName>}*

**Consejo:** SPEL es un lenguaje de expresiones estándar de la industria incluido en el JSP Standard Tag Library (JSTL). Si está interesado en obtener más información sobre esto (aunque esto no es necesario para el uso limitado en el Marco de OA), la búsqueda del "lenguaje de expresión más simple posible (SPEL)" en la web genera numerosos recursos.

El uso de expresiones SPEL se describe en detalle en el capítulo 4 de [la interfaz de usuario dinámica](https://translate.googleusercontent.com/feat/feat_ppr.htm) .

**Componentes reutilizables**

Una de las ventajas clave del entorno de desarrollo declarativo de componentes OA es la facilidad con la que puede reutilizar definiciones comunes de página, región y artículo.

Vea la [sección de regiones compartidas](https://translate.googleusercontent.com/translate_f#shared) en este capítulo para ver cómo se crea un módulo común con su propio módulo de la lógica y la aplicación y utilizar.

**Regiones compartidas**

**Cumplir con los estándares de reutilización**

Si desea crear una región compartida, debe cumplir con los siguientes estándares.

**Nota:** una región compartida puede incluir uno o más subregiones.

* La región de nivel superior (compartida) se debe guardar en su propio archivo XML.
* Puede diseñar su región compartida para aceptar valores de una región en uso. Los valores pueden ser pasados a la solicitud utilizando cualquiera de los métodos descritos en [la OA de Gestión de Estado Marco](https://translate.googleusercontent.com/ess/ess_state.htm) , o como un valor almacenado en caché en la transacción de la página (también se describe en el documento *Gestión del Estado).*
* La región compartida debe implementarse para fallar con elegancia. Por ejemplo, si los parámetros apropiados no se pasan desde una región en uso, la región compartida debe establecer valores predeterminados aceptables o generar una excepción significativa (y documentada).
* Si el ámbito de la región se establece en **Público** (consulte *Creación de una región compartida* continuación para obtener información adicional acerca de este):
  + La región de nivel superior debe tener su propio módulo de aplicación. El módulo de la aplicación debe incluir solo aquellos objetos de vista que sean relevantes para la región compartida.
  + La región de nivel superior debe tener su propio controlador. Puede asociar controladores adicionales con subregiones según sea necesario.
* La región compartida debe estar completamente documentada como se describe a continuación.

**Crear una región compartida**

Para crear una región compartida :

1. En JDeveloper Navigator, seleccione el proyecto OA donde desea crear su región.
2. En el menú principal, seleccione Archivo> Nuevo para abrir la nueva galería de objetos.
3. En el árbol de categorías, expanda el nodo de nivel Web y seleccione Componentes OA.
4. En la lista Elementos, seleccione Región para abrir la ventana nueva región.
5. Introduzca un nombre y un paquete de acuerdo con las [Normas Marco de la OA de archivos](https://translate.googleusercontent.com/cs/cs_filedir.htm) , y especificar el estilo de la región que desea crear (seleccionar su estilo cuidadosamente, ya que no se puede cambiar una vez que se crea la región).Seleccione Aceptar para guardar crear su <paquete>. <Nombre> .xml documento OA componentes.
6. Seleccione la nueva región en el panel Estructura JDeveloper y establezca la propiedad de documentación comentario con el contenido descrito a continuación.
7. Establecer la propiedad Ámbito de acuerdo con su uso previsto de la región compartida. Por ejemplo, para una región privada para uso exclusivo dentro del paquete actual, seleccione la opción **del paquete actual** (nota que se puede limitar el acceso a sólo el producto si lo desea). Alternativamente, ponga esto en **público** a dejar a nadie en ella en cualquier página.
8. Establezca las otras propiedades en su región.
9. Cuando es hora de empaquetar y enviar su región compartida, debe generar documentación HTML Javadoc-como de la región mediante el generador de documentación de desarrolladores de la herramienta en JDeveloper (véase Introducción a la extensión de la OA> Herramientas de línea de comandos para la extensión de la OA> Sobre el desarrollador Generador de documentación de la herramienta en la ayuda en línea de Oracle JDeveloper para obtener información adicional).

**Advertencia:** Preste mucha atención a las normas de nomenclatura de la [Normas Marco de archivos OA](https://translate.googleusercontent.com/cs/cs_filedir.htm) documento cuando el nombramiento de su región compartida y cualquiera de sus elementos y subregiones. Dado que todos los componentes OA en una página deben tener un nombre único, el cumplimiento de los estándares de nomenclatura ayudará a garantizar que su región reutilizable se pueda volver a utilizar realmente.

**Nota:** Debido a las restricciones de nomenclatura, una sola región no se puede usar más de una vez en una página. Esta restricción se eliminará en algún momento en el futuro.

*Documentación Comentario Contenido*

Debe añadir el contenido siguiente t en la propiedad de documentación Comentario de la región:

/ \*\*

\* Controlador para: <nombre de página / región compartida, incluido el paquete>

\*

\* Alcance: **<privada** (para uso poseer equipo de producto - este es el ámbito predeterminado),

\* **Público** (para su uso por cualquier persona) u **Oracle** (por Suite de Oracle E-Business utilización desarrollo solamente)>

\*

\* Uso: <describe el propósito y el uso del componente, incluidos los mensajes de error que

\* podría ser levantado>

\*

\* @param <parámetro de objeto 1> <descripción del parámetro de objeto 1 y valores aceptables>

\* @param <objeto parámetro 2> <objeto parámetro descripción 2 y valores aceptables>

\* @param <parámetro de objeto N> <parámetro de objeto N descripción y valores aceptables>

\* @see <opcionalmente incluye referencias a otros objetos como otros controladores para niños compartidos, si corresponde>

\* /

**Nota:** Cuando se describe un parámetro, indicar claramente si el parámetro debe ser transmitida a la región en la solicitud, o en la transacción módulo de aplicación.

El siguiente ejemplo ilustra el contenido apropiado para un controlador de componente compartido:

Controlador para: ford.oracle.apps.xyz.webui.FordDistributorAddressRN

Alcance: Público

Uso: Implementa una región de dirección localizada para distribuidores.

@param: id ID disitributor ID que se pasa en la solicitud;   
 requerido para inicializar la región

@param: locale locale que se pasa en la solicitud; requerida para   
 localizar correctamente la dirección

@param: enviar aprobado en la solicitud; si se especifica, esta región

cometer sus cambios

**Extender una región reutilizable**

Como se mencionó en [Anatomía de un marco de OA](https://translate.googleusercontent.com/ess/ess_anatomy.htm) página, utilizar una región compartida en su página, sólo tiene que extenderla:

1. En el panel Estructura de JDeveloper, seleccione la región a la que desea agregar la región compartida.
2. Utilice el botón derecho del ratón para seleccionar Nuevo> Región del menú contextual.
3. Coloca el cursor en la nueva región se extiende campo en el inspector de propiedades y seleccione el botón ... para abrir el navegador de componentes. Buscar o navegar como lo prefiere, seleccione la región que desea extender en la lista de resultados y seleccione Aceptar para hacer su elección.
4. JDeveloper entra el nombre completo de la región compartida en el campo se extiende (por ejemplo, / Oracle / aplicaciones / enc / marco / caja de herramientas / tutorial / webui / PoSummaryRN). Observe que puede editar la mayor parte de las propiedades de la zona de fondo que ha creado (no se puede cambiar su estilo), pero la región extendida no puede ser modificado. De hecho, sus contenidos se muestran con una fuente gris en el panel Estructura y en el Inspector de propiedades.
5. Guarda tu trabajo.

