

Unidad Didáctica IX.- Herramientas Case

*José se reúne con **María** y **Víctor** para ver los avances que ha experimentado su aprendizaje con Oracle Express, como herramienta CASE concreta que está aprendiendo a manejar, y con la programación de aplicaciones de arquitectura cliente-servidor en general. Ya controla bastantes aspectos de cómo se puede construir una aplicación con **arquitectura cliente-servidor** n-capas, y ha visto bastantes ejemplos de cómo desarrollaba **Carmen** la capa de negocio que trabajaba sobre la capa de datos. Pero sabe que falta una de las "patas del banco": Esas aplicaciones también requieren programar o desarrollar la **capa de cliente o de presentación**, y quiere completar su aprendizaje viendo cómo se desarrolla esta capa haciendo uso además de las herramientas CASE que nos proporciona Oracle Application Express.*



***Carmen** se incorpora a la reunión, y está de acuerdo en que ha llegado el momento, que tienen que pasar a la acción. Por eso empieza a mostrarle a **Víctor** la herramienta que han elegido para trabajar en el desarrollo de la capa del cliente. Esta herramienta integrada en Oracle Application*

*Express se llama **Creador de Aplicaciones**, y dedican un cierto tiempo a familiarizarse con ella, y con sus posibilidades, los distintos métodos que proporciona para crear una aplicación, los tipos de páginas que pueden crearse, los separadores, los componentes compartidos, el interfaz de usuario, etc. Todos estos conceptos son un poco nuevos para **Víctor**, pero sabe que pronto los controlará, porque ya está seguro de su capacidad de aprender conceptos y herramientas nuevas, a los que sabe que se adapta con rapidez y facilidad.*

*Para conseguir sus objetivos, **María** propone que **Carmen** y **Víctor** sigan desarrollando una de las aplicaciones que han venido usando desde el principio, para clientes, departamentos, pedidos, etc. A esta aplicación le han añadido una tabla de empleados, y van a trabajar sobre ella en el desarrollo de la capa de cliente.*

Herramientas CASE

Como recordarás,

- en la **unidad 6** hemos estudiado las herramientas CASE de modelado de datos y has aprendido a usarlas para que te ayuden en las tareas de **diseño de bases de datos** correctas, es decir, has aprendido a servirte de herramientas CASE para



diseñar e implementar de forma correcta la capa de datos de una aplicación.

- en la **unidad 8**, aprendimos en qué consisten las distintas **arquitecturas** centrándonos en la arquitectura cliente-servidor n-capas, aprendiendo a desarrollar la capa o lógica de negocio.
- En **esta unidad**, por ser la última, además de aplicar todo lo aprendido durante este módulo, vas a aprender a diseñar y desarrollar la capa de cliente de una aplicación que presenta arquitectura cliente- servidor de n capas, dando por supuesto que ya se ha desarrollado la capa de datos y la lógica de negocio necesaria para desarrollar esa aplicación.

Si lo prefieres, podemos decir también que vamos a aprender a **desarrollar el interface** mediante el cual el usuario podrá interactuar con los datos. Para ello, usaremos la herramienta CASE de diseño gráfico de interface que nos suministra el Oracle Application Express y que nos permite el desarrollo y mantenimiento de aplicaciones. Esta herramienta se denomina **Creador de Aplicaciones**.

Esperamos que al final de esta unidad poseas los conocimientos necesarios para desarrollar tus propias aplicaciones.

Herramientas CASE

Ya conoces las particularidades de las herramientas CASE para diseñar e implementar de forma correcta la capa de datos de una aplicación, si bien en esta unidad nos centraremos en la utilidad de **diseño gráfico** de interface que nos suministran algunas de estas herramientas centrándonos en la del Oracle Application Express.



La tecnología CASE supone la **automatización del desarrollo del software**, contribuyendo a mejorar la calidad y la [productividad](#) en el desarrollo de sistemas de información y se plantea los siguientes objetivos:

- Permitir la aplicación práctica de metodologías estructuradas, que al ser realizadas con una herramienta consiguen agilizar el trabajo.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.
- Aumentar la [portabilidad](#) de las aplicaciones.
- Facilitar la reutilización de componentes software.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos.



Estas herramientas son fundamentales a la hora de desarrollar una aplicación, facilitando enormemente los procesos de su diseño, desarrollo, implementación y mantenimiento, como podrás apreciar a lo largo de esta unidad, permitiendo **desarrollar aplicaciones con unos recursos humanos y un tiempo de desarrollo mucho más reducidos** que si tuviéramos que trabajar sin estas

herramientas.

Herramientas CASE

Características deseables de una herramienta CASE

¿Recuerdas cuáles eran las características que más nos interesan de las herramientas CASE?

En este apartado vamos a enumerarlas para que las tengas presentes en todo momento.

Las características más reseñables de una herramienta CASE cliente / servidor son:



- **Proporcionar topologías de aplicación flexibles.** La herramienta debe proporcionar facilidades de construcción que permita separar la aplicación (en muchos puntos diferentes) entre el cliente y el servidor y también, entre servidores.
- **Proporcionar aplicaciones portátiles.** La herramienta debe generar código para Windows, OS/2, Macintosh, Unix y todas las plataformas de servidores conocidas.
- **Control de versión.** La herramienta debe reconocer las versiones de código que se ejecutan en los clientes y servidores, y asegurarse que sean consistentes. También, la herramienta debe ser capaz de controlar un gran número de tipos de objetos incluyendo texto, gráficos, documentos complejos y objetos únicos, tales como definiciones de pantallas, definiciones de informes, archivos, datos de prueba y resultados.
- **Crear código compilado en el servidor.** La herramienta debe ser capaz de compilar automáticamente código 4GL en el servidor.
- **Trabajar con una variedad de software intermedios.** La herramienta debe adaptar sus comunicaciones cliente / servidor al software intermedio existente. Como mínimo la herramienta debería adaptar así el tráfico que se está moviendo en una LAN o WAN.
- **Soporte multiusuarios.** La herramienta debe permitir que varios diseñadores trabajen en una aplicación simultáneamente. Debe gestionar los accesos concurrentes a la base de datos por diferentes usuarios.
- **Seguridad.** La herramienta debe proporcionar mecanismos para controlar el acceso y las modificaciones a los datos que contiene. La herramienta debe, al menos, mantener contraseñas y permisos de acceso en distintos niveles para cada usuario.



Herramientas CASE

Las herramientas CASE evolucionan hacia tres tipos de integración:



1. La **integración de datos** permite disponer de herramientas CASE con diferentes estructuras de diccionarios locales para el intercambio de datos.
2. La **integración de presentación** confiere a todas las herramientas CASE el mismo aspecto.
3. La **integración de herramientas** permite disponer de herramientas CASE capaces de invocar a otras CASE de forma automática.

Para saber más

En el siguiente enlace encontrarás abundante información sobre las herramientas CASE

Herramientas CASE

<http://www.monografias.com/trabajos14/herramicase/herramicase.shtml#act>
[versión en cache]

En los siguientes enlaces encontrarás un muy interesante resumen de las principales herramientas CASE.

Herramientas CASE

<http://www.lsi.us.es/docencia/get.php?id=1130> [versión en cache]

Herramientas CASE más utilizadas

<http://www.monografias.com/trabajos24/herramientas-case/herramientas-case.shtml> [versión en cache]

Autoevaluación

De las siguientes afirmaciones referidas a la integración de las herramientas CASE, señala la que es correcta.

- a) La integración de herramientas permite disponer de herramientas CASE con diferentes estructuras de diccionarios locales para el intercambio de datos.
- b) La integración de datos confiere a todas las herramientas CASE el mismo aspecto.
- c) La integración de presentación permite disponer de herramientas CASE con diferentes estructuras de diccionarios locales para el intercambio de datos.
- d) La integración de herramientas permite disponer de herramientas CASE capaces de invocar a otras CASE de forma automática.

Comprobar

Tras varias explicaciones de **Carmen**, **Víctor** se ha puesto a trabajar y ha creado una aplicación usando el **Creador de Aplicaciones** de Oracle Express, es el comienzo del desarrollo de la capa de cliente, y la verdad, le ha resultado sumamente fácil. Esta herramienta CASE le va pidiendo los datos, y él sólo ha tenido que preocuparse de ir dando los valores adecuados, y seleccionando las opciones oportunas.



Ha elegido la opción de **Crear Aplicación**, le ha dado **NOMBRE** y la ha identificado con un **código numérico** único,

ha indicado que se trata de una aplicación **NUEVA**,

ha indicado que **el tipo de página es PANTALLA**, para que le permita actualizar una única fila de una tabla, haciendo inserciones, actualizaciones y supresiones,

ha indicado que la aplicación va a trabajar sobre la **tabla EMPLEADOS** de la base de datos, y que la aplicación va a tener **un nivel de separadores**.

Ha indicado también algunos atributos adicionales, concretamente como **ESQUEMA DE AUTENTICACIÓN** ha elegido **CUENTA DE BASE DE DATOS**, para indicar que los usuarios **que podrán usar la aplicación son los mismos que tienen acceso a la base de datos** sobre la que corre la aplicación, es decir, la base de datos que contiene la tabla Empleados.

Otro atributo es el **IDIOMA**, que ha elegido **Español**.

Otro elemento importante de la aplicación es el **INTERFAZ DE USUARIO**.

Como el Creador de Aplicaciones nos ofrece una serie de **TEMAS**, que no son más que recopilaciones de plantillas que fijan el aspecto de los distintos elementos de la aplicación para adaptar el interfaz a nuestros gustos y necesidades, **Víctor** no ha tenido más que elegir el que le ha parecido más adecuado. En este caso ha elegido el **Tema1**, que le ha parecido que da un aspecto muy profesional a su aplicación.



Considera que muchas aplicaciones de las que tendrá que desarrollar en el futuro seguirán el mismo modelo que ésta, así que ha decidido que va a **guardar la definición como modelo para reutilizarla**. Y ya lo único que le quedaba por hacer era darle al **botón de CREAR** dentro de la pantalla de Crear Aplicación, para que la aplicación esté dispuesta para ser ejecutada y probada.

Vamos a comenzar a desarrollar la capa cliente para nuestra aplicación usando el Oracle Express.



¿Disponemos de alguna herramienta CASE integrada en Oracle Express que nos permita hacer este trabajo de forma cómoda y sencilla?

La respuesta es que sí.

El Oracle Express nos suministra una herramienta para la creación y mantenimiento de las aplicaciones. Esta herramienta es el **Creador de Aplicaciones**.

En esta unidad nos vamos a centrar en el manejo de dos opciones de esta herramienta, que puedes ver en el menú de la imagen que acompaña a este texto:



- **Crear Aplicación**
- **Ver Aplicaciones**

Con ellas seremos capaces de crear y mantener la capa cliente de cualquier aplicación cliente-servidor.

Herramientas CASE

Crear Aplicación

Este menú nos da acceso a la herramienta con la que conseguiremos crear la aplicación que deseamos desarrollar. No hemos de perder de vista **que estamos realizando la capa cliente de una aplicación cliente-servidor**, es decir, partimos de la premisa de que ya hemos desarrollado la capa de datos y la capa de negocio. Nos queda por tanto dotar a la aplicación de un interface de usuario.

En esta unidad, las explicaciones se apoyan mucho sobre **imágenes** de ejemplo. Para mejorar la visualización del texto, las hemos incorporado en tamaño reducido, como la siguiente, pero podrás ampliarlas a pantalla completa sin más que hacer clic sobre la propia imagen, que actuará de enlace para verla aumentada.



Antes de comenzar a crear nuestra aplicación debemos entender qué es una **aplicación** desde el punto de vista del uso de la herramienta de desarrollo que nos suministra Oracle Express.

Una aplicación es **una recopilación de páginas enlazadas entre sí mediante separadores, botones o enlaces de hipertexto**. Las páginas de una aplicación comparten una definición de estado de sesión y un método común de **autenticación** (también se suele llamar autenticación).

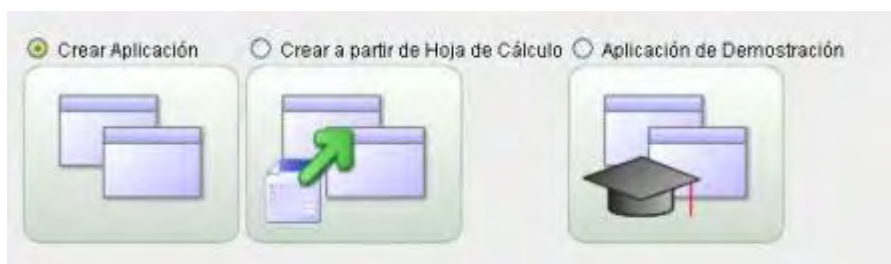


Una vez entendido, vamos a conocer las principales funcionalidades que nos suministra la herramienta de desarrollo de aplicaciones del Oracle Express.

Herramientas CASE

Método

La herramienta del Oracle Express nos facilita tres métodos o formas de crear nuestra aplicación.



- **Crear Aplicación.**
Crea una aplicación definiendo páginas, seleccionando un esquema de autenticación y especificando una interfaz de usuario. Las páginas se pueden basar en tablas, consultas o consultas desplegables.
- **Crear a partir de Hoja de Cálculo.**
Crea una aplicación basada en datos de hoja de cálculo. Permite cargar o pegar datos de una hoja de cálculo para crear una tabla. La aplicación mostrará las capacidades de consulta, inserción, actualización y análisis en la tabla recién creada.
- **Aplicación de Demostración.**

Permite instalar o desinstalar las aplicaciones de demostración. Esta opción es muy recomendable para usar estas aplicaciones con el propósito de obtener información sobre cómo crear nuestras propias aplicaciones.

Autoevaluación

De las siguientes afirmaciones referidas a los métodos para crear aplicaciones, señala la correcta.

- a) Las páginas se pueden basar en tablas pero nunca en consultas.
- b) Crear a partir de hoja de cálculo permite cargar o pegar datos de una tabla para crear una hoja de cálculo.
- c) Crear aplicación crea una aplicación definiendo páginas, seleccionando un esquema de autenticación y especificando una interfaz de usuario.
- d) Aplicación de demostración. Permite instalar o desinstalar las aplicaciones de demostración y la herramienta de desarrollo del Oracle Express.

Comprobar

Herramientas CASE

Páginas

Para crear la capa cliente desarrollaremos una aplicación cuyo interface principal con el usuario será el uso de páginas, es decir, nuestra aplicación estará compuesta de páginas o formularios que englobarán una o varias funcionalidades y que además podrán llamar a otras páginas.



Pero antes de proseguir, hemos de entender bien qué es una página.

Probablemente, al mencionar el término **página** te viene a la mente lo que ves cuando entras a un sitio Web. Esto es así, indudablemente, pero no nos explica realmente qué es una página. Concretando más y aplicando los conocimientos adquiridos, podríamos afirmar que **una página es un documento de hipertexto (html)**, donde la página es cada uno de los elementos que nos puede presentar un cliente Web. Las páginas contienen texto, enlaces, imágenes, y otros elementos multimedia.

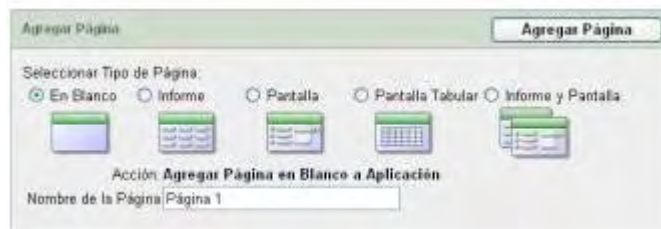
Esta definición es totalmente correcta, pero desde el punto de vista del desarrollo de la capa cliente de una aplicación cliente-servidor, definiremos una página como **cada segmento de un programa o aplicación que se carga en la memoria principal sólo si es necesario que se ejecute.**

Una vez entendido qué es una página, necesitamos conocer cuáles son los **tipos** de los que disponemos para desarrollar nuestra aplicación.

Los **tipos de páginas**, atendiendo a las funcionalidades que presentan, que nos suministra el Oracle Express son las siguientes:



- **Página en blanco.** Crea una página sin ninguna funcionalidad incorporada.
- **Página Informe.** Crea una página que contiene el resultado con formato de una consulta SQL. Puede generar un informe basado en una tabla seleccionada, en una sentencia SELECT de SQL personalizada o en una función PL/SQL que devuelva una sentencia SELECT de SQL proporcionada. Las páginas de informe pueden incluir además **Páginas de Análisis** para crear informes y gráficos de resumen adicionales.
- **Página Pantalla.** Crea una pantalla para actualizar una única fila de una tabla de base de datos.
- **Página Pantalla Tabular.** Crea una pantalla para realizar operaciones de actualización, inserción y supresión en varias filas de una tabla de base de datos.
- **Página Informe y Pantalla.** Crea una combinación de pantalla e informe de dos páginas. En la primera página, los usuarios seleccionan la fila que se vaya a actualizar. En la segunda, los usuarios pueden actualizar la tabla o vista seleccionada.



Herramientas CASE

Separadores

¿Tenemos ya todos los elementos necesarios para construir una aplicación real?

Para poder seguir construyendo nuestra primera **aplicación**, nos hace falta conocer otros términos.

Vamos ahora a conocer qué son los **separadores** usando una de las aplicaciones de ejemplo que proporciona el propio Oracle Express.

Definir qué son los separadores es algo complicado, ya que no existe una definición concreta sobre los mismos, pero con la siguiente imagen comprenderás fácilmente

en qué consisten.



Como puedes ver, en esta aplicación estamos en la página principal, **mediante los separadores el usuario puede acceder a las distintas páginas de la aplicación con sólo pulsar en la opción deseada.**

Por ejemplo, el usuario pulsando en el separador de pedidos (orders) accede (carga) la página asociada a ese separador.



Es decir, podemos definir un separador como **un enlace a una página para acceder a dicha página rápida y fácilmente.**

Pero además de permitirnos un acceso óptimo a las páginas de nuestra aplicación, el uso de **separadores** en los interfaces nos suministran una cualidad añadida, y es que gracias al uso de separadores, el usuario conoce las funcionalidades principales de la aplicación (o de una página) con un solo golpe de vista. Es decir, **el uso de separadores mejora notablemente la ergonomía de nuestra aplicación.**



Oracle Express, mediante su herramienta CASE para el diseño de la capa cliente, nos permite añadir hasta **dos niveles de separadores** en una página. Si deseamos añadir más niveles (lo cual no es aconsejable atendiendo a la ergonomía) deberemos, una vez creada la página, editarla para después modificarla.



Una aplicación con sólo un nivel de separadores utiliza un **juego de separadores estándar**. Un juego de separadores estándar se asocia a una página e identificador de página concretos. Se puede utilizar separadores estándar, por ejemplo, para enlazar usuarios a una página concreta.

Un **juego de separadores principales** funciona como un contenedor para albergar un grupo de separadores estándar. Los separadores principales ofrecen a los usuarios otro nivel de navegación, así como un contexto. Los separadores principales los usaremos para enlazar usuarios a una dirección URL concreta asociada a una página determinada.

Autoevaluación

De las siguientes afirmaciones referidas a los separadores, señala la correcta.

- a) Un juego de separadores estándar se asocia a una o a varias páginas e identificadores de página genéricos.
- b) Una aplicación con sólo un nivel de separadores utiliza un juego de separadores principales.
- c) Un juego de separadores estándar funciona como un contenedor para albergar un grupo de separadores principales.
- d) Los separadores principales los usaremos para enlazar usuarios a una dirección URL concreta asociada a una página determinada.

[Comprobar](#)

Herramientas CASE

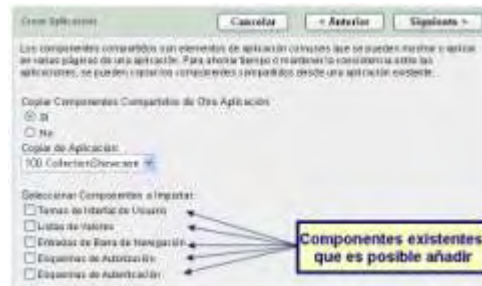
Componentes compartidos

Con lo visto hasta ahora y lo que nos queda por aprender, comprobamos que el trabajo del desarrollo de los **interfaces**, en nuestro caso las páginas, aún con la ayuda incalculable que nos suministra la herramienta del Oracle Express, es una tarea farragosa. Sería de gran ayuda que pudiéramos reutilizar los componentes que hemos desarrollado o que



existen en otra aplicación para usarlos en nuestro programa.

Pues bien, el Oracle Express nos posibilita esta reutilización mediante una de sus funcionalidades, en concreto mediante el uso de los **componentes compartidos**, que **son elementos de aplicación comunes que se pueden mostrar o aplicar en varias páginas de una aplicación**. Es decir, usamos elementos existentes para **ahorrar tiempo y/o mantener la consistencia** entre las aplicaciones, ya que se pueden copiar los componentes compartidos desde una aplicación existente.



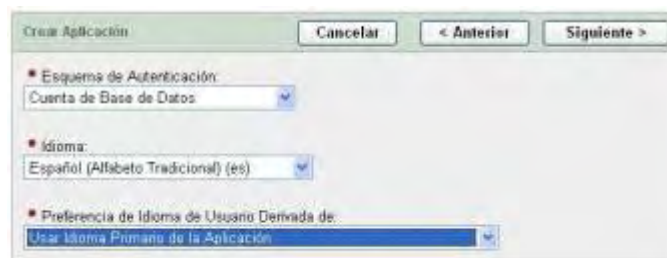
Herramientas CASE

Atributos

Las aplicaciones que creemos con el Creador de Aplicaciones de Oracle Express tendrán una serie de atributos que deben ser definidos. ¿Cuáles son esos **atributos** y cómo los definimos? Ese va a ser el argumento a desarrollar en este apartado.



Es aquí donde definiremos los atributos de nuestra aplicación.



Vamos a estudiar cada uno de ellos:

- **Esquema de Autenticación.** Como ya sabemos, la autenticación es el proceso de establecer las identidades de los usuarios para que puedan acceder a una aplicación. Es aquí donde especificamos el esquema de autenticación que queremos que se utilice por defecto.

Los esquemas de autenticación disponibles son:



- **Autenticación de Oracle Express.** Sólo tendrán acceso a la aplicación los mismos usuarios que los autenticados dentro del Oracle Express.
- **Sin Autenticación.** Los usuarios no necesitan autenticación para acceder a la aplicación.
- **Cuenta de base de datos.** Los usuarios que tendrán acceso a la aplicación son los "autenticados" para la base de datos sobre la que corre la aplicación.
- **Idioma.** Este atributo identifica el idioma en el que se desarrolla una aplicación. Se trata del idioma base del que se realizan todas las traducciones. Es decir, el idioma que vamos a utilizar para todo el desarrollo de nuestra aplicación.
- **Preferencia de Idioma de Usuario Derivada de.** Especifica el modo en que el sistema determina el idioma de la aplicación. El idioma primario de la aplicación puede ser estático (es decir, derivado del idioma del explorador Web) o determinado a partir de un elemento o de una preferencia del usuario. El valor de idioma de la base de datos determina la visualización de fecha y las características de ordenación.



Herramientas CASE

Interfaz de usuario

¿Es posible disponer de distintos aspectos y **estilos** para construir nuestra aplicación, sin necesidad de perder demasiado tiempo definiendo el estilo que más nos gusta o interesa? ¿Nos proporciona el Creador de Aplicaciones esos estilos? Naturalmente, Oracle Express ha pensado también en ello.



La herramienta de Oracle Express nos permite seleccionar doce temas distintos para nuestra aplicación. Los **temas son recopilaciones de plantillas que definen el diseño y el estilo de toda una aplicación.**

Usar un estilo definido en toda la aplicación (en todas las páginas) es muy importante ya que es preciso, siempre que sea posible, mantener la uniformidad de la aplicación (del interface) atendiendo a los principios de ergonomía y facilidad de uso.

Es recomendable que antes de comenzar a desarrollar una aplicación, diseñes los

interfaces, que en tu opinión, dotarían de una mejor funcionalidad a tus páginas, siempre teniendo en cuenta que dichos interfaces deben de conseguir los principales objetivos de la ergonomía. Es decir, debes de dotar a tu aplicación de **un interface lo más amigable posible**.

Bien, ya tenemos los conceptos necesarios para realizar nuestra primera aplicación cliente-servidor mediante Oracle Express.

Por ahora lo que vamos a hacer es una primera toma de contacto con la herramienta de desarrollo para luego ir profundizando en la misma.



Demo: Crear aplicación con la Interfaz de usuario

Herramientas CASE

Desarrollar una aplicación

***Víctor** está contento por haber creado la primera aplicación con su capa cliente para permitir inserciones, actualizaciones y borrados de registros, pero en el fondo sabe que es una aplicación un poco simple. Por lo pronto necesita que la aplicación que está desarrollando debe tener una página donde se muestren los distintos departamentos de la empresa, para que el usuario seleccionando un departamento en concreto, acceda a una pantalla que le muestre todos los empleados que trabajan en ese departamento, y que en ese momento, el usuario pueda actualizar los datos de un empleado determinado. **Carmen** le comenta que eso se puede conseguir de forma casi igual de fácil mediante el uso del Creador de Aplicaciones, sólo hay que desarrollar una **página INFORME Y PANTALLA**, y saber las opciones que hay que ir seleccionando en el asistente. Por esta vez, **Carmen** se va a poner a hacer esta aplicación, para que **Víctor** siga el ejemplo, pero él debe atender con atención, porque la próxima vez, seguramente tendrá que hacer el desarrollo él solo.*



*Pero **Víctor** sigue con dudas. A fin de cuentas, con la página informe y pantalla siguen haciendo las actualizaciones de registro en registro, y hay operaciones, como el cálculo del incremento del sueldo de los empleados de un departamento que se hacen de hecho de la misma forma para todos, y es un poco pesado tener que recorrerlos todos uno a uno haciendo la misma actualización. ¿No dispone el creador de Aplicaciones de alguna forma de facilitar el proceso?*

***Carmen** le informa de que por supuesto, permite hacerlo. Pero para ello hay que usar otro tipo de páginas, concretamente **páginas de tipo PANTALLA TABULAR**.*

*Y evidentemente, **Víctor** ya quiere aprender a hacerlo, porque también es*

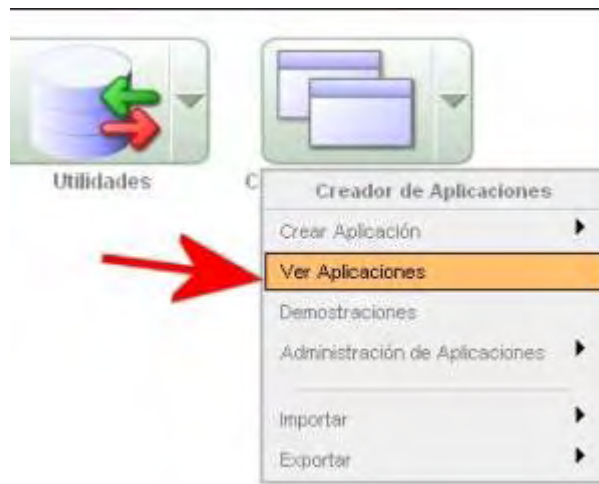
algo que necesita incorporar a la aplicación que está desarrollando.

Víctor: "Y esto, ¿es ya muy complicado?"

Carmen: "Casi tan fácil como todo lo que hemos hecho hasta ahora."

Ya hemos creado nuestra primera aplicación como hemos aprendido en los apartados anteriores, y nos ha resultado sumamente fácil, tanto que incluso podríamos hacernos la siguiente pregunta:

¿Qué es lo que hace esta aplicación?



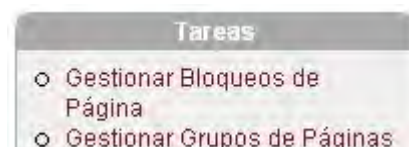
Vamos a ejecutarla para ver el resultado del proceso que hemos seguido. Para ello debemos ver la aplicación mediante la herramienta **VER APLICACIONES** como se muestra en la imagen que acompaña al texto.

Al entrar en **VER APLICACIONES** se nos muestra el menú de **CREADOR DE APLICACIONES**.

Como podemos apreciar, este menú, además de darnos la posibilidad de acceder a las aplicaciones existentes, nos permite la ejecución de una serie de tareas de administración de esas aplicaciones. Nosotros nos vamos a centrar en acceder a la aplicación creada, para ello hemos de pulsar en la aplicación a la que queremos acceder, en nuestro caso Primera, tal como se muestra en la imagen siguiente:



Como vemos, al entrar en el menú de nuestra aplicación, también aquí disponemos de **herramientas de administración**, en este caso, la aplicación que hemos seleccionado.



No es el objetivo de esta unidad, debido a que sería de una extensión excesiva, analizar las herramientas de administración, pero sería conveniente que te familiarizaras con las mismas. Recuerda que dispones de la ayuda de tu tutor o tutora siempre que te sea necesaria.

Como puedes observar, al acceder a la aplicación, disponemos de las siguientes opciones:

- **Ejecutar Aplicación**
- **Editar Atributos**
- **Componentes Compartidos**
- **Exportar/Importar**

Además tenemos la opción de **Crear Página**. Nos vamos a centrar en este punto en las herramientas **Ejecutar Aplicación y Crear Página**, dejando las demás para cuando aprendamos a mantener nuestra aplicación.



Como puedes comprobar, se ejecuta la aplicación accediendo a la primera página de la misma, es decir, la de **autenticación**.

A screenshot of an authentication form. It has a red header bar. Below it, there are two input fields: 'Nombre de Usuario' with the value 'tpe' and 'Contraseña' with masked characters '...'. To the right of the password field is a 'Conectar' button.

La página de autenticación tendrá el formato que hayamos seleccionado cuando creábamos la aplicación. Si recuerdas, cuando la creábamos, elegimos como esquema de autenticación el de cuenta de la base de datos. Quiere esto decir que para acceder a la aplicación hemos de identificarnos como un usuario que hayas creado en la base de datos, como has aprendido en unidades anteriores.

A screenshot of a form titled 'EMPLEADOS'. At the top right are three buttons: 'Cancelar', 'Suprimir', and 'Crear'. The form contains several input fields: 'Cod Emp' (999), 'Nombre' (Nuevo), 'Sueldo' (99999), 'Dpto' (20), 'Email' (tpe@tpe.es), and 'Puesto Trabajo' (FPE).

Una vez autenticados, al acceder a la página empleados, apreciamos las

funcionalidades que nos suministra la aplicación. Desde esta página, el **usuario** podrá crear nuevos empleados o suprimir empleados existentes de la tabla de datos empleados. Vamos a comprobarlo. Para ello introduciremos un nuevo empleado, por ejemplo uno de nombre "Nuevo". Escribiremos los distintos campos y pulsaremos el botón Crear.

Una vez creado, pulsamos sobre el botón Aplicar Cambios, para así grabar el nuevo registro, tal y como se muestra en la siguiente imagen



Con esto ya hemos visto la aplicación que hemos creado. Como has comprobado, gracias a la herramienta del Oracle Express, hemos desarrollado la capa cliente de una aplicación cliente-servidor que permite al usuario crear y suprimir empleados de una forma totalmente transparente y con un interfaz bastante amigable. Por nuestra parte, la del desarrollador, lo hemos conseguido sin tener que escribir una sola línea de código. Imagina por un momento el trabajo que tendrías que haber realizado si no hubiéramos utilizado esta herramienta. Bastante trabajo ¿verdad?



Herramientas CASE

Página informe y pantalla

Pero, ¿no es un poco simple esa aplicación? ¿No nos permite hacer aplicaciones con una funcionalidad más amplia y algo más complicada?

La aplicación que hemos desarrollado tiene una **única página pantalla** (empleados). Si recuerdas, las páginas de tipo pantalla crean una pantalla para actualizar una única fila de una tabla de base de datos. Por ser nuestra primera aplicación es bastante limitada.



Vamos a complicarlo un poco. Imagina que deseas crear una aplicación que tenga una página donde se muestran los distintos departamentos de una empresa, el usuario, seleccionando un departamento en concreto, accede a una pantalla que le muestra todos los empleados que trabajan en ese departamento, y que en ese momento, el usuario podrá actualizar los datos de un empleado determinado. Es decir, necesitamos desarrollar una página informe y pantalla.

DEPT

Restablecer Boton1

Buscar Mostrar 15 Ir

Depto #	Loc	Dname
10	NEW YORK	ACCOUNTING
20	DALLAS	RESEARCH
30	CHICAGO	SALES
40	BOSTON	OPERATIONS

Hoja de Cálculo

1 - 4

Lo primero es comprender bien qué es lo que pretendemos hacer.

Como se ha indicado, la primera página debe mostrar **todos los departamentos** de la empresa. Una posible página sería la de la imagen de la derecha

El usuario selecciona un departamento, en este caso **"RESEARCH"** (investigación). Como resultado deben aparecernos en otra página solamente los empleados adscritos a dicho departamento.

Es decir, el campo departamento es un enlace a la página empleados, que será el reflejo de la consulta "buscar empleados por departamento"

La página informe podría tener la apariencia de la siguiente imagen:

DEPT > EMPLEADOS

Restablecer Crear

Buscar Ir

Depto	Sueldo	Nombre #
20	333	antonio sanchez
20	2	diego rodríguez
20	233	francisco perez
20	194	huan gomez
20	99999	Nuevo
20	2030	paco perez

Hoja de Cálculo

1 - 6

Como puedes apreciar en el campo **Dpto**, sólo se muestran en la página los empleados del departamento investigación, que tiene el código 20. Ahora el usuario puede editar un empleado, pulsando la edición del mismo, y así poder modificar los campos que desee accediendo a la siguiente página.

DEPT > EMPLEADOS > EMPLEADOS

Cancelar Suprimir Aplicar Cambios

Nombre

Sueldo

Dpto

Me Voy

Una vez entendido qué es lo queremos hacer, vamos a aprender cómo hacerlo. Para ello, abre el enlace que aparece a continuación, en el que encontrarás una

presentación de todo el proceso.



Demo: Página informe y pantalla

Herramientas CASE

Página pantalla tabular

Tal vez te hayas planteado que si deseamos modificar o insertar un número elevado de registros de una tabla, el manejo de los mismos mediante las páginas pantalla puede ser un trabajo farragoso, ya que en cada pantalla tendremos un solo registro cada vez. Debería de existir alguna forma de que estas operaciones fueran más simples.



Esa forma existe y es lo que en Oracle Express se denomina **página pantalla tabular**. Si recuerdas, este tipo de página crea una pantalla para realizar operaciones de actualización, inserción y supresión en varias filas de una tabla de base de datos.

Veamos la siguiente imagen para comprender mejor en qué consiste este tipo de página.

ID	NOMBRE	Apellido	Nombre	Direccion	Capital	Cx	Provincia
1	LOPEZ	LOPEZ	JUAN	COLACON 22	10000	10000	BOGOTA
2	PEREZ	PEREZ	PEDRO	AV. 100	20000	20000	BOGOTA
3	RODRIGUEZ	RODRIGUEZ	MARIA	AV. 100	30000	30000	BOGOTA

Como puedes apreciar, se trata de una página estructurada de manera que muestra todos los registros de la tabla y además permite modificar, suprimir o insertar un nuevo registro de una forma muy cómoda para el usuario.

Ahora vamos a aprender a crear una página de tipo pantalla tabular.



Demo: Crear una página de tipo pantalla tabular

Para saber más

Este enlace contiene un muy completo manual de Oracle Express 10g, desgraciadamente está en inglés pero el esfuerzo merece la pena.

Manual de Oracle Express

<http://st-curriculum.oracle.com/tutorial/DBXETutorial/index.htm>
[versión en caché]

Herramientas CASE

Ejecutar la lógica de negocio desde la capa de cliente

Hasta ahora hemos estado conociendo los distintos **tipos de pantallas** que nos proporciona el Oracle Express, es decir, cómo aprovechando la herramienta, podemos decidir la forma en la que se van a presentar y manejar los datos desde la capa cliente. Como hemos aprendido en la unidad ocho, mediante la arquitectura cliente-servidor n-capas, podemos definir la capa cliente, la lógica de negocio y la de datos. Es interesante **poder lanzar la lógica desde el cliente**. Eso es lo que vamos a aprender en este apartado.



Para ello supondremos que uno de los requisitos de nuestra aplicación es que se pueda **calcular** (que el usuario pueda ver) cómo quedarían los sueldos de los empleados si se les aplicara un cierto aumento.



Si recuerdas, en la **unidad ocho** aprendimos a desarrollar un procedimiento almacenado que, dado un código de empleado y un tanto por ciento de incremento, modificaba el sueldo del empleado calculando y aplicando la subida del sueldo (siempre que no se violaran unas restricciones determinadas), es decir, ya podríamos desarrollar directamente una funcionalidad que lanzara ese procedimiento, o si prefieres, solamente cambiando el parámetro del código del empleado por el del departamento al que pertenece podríamos desarrollar una funcionalidad que aplicara el tanto por ciento de subida a todos los empleados de un departamento (de esta manera el usuario no tendría que ir modificando uno por uno a cada empleado).

Pero vamos a complicarlo un poco. Como ya hemos dicho, vamos a suponer que no se desea aplicar la subida directamente, sino que antes se prefiere conocer como quedaría el sueldo del empleado antes de tomar una decisión. Todo ello de una manera que sea cómoda para el usuario.

Bien, lo primero que debemos desarrollar es la función que nos va a permitir esta funcionalidad. El código de la misma es el siguiente:

```
CREATE or REPLACE FUNCTION get_sal
(p_id IN empleados.cod_emp%TYPE,
p_increment IN NUMBER := 1)
RETURN NUMBER
IS
    v_sal empleados.sueldo%TYPE := 0;
BEGIN
    SELECT sueldo * p_increment
    INTO v_sal
    FROM empleados
    WHERE empleados.cod_emp=p_id;
RETURN v_sal;
END get_sal;
```

Vamos a **recordar** un poco lo aprendido en la unidad anterior. Como vemos, la función tiene dos parámetros, uno que es el código del empleado y otro el incremento que queremos comparar. Se usa también una variable local **v_sal**.

La función calcula el incremento de sueldo (por defecto es 1) aplicando una subida que es almacenada en **v_sal**.

Vamos a implementar esta función de una manera distinta a como hemos aprendido en la unidad ocho. Vamos a implementarla desde Introducir Comando tal y como vemos en la siguiente imagen.



Ahora introducimos las sentencias de creación de la función como un conjunto de instrucciones SQL



Como nos muestra la imagen anterior, la función se ha creado correctamente.

Ahora vamos a comprobar el resultado de la función. Para ello introduciremos una instrucción que nos muestre cómo quedarían los sueldos (Sueldo Propuesto) si se aplicara una subida de un 1.4%



Como ves, nos aparece la columna "Sueldo Propuesto", que es la que indica cómo quedarían los sueldos una vez aplicada la subida, pero que no afecta a ninguno de los campos de la tabla.

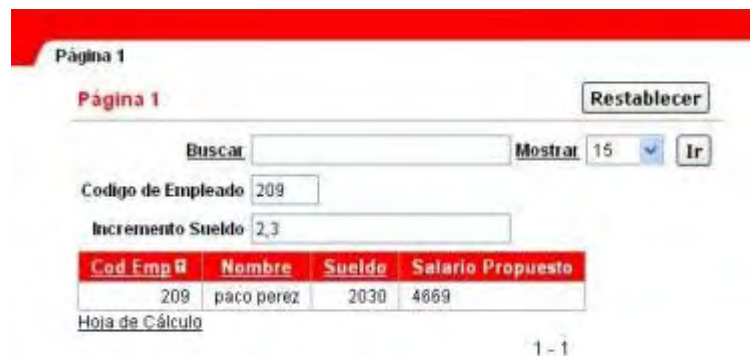
Si realizamos un **select** sobre la tabla **empleados** veremos que el campo **SUELDO** no ha sido modificado.



COD_EMP	NOMBRE	SUELDO
430	Juan gomez	194
209	paco perez	2030
110	antonio sánchez	352,98
999	Nuevo	99999
424	Isidoro perez	233
202	diego rodríguez	2

Ahora lo que nos queda es **aplicar una funcionalidad a nuestro programa que nos permita lanzar la función desde la capa cliente**, es decir, que el usuario ejecute la función implementada de una manera totalmente transparente para él.

Pretendemos diseñar la siguiente pantalla:



Página 1

Página 1 Restablecer

Buscar Mostrar 15

Código de Empleado

Incremento Sueldo

Cod Emp B	Nombre	Sueldo	Salario Propuesto
209	paco perez	2030	4869

[Hoja de Cálculo](#)

1 - 1

Como vemos, el usuario introducirá un **código** y el incremento de sueldo de ese empleado que desea calcular. La aplicación debe mostrar los datos deseados del usuario, su sueldo actual y cómo quedaría una vez aplicado el incremento.

Esta funcionalidad es muy útil. Imagina que cambiando el parámetro del código de empleado, por ejemplo, por departamento al que pertenece, podrías mostrar o aplicar cambios a los registros agrupándolos por algún criterio, en vez de ir registro por registro. Bien, vamos a aprender a realizarlo.



[Demo: Ejemplo para Crear la aplicación](#)

Desarrollo avanzado de una aplicación

Lo único que todavía no tiene claro **Víctor** es cómo integrar varias páginas como las que han venido desarrollando hasta ahora en aplicaciones independientes dentro de la misma aplicación. Naturalmente **Carmen** le dice que no es nada complicado, y que también se hace con la ayuda de la herramienta CASE que vienen usando, el Creador de Aplicaciones de Oracle Application Express.



Sin demasiado esfuerzo, y de una forma que a **Víctor** le ha parecido de lo más claro que puede pedirse, **Carmen** le ha ido mostrando:

cómo crear una tabla maestro-detalle,

cómo agregar nuevas páginas a la aplicación que ya tenían,

cómo permitir el acceso a esas nuevas páginas,

cómo agregar nuevas regiones dentro de una página, y las opciones disponibles para ello, además de definir algunos de los atributos de nuestra aplicación.

No es que sea un experto, pero **Víctor** siente que puede enfrentarse al problema de desarrollar aplicaciones con estructura cliente-servidor de n-capas, y que es capaz de construir para las aplicaciones que tenga que desarrollar en el futuro una capa de usuario o de presentación bastante funcional. La oportunidad de aplicar de forma práctica todo lo que ha aprendido desarrollando este ejemplo no va a tardar en llegar, de eso está seguro, porque en **Si Andalucía** tienen montones de trabajos por realizar, y sabe que la política de **José** es potenciar la formación de los trabajadores de la empresa, pero también rentabilizar todo lo que sus trabajadores hayan aprendido. Seguro que ya le tiene algún trabajo reservado. Y no se equivoca, porque **José** les convoca a **Carmen** y a él a una reunión para asignarles un nuevo proyecto ese mismo día.

Hasta ahora, como habrás comprobado, para explicar cada uno de los tipos más importantes de páginas, con el objetivo de simplificar la explicación, hemos creado una nueva aplicación para cada caso.



Tal vez te estés preguntando que si necesitaras distintos **tipos de páginas** para una aplicación que fueras a desarrollar, deberías crear una nueva aplicación para cada una de ellas. O que si una vez creada una página ésta ya no puede ser modificada, añadiéndole por ejemplo nuevas funcionalidades.

La respuesta es que el desarrollador de aplicaciones de Oracle Express permite todo esto, y ahora vamos a aprender cómo hacerlo.

Agregando nuevas páginas a nuestra aplicación

Vamos a **modificar** una de las aplicaciones que ya hemos desarrollado, por ejemplo la de pantalla tabular. Para acceder a ella la seleccionaremos desde el Creador de Aplicaciones, tal y como se ve en la imagen siguiente:



Accedemos al **menú de mantenimiento** de la aplicación seleccionada. Vamos a suponer que queremos **añadir una funcionalidad** a la aplicación mediante la cual, dada una tabla determinada, podemos acceder a otra tabla que nos muestra los detalles de la primera, permitiéndonos realizar las operaciones de inserción, modificación y borrado en la tabla "subordinada" todo ello en una sola página. Este tipo de página se conoce como **maestro-detalle**.



Es decir, vamos a aprender **dos conceptos** muy importantes:

- **Añadir una página a una aplicación.**
- **Aprender a desarrollar una página de tipo maestro-detalle**

Lo primero que necesitamos es que comprendas lo que pretendemos hacer. Queremos añadir una nueva página a nuestra aplicación "pantalla tabular" en la que podamos, por ejemplo, dados todos los departamentos de la empresa, pulsando en el deseado, acceder a todos los empleados que pertenezcan al mismo y así poder realizar las operaciones de inserción, modificación y borrado sobre los mismos, todo ello en una sola página tal y como muestra la figura de la izquierda.

Pulsando en Editar del departamento de ventas (SALES en la imagen) accedemos a todos los empleados del mismo. La imagen siguiente muestra esos empleados. Vemos que el cuadro de texto **Dname** (nombre de departamento) tiene como contenido SALES (Ventas) y que la tabla muestra todos los empleados de ese

departamento con sus atributos para cada empleado, como Nombre de empleado (Ename), Puesto de Trabajo (Job), etc.

Dept Detail

	Emp	Job	Mgr	Hiredate	Sal	Comm
<input type="checkbox"/>	BLAKE	MANAGER	7818	01/05/81	2900	
<input type="checkbox"/>	ALLEN	SALESMAN	7818	20/03/81	1600	.300
<input type="checkbox"/>	WARD	SALESMAN	7818	22/05/81	1200	.500
<input type="checkbox"/>	WARD	SALESMAN	7818	20/05/81	1200	.500
<input type="checkbox"/>	TURNER	SALESMAN	7818	08/09/81	1500	
<input type="checkbox"/>	JAMES	CLERK	7818	03/12/81	800	

1 de 1

Anterior Siguiente

Suprimir Aplicar Cambios

Abrir Fichero

Una vez comprendido qué queremos hacer vamos a aprender a realizarlo.



Demo: Agregando nuevas páginas a nuestra aplicación

Como puedes apreciar, se han añadido dos nuevas páginas a nuestra aplicación: Dept y Master Detail.



Herramientas CASE

Permitiendo el acceso a las nuevas páginas

Bien, ya hemos aprendido a añadir páginas a nuestra aplicación según vayan aumentando las necesidades de la misma, pero ahora surge un problema. Cuando el usuario ejecuta la aplicación accede a la página Conectar y una vez autenticado es direccionado a la página clientes. Es decir, no tiene acceso a la página nueva que hemos añadido.



¿Cómo podemos permitir el acceso del usuario a las nuevas páginas?

Esto lo haremos **modificando los separadores** de la página clientes.

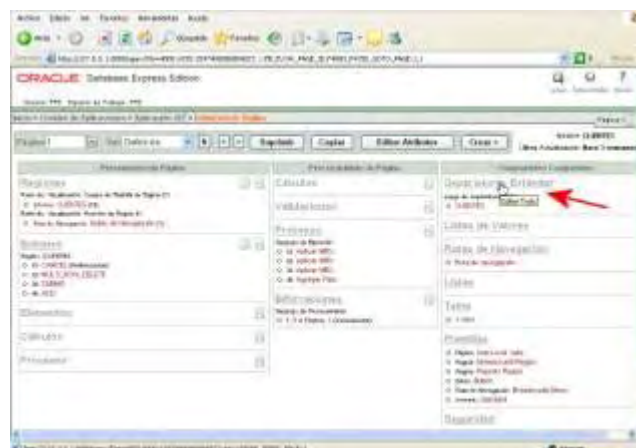
Como puedes comprender, es muy aconsejable que a la hora de diseñar una aplicación, dotes a la misma de **una página principal** desde la cual el usuario pueda acceder a las distintas páginas. Así , cuando añadas una nueva página sólo tienes que añadir un acceso en el separador de la página principal. Un ejemplo del uso de una página principal lo puedes ver en la imagen del apartado 4.1.3 "SEPARADORES", de esta misma unidad.

Vamos a aprender a modificar los separadores de nuestras páginas.

Lo primero es acceder a las propiedades de la página, en este caso clientes, pulsando en ella.



Así accedemos a las propiedades de la página. Ahora, dentro de **Componentes Compartidos**, hemos de pulsar en **Separadores Estándar** tal y como se muestra en la imagen siguiente:



Ahora hemos de **agregar un nuevo enlace en el separador**.



Vamos a denominar a este enlace **Maestro / Detalle** y pulsamos **siguiente**.



Pinchamos en **Separador Actual para Página** para seleccionar a dónde debe enlazar el separador. Elegiremos la página **Dept**



Dejamos los valores por defecto y pulsamos **siguiente**.



En las sucesivas pantallas que nos aparecerán, dejaremos los valores por defecto y pulsaremos **siguiente** hasta que lleguemos a la pantalla de **Crear el separador**.

Crear Separador Estándar

Cancelar < Anterior **Crear Separador**

Confirmar Creación de Separador Estándar en Aplicación 107.

La página 8 utilizará el juego de separadores existente "TS1".

Juego de Separadores	TS1
Texto de Separador	Maestro / Detalle
Página Actual	8
Juego de Separadores Principales	TS1
Nuevo Juego de Separadores Principales	TS1
Texto de Separador Principal	=
Secuencia del Separador	29
Imagen Actual	
Imagen no Actual	
Atributos de Imagen	
Condición de Visualización	
Mostrar Texto de Condición	

Cuando ejecutemos nuestra aplicación accederemos a la página clientes, en la cual existe un nuevo separador, Maestro/Detalle, mediante el cual el usuario podrá acceder a dicha página tal y como se muestra en la figura.

CLIENTES

CLIENTES

Cancelar Suprimir Ejecutar

	NIF_DISPLAY	Apellidos	Nombre	Domicilio	Localidad	Cp	Provincia
<input type="checkbox"/>	1	LOPEZ LOPEZ	LUIS	C/LA CALLE	SEVILLA	99999	SEVILLA
<input type="checkbox"/>	2	RODRIGUEZ GRAC	DIEGO	I.E.S. AGUADULCE	AGUADULCE	04720	ALMERIA
<input type="checkbox"/>	21	PEREZ PEREZ	PEDRO	AVDA/LA AVENIDA	GRANADA	00000	GRANADA

1 - 3

Agregar Fila

Herramientas CASE

Agregando nuevas regiones a nuestras páginas

¿Qué son las regiones de una página? ¿Podemos cambiar las regiones de una página para adaptarlas a nuestras necesidades?

Las **regiones** de una página las podemos definir como las **partes en las que se divide la página para ofrecer funcionalidades distintas en cada una de esas partes.**



Es decir, que a lo largo del desarrollo de nuestra aplicación es posible que tengamos que añadir, modificar o suprimir regiones existentes debido a nuevas necesidades.

Para aprender a realizar estas operaciones vamos a partir de un supuesto:

Imaginemos que una vez que hemos desarrollado la página maestro/detalle se nos pide que a la vez que se ven los sueldos de los empleados, la aplicación nos

muestre qué tanto por ciento del total supone un sueldo sobre la suma de todos los sueldos.

Una solución posible sería la siguiente:



Como puedes observar, hemos añadido una nueva región que contiene un gráfico. Vamos a aprender cómo se hace.



Demo: Agregando nuevas regiones a nuestras páginas

Para saber más

Aquí encontrarás una muy buena explicación sobre las aplicaciones web y sus posibilidades de negocio.

Aplicaciones Web

<http://www.monografias.com/trabajos5/laweb/laweb.shtml> [versión en cache]

Autoevaluación

De las siguientes afirmaciones referidas a las regiones, señala la correcta.

- a) Al seleccionar Control Compartido estamos indicando que la nueva región debe de estar incluida en la página actual.
- b) Al seleccionar Control Compartido estamos indicando que la nueva región será un componente que se podrá usar en las demás páginas de la aplicación.
- c) El control compartido puede ser, entre otros, de los siguientes tipos: Separador Principal, Separador Estándar, Lista de Valores y Validación.
- d) El control compartido puede ser, entre otros, de los siguientes tipos: Lista, Ruta de Navegación y Lista de Valores.

Comprobar

Opciones al agregar nuevas regiones a las páginas

Has observado que cuando añadimos una **nueva región**, la herramienta de Oracle nos permite varias opciones. Vamos a analizarlas.

En la pantalla de creación podemos observar los siguientes tipos.

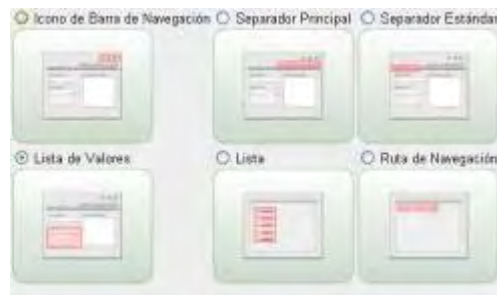


- Si seleccionamos **Nueva Página** accederemos a la pantalla de creación de páginas que ya conocemos.
- Al seleccionar **Región en esta Página** estamos indicando que la nueva región debe de estar incluida en la página actual. Podremos seleccionar el tipo de página que se va a incluir y que puede ser:
 - HTML
 - Informe
 - Pantalla
 - Gráfico
 - Ruta de Navegación
 - Contenido dinámico PL/SQL
 - Árbol
 - URL
 - Calendario
 - Varios HTML
 - Texto de Ayuda
- Al seleccionar **Control de Página** estamos indicando que la nueva región se encargará de realizar funciones de control en la propia página y que pueden ser utilizados para direccionar la aplicación a otra página.

El control de página puede ser del tipo que se muestra en la figura:



- Elemento.
- Botón.
- Bifurcación.
- Cálculo.
- Proceso.
- Validación.



- Al seleccionar **Control Compartido** estamos indicando que la nueva región será un componente que se podrá usar en las demás páginas de la aplicación. Los tipo de controles compartidos que podemos insertar son:
 - Icono de Barra de Navegación.
 - Separador Principal.
 - Separador Estándar.
 - Lista de Valores.
 - Lista.
 - Ruta de Navegación.

Herramientas CASE

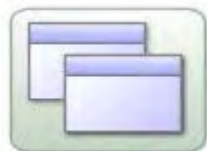
Atributos

Para finalizar, vamos a hacer un breve recorrido por los atributos y componentes compartidos de nuestra aplicación.

Podemos definir los **atributos de aplicación** como los identificadores de la misma. Como hemos visto, el identificador de una aplicación es un número.

Los tipos de atributos son los siguientes:

- **Atributos estándar.**
- **Atributos de seguridad.**
- **Atributos de globalización.**



Editar Atributos Estándar



Editar Atributos de Seguridad



Editar Atributos de Globalización

■ **Atributos estándar.**

Los atributos estándar de aplicación. Se usan estos atributos para controlar el **nombre de la aplicación** y disponibilidad al igual que se pueden definir las cadenas de sustitución. También muestran opciones definidas de construcción, el tema asociado, plantillas por defecto y componentes por defecto.



■ **Atributos de seguridad.**

Estos atributos se usan para definir valores de seguridad a nivel de aplicación. También se utilizan las páginas de definición de componentes de aplicación para gestionar valores más específicos. Los atributos son los siguientes:

1. **Autenticación.** La autenticación es el proceso que establece la identidad de cada usuario antes de que pueda acceder a la aplicación. Puede definir varios esquemas de autenticación para la aplicación, pero sólo uno de ellos puede ser el actual. Cuando la aplicación se ejecuta, se utiliza la lógica de autenticación del esquema actual.
2. **Autorización.** Los esquemas de autorización de la aplicación controlan el acceso a todas las páginas de una aplicación. El acceso no autorizado a la aplicación, independientemente de la página que se haya solicitado, hace que se muestre una página de error.
3. **Esquema de Base de Datos.** Todos los comandos SQL y PL/SQL emitidos por una aplicación se ejecutan con los derechos y privilegios del esquema de base de datos definido. El dominio de esquemas disponibles se define por espacio de trabajo.
4. **Protección de Estado de la Sesión.** La activación de la protección de estado de la sesión puede evitar que los piratas informáticos manipulen las direcciones URL de la aplicación. La manipulación de la dirección URL puede afectar de forma negativa a la lógica del programa, al contenido del estado de la sesión y a la privacidad de la información.
5. **Base de Datos Privada Virtual (VPD).** La Base de Datos Privada Virtual (VPD) proporciona una interfaz de programas de aplicación ([API](#)) que permite a los desarrolladores asignar políticas de seguridad a las tablas y vistas de bases de datos. Mediante PL/SQL, los desarrolladores pueden crear políticas de seguridad con procedimientos almacenados y enlazar los procedimientos a una tabla o vista mediante una llamada a un paquete [RDBMS](#). Tales políticas



se basan en el contenido de los datos de la aplicación almacenados en la base de datos, o en variables de contexto proporcionadas por la base de datos Oracle. De este modo, VPD permite eliminar los mecanismos de seguridad de acceso de las aplicaciones y realizar una gestión centralizada. La llamada de VPD se utiliza para definir el contexto de seguridad de las aplicaciones que necesitan VPD. Si la aplicación no necesita soportar varios clientes en la misma base de datos, dejaremos este atributo como nulo.

■ Atributos de globalización.

Mediante los atributos de globalización de la aplicación se puede acceder a las traducciones de la aplicación siguiendo los siguientes enlaces:

1. **Página Inicial de Traducción.** Las aplicaciones se pueden traducir del idioma primario a otro idioma.
2. **Asignaciones de Traducción.** Las aplicaciones traducidas se publican como nuevas aplicaciones. Debe especificar un identificador de aplicación de idioma primario y un identificador de aplicación de idioma traducido para cada idioma que se desee traducir.
3. **Exportar Traducciones.** Para traducir una aplicación, primero debe extraer el texto para traducir. Al extraer la traducción, se copia todo el texto traducible en un repositorio de texto de traducción. Cuando se extrae el texto traducible, se puede iniciar el proceso de traducción de las cadenas de texto traducibles. Después de extraer el texto de traducción de la aplicación e idioma específico y copiarlo en el repositorio de texto de traducción, puede generar y exportar un archivo [XLIFF](#) para traducirlo. El proceso de extracción mantiene la aplicación en el idioma primario sincronizada con el repositorio de texto de traducción. Por tanto, debe ejecutarse cada vez que haya cambios en la aplicación en idioma origen.
4. **Importar Traducciones.**
5. **Definición de Mensaje.** Se pueden utilizar mensajes de texto para crear cadenas de texto traducibles con variables de sustitución que se pueden llamar desde paquetes, procedimientos y funciones PL/SQL.
6. **Traducción de Mensaje.** Los mensajes son cadenas de texto con nombre que se pueden llamar desde el código PL/SQL escrito por el usuario. Este código PL/SQL puede constituir bloques anónimos dentro de los procesos de página y regiones de página o en paquetes y procedimientos.
7. **Traducciones Dinámicas.** Las traducciones dinámicas permiten que el código PL/SQL realice traducciones con la función `HTMLDB_LANG.LANG`. Esta tabla asigna valores del idioma primario a otro idioma nacional.



Herramientas CASE

Componentes compartidos: tipos posibles

Los **componentes compartidos**, como ya hemos indicado, son los componentes de la aplicación que **se pueden utilizar en cualquier**



página de la aplicación.

Los componentes compartidos pueden ser de tipo:

- **Lógico.**
- **De navegación**
- **De seguridad.**
- **De interfaz de usuario.**
- **De globalización.**
- **Archivos**

Veamos cada uno de ellos de forma un poco más detallada en los siguientes apartados.

Herramientas CASE

Componentes compartidos: tipos "lógico" y "navegación"

- **Lógico.**

- **Elementos de Aplicación.** Se utilizan para mantener el estado de la sesión. Los elementos de la aplicación se pueden definir mediante cálculos o procesos, o bien transfiriendo valores en una dirección [URL](#). Se usan para mantener el estado de la sesión que no se muestra ni es específico de ninguna página.
- **Procesos de Aplicación.** Ejecutan lógica PL/SQL en puntos específicos para cada página de una aplicación. También definen las condiciones que deben cumplirse tanto para que se ejecute la página como para la forma de hacerlo.
- **Cálculos de Aplicación.** Se usan para asignar valores a elementos de página y aplicación.
- **Referencias de Servicio Web.** El sistema de Application Express puede utilizar las referencias de servicio Web para acceder a un servicio Web a través de la red. El servicio Web realiza una acción y, a continuación, devuelve una respuesta.
- **Opciones de Creación.** Se usan para mostrar de forma condicional la funcionalidad específica en una aplicación.



- **Navegación.**

- **Árboles.** Un árbol es un mecanismo de navegación jerárquica. Se puede crear uno a partir de una consulta que especifique una relación jerárquica mediante la identificación de la columna de identificador y de identificador principal en una tabla o vista.
- **Entradas de Barra de Navegación.** Las entradas de barra de navegación proporcionan una navegación basada en enlaces de [hipertexto](#). La ubicación se determina mediante



plantillas de página.

- **Listas.** Una lista es una recopilación de enlaces compartidos y controlados por plantillas. Se usan las listas para agregar navegación a la aplicación.
- **Rutas de Navegación.** Las rutas de navegación proporcionan una navegación jerárquica hasta un número infinito de niveles. Se puede utilizar rutas de navegación para mostrar rutas de navegación tradicionales de un nivel u otros tipos de jerarquías de estilo similar.
- **Separadores.** Como ya hemos visto, podemos definir un separador como un enlace a una página para acceder a dicha página rápida y fácilmente.

Herramientas CASE

Componentes compartidos: tipos "seguridad" e "interfaz de usuario"

■ Seguridad.

- **Esquemas de Autenticación.** Un esquema de autenticación es una configuración guardada que se puede aplicar a la aplicación.

- **Esquemas de Autorización.** La autorización para aplicaciones, páginas y para la mayor parte del resto de los componentes se gestiona mediante los esquemas de autorización.

El esquema de autorización se asocia a una aplicación, página o cualquier otro componente en la página de atributos del componente.

Los esquemas de autorización son comprobaciones de aprobación/fallo. Los tipos de esquema de autorización comunes son: consultas SQL Existe y No Existe y Función PL/SQL que devuelve valor booleano.



- **Protección de Estado de la Sesión.** La activación de la protección de estado de la sesión puede evitar que los piratas informáticos manipulen las direcciones URL de la aplicación. La manipulación de la dirección URL puede afectar de forma negativa a la lógica del programa, al contenido del estado de la sesión y a la privacidad de la información.
- **Editar Atributos de Seguridad.** Ya lo hemos visto en los atributos. Estos atributos son también componentes compartidos.

■ Interfaz de usuario.

- **Temas.** Un tema es una recopilación con nombre de plantillas utilizadas para definir la interfaz de usuario de una aplicación.
- **Plantillas.** El sistema Application Express crea el aspecto de cada página de una aplicación mediante plantillas. Las plantillas definen la forma en que se van a mostrar las páginas, los

Lista de Valores

Mostrar	Devolver
FPE	FPE
HERRAMIENTAS 4GL	4GL
S.I.M.R.	SIMR
PLE	PLE
ANALISIS	ANALISIS

controles de página o los componentes de páginas.

- **Valores por Defecto de Interfaz de Usuario.** Los valores por defecto de interfaz de usuario se utilizan para rellenar con valores iniciales las propiedades de región y elemento, a fin de proporcionar consistencia entre páginas de una aplicación o de varias aplicaciones.
- **Listas de Valores.** Tanto los elementos de página como los campos de informe pueden hacer referencia a la lista de valores. La lista de valores controla los valores mostrados y limita la selección del usuario. Las listas de valores pueden ser estáticas (basadas en los valores introducidos) o dinámicas (basadas en la consulta SQL).
- **Métodos Abreviados.** Se utilizan los métodos abreviados para escribir una vez el código que se utiliza con frecuencia y hacer referencia a éste en muchas ubicaciones de la aplicación. Es decir, permiten la reutilización de los componentes software.

Herramientas CASE

Componentes compartidos: tipos "de globalización" y "archivos"

- **De globalización.**
 - Ya los hemos visto en los atributos, por lo que no lo vamos a repetir aquí. .
- **Archivos.**
 - **Archivos Estáticos.** Los [archivos estáticos](#) los podemos definir como los archivos que no deben ser modificados por el usuario, es decir, que no son modificables en tiempo de ejecución de la aplicación, como son, por ejemplo, imágenes y hojas de estilos.
 - **Hojas de Estilo en Cascada.** Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o [XML](#) (y por extensión en [XHTML](#)). El [W3C](#) (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores.
 - **Imágenes.** Las imágenes de Application Express se dividen en dos categorías:
 1. **Imágenes de Espacio de Trabajo** que están disponibles para todas las aplicaciones de un espacio de trabajo determinado.
 2. **Imágenes de Aplicación** que están disponibles sólo para una aplicación.



Autoevaluación

Empareja cada uno de los siguientes componentes compartidos de

tipo navegación con su correspondiente descripción.

- | | | |
|---|---------------|--------------------------|
| a) Proporcionan una navegación jerárquica hasta un número infinito de niveles. | Selecciona... | <input type="checkbox"/> |
| b) Son una recopilación de enlaces compartidos y controlados por plantillas. Se usan para agregar navegación a la aplicación. | Selecciona... | <input type="checkbox"/> |
| c) Es un mecanismo de navegación jerárquica. Se puede crear uno a partir de una consulta que especifique una relación jerárquica mediante la identificación de la columna de identificador y de identificador principal en una tabla o vista. | Selecciona... | <input type="checkbox"/> |
| d) Es un enlace a una página para acceder a dicha página rápida y fácilmente. | Selecciona... | <input type="checkbox"/> |

Comprobar

Herramientas CASE