



ORACLE APEX

Desarrollo de Aplicaciones Web de Base de Datos

ABSTRACCIÓN

Guía de apoyo para utilizar durante la formación guiada sobre Oracle Application Express, para crear aplicaciones Web de una manera sencilla y rápida sin prácticamente escribir una sola línea de código.

Juan López García

Novoshore Europe S.L.

Generando apps con Oracle Application Express

Antes de empezar...

Este documento es una referencia de apoyo para la realización de una serie de prácticas, que pretende facilitar y agilizar la formación sobre Oracle Application Express (APEX) en su versión 19.2.

La primera recomendación para llevar a cabo esta introducción a APEX pasa por disponer de un workspace “limpio” donde realizar las prácticas que servirán como referencia guiada a la hora de conocer el entorno, evitando así manipular por error objetos de base de datos que usen otras aplicaciones. Quizás, lo más cómodo sea solicitar un espacio gratuito en apex.oracle.com para desarrollo y hacer las prácticas en el mismo, minimizando así el riesgo de modificar o eliminar otros elementos de base de datos de un workspace ya existente.

Del mismo modo, esta documentación se apoya sobre algunos ficheros, se adjunta un fichero .zip utilizados durante las prácticas.

Práctica 1. Creación de un WS en APEX.ORACLE.COM

Disponer de un entorno de prueba siempre es una opción recomendable, para hacer pruebas sin riesgo de corromper los objetos de base de datos de otros entornos.

1. Visitar apex.oracle.com
2. Solicitar un Workspace gratuito.
3. Confirmar en el correo.
4. Manipular usuarios del WorkSpace.

Práctica 2. Instalación de una Packaged App

Oracle APEX cuenta con una batería de aplicaciones completas y funcionales, soportadas por la comunidad de APEX, que pueden constituir buenos ejemplos a la hora de aprender a desarrollar apps propias.

1. Seleccionar la opción **Packaged Apps**.
2. Seleccionar la aplicación de nuestro interés; por ejemplo, Sample Charts.
3. Manipular la aplicación como deseemos. Se trata de aplicaciones completamente funcionales y constituyen buenos ejemplos acerca de “Cómo hacer las cosas en APEX”.

Practica 3. Importación de un Script y ejecución. Cargar Datos.

Aunque la mayoría de desarrolladores utilizan herramientas como Oracle DB Developer, también puede usarse el IDE de APEX para gestionar nuestros objetos de base de datos.

1. Importar usando SQL Workshop > SQL Scripts el fichero ‘**Project_Basic_Tables.sql**’. Dicho fichero se puede revisar, editar y manipular si es necesario. Ejecutar el script.
2. Hacer lo mismo con el fichero ‘**Project_Data.sql**’.
3. Ahora, vamos a rellenar algunos datos mediante el siguiente comando, en SQL Workshop > SQL Commands:

```
begin
demo_projects_data_pkg.load_sample_data;
end;
```

NOTA: Este pequeño script se encarga de simular una inserción de datos como si se estuviese usando la estructura de BD recién creada, haciendo uso de la función contenida en el correspondiente paquete.

4. Accediendo a SQL Workshop > Object Browser, podemos ver los datos introducidos, por ejemplo, en la table DEMO_PROJ_TEAM_MEMBERS, seleccionando la tabla y pulsando en la pestaña Data.



Ahora vamos a cargar datos desde un fichero y a crear una tabla de Lookup usando las herramientas que nos facilita APEX:

1. Vamos a SQL Workshop > Utilities > Data Workshop.
2. Data Load > Spreadsheet Data.
3. Aparece un asistente, y seleccionamos New Table y Upload File.
4. Seleccionamos el fichero **harware.csv**
5. Tras verificar los tipos de las columnas, cargamos los datos. Así, hemos creado una nueva tabla con los datos del fichero.
6. Ahora en SQL Workshop > Object Browser, sobre la tabla HARDWARE, buscamos el botón Create a Lookup Table.
7. Seleccionamos la columna DEPARTMENT y tras seguir el asistente, hemos creado una tabla de lookup o consulta.

Práctica 4. Creación de una App

A la hora de crear una app, podemos utilizar el asistente de creación para generar la app desde cero, o podemos hacerlo a partir de una hoja de cálculo.

Creación de una aplicación usando el Asistente

En este caso, crearemos una app utilizando solamente las herramientas que nos brinda el asistente de creación de Apps.

1. Desde el App Builder, seleccionamos la opción de crear (Create) aplicaciones.
2. Elegimos ahora el tipo de App. En nuestro caso, queremos que sea una app de escritorio (Desktop).
3. Asignamos un nombre a la app.
4. En la página de añadir páginas, incluiremos reports y formularios para las tablas. Para crear el correspondiente formulario asociado, solo tenemos que seleccionar en la correspondiente casilla.
 - a. DEMO_PROJ_TEAM_MEMBERS – Interactive Report
 - b. DEMO_PROJECTS – Interactive Report
 - c. DEMO_PROJ_MILESTONES – Interactive Report
 - d. DEMOS_PROJ_TASKS – Interactive Report
5. Marcamos las opciones “Check all”, para luego visualizar su efecto en la app.
6. En Advanced, seleccionamos mostrar
7. Pulsar el botón Create Application

Ahora podemos navegar dentro de la app y ejecutarla para observar el resultado de lo que hemos realizado.

Creación de una aplicación a partir de una hoja de cálculo

Ahora crearemos de forma muy rápida una app, apoyándonos en una hoja de cálculo que contiene datos.

1. Desde el App Builder, seleccionamos la opción de crear (Create) aplicaciones.
2. Ahora, usamos el enlace etiquetado como “From a spreadsheet”.
3. Cargamos un archivo .csv, separado por comas o delimitado por tabulaciones: **‘budget.csv’**.
4. Seleccionamos el fichero, así como el separador, etc y pulsamos en siguiente.
5. Nombramos la tabla como PROJECT_BUDGET – Nota: nos aseguramos que todas las columnas tienen el tipo correcto (START_DATE / END_DATE).
6. Seleccionamos Interactive Report como tipo de formulario.
7. Pulsamos el botón de Create Application.

Ahora podemos navegar dentro de la app y ejecutarla para observar el resultado de lo que hemos realizado.



Práctica 5. Manejo de páginas en APEX.

En este punto podemos navegar por la interfaz de Page Designer y examinar las herramientas de trabajo de que se disponen. Además, podemos insertar en la página Home de **Demo Projects** una región Chart para comprobar lo sencillo que puede llegar a ser el método de Drag&Drop.

1. En la página Home, insertamos un Chart en la región Content Body.
2. Llamamos al gráfico Project Tasks.
3. En el Editor de Propiedades (Property Editor), definimos el gráfico como Bar – Horizontal – Stack: Yes.
4. La serie Y se llamará Tasks, y usará una columna de la tabla DEMO_PROJ_TASKS llamada IS_COMPLETE_YN, que indica si una tarea está o no terminada.
5. Ahora entramos en la sección Series, y nombramos la serie como Tasks.
6. Incluimos la siguiente consulta en el campo Source, con el tipo SQL Query:

```
select p.id
, p.name as label
, (select count('x') from demo_proj_tasks t
where p.id = t.project_id
and nvl(t.is_complete_yn, 'N') = 'Y'
) value
, 'Completed Tasks' series
, p.created
from demo_projects p
UNION ALL
select p.id
, p.name as label
, (select count('x') from demo_proj_tasks t
where p.id = t.project_id
and nvl(t.is_complete_yn, 'N') = 'N'
) value
, 'Incomplete Tasks' series
, p.created
from demo_projects p
order by 5
```

7. En el mapeo de columnas seleccionamos:
 - a. Series Name: **SERIES**
 - b. Label: **LABEL**
 - c. Value: **VALUE**

Ahora, al navegar a la página Home, aparecerá un gráfico de barras horizontal que muestra la información de la consulta. Un poco más adelante podremos incidir algo más sobre los gráficos en APEX.

Práctica 6. Creación de Reports (Classic & Interactive report – Interactive Grid)

En Oracle APEX existen, por defecto, tres tipos de Informes o Reports. A continuación, procederemos a la creación de cada uno de estos tipos, y más adelante analizaremos el uso de Interactive Reports y de Interactive Grids.

Creación de un Classic Report

Vamos a insertar un informe clásico en la página 1 de nuestra app.

1. Creamos una nueva región en Content Body de Página 1, de tipo **Clasic Report** y con el Título **My Outstanding Tasks**.



2. Incluimos la siguiente consulta como fuente (Source):

```
select p.name project
, t.name task
, t.end_date
from demo_proj_tasks t
, demo_projects p
, demo_proj_milestones m
, demo_proj_team_members tm
where p.id = t.project_id
and m.id = t.milestone_id (+)
and tm.id = t.assignee and nvl(t.is_complete_yn, 'N') = 'N'
and upper(tm.username) = upper(:APP_USER)
order by t.end_date
```

3. Editamos ahora los atributos del informe para que se vea un poco mejor:
 - a. En la Sección de Appearance, seleccionamos Template Options > Remove Body Padding.
 - b. Ahora en Attributes, dentro de la región Template, seleccionamos Stretch Report / No Outer Borders.

Si ejecutamos ahora la app, podremos ver un informe generado sobre la consulta del ejemplo, en la que se muestran las tareas pendientes del usuario logueado.

Creación de un Interactive Report

Ahora, vamos a generar un informe interactivo a partir de la tabla **HARDWARE** que generamos en la práctica 3.

1. Creamos una nueva aplicación de escritorio llamada **HARDWARE**.
2. Ahora, dentro de App Builder, pulsamos en el botón de Create Page.
3. Seleccionamos ahora Report > Interactive Report. Al seguir los distintos pasos del asistente, seleccionamos las siguientes opciones:
 - a. Page Name: Hardware Interactive Report / Select Breadcrumb / Parent: Home.
 - b. Create New Navigation Menu entry / Parent: Home
4. Seleccionamos la tabla **Hardware** como fuente, y creamos la página.

Si navegamos ahora a la app, podremos encontrarnos el informe que acabamos de generar.

Creación de un Interactive Grid

Para crear este tipo de informe, navegamos hasta la app Budget App.

1. Creamos una nueva página dentro de la app del tipo Report > Interactive Grid.
2. Seleccionamos las siguientes opciones de configuración:
 - a. Create New Navigation Menu entry: Interactive Grid
3. Este informe usará la table **Project_Budget** como fuente, y creamos la página.

Si navegamos ahora a la app, podremos encontrarnos el informe que acabamos de generar.

Creación de otros tipos de informes

Además de estos tres informes principales, podemos encontrar otro tipo de informes disponibles en Oracle APEX. Dentro de la aplicación Hardware creamos, a partir de la tabla Hardware, los siguientes informes, incluyendo las correspondientes entradas de navegación:



1. Toggle Report
2. Reflow Report
3. Faceted Search.

Si navegamos ahora a la app, podremos encontrarnos los informes que acabamos de generar.

Práctica 7. Uso de un Interactive Report

En esta práctica, vamos a abordar la manipulación de los Interactive Reports desde dos puntos de vista: a nivel de usuario final o como desarrollador.

Utilizando un informe interactivo como usuario final

En esta primera parte, veremos como el usuario final de nuestra app puede manipular un informe interactivo. Para ello, recorreremos las funcionalidades principales que pueden ser de nuestro interés. Manipularemos el informe Interactivo de la app Hardware Interactive Report.

1. Añadir / quitar columnas del informe (Quitar: Id – Form Factor)
2. Reordenar orden de columnas (bajar dos posiciones Purchase Date)
3. Añadir filtrados (Purchase Price >= 5330 – CPU Type = Pentium III)
4. Añadir Filtrado con ambas condiciones a la vez (row type: C='Pentium III' AND I>=5300)
5. Ordenar por valor de una columna (Short, Purchase Price, Ascending)
6. Mostrar una Suma de una columna (Agregate, Sum, Purchase Price)
7. Realizar una computación para sumar tasas (Compute, Price with Tax, máscara de formato, Purchase Price * 1.05)
8. Ruptura de grupo (Control Break, CPU Type)
9. Remarcar o subrayar (Highlight, Price with Tax less than \$2400, yellow/red)
10. Chart (Horizontal Column, CPU Type, Price with Tax, Sum, CPU Type, Price with Tax)
11. Group by (CPU Type, Sum, Purchase Price, Total Price, máscara de formato)
12. Pivotar (CPU Type, Department Id, Count, CPU Type)
13. Guardar informe / Resetear informe
14. Descargar Informe (csv, pdf)

Como se puede observar, hay muchas opciones de manipulación de informes por parte del usuario final que utiliza la app, una vez dispuestos los datos de la consulta. Además, existe una guía de ayuda en el propio menú de Acciones del informe interactivo.

Customizando un informe interactivo como desarrollador

Como desarrollador, también se puede editar ciertas opciones que queremos (o no) que se utilicen en los Informes interactivos. Para ver algunas de estas funciones, vamos a editar alguno de esos elementos.

1. En el informe anterior, no queremos que se muestren las columnas Id y Form Factor. Para ello, introducimos el siguiente código en la región de Source:

```
select SERIAL,  
CPU_TYPE,  
CPU_SPEED,  
PURCHASE_DATE,  
BRAND,  
MODEL,  
PURCHASE_PRICE,  
DEPARTMENT_ID  
from HARDWARE
```

2. Incluimos vista a una sola línea del informe con todos sus detalles. Para ello, lo habilitamos en Atributos – Link to single Row View.



3. Elegimos el tipo de paginación, cambiándola a Row Ranges X to Y of Z.
4. Elegimos el máximo número de filas por página (maximum rows per page – 10)
5. Habilitamos la opción de guardar informes públicos (Actions Menu – Save Public Report)
6. Cambiamos las opciones de descarga de los informes (Quitar Email, RTF)

Práctica 8. Uso de un Interactive Grid

Del mismo modo que hemos visto con los Informes Interactivos, hay muchas opciones para manipular un Grid Interactivo. Se puede abordar esta manipulación desde el punto de vista de un desarrollador, así como usuario final. Muchas de las acciones que se van a mostrar en este punto, también son aplicables a la edición y customización de los Interactive Reports, procediendo de un modo similar al que se aborda en esta sección.

Customización de un Interactive Grid: Project Team Members

En este apartado, modificaremos como desarrolladores las columnas que queremos que se muestren o no en un Grid Interactivo.

1. Editamos la página 2 de Demo Projects: Demo Proj Team Members.
2. En la consulta, insertamos el siguiente código:

```
select
  "ID",
  "USERNAME",
  "FULL_NAME",
  "EMAIL",
  "PROFILE",
  "CREATED",
  "CREATED_BY",
  "UPDATED",
  "UPDATED_BY"
from "DEMO_PROJ_TEAM_MEMBERS"
```

3. Al ejecutar el informe, entramos en columnas para ocultar por defecto las que no queremos que se muestren (Username, Created, Created by, Updated, Updated by)
4. Guardamos el Informe para que se muestre de este modo por defecto para todos los usuarios.

Customizando Projects Interactive Grid

En esta ocasión, incluimos en el informe una agregación en la columna de tareas y eliminamos las columnas que no queremos que sean visibles, del mismo modo que cambiaremos la forma de enlazar al formulario.

1. Editamos la página 4 de Demo Projects: Demo Projects.
2. Incluimos la siguiente consulta:



```

select
  "ID",
  "NAME",
  "DESCRIPTION",
  "PROJECT_LEAD",
  "COMPLETED_DATE",
  "STATUS_CD",
  "CREATED",
  "CREATED_BY",
  "UPDATED",
  "UPDATED_BY",
  (select count('x')
   from demo_proj_milestones m
   where m.project_id = p.id
  ) milestones,
  (select count('x')
   from demo_proj_tasks t
   where t.project_id = p.id
  ) tasks
from "DEMO_PROJECTS" p

```

3. En la columna ID, eliminamos el enlace a página 5.
4. Al ejecutar el informe, eliminamos la selección de las columnas ID, Description, Created, Created By, Updated y Updated By. Del mismo modo, reordenamos la columna Project Lead tras la de Name.
5. Añadimos una Agregación: Actions > Data > Agregate.
 - a. Pulsamos Añadir.
 - b. Seleccionamos Milestones, Sum, Show Overall Value.
 - c. Pulsamos Añadir de nuevo.
 - d. Tasks, Sum for Agregation, Show Overall Value.
6. Guardamos el informe.
7. Ahora, editamos de nuevo la página para convertir la columna Name en un enlace.
8. En la columna Name, seleccionamos que sea de tipo Link y configuramos el Target:
 - a. Page: 5
 - b. Set Items: P5_ID - &ID.
 - c. Clear cache: 5
 - d. Al cerrar el dialogo, decimos que se muestre como enlace Link Text: NAME.



Customización de un Interactive Grid: Milestones

Ahora, vamos a customizar del mismo modo el Interactive Grid de página 6. Este ejemplo es prácticamente igual al anterior, de modo que sería conveniente realizarlo sin ayuda a partir de la consulta facilitada.

1. Cambiamos la consulta SQL del Grid Interactivo:

```
select
  "ID" ,
  "PROJECT_ID",
  "NAME",
  "DESCRIPTION",
  "DUE_DATE",
  "CREATED",
  "CREATED_BY",
  "UPDATED",
  "UPDATED_BY",
  (select count('x')
   from demo_proj_tasks t
   where t.milestone_id = m.id
  ) tasks
from "DEMO_PROJ_MILESTONES" m
```

2. Ocultamos las columnas Description, Created, Created By, Updated y Updated By. Guardamos el informe.
3. Ordenamos el informe por Due Date, Ascendente.
4. Añadimos una agregación:
 - a. Botón Añadir.
 - b. Escogemos Milestones, Sum, Show Overall Value. Guardamos el informe.

Customización de un Interactive Grid: Tasks

Ahora, volvemos a editar las características de un grid interactivo para que se adecue a la idea de representación que nosotros tenemos. Para ello, en el grid interactivo Tasks.

1. Mostramos / Ocultamos columnas para que las siguientes aparezcan en orden: ID, Assignee, Name, Start Date, End Date, Is Complete Yn. Guardamos.
2. Añadimos ahora Control de Ruptura: Actions > Format > Control Break.
 - a. Añadimos uno para la columna Project ID.
 - b. Añadimos otro para la columna Milestone ID.
3. Ahora, ordenamos de forma ascendente por la fecha de inicio: Actions > Data > Sort:
 - a. Añadimos uno para Start Date, Ascending.
4. Guardamos el informe y lo visualizamos.

Customización de un Grid Interactivo como Desarrollador

Para preparar como desarrollador un grid interactivo para los usuarios finales, vamos a editar la Página 3 de la app Budget App.

1. En el editor de páginas de la app, buscamos Rendering > Regions. Pulsamos con el botón derecho sobre Attributes. Escogemos la opción de Create Column Group.
 - a. Generamos un Grupo para Work Breakdown.
 - b. Generamos un Grupo para Schedule.
 - c. Generamos un Grupo para Project Financing.
2. Una vez creados los distintos grupos, debemos asignar columnas a los mismos. Para ello, desplegamos la región de columnas y seleccionamos **Project** y **Task_Name**. Las asignamos a Group: **Work Breakdown**.



3. Hacemos la misma operación para asignar **Start_Date** y **End_Date** al grupo **Schedule**.
4. Repetimos con las columnas **Cost** y **Budget**, y las asignamos al grupo **Project Financing**.
5. Guardamos y ejecutamos la página. Vemos los grupos que hemos generado.
6. Reordenamos los grupos, de modo que aparezcan Work Breakdown, Schedule y Project Financing. Para ello, solo tenemos que arrastrar el grupo restante al final.
7. Ajustamos el ancho de la columna Id.
8. Ahora, desactivamos la opción de descargar el informe como HTML, dentro de Atributes > Download > HTML.
9. Permitimos a los usuarios que guarden informes públicos: Atributes > Enable Users To > Save Public Report: Yes.
10. Ahora, vamos a habilitar la edición de este Interactive Grid. Para ello, habilitamos esta característica en Atributes > Edit > Enabled: Yes. Además, en Atributes > Pagination, establecemos Type: Page y Show Total Row Count: Yes.
11. Eliminamos ahora los grupos de columnas, seleccionándolos con el botón derecho y seleccionando la opción de borrar en el menú contextual. Guardamos y ejecutamos la página para observar el resultado.

Customizando un Grid Interactivo como Usuario final

El informe que hemos personalizado en el apartado anterior es ahora editable, por lo que se pueden observar botones adicionales que antes no estaban, como son Edit, Save y Add Row.

1. Realizamos una búsqueda en el cuadro de texto: server. Como vemos, esta búsqueda puede ser sensible a caracteres mayúsculas y minúsculas o no. Limpiamos el filtro.
2. Si pulsamos ahora en la lupa, seleccionamos la columna Task Name y buscamos nuevamente: server. Eliminamos el filtro.
3. Ahora queremos editar el valor Budget de la línea con ID= 1; lo pulsamos y sobrescribimos 300 en lugar del valor de 100. Hasta que no pulsamos en el botón de salvar, no se guardan los cambios.
4. Otra forma de editar la entrada es seleccionando Single Row View en la entrada con ID=2. Cambiamos el valor de Budget a 500.
5. Añadimos una nueva fila y rellenamos todos los campos requeridos.
6. Borramos la fila recién añadida.
7. Si queremos ver los datos que había en el informe hace unos minutos, Actions > Data > Flashback. Introducimos un valor de 5 minutos y vemos el resultado. Obsérvense los valores de la columna Budget para las filas con ID = 1 y 2.
8. Generamos ahora un Control Break para la columna Project.
9. Ahora, remarcamos líneas en base a las siguientes condiciones:
 - a. Actions > Format > Highlight
 - b. Project costing greater than 750 / Yellow & Red.
10. Guardamos ese informe como My Private Report, con tipo Private.
11. Volvemos al informe principal (Primary Report), y ocultamos las columnas de fecha, y Assigned to. Después, reajustamos el ancho de las columnas con el ratón.
12. Incluimos ahora un gráfico. Para ello, seleccionamos Actions > Chart e introducimos las siguientes características:
 - a. Tipo: **Bar**
 - b. Label: **Project**
 - c. Value: **Cost** / Agregation: **Sum**.
13. Tras guardar, vemos en pantalla el gráfico que hemos generado. Guardamos ahora este informe, con el título My Custom Report, de tipo Private.
14. Si queremos descargar el informe ahora, nótese que ha desaparecido la opción HTML.

Práctica 9. Creación y uso de Formularios

A lo largo de la práctica, veremos cómo crear y utilizar formularios. Para ello, manipularemos un formulario ya creado, y crearemos uno nuevo a partir de una tabla para enlazarlo a un informe.



Actualización / Edición de un Formulario

1. Entramos en la aplicación Demo Projects. Navegamos hasta la página que contiene el informe Demo Proj Team Members, y abrimos un elemento del informe para acceder al formulario. Ahora, editamos esta página. También podemos acceder a través del menú de App Builder, navegando hasta la página 3.
2. Cambiamos el título y nombre de la página modal: **Maintain Team Member**.
3. Dentro de Content Body, seleccionamos el elemento Edit DEMO_PROJ_TEAM_MEMBERS y pulsamos en Template Options. Modificamos el atributo Item Width: **Stretch Form Fields**. Guardamos los cambios.
4. Modificamos ahora el formulario de Demo Projects (formulario en página 5).
5. Cambiamos el título y nombre de la página a **Maintain Project**.
6. Modificamos el atributo Navigation > Cursor Focus a **First Item on page**. Guardamos los cambios.
7. Ahora modificaremos el formulario de página 7, Demo Proj Milestones.
8. Cambiamos el título a **Maintain Milestones**.
9. Fijamos el foco del cursor en el primer elemento de la página: Navigation > Cursor Focus: First Item on page.
10. Realizamos todas estas acciones con el formulario asociado a Proj Tasks, en página 9.
 - a. Título y nombre: **Maintain Tasks**.
 - b. Fijamos el foco del cursor en el primer elemento de la página: Navigation > Cursor Focus: First Item on page.

En los formularios, podemos hacer otras muchas manipulaciones de elementos, pero las veremos más adelante.

Creación de un formulario a partir de una tabla, y enlazado con un informe

Para este ejemplo, trabajaremos dentro de la app Budget App. Crearemos un Informe clásico a partir de la tabla PROJECT_BUDGET, después crearemos un formulario basado en la misma tabla y, finalmente, enlazaremos en informe y el formulario.

1. En primer lugar, dentro de la aplicación Budget App, creamos un informe con las siguientes características:
 - a. Classic Report. Título: Project Budget Report.
 - b. En navegación, creamos una nueva entrada en el menú.
 - c. Seleccionamos PROJECT_BUDGET como tabla / vista y creamos el informe.
2. Para crear ahora el formulario, realizamos pasos similares.
 - a. Dentro de Page Designer, pulsamos el botón de la barra Añadir, y escogemos Page.
 - b. Seleccionamos Formulario.
 - c. Escogemos el tipo: Basado en una Tabla.
 - d. Lo llamamos **Project Budget Details** y escogemos que sea tipo **Modal Dialog**.
 - e. Escogemos la tabla PROJECT_BUDGET y creamos el formulario.
3. La página que acabamos de crear es de tipo modal, por lo que no podemos ejecutarla directamente. Para acceder a esta página, navegamos a página 4, donde construimos el informe.
4. Seleccionamos la columna ID, y cambiamos su tipo a Enlace (Link). Como Link Target, ponemos la página del formulario, incluyendo los atributos necesarios para el salto:
 - a. Target: Page in this application
 - b. Target Page: 5
 - c. Set Items: P5_ID / Value: #ID#
5. Como Texto del enlace, seleccionamos un icono de la lista.
6. Si ejecutamos ahora el informe que hemos creado, observamos que la columna ID contiene el icono que hemos establecido, y al pulsarlo podemos acceder al formulario que acabamos de crear.



Práctica 10. Creación y uso de controles y elementos de página

En este epígrafe, se cubre la creación y actualización de elementos de página y propiedades de botones. Para ello, trabajaremos sobre cuatro apartados, customizando los distintos formularios de la aplicación Demo Projects.

Actualizando el formulario de Team Members

1. Navegamos a la página 2, Demo Proj Team Members. Por defecto, el botón se encuentra dentro de la región a la que está asociado. Lo arrastramos a la región Breadcrumbs, y lo editamos con las siguientes características:
 - a. Label: Add Team Member
 - b. Layout > Button position – Create.
 - c. Appearance – Hot: Yes.
2. Dentro del formulario, en página 3, realizamos los siguientes cambios:
 - a. Hacemos los campos Username / Full Name obligatorios.
 - b. Convertimos Profile en Text Area.
 - c. Cambiamos la etiqueta del elemento P3_PHOTO_BLOB a Photo.
 - d. Ocultamos los elementos P3_PHOTO_FILENAME, P3_PHOTO_MIMETYPE, P3_PHOTO_CHARSET, P3_PHOTO_LAST_UPDATED.
3. La tabla en la que se basa este formulario cuenta con columnas de auditoría. Vamos a ocultar esos datos en una subregión. Para ello, creamos una Sub-Región dentro de la región que hay en el Content Body:
 - a. Title: Audit Details.
 - b. Appearance > Template: Collapsible
 - c. Use Template Defaults, Expanded, Scroll-Default, Collapsed, Remove UI Decoration.
 - d. Añadimos una condición Server Side Condition: Item is NOT Null – P3_ID.
 - e. Arrastramos los elementos P3_CREATED, P3_CREATED_BY, P3_UPDATED, P3_UPDATED_BY a esta sub-región.
 - f. Seleccionamos el tipo Display Only para estos elementos.

Actualizando el formulario de Projects

1. Para editar el formulario de Projects, vamos a utilizar una Lista de Valores.
 - a. Vamos a Shared Components > List of Values – Create
 - b. From Scratch
 - c. Team Members – Dynamic.
 - d. Insertamos la siguiente consulta:

```
select full_name as display,  
id as return  
from demo_proj_team_members  
order by 1
```

2. Navegamos a Página 4, y seleccionamos la columna PROJECT_LEAD
 - a. La definimos como tipo: Select List, con alineamiento al inicio (Start) en Heading y Layout.
 - b. List of Values > Tipo: Shared Component y, como valor, la lista que acabamos de crear: Team Members.
 - c. Al igual que antes, podemos mover el botón CREATE a la región Breadcrumb, con la etiqueta Create Project, en la zona Create y de tipo Hot.
3. Navegamos a página 5 para editar ahora el formulario.



- a. P5_NAME: required, width: 60, value required: Yes.
- b. P5_DESCRIPTION: Text Area, Width: 70
- c. P5_PROJECT_LEAD: Select List (Team Members), Display extra values: NO, Display null value: - Select Team Members -.
- d. P5_STATUS: Generamos una lista estática. Select List, Static Values, introducimos los valores: STATIC2:Assigned,In-Progress,Completed, No mostramos Valores extra ni el valor nulo.
- e. P5_COMPLETED_DATE: Required.
- f. Para realizar el mismo efecto de los campos de auditoría, vamos a copiar la región de página 3. Como los Items tienen el mismo nombre, eliminamos los de página 5 para evitar duplicados, y copiamos la región de página 3 (navegamos allí, seleccionamos la región y pulsamos con el botón derecho para desplegar el menú contextual, y copiamos la región a página 5 rellenando los campos requeridos en el asistente). Una vez en página 5, colocamos la nueva región como sub-región de la región principal. Guardamos y vemos el resultado.

Actualizando el formulario de Milestones

1. Para editar los elementos que intervienen en esta sección, vamos a empezar por crear una lista de valores.
 - a. Desde Shared Components, > List of Values, creamos una lista como copia de otra existente. Le damos el nombre Projects.
 - b. La editamos, y cambiamos la consulta a la siguiente:

```
select name as display
, id as return
from demo_projects
order by 1
```

2. De vuelta en página 6, usamos la lista en la columna PROJECT_ID, alineándola al principio (start) y con la etiqueta Project.
3. Colocamos el botón CREATE del mismo modo que en los apartados anteriores: en la región Create del Breadcrumb, Create Milestone como etiqueta y de tipo Hot.
4. Editamos ahora los elementos de Página 7.
 - a. P7_PROJECT_ID: En primer lugar, utilizamos la Lista de Valores recién creada dentro del formulario, asignamos la etiqueta Project, el campo será Required, no muestra valores extra y tiene como Null Value: - Select Project -.
 - b. P7_NAME: Label: Milestone, Required, Width: 60, Value Required: Yes.
 - c. P7_DESCRIPTION: Textarea, Width: 70
 - d. Eliminamos los elementos de auditoría y copiamos la región de página 3. Tras asignar la anidación correspondiente a la región copiada, guardamos y vemos el resultado.

Actualizando el formulario de Tasks

1. Creamos una lista de valores para mejorar la experiencia de uso. Será una lista dinámica (copia o desde cero), de título Milestones que ejecuta la consulta siguiente:

```
select name as display , id as return
from demo_proj_milestones
order by 1
```

2. De vuelta en página 8, usamos algunas listas ya construidas y las alineamos al principio con la etiqueta que corresponda:
 - a. ASSIGNEE: Team Members



- b. PROJECT_ID: Projects, con título Projects
- c. MILESTONE_ID: Milestones, con título Milestone
- 3. El ítem llamado NAME tiene un nombre algo ambiguo, por lo que cambiamos tu título a Task.
- 4. Para la columna IS_COMPLETE_YN, sería más conveniente mostrar un valor más fácil de entender por el usuario final, modificamos la línea correspondiente de la consulta por el siguiente valor, y su cabecera por “Completed?”.

```
decode (IS_COMPLETE_YN, 'Y', 'Yes', 'No') as "IS_COMPLETE_YN",
```

- 5. Colocamos el botón CREATE del mismo modo que en los apartados anteriores: en la región Create del Breadcrumb, Create Task como etiqueta y de tipo Hot.

Ahora, pasamos a la edición de los ítems en página 9. En este caso, no se trata de una página modal. Reordenamos los ítems para que sigan la siguiente secuencia en que se muestran sus características:

- 1. P9_PROJECT_ID: Select List, Project, Required, Value Required, Shared Components, PROJECTS, No extra values, - Select Project – as default.
- 2. P9_MILESTONE_ID: Aquí haremos algo nuevo. Será un Select List, de nombre Milestone. Como valor, tendrá la siguiente consulta SQL:

```
select name as display, id as return
from demo_proj_milestones
where project_id = :P9_PROJECT_ID
order by 1
```

No mostramos valores extra, mostramos – Select Milestone – como valor null, y seleccionamos P9_PROJECT_ID. Como Cascading LOV Parent Item(s). De este modo, si no hay un proyecto seleccionado, no se puede seleccionar este elemento.

- 3. P9_NAME: Le ponemos una etiqueta de Task, hacemos el campo Required, con una anchura de 60 caracteres y es un valor requerido.
- 4. P9_DESCRIPTION: Utilizamos un Textarea para este elemento, con una anchura de 70.
- 5. P9_ASSIGNEE: Utilizamos la lista de valores TEAM_MEMBERS, que no muestre valores adicionales y tenga un valor por defecto de – Select Assignee –.
- 6. P9_START_DATE
- 7. P9_END_DATE
- 8. P9_IS_COMPLETE_YN: Seleccionamos el tipo Switch, y cambiamos la etiqueta a “Completed?”.
- 9. Eliminamos los ítems de auditoría y los copiamos de página 3 como hicimos anteriormente, copiándolo como sub-región de la principal.
- 10. Modificamos la apariencia de la región DEMO_TASKS:
 - a. Header: Hidden but accesible
 - b. Style: Remove UI Decoration
- 11. Añadimos un elemento Display Only, colocado a la izquierda de P9_MILESTONE_ID: P9_MILESTONE_DUE_DATE, con etiqueta Due Date, que no almacene Estado de Sesión y con Source Null. Dado que no existe ninguna columna en base de datos que tenga este valor, es importante

Práctica 11. Computaciones, Procesos y Validaciones.

Implementación de una Validación

- 1. En la app Demo Projects, navegamos a página 5(Maintain Project Page).
- 2. Ahora, vamos a añadir una validación en la región Validating, dentro de la pestaña de procesos del panel derecho de Page Designer. Para ello, pulsamos sobre Validations con el botón derecho y añadimos una validación con las siguientes características:
 - a. Title: Completed Date is not Null



- b. Type: Item is NOT NULL
 - c. Item: P5_COMPLETED_DATE
 - d. Error Message: #LABEL# must have some value
 - e. Condition: Item = Value / P5_STATUS_CD / Completed
- 3. Añadimos una segunda validación con los siguientes valores:
 - a. Title: Completed Date is not Forward Dated
 - b. Type: PL/SQL Expresion: :P5_COMPLETED_DATE <= sysdate
 - c. Error Message: #LABEL# cannot be forward dated
 - d. Condition: Item is NOT NULL / P5_COMPLETED_DATE
- 4. Guardamos y comprobamos el funcionamiento correcto de nuestras validaciones.

Creación y Uso de una Computación

En este ejercicio, lo que haremos será crear una computación encargada de convertir en mayúsculas el campo Assigned To cuando se envía un formulario.

1. En la Aplicación Budget App, vamos a la página 2.
2. En el gestor de procesos de Page Designer, pulsamos con el botón derecho sobre After Submit y escogemos Create Computation, y la editamos con los siguientes atributos:
 - a. Item Name: **P2_ASSIGNED_TO**
 - b. Execution Point: **After Submit**
 - c. Computation Type: PL/SQL Expresion, con el código **upper(:P2_ASSIGNED_TO)**
3. Guardamos y comprobamos el resultado.

Creación y uso de un Proceso

Para entender los procesos, generaremos uno en la aplicación Project Budget que inserte entradas en la tabla de auditoría que vamos a llamar AUDIT_DETAILS cada vez que se añada un proyecto.

1. En la aplicación Budget App, vamos a la página 2.
2. En el gestor de proceso de Page Designer, seleccionamos el proceso Process Row of PROJECT_BUDGET.
3. En Settings, asignamos al atributo Return Key into Item el valor **P2_ID**. Cambiamos el mensaje de éxito por **Project added successfully!**
4. Nos dirigimos ahora a SQL Workshop y ejecutamos el siguiente código para crear la tabla AUDIT_DETAILS:

```
CREATE table "AUDIT_DETAILS" (
  "ID" NUMBER,
  "CREATED_BY" VARCHAR2(60),
  "CREATED_ON" DATE,
  constraint "AUDIT_DETAILS_PK" primary key ("ID") )
```

5. Volvemos a la página 2, a la región de Procesos. Pulsamos con el botón derecho para desplegar el menú contextual y seleccionamos Create Process, con los siguientes detalles:
 - a. Identification > Name: Audit Details
 - b. Type: PL/SQL Code
 - c. Source: **INSERT INTO Audit_details VALUES (:P2_ID, :APP_USER, sysdate);**
 - d. Sequence: 40
 - e. Point: Processing.
 - f. Success Message: Success!!
 - g. Error Message: Error!
 - h. Server-side Condition > When Button Pressed: CREATE
6. Guardamos y comprobamos que el proceso realiza la acción que queríamos generando un nuevo proyecto, con los siguientes detalles (por ejemplo):
 - a. Project: New Packaged App



- b. Task name Prepare requirements doc
 - c. Start Date: 15-Feb-17
 - d. End Date: 20-Feb-17
 - e. Status: Open
 - f. Assigned To: Pam King
 - g. Cost: 500
 - h. Budget: 800
7. Tras crear la entrada, visitamos la tabla para observar que, efectivamente, cuenta con los datos de auditoría que indicamos en el PL/SQL.

Creación y uso de Validaciones

Ahora vamos a incluir dos validaciones en el formulario con el que estamos trabajando, dentro de la aplicación Project Budget.

1. En la región de procesado, añadimos una nueva validación con los siguientes detalles:
 - a. Name: Project is not null.
 - b. Type: Item is NOT NULL.
 - c. Item: P2_PROJECT
 - d. Error Message: Project should contain a value!!
 - e. Display Locations: Inline in Notification.
2. Creamos otra validación con los siguientes detalles:
 - a. Name: Cost should be numeric
 - b. Type: Item is numeric
 - c. Item: P2_COST
 - d. Error Message: Cost should be numeric
 - e. Display Location: Inline with Field.
 - f. Associated Item: P2_COST
3. Guardamos los cambios y navegamos hasta el formulario para comprobar el efecto de nuestras validaciones.

Práctica 12. Gestionar la navegación de la App

Los elementos de navegación de que disponemos en la aplicación de ejemplo que estamos construyendo son Breadcrumb y Navigation List. Vamos a editar las entradas del Breadcrumb para nuestras páginas, así como las entradas del Menu de Navegación.

1. Navegamos a Página 2, y editamos el título y nombre de la página: Team Members.
2. En la pestaña de Shared Components, desplegamos la región Breadcrumb y editamos el componente, asignando el mismo nombre que a la página: Team Members.
3. Repetimos el proceso con las demás páginas: Projects, Milestones y Tasks.

Ahora vamos a editar la lista del menú de Navegación, añadiendo algunos iconos.

1. Navegamos hasta Shared Components > Navigation Menu
2. Entramos en la entrada Home, y le asignamos el icono fa-home en el atributo Image/Class.
3. Arreglamos el resto de entradas asignando los siguientes valores:
 - a. Team members – fa-users
 - b. Projects – fa-folder
 - c. Milestones – fa-flag
 - d. Tasks – fa-check-square-o

Finalmente, podemos ejecutar la app y observar el resultado de nuestros cambios.



Práctica 13. Temas y Estilos

A lo largo de esta práctica, podremos ver cómo editar algunos componentes del estilo de la aplicación.

Actualización de la Lista de Navegación

1. Dentro de la aplicación Demo App, navegamos hasta Shared Components > Lists.
2. Creamos una nueva entrada con los siguientes atributos:
 - a. Image / Class: fa-user
 - b. List Entry Label: &APP_USER.
 - c. Target type: URL
 - d. URL Target: #
 - e. User defined Attributes 2: has-username. Nota: para asegurar que Atributos están definidos, entramos en List Template Attributes y marcamos Universal Theme 42 / Navigation Bar.
 - f. Ahora, entramos en la entrada Log Out, y seleccionamos como atributo Parent List Entry: &APP_USER.
 - g. Guardamos y vemos los cambios al ejecutar la app.

Actualizando la apariencia de un Informe

1. Accedemos a Página 2 de Demo Projects, Team Members.
2. Incluimos el siguiente código CSS en la descripción de la página en CSS - Inline:

```
.t-Card-icon img {  
width: 48px;  
height: 48px;  
}
```

3. Ahora, vamos a cambiar el Interactive Grid por un Classic Report, de modo que podamos aprovecharnos de las plantillas predefinidas. Para ello, modificamos el Tipo del informe por Classic Report, y cambiamos su título a Team Members.
4. Cambiamos la consulta por el siguiente código SQL. Cambiamos la consulta para que las columnas empleen los nombres correspondientes con las columnas requeridas por la plantilla de tarjetas que vamos a utilizar.

```
select id  
, full_name card_title  
, profile card_text  
, nvl(email, ' ') card_subtext  
, dbms_lob.getlength('PHOTO_BLOB') card_initials  
, apex_util.prepare_url('f?p=||:APP_ID||:3:||:APP_SESSION||::::P3_ID:'||id)  
card_link  
, photo_filename  
, photo_mimetype  
, photo_charset  
, photo_last_updated  
from demo_proj_team_members  
order by created
```

5. La tabla sobre la que hacemos la consulta cuenta con un campo de imagen, por lo que podemos hacer que se muestre. Para ello, establecemos los siguientes atributos para la columna CARD_INITIALS:
 - a. Type: Display Image.



- b. BLOB Attributes > Table Name: DEMO_TEAM_MEMBERS
 - c. BLOB Attributes > BLOB Column: PHOTO_BLOB
 - d. BLOB Attributes > Primary Key Column 1: ID
 - e. BLOB Attributes > Mime Type Column: PHOTO_MIMETYPE
 - f. BLOB Attributes > Filename Column: PHOTO_FILENAME
 - g. BLOB Attributes > Last Updated Column: PHOTO_LAST_UPDATED
6. Entramos ahora en Attributes, y seleccionamos Template: Cards. Entramos en las opciones y seleccionamos:
- a. Style: Basic
 - b. Icons: Display Initials
 - c. Layout: 4 Columns
 - d. Body Text: 3 Lines
7. Guardamos y vemos el resultado al ejecutar la aplicación.

Creación y uso de los Estilos del Tema

En este apartado vamos a utilizar la herramienta Theme Roller para cambiar la apariencia general de la aplicación.

1. Ejecutamos la aplicación y, en la barra de Desarrollador, abrimos Theme Roller. Podemos manipular los colores de las distintas regiones y Guardar el tema con el botón Save As. Guardamos con el nombre que nos apetezca.
2. Navegamos ahora, en el escritorio de APEX, a Shared Components > User Interface Attributes y dentro, editamos las características de Desktop, para permitir a los usuarios finales elegir su propio estilo.
3. Ahora, al ejecutar la aplicación, podemos observar un enlace en el footer de la aplicación con la etiqueta Customize. Al pulsar