**Consejo:** Cuando se agrega una región compartida con su propio módulo de aplicación a su página, el Marco OA anida automáticamente el módulo de aplicación de componentes por debajo de su módulo de aplicación radicular. No necesita crear ninguna relación explícita de tiempo de diseño entre estos módulos de aplicación.

Para borrar una extensión, coloque el cursor en el campo amplía y seleccione Definir la propiedad del inspector botón de la barra de defecto.

**Caso especial: Lista de valores (LOV)**

Aunque los pasos de implementación se describen en otro lugar (véase la [Lista de Valores](https://translate.googleusercontent.com/feat/feat_lov.htm) tema en el Capítulo 4), vale la pena señalar que la lista de valores puede ser implementado como una clase especial de región compartida (también se puede crear una lista de valores de un solo uso):

* Puede crear una Lista de valores reutilizable utilizando el mismo procedimiento que para cualquier otra región compartida, aunque no requiere un controlador asociado.
* Cuando desee usar el LOV compartido en una página, no lo extienda como se describe arriba. En su lugar, se establece LOV externa del campo de página de base   propiedad y configurar el intercambio de datos entre la página base y el LOV.

**Páginas compartidas**

Una página es en realidad una región compartida cuya región de nivel superior pasa a ser designada como un componente **pageLayout.** Como tal, una página compartida debe seguir todos los estándares de creación de la región y las instrucciones descritas anteriormente.

* Si desea reutilizar una página independiente o un flujo de página, simplemente cree una nueva función de menú y apúntela a la página principal (las funciones del menú se describen a continuación).
* Si desea insertar una página compartida en otro flujo de la página con un módulo de aplicación raíz diferente, debe crear una nueva página, y luego extender el contenido de la página compartida por debajo de la región **pageLayout.** Recuerde establecer el módulo de aplicación raíz correcto en su nueva página.

**Artículos compartidos**

También puede extender artículos individuales desde cualquier región, aunque le recomendamos que coloque los artículos que desea compartir en una región reutilizable. Compartir la región contenedora ayudará a garantizar que alguien no cambie las propiedades en ningún elemento arbitrario sin darse cuenta de que el cambio podría tener un impacto adverso en las páginas que usan el elemento.

* En el panel Estructura de JDeveloper, seleccione el elemento que extenderá otro elemento.
* Coloca el cursor en el elemento de Extiende campo en el inspector de propiedades y seleccione el botón ... para abrir el navegador de componentes. Buscar o navegar como lo prefiere, seleccione el elemento que desea extender la lista de resultados y seleccione Aceptar para hacer su elección.
* JDeveloper entra el nombre completo del artículo en el campo se extiende (por ejemplo, /oracle/apps/fnd/framework/toolbox/tutorial/webui/PoSummaryRN.OrderTotal). Observe que puede editar la mayor parte de las propiedades del elemento extendido, pero no se puede cambiar su estilo de artículos.
* Guarda tu trabajo.

**Compartir Logística**

Los equipos de productos de Oracle E-Business Suite deben producir ARU internas para compartir objetos reutilizables. Para los equipos con una gran cantidad de objetos compartibles que todavía están cambiando rápidamente, contáctese con el equipo de OA Framework para investigar la posibilidad de aprovechar el proceso de congelamiento diario de OA Framework.

**Conjuntos de atributos**

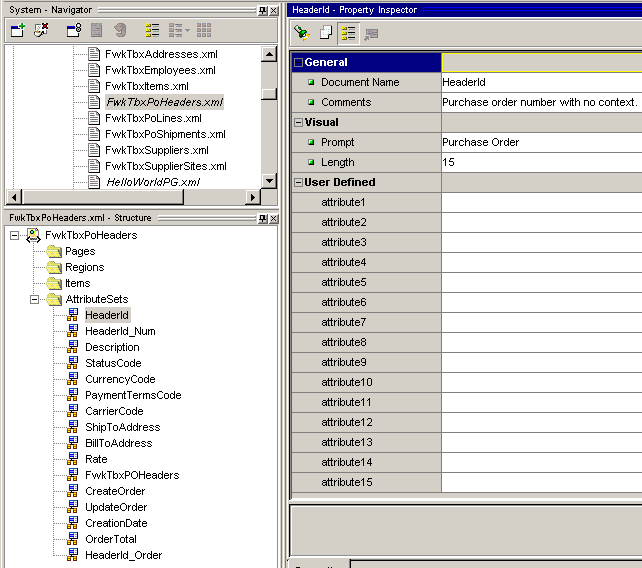
Los conjuntos de atributos son colecciones de propiedades reutilizables (solicitud, longitud máxima de visualización, tipo de datos, etc., según corresponda para el tipo de conjunto de atributos) que puede utilizar cualquier tipo de componente OA, incluidas regiones, elementos y otros conjuntos de atributos. Están diseñados para facilitar la reutilización de estos componentes a través de Oracle E-Business Suite, lo que supone un importante ahorro de costes tanto para Oracle como para sus clientes:

* Oracle ahorra en costos de traducción y mantenimiento.
* Los clientes pueden realizar personalizaciones rápidas y globales en sus aplicaciones. Además, menos elementos de IU se traducen en menos consumo de memoria de nivel medio, y en última instancia, esto significa un mejor rendimiento y escalabilidad.

En términos generales, los conjuntos de atributos están organizados en paquetes de componentes OA (archivos de paquetes XML individuales), donde tiene un archivo de paquete por tabla de base de datos en su aplicación:

* El nombre del paquete coincide con el nombre de la tabla de la base de datos subyacente sin guiones bajos. Por ejemplo, en el marco de la OA caja de herramientas, tenemos una tabla llamada FWK\_TBX\_PO\_HEADERS. El paquete de establecer el atributo correspondiente se denomina FwkTbxPoHeaders.
* Los conjuntos de atributos individuales se crean para cada columna visualizable en la tabla. Los conjuntos de atributos de columnas de traducción TL se incluyen con la tabla base, al igual que los valores visualizables para códigos de búsqueda (por ejemplo, si una tabla incluye un código de búsqueda para términos de flete, el paquete incluiría un conjunto de atributos para el valor mostrado de FreightTerms).
* Los conjuntos de atributos basados ​​en columnas se nombran para la columna correspondiente. Por ejemplo, en la tabla FWK\_TBX\_PO\_HEADERS tenemos una columna HEADER\_ID. El conjunto de atributos correspondiente es llamado HeaderId(esto siempre se utiliza en el contexto del nombre de paquete completo como se muestra a continuación, por lo que no tiene que preocuparse de varias tablas que tienen HeaderId conjuntos de atributos). Si hay múltiples atributo establece para la misma columna (para que el valor se utiliza comúnmente en varios contextos diferentes con diferentes indicaciones, por ejemplo) que se diferencian con sufijos rápidas como se ilustra para el caso HeaderId en el ejemplo FwkTbxPoHeaderscontinuación. El caso de uso más común para el id del encabezado utiliza el mensaje "Pedido de compra". El atributo HeaderId\_Order set de petición de orden es "orden" y el símbolo HeaderId\_Num es "Número".
* El paquete de conjunto de atributos de tabla también puede incluir conjuntos de atributos para encabezados y botones de región común, que reciben el nombre de sus etiquetas asociadas.

Figura 1: conjuntos de atributos en el paquete FwkTbxPoHeaders.xml.



Ver [Creación de conjuntos de atributos](https://translate.googleusercontent.com/build_attrsets.htm) para obtener instrucciones detalladas si es necesario para crear o mantener sus propios conjuntos de atributos.

**Uso de conjuntos de atributos**

Los desarrolladores de Oracle E-Business Suite deben usar conjuntos de atributos en todos los casos siguientes:

* Todos los artículos asociados con los valores de datos (incluyendo table.column ningún artículo transitorio que se relacionan con valores de la tabla). Por ejemplo, tanto un campo de búsqueda de nombre de proveedor como un campo de entrada de datos de nombre de proveedor deben usar el mismo conjunto de atributos.
* Todos los botones comunes (Ir, Aplicar, Cancelar, etc.). Esto también asegura que se hereda correctamente el criterio de [teclas de aceleración](https://translate.googleusercontent.com/feat/feat_accesskeys.htm) para estos botones.

**Propina:** La OA Framework botones comunes atributo de paquete es / Oracle / aplicaciones / enc / attributesets botones / / <AttributeSetName>.

* Todas las columnas comunes de acción de tabla (como Eliminar, Actualizar, etc.) deben usar el conjunto de atributos del botón OA Framework correspondiente.
* Cualquier botón compartido en su producto para el cual se hayan creado conjuntos de atributos; no debería crear o usar conjuntos de atributos para botones de un solo uso
* Cualquier región de encabezado compartido en su producto para la cual se hayan creado conjuntos de atributos; no debería crear o usar conjuntos de atributos para encabezados de uso único

Para usar un atributo establecido para un artículo:

* En el panel Estructura de JDeveloper, seleccione el elemento para el que desea especificar un conjunto de atributos.
* Coloca el cursor en establecer el atributo del elemento   campo en el inspector de propiedades y seleccione el botón ... para abrir el navegador de componentes. Buscar o navegar como lo prefiere, seleccione el conjunto de atributos que desea extender en la lista de resultados y seleccione Aceptar para hacer su elección.
* JDeveloper entra el nombre completo del conjunto de atributos en el conjunto de atributos   campo (por ejemplo, / Oracle / Aplicaciones / enc / attributesets botones / / Aplicar).

Aunque puedes anular propiedades del conjunto de atributos cuando los asocia con sus elementos, debe evitar hacer esto para los valores traducidos. Si usted se encuentra, por ejemplo, que necesita una variante de un atributo ya existente configurado para mostrar un nuevo sistema, se debe crear un atributo adicional que se indica como se describe en el [conjuntos de atributos Creación](https://translate.googleusercontent.com/build_attrsets.htm)de documentos. Anular algo así como un ancho de pantalla está bien.

Para borrar un conjunto de atributos, coloque el cursor en Atributo campo Set y seleccione Definir la propiedad del inspector botón de la barra de defecto.

**Acceso programático a los conjuntos de atributos**

También puede acceder a conjuntos de atributos en su controlador . Por ejemplo, el siguiente código muestra cómo obtener un mensaje traducido desde el conjunto de atributos del botón Crear común:

importar oracle.apps.fnd.framework.webui.AttributeSet;

...

public void processRequest (OAPageContext pageContext, OAWebBean webBean)   
 {

super.processRequest (pageContext, webBean);

AttributeSet attrSet =

nuevo AttributeSet (pageContext, "/ oracle / apps / fnd / attributesets / Buttons / Create");

String createPrompt = (String) attrSet.getAttributeValue (pageContext, PROMPT\_ATTR);

}

**Parámetros de URL: Tokens, Encriptación, Codificación**

**Tokens**

Cuando especifica parámetros de URL en sus definiciones de página declarativa, puede especificar valores literales y sustituidos por token que obtienen sus datos de atributos de objetos de vista asociados en el tiempo de representación (en este caso, el elemento debe estar vinculado a un objeto de vista). Esto se usa comúnmente, por ejemplo, en una columna de tabla para pasar un valor de clave primaria a una página de detalles para consultar.

Ejemplo de sustitución de tokens (utilizando el nombre de atributo de objeto de vista "OrderNum"):   
OA.jsp?OAFunc=FWK\_TBX\_T\_PO\_PAGE&order={@OrderNum}

Ejemplo literal:? OA.jsp OAFunc = FWK\_TBX\_T\_PO\_PAGE y orden = 123

**Tipos de tokens**

Los tokens usan un prefijo de carácter especial para indicar al OA Framework cómo resolver el valor en tiempo de ejecución (tenga en cuenta que los conceptos de "codificación" "y" encriptación "se describen a continuación):

* **{!** ***Attr}*** - cifra el valor de atributo, dejando el {} en la URL (por ejemplo, OA.jsp ... y SSN = {} SSN & ...?!)!. Usando OAPageContext.getParameter ( "SSN") devolverá el valor descifrado.
* **{} @Attr** - codifica el valor del atributo, dejando la {@} en la URL (por ejemplo, OA.jsp ... y dir = {@} EmpAdd & ...?). Usando OAPageContext.getParameter ( "addr") para obtener el valor del parámetro devolverá el valor decodificado.
* **{} $ Atr** - llano sustitución testigo (sin codificación o encriptación) por lo que es su responsabilidad asegurarse de que un valor sustituido no se rompe la URL.
* **{} @@ RETURN\_TO\_MENU** - Puede ser utilizado exactamente como se muestra para especificar la propiedad URI de destinode un componente de aplicación si quieres que vuelva al usuario de la E Business Suite páginas personales. Si necesita especificar esta cuando se realiza una JSP hacia adelante, la constante correspondiente para esto es OAWebBeanValues.RETURN\_TO\_MENU\_URL.
* **{} @@ RETURN\_TO\_PORTAL** - Puede ser utilizado exactamente como se muestra para especificar la propiedad URI de destinode un componente de aplicación si quieres que vuelva al usuario a una página de lanzamiento de Portal. Si necesita especificar esta cuando se realiza una JSP hacia adelante, la constante correspondiente para esto es OAWebBeanValues.RETURN\_TO\_PORTAL\_URL.

**Codificación**

Cualquier valor que especifique para un parámetro de solicitud debe cumplir con las reglas de sintaxis de HTTP. Por ejemplo, no puede pasar un valor de parámetro de URL con un espacio en blanco ; el siguiente valor de parámetro podría causar un error de ejecución cuando se accede al URL correspondiente: BUYERNAME = John Doe.

Para solucionar esto, *codificamos* estos valores, lo que significa que la rutina de codificación reemplaza los caracteres problemáticos con sustituciones estándar como se muestra en este ejemplo: BUYERNAME = John% 20Doe.

* Cuando el Marco de OA agrega parámetros a la solicitud (valores de campo de formulario, por ejemplo), los codifica automáticamente.
* Cuando se pone parámetros en la petición durante una llamada a un método setForward \*, el Marco de la OA codifica automáticamente estos valores.
* Cuando se pone parámetros en una URL que se monten a sí mismo (si, por ejemplo, se establece la dirección URL de un grano llamando a su método establecer destino), debe codificar cualquier parte de la cadena que podría incluir caracteres no válidos. Para ello, se pasa la cadena a un método de codificación de la clase de utilidad oracle.apps.fnd.framework.webui.OAUrl.

**Consejo:** Si establece manualmente un valor de parámetro de URL que no puede incluir caracteres no válidos (por ejemplo, "value = Y"), entonces usted no tiene que molestarse con la etapa de codificación.

* Cuando se pone en los valores de la solicitud utilizando OAPageContext.putParameter, debe codificar la cadena si es necesario.

El OA Framework decodifica automáticamente valores de los parámetros cuando se llama a los métodos OAPageContext.getParameter \*, a excepción de los siguientes casos:

* Cuando se utiliza el carácter "#" para los tokens función de JavaScript, el Marco de la OA codifica los valores de los símbolos, pero no decodifica automáticamente cuando se llama pageContext.getParameter ( "<tokenName>"). Para hacerlo usted mismo, tendrá que utilizar el método de decodificación OAUrl en el valor que GetParameter devoluciones.
* Cuando llama a putParameter con un valor codificado, OA Framework no lo decodifica. También debe utilizar el método de decodificación OAUrl en este caso sobre el valor del getParameter regresa.

**Cifrado**

El cifrado es el proceso de ofuscación de datos para hacerlo ilegible. Dado que los valores de los parámetros de solicitud de URL pueden ser visibles para el usuario (y los valores de campo de formulario ocultos si el usuario opta por ver el origen de la página HTML), siempre debe cifrar datos confidenciales si se almacenan en un parámetro de URL o en un campo oculto.

Además de la encriptación declarativa, basada en token descrito anteriormente, el marco OA también proporciona métodos en oracle.apps.fnd.framework.webui.OAPageContext para cifrar y descifrar cualquier valores de los parámetros que se pone en la solicitud de programación manualmente.

**Hojas de estilo**

Una de las razones por las que las aplicaciones de OA Framework tienen una interfaz de usuario agradable y uniforme es la apariencia y el sentido de cada página definida por Oracle Corporate. Look navegador y hoja de estilos Feel (BLAF) (blaf.xss). Ver el [BLAF interfaz de usuario Guía: Texto y estándares CSS](https://translate.google.com/translate?hl=es&prev=_t&sl=en&tl=es&u=http://uistandards.us.oracle.com/blaf/history/3.1/specs/textStandards.html) [ [Versión OTN](https://translate.google.com/translate?hl=es&prev=_t&sl=en&tl=es&u=http://otn.oracle.com/tech/blaf/specs/textStandards.html) ] para una consulta rápida de los estilos.

**Usando estilos**

Todas las regiones (y la mayoría de los elementos) que agrega a la página tienen sus estilos establecidos automáticamente; no necesita hacer nada extra (ni debería hacerlo). Como se describió anteriormente, debe establecer las propiedades de región y elemento SÓLO si debe anular el comportamiento predeterminado.

Dicho esto, hay varios casos en los que debe establecer la clase CSS   propiedad para sus artículos:

* Si se crea un elemento **staticStyledText** a utilizar para el texto de instrucciones, debe configurar su clase CSS a **OraInstructionText.**
* Para cualquier campo de entrada de texto, casillas de verificación y botones de radio poplists debe establecer la clase CSS   a **OraFieldText.** *No* utilice **OraPromptText** para los botones de radio y casillas de verificación.
* Si se crea un elemento **messageStyledText** para la visualización de datos de sólo lectura, debe establecer la clase CSS   a **OraDataText** para los datos que se procesan en negrita (tenga en cuenta que no es necesario establecer este valor para las columnas de la tabla)

**Consejo:** los desarrolladores nuevo marco de OA a menudo cometen el error de tratar de cambiar significativamente el componente de representación "nativo" cambiando el estilo CSS. Si te encuentras cayendo en esta trampa (y te sientes frustrado porque tu configuración de estilo no parece tener ningún impacto en el aspecto del tiempo de ejecución del bean):

* Asegúrese de estar utilizando el bean correcto (región o estilo de elemento) para el trabajo.
* Si está seguro de que está usando el frijol derecho, verifique si publica un método que le permita lograr el resultado deseado.Por ejemplo, un oracle.apps.fnd.framework.webui.beans.layout.OAHeaderBean hereda un método setSize (int size) que le permite controlar el tamaño del texto de cabecera (lo cual es útil cuando se representa en las cabeceras de página de inicio "En una Vistazo "regiones o en navegación lateral" Buscar "regiones, por ejemplo). No se puede lograr este efecto al tratar de establecer la clase CSS de la cabecera para **OraHeaderSubSub** ya que algunos están tentados a probar después de leer la especificación BLAF que implementan los granos.

**Creando Estilos**

**Clientes**

El OA Framework establece automáticamente custom.xss sea su hoja de estilo principal. Cualquier personalización que tenga se debe agregar a esta hoja de estilos.

Para obtener información detallada acerca de las hojas de estilo (incluidas instrucciones de personalización), ver [hojas de estilo](https://translate.googleusercontent.com/persguide/T401443T401447.htm#cust_perssys_StyleSheets) .

**Desarrolladores de aplicaciones de E-Business Suite**

La hoja de estilo custom.xss mencionado anteriormente incluye oa.xss, que a su vez incluye blaf.xss.

La hoja de estilo oa.xss pretende incluir cualquier extensión que usted tiene que la hoja de estilo BLAF (en contacto con el equipo de OA Framework si tiene adiciones que han sido aprobados por el equipo de diseño de interfaz de usuario y la usabilidad). NO deberías tratar de crear tu propia hoja de estilo.

**Accesibilidad**

Aplicaciones OA Framework son *accesibles,* lo que significa que pueden ser utilizados por personas con discapacidad como la ceguera, baja visión, ceguera al color y la sordera. En términos simples, los productos accesibles cumplen con las siguientes pautas:

* el producto debe ser utilizable sin un mouse (solo teclado)
* el producto debe ser utilizado por un usuario ciego (con un lector de pantalla o un lector de Braille)
* no debe haber confianza en el sonido
* no debe haber confianza en el color
* no debe haber confianza en la animación o el tiempo

**Nota:** Teclado-sólo los usuarios deben configurar su modo de accesibilidad a "Estándar" para asegurarse de que el teclado sólo la versión accesible de mejoras en la interfaz de usuario introducidas en Oracle E-Business Suite Release 12.1.3 están habilitadas. Esto se puede hacer mediante el establecimiento de la opción de perfil [Características de autoservicio Accesibilidad / ICX\_ACCESSIBILITY\_FEATURES](https://translate.googleusercontent.com/options.htm#accessibility) al **estándar de accesibilidad.**

Para crear páginas accesibles:

* Los Desarrolladores de Oracle deben seguir las Pautas de Accesibilidad HTML Global de Oracle (OGHAG). La lista de verificación de las Pautas de Accesibilidad de Oracle Global HTML es una combinación de los Estándares de la Sección 508 de la Junta de Acceso de los Estados Unidos y Pautas de Accesibilidad al Contenido Web (WCAG) que Oracle ha adoptado a seguir. Para cumplir con las normas específicas en esas directrices, debe cumplir con las directrices de accesibilidad descritas en la [OA marco Ver normas de codificación](https://translate.googleusercontent.com/cs/cs_viewgeneral.htm) . También debe probar su producto para verificar el cumplimiento de las normas de accesibilidad antes de enviarlo. Ver [Prueba de aplicaciones OA Framework](https://translate.googleusercontent.com/test/test_test.htm) para obtener información adicional.
* Los clientes pueden seguir las [Sección 508 Normas](https://translate.google.com/translate?hl=es&prev=_t&sl=en&tl=es&u=http://www.access-board.gov/sec508/guide/) y las [Directrices de Accesibilidad para el Contenido Web (WCAG)](https://translate.google.com/translate?hl=es&prev=_t&sl=en&tl=es&u=http://www.w3.org/TR/WAI-WEBCONTENT/) .

**Internacionalización**

Aplicaciones OA Framework están diseñados ser localizado por completo. En su mayor parte, esto es transparente para usted, siempre y cuando cumpla con todos los estándares de internacionalización descritos en el Capítulo 8. Véase también la [internacionalización](https://translate.googleusercontent.com/build_i18n.htm) de documentos en este capítulo para obtener información detallada sobre el idioma, la zona horaria, la fecha y el apoyo número en el marco de la OA .

**Interacción del modelo**

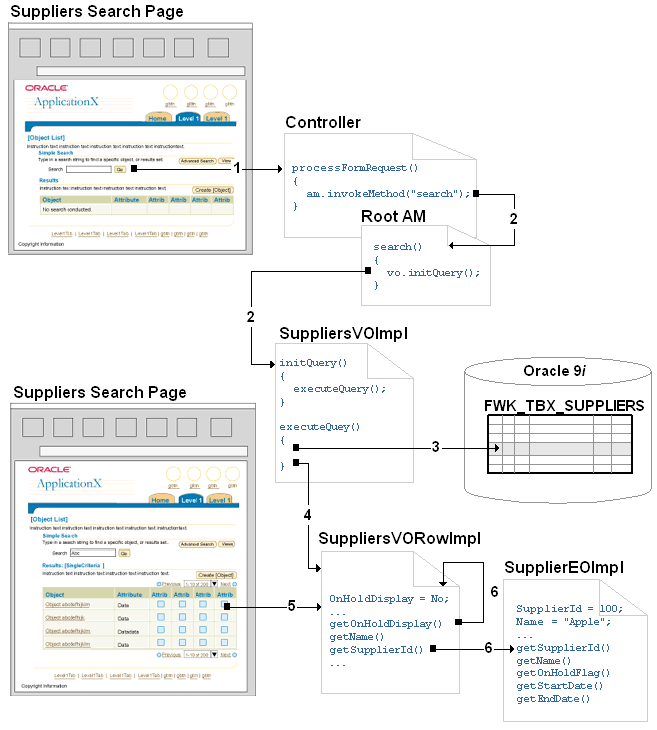
Suponiendo que haya especificado los enlaces de origen de datos necesarios en el marco OA lee automáticamente los datos del modelo para su visualización en la vista, y escribe datos introducidos por el usuario en la vista de nuevo al modelo. No es necesario escribir una sola línea de código (a excepción de ningún tipo de validación que desea realizar en los objetos de entidad subyacentes, por supuesto).

**Lectura de datos Modelo**

En términos simples, cada vez que el OA Framework hace que una página, se llama la corriente objeto de vista de la fila get <*AttributeName* > método para el atributo objeto vista asociada de cada grano web.

Considere una página de ejemplo, con una tabla "Proveedores", que se une a un SuppliersVO objeto de vista. El SuppliersVO se asigna a un subyacente SupplierEOImpl , aunque también incluye un "calculado" atributo transitorio ( "OnHoldDisplay") que no tiene un atributo de objeto correspondiente entidad .

Figura 2: Ilustración de cómo el marco OA lee datos de modelo después de que se ejecuta una consulta



1. El usuario selecciona el botón de la región "Buscar" "Go" para poblar resultados en la tabla "Proveedores".
2. El controlador de "Buscar" de la región maneja el botón de prensa invocando una búsqueda método en el módulo de aplicación de la raíz, que a su vez los delegados a la SuppliersVOImpl clase por lo que puede consultar en sí.
3. Dentro del executeQuery método, el SuppliersVOImpl objeto de vista lleva a cabo su SELECT SQL en la base de datos.
4. Para cada fila devuelta en nuestro conjunto de ejemplo En consecuencia, el objeto de vista instancia un SupplierEOImpl objeto de entidad y establece sus valores de atributos con los resultados de la consulta.

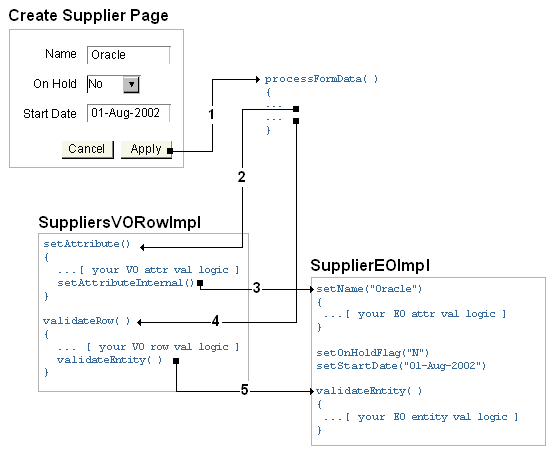
**Nota: Los** valores de los atributos basados en objetos Entidad no se almacenan realmente en cualquier parte del objeto de vista. Que "viven" en el objeto de entidad, y se recuperan cuando sea necesario por el objeto de vista. "Calculado" (es decir, los valores son simplemente seleccionado de una instrucción SQL y no tienen relación a un objeto entidad) o "Transient" vista de objetos de valores de atributos se almacenan en el SuppliersVORowImpl objeto. Ver [Capítulo 5: Implementación de Java objetos de entidad](https://translate.googleusercontent.com/bus/bus_java.htm) para obtener información adicional acerca de la memoria caché de objetos entidad.

1. Durante la renderización de páginas (después de todo el procesamiento de consultas), el Marco de la OA utiliza los enlaces de datos de visualización de objetos definidos para cada bean web para llamar la correspondiente SuppliersVORowImpl del objeto getAttribute ( "<attributeName">) que a su vez llama a su get < *AttributeName* > método.
2. El SuppliersVORowImpl llegar <A ttributeName> método a su vez llama a la correspondiente SupplierEOImpl obtener < *AttributeName* > método para recuperar el valor. Para el atributo OnHoldDisplay "calculada", la fila Vista del objeto recupera el valor de su propia caché.

**Modelo de escritura de datos**

Cada vez que el explorador emite una POSTAL solicitud , el Marco de la OA escribe automáticamente los valores del formulario introducidos por el usuario de nuevo a la vista Objetos subyacentes, que a su vez actualizar los objetos de cualquier entidad correspondiente, como se muestra a continuación.

Figura 3: Ciclo HTTP datos POST



**Nota:** Los siguientes pasos se supone que el objeto de entidad para la fila ya se ha creado una instancia e inicializado (como en el método de crear en el objeto de vista subyacente para la página de éxito cuando el usuario entra inicialmente en una página de crear la vista objeto crear método. llama al método correspondiente crear en el objeto de entidad detrás de las escenas).

1. UIX realiza onSubmit validación de Javascript (campos obligatorios, tipos de datos, formatos) y emite el POSTE solicitud solamente si esta validación se realiza correctamente.
2. El navegador envía una POSTAL solicitud y la OA Framework llama a los processFormData métodos en todos los granos de la tela en la jerarquía como se describe en [Anatomía de un marco de páginas OA](https://translate.googleusercontent.com/ess/ess_anatomy.htm) .
3. Dentro processFormData , la OA Framework llama automáticamente setAttribute (String nombre, valor del objeto) en la fila actual del objeto de vista subyacente para cada grano. Esto ejecuta cualquier *nivel de atributo* de validación que usted ha escrito en la fila objeto de vista.
4. Dentro de este setAttribute método, la vista de primera fila objeto llama automáticamente al correspondiente conjunto < *AttributeName* > método en el objeto entidad subyacente. Esto ejecuta cualquier asociado *de nivel de atributo* de validación en el objeto de entidad.
5. Una vez que se han establecido todos los valores de atributo, el Marco de la OA llama al objeto vista validar para cada fila se modificó para ejecutar cualquier asociada *a nivel de registro* de validación.
6. Finalmente, dentro de la Validar método, la fila de objeto de vista llama validateEntity para el objeto entidad subyacente que ejecuta cualquier asociado *de nivel de entidad* de validación.

**Nota:** El Marco de OA muestra automáticamente mensajes de error para cualquier excepciones producidas por la capa de modelo durante processFormData y no pasar a la siguiente fase de llamar processFormRequest . Ver [Control de errores](https://translate.googleusercontent.com/build_error.htm) para obtener información adicional acerca de cómo el Marco OA muestra mensajes de error.

**Validación pasando por alto**

Como se describió anteriormente, el Marco de la OA escribe los datos del modelo para cada formulario de envío - lo que significa que se ejecuta toda su validación de atributos y de las entidades. Hay momentos en los que hay que "cortocircuito" este proceso lo que los errores no se notifican al usuario en un momento inapropiado.

Ver [Implementación del controlador: Modelo de Interacción](https://translate.googleusercontent.com/build_control.htm#model) para obtener instrucciones específicas sobre la prevención de esta.

**Defaulting**

Cuando se crea una nueva fila para ser utilizado en una página "Crear", como se muestra en la Figura 3 anterior, puede especificar los valores por defecto en tres lugares:

* [Modelo] Si su objeto vista se basa en uno o más objetos de entidad, puede anular su create () método para establecer mediante programación por defecto a nivel de atributo. Ver los [objetos de Java Entidad Crear](https://translate.googleusercontent.com/bus/bus_java.htm#create) tema para obtener información adicional.
* [Modelo] También puede asociar de forma declarativa por defecto con objeto de entidad atributos utilizando el asistente BC4J entidad objeto. Tenga en cuenta que los desarrolladores de E-Business Suite de Oracle internos no deben usar esta opción.
* [Ver] Como alternativa, se puede establecer valores por defecto para los elementos individuales configurando sus valores iniciales propiedad en el Oracle JDeveloper 10 *g* de Extensión OA. La ventaja de este enfoque - para los valores estáticos que puede ser determinado en tiempo de diseño - es los clientes pueden personalizar fácilmente estos valores predeterminados. Esta característica se puede utilizar con los objetos que lo hacen, y no lo hacen, mapa para una vista de objeto subyacente. Por ejemplo, podría utilizar esto para un campo de criterios de búsqueda a pesar de que no tiene una instancia de vista de objeto asociado.

Incumplidora está apagado. Debe establecer el valor de la opción de perfil [FND: OA: Habilitar predeterminados / FND\_OA\_ENABLE\_DEFAULTS](https://translate.googleusercontent.com/options.htm#default) a **Y** para encender el impago.

Si el perfil está habilitado y un valor predeterminado se especifica en la vista (ya sea en la extensión de OA o como personalización) en un elemento de forma que **no es un messageChoice o messageRadioGroup** , a continuación, OA marco establece el valor del elemento de acuerdo con las siguientes reglas :

* Si el artículo no tiene ninguna fuente de datos asociada objeto de vista, la opción de perfil no tendrá ningún efecto y OA marco establece automáticamente el valor por defecto directamente sobre el elemento cuando se representa la página. **Nota:**Si usted está tratando de establecer el "valor inicial" de una forma de grano con el tipo de datos como "fecha", a continuación, utilizar el formato canónico aaaa / MM / dd. De lo contrario, el valor por defecto no se aplicará.
* Si el artículo tiene una vista de objeto asociado, OA Marco establece el valor por defecto cuando se llama [createRow ()](https://translate.googleusercontent.com/translate_f" \l "createrow) en el objeto de vista.

Si el perfil está habilitado y un valor predeterminado se especifica en la vista (ya sea en la extensión de OA o como personalización) en un elemento de formulario que **es un messageChoice o messageRadioGroup** , a continuación, OA marco establece el valor de acuerdo a las siguientes reglas:

* Si el valor de la fila actual para el atributo de vista **no es nulo** , el valor predeterminado especificado se establece como el valor por defecto de la selección.
* Si el valor de la fila actual para el atributo de vista **es nulo** , el valor predeterminado se muestra en la selección se determina por el valor predeterminado especificado y **se** aplica al objeto vista.
* **Nota:** Si la opción de perfil no está activado, y el valor de la fila actual para el atributo de vista **es nulo** , el valor por defecto se muestra en la selección también será determinado por el valor predeterminado especificado y **no se** aplica al objeto vista.

El siguiente ejemplo muestra el código típico para la creación de una nueva fila:

createSupplier public void ()

{

OAViewObject vo = getSuppliersVO ();

// El Marco de OA se aplica por defecto la interfaz de usuario durante el alcance de esta

// llamada al método.

**Fila fila = vo.createRow ();**

vo.insertRow ();

// Tal como se especifica en la OA Modelo Marco de Estándares de Codificación,   
// siempre establecer el nuevo estado fila para STATUS\_INITIALIZED

// después de crearlo.   
row.setNewRowState (Row.STATUS\_INITIALIZED);

}

Específicamente, el createRow () llamadas de método Create () en cada ViewRowImpl . En el ámbito de la creación ()llamada, la OA Framework llama al colocador atributopor defecto para cada interfaz de usuario especificado tiene que aplicar. Esto asegura que cualquier fila vista de atributo de validación - y validación atributo de objeto entidad asociada - se ejecuta como se describe en la Figura 3 anterior. La OA Framework entonces se restablece el estado de su vista de primera fila a STATUS\_INITIALIZED por lo que parece sin tocar a BC4J. Esto asegura que los usuarios pueden navegar a páginas y dejar que sólo tienen valores por defecto sin ser advertido sobre la posible pérdida de trabajo. Cualquier excepción de validación detectados durante el proceso de morosos se muestran como normal.

**Consejo:** Los valores predeterminados se aplican *una vez y sólo una vez* cuando se crea la fila. Si usted tiene un flujo de varias páginas con las regiones que se unen a la misma vista objeto subyacente - y cada región especifica un diferente valor inicial para uno de los atributos de vista - sólo el defecto asociado con la primera región que hace que cuando se crea la fila es aplicado. Los otros son ignorados. Del mismo modo, si crea una nueva fila en la página A, y luego navegar a la página B, donde tiene un valor inicial establecido para uno de sus atributos, el valor por defecto no se aplica debido a que la fila se ha creado antes Página B hace (tenga en cuenta que la creación de una fila antes de navegar a la página donde el usuario entra realmente en valores para la nueva fila no es consistente con[la aplicación recomendada Volver botón para un flujo Crear](https://translate.googleusercontent.com/adv/adv_backbutton.htm) ).

Suponiendo que los valores predeterminados están usando cada uno de estos tres enfoques, el Marco de la OA observa la siguiente prioridad de precedencia:

1. (Más alta) por defecto propiedad del elemento declarativos como se especifica en la extensión de la OA o en el módulo de personalizaciones
2. Objeto de entidad programática de atributos por defecto (que se aplican durante el alcance de la vo.createRow () llamada al método que a su vez delega en el objeto de entidad de crear () método)
3. (Más bajo) Declarative objeto de entidad de atributos por defecto

Si desea asegurarse de que un valor predeterminado se establece siempre, independientemente de qué valores pueden ser especificados de forma declarativa, puede anular la insertRow () método en su fila objeto de vista como se muestra a continuación:

pública EmployeesVOImpl vacío se extiende OAViewObjectImpl

{

...

insertRow pública vacío (fila de la fila)

{

// Siempre llame super.insertRow () en primer lugar.

super.insertRow ();

// Ahora llama a cualquier atributo de los emisores es necesario llamar para asegurar

// que sus valores por defecto siempre tienen prioridad.

row.setAttribute ( "< *attributeName* >", < *attributeValue* >);

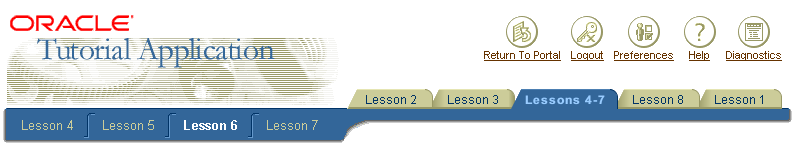
...

}   
 }

**Menús y Seguridad Página**

En una aplicación Marco de la OA, el menú de páginas disponibles se presentan al usuario en un modelo basado en pestañas, como se ilustra en el siguiente ejemplo de la aplicación OA Framework Caja de Herramientas Tutorial:

Figura 4: menús OA marco de cuadro de herramientas de aplicación Tutorial.



Dentro de este modelo básico, usted es libre de elegir entre una gama de opciones de diseño válidas basadas en la complejidad de su aplicación y espera que los patrones de uso (cuando esté listo para comenzar a diseñar menús para su aplicación, consulte la [Oracle Browser Look-and-feel ( BLAF) Pauta de interfaz de usuario: aquí / Navegación](https://translate.google.com/translate?hl=es&prev=_t&sl=en&tl=es&u=http://uistandards/blaf/guidelines/tabs.html) [ [OTN Versión](https://translate.google.com/translate?hl=es&prev=_t&sl=en&tl=es&u=http://otn.oracle.com/tech/blaf/specs/tabs.html) ] en primer lugar para una introducción detallada a las distintas configuraciones de menú).

Esta estructura de menús tiene dos finalidades distintas en una aplicación marco OA:

* Se organiza el contenido en unidades significativas.
* Permite a los usuarios navegar fácilmente entre estas unidades significativas.

**Implementación de menú**

Detrás de las escenas, un menú Marco de la OA es en realidad compuesto de la Suite Oracle E-Business *funciones* y *menús* .

**Funciones de navegación**

Las funciones de navegación representan las páginas individuales dentro de su aplicación; cada página que un usuario tiene acceso en tiempo de ejecución está asociada a una función predefinida , que se utiliza únicamente con el fin de navegar a través de su aplicación. Tal vez lo más importante, esta función incluye la Web HTML de llamadas para la página. Por ejemplo, en el Tutorial Application Toolbox, cuando el usuario selecciona la lección 3 entrada de menú, se visualiza la página de la orden de compra de búsqueda. Hemos creado una función para esta página y establezca su Web HTML llamada para que apunte a la página XML queremos mostrar :

OA.jsp? Page = / oracle / aplicaciones / enc / marco / caja de herramientas / webui / PoSearchPG

Cuando el usuario selecciona la lección 3 entrada de menú, la petición se encamina al OA.jsp que inicializa un oracle.apps.fnd.framework.webui.OAPageBean objeto de procesar el archivo de página XML correspondiente como se describe en [Anatomía de una página de marco OA](https://translate.googleusercontent.com/ess/ess_anatomy.htm) (en caso de que se esté preguntando, OA.jsp es la única JSP que se invoca cuando se accede a cualquier página de la aplicación marco OA).

**Nota:** Una sola página puede ser llamado por muchas funciones (cada uno potencialmente pasan diferentes parámetros a través de la URL), lo que significa que se puede utilizar en muchos menús diferentes.

**Menús de navegación**

Navegación menús son agrupaciones reutilizables de funciones y submenús que en última instancia, crean la estructura pestaña que hemos descrito anteriormente. Cada menú Marco de OA que se crea tiene asociado un tipo que determina cómo debe ser prestados. Por ejemplo, la pestaña de la lección 2 en la Figura 1 anterior es de tipo "HTML Tab."

Menús de navegación incluyen ***todas las*** funciones que se pueden mostrar en su aplicación. También puede conceder selectivamente el acceso a funciones individuales dentro de este menú de navegación. Esto se describe con más detalle a continuación en la sección de seguridad de aplicaciones.

Para obtener instrucciones detalladas sobre cómo diseñar y crear menús de navegación OA Framework, consulte [el Capítulo 4: aquí / navegación](https://translate.googleusercontent.com/feat/feat_tabs.htm) .

**Seguridad de aplicaciones**

Las características de seguridad de la aplicación son amplios; en este capítulo introductorio vamos a tocar en algunos conceptos clave para que tenga una idea general de lo que está apoyado en lo que respecta a las definiciones de menú. Cuando esté listo para empezar a diseñar su aplicación, se recomienda familiarizarse con estas características mediante la revisión de [seguridad de página](https://translate.googleusercontent.com/feat/feat_pagesecurity.htm) en el capítulo 4.

**Usuarios y Responsabilidades**

Una suite el reponsibility Oracle E-Business es un conjunto de tareas que se concede a uno o más usuarios como un conjunto. Por ejemplo, puede crear un **administrador de beneficios** y una genérica **Empleado** responsabilidad, cada uno con las aplicaciones adecuadas relacionados con recursos humanos. A continuación, asignar estas responsabilidades a los usuarios individuales para conceder rápidamente el acceso a estos módulos.

Todas las responsabilidades se asocian con el menú de navegación de nivel superior solo para su aplicación. Como se describió anteriormente, el menú de navegación en última instancia, incluye todas las pestañas con el apoyo de su aplicación.

Anteriormente, la responsabilidad era el principal mecanismo para agrupar a los usuarios en grupos basados ​​en roles. A continuación, asignar responsabilidades a los menús, y crear reglas de seguridad mediante la exclusión de las funciones del menú individuales de su responsabilidad. En tiempo de ejecución, el grupo de la responsabilidad, organización y seguridad actual conjunto compuesto por el contexto de seguridad.

Con la versión 12, el concepto de responsabilidad se ha ampliado a una más genérica **papel** . Los usuarios pueden pertenecer a una o varias funciones. Todos los usuarios asignados a una responsabilidad particular también se asignan a un correponding **papel** . Las reglas de seguridad se basan en **subvenciones de permiso** en lugar de las reglas de exclusión función. En tiempo de ejecución, estas ayudas se evalúan para el contexto actual de la seguridad, que ahora incluye funciones (también conocido como un "concesionario"), además de la responsabilidad, organización y grupo de seguridad.

La OA Framework recomienda el uso de papeles de permisos y concesiones para la configuración de seguridad en lugar de las responsabilidades y las reglas de exclusión.

**Grants y permisos**

Además de la creación de funciones de navegación, también debe crear funciones de autorización (conocidos como "permisos") para cada una de sus páginas. A continuación, estos permisos de grupo en una estructura "plana" del menú (conocido como un "conjunto de permisos") con el propósito de facilitar el acceso del usuario a las páginas asociadas.

La forma más sencilla de introducir el uso de conjuntos de permisos está caminando a través de un pequeño caso de uso. Por ejemplo, suponga que tiene una aplicación muy simple beneficios incluyendo las cuatro páginas siguientes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Página** | **Descripción** | **Beneficios de Access Manager?** | **El acceso de los empleados?** |
| administrar los beneficios | Ver, actualizar, aprobar y suspender beneficios. | **Sí** | No |
| crear Beneficio | Crear un nuevo beneficio. | **Sí** | No |
| Mis beneficios | Ver selecciones de beneficios actuales y hacer nuevas selecciones según el caso. | **Sí** | **Sí** |
| Actualizar beneficiarios | Actualizar los beneficiarios designados. | **Sí** | **Sí** |

Como se describió anteriormente, se crearía Funciones de Navegación en cada una de estas páginas y organizarlas en un menú de navegación completa. Para garantizar que los usuarios tengan acceso a las páginas de la derecha, a continuación, procederá del siguiente modo:

Paso 1: Crear permisos.

Al igual que las funciones de navegación, **los permisos** son funciones de forma FND, pero en este contexto, se utilizan exclusivamente para la seguridad de las aplicaciones.

En nuestro ejemplo, podemos utilizar las funciones de navegación que hemos creado para cada página como permisos. No hay necesidad de crear funciones de permisos adicionales.

Paso 2: Crear roles o concesionarios.

Un concesionario puede ser o bien un **usuario** (FND\_USER) ***,*** o un **grupo de usuarios**(también conocido como **papel** ), o **"global** ". **Usuarios** identidades se crean en FND\_USERS, y deben asignar uno-a-uno con los seres humanos o los sistemas individuales. Los usuarios pueden pertenecer a **grupos** o **funciones** que se forman mediante la agrupación de las relaciones organizativas o de posición modelados en productos tales como Recursos Humanos. Los roles se definen en WF\_ROLES, y en el futuro pueden asignar a grupos de usuarios en el sistema LDAP de un cliente. A pesar de que sus miembros no se rellena de forma explícita, hay un **Global**grupo que incluye "todo el mundo".

Se necesitan dos roles de usuario para el ejemplo anterior: uno que agrupa a todos los gerentes en función de un gerente, y otra que agrupa a todos los empleados. Dado que todos los empleados incluye a todos, se puede utilizar una función global para este fin.

Como alternativa, puede crear una responsabilidad que se asigna a todos los gerentes, y el uso que para la configuración de las subvenciones.

Vamos a discutir tanto las alternativas anteriores, cuando se procede al Paso 4 para crear las subvenciones.

Paso 3: Crear conjuntos de permisos.

**Los conjuntos de permisos** se implementan como menús, pero son sólo existen para agrupar una lista plana de permisos en conjuntos de conceder el acceso fácil. Lo ideal es que los permisos de grupo que son requeridos por un papel en uno o más conjuntos de permisos.

Se necesitan dos conjuntos de permisos para el ejemplo anterior:

* A Permiso Administrador de conjuntos de todas las tareas a las que sólo los gerentes deben tener acceso. Esto incluye las funciones de navegación "administrar los beneficios", y "crear un beneficio".
* Un permiso global de conjunto con los permisos que sean accesibles por todos. Esto incluye las funciones de navegación "Mis Beneficios" y "beneficiarios de actualización".

Paso 4: Crear Grants

Una **subvención** define las reglas de seguridad que permite que sólo determinados usuarios de su sistema de acceso a las funciones o las páginas de su aplicación específica. Una **subvención** da un **concesionario** acceso a los conjuntos de permisos descritos anteriormente. En términos simples, las subvenciones enlace para que los beneficiarios de sus conjuntos de permisos.

Se necesitan dos subvenciones para el ejemplo anterior:

* Una subvención Gestor de asociar el permiso Administrador de conjunto con la función de administrador.
* Una subvención del empleado que se asocia con su permiso global establecido con un concesionario global.

Dado que esta subvención está asociado a un concesionario mundial (en otras palabras, todo el mundo) y no tiene restricciones de seguridad adicionales (en otras palabras, no se limita a ninguna responsabilidad, organización o grupo de seguridad), sino que también se puede llamar una **subvención global** .

Además de especificar un concesionario, también se puede restict su nueva subvención con el contexto de seguridad adicional. Esto incluye la responsabilidad del usuario actual, la organización y grupo de seguridad. Así, por ejemplo, para restringir la concesión encargado a una organización específica, se puede asociar un contexto de la organización con la subvención.

En lugar de conceder el permiso Administrador de conjunto a la función de administrador, puede conceder a un concesionario global. A continuación, puede restringir a los administradores solo asociando un contexto de seguridad de la responsabilidad a la que sólo tienen acceso los administradores. Sin embargo, tenga en cuenta que el Marco OA recomienda el uso de subvenciones basado en roles en lugar de las responsabilidades.

En tiempo de ejecución, un usuario tiene acceso a una página si el permiso asociado a la página se concede acceso al contexto de seguridad del usuario actual. contexto de seguridad del usuario como se ha descrito anteriormente incluye la función de, responsabilidad, organización y grupo de seguridad del usuario.

**Seguridad página**

Si nos fijamos en el ejemplo anterior, cabe mencionar que se puede vincular los permisos con sus páginas para restringir el acceso. Este es uno de los casos en que es necesario para garantizar la prestación de su página con un permiso.

Otros casos en los que es posible que desee para asegurar la prestación de su página con un permiso incluyen las páginas de inicio de sesión anónimos, las páginas que requieren el establecimiento de la responsabilidad o el cambio automático y páginas compartidas / reutilizables.

Para obtener instrucciones detalladas, ver [el Capítulo 4: Página de Seguridad](https://translate.googleusercontent.com/feat/feat_pagesecurity.htm) .

[Copyright © 2000,2010, Oracle y / o sus afiliados.](https://translate.googleusercontent.com/copyright.htm) [Todos los derechos reservados.](https://translate.googleusercontent.com/copyright.htm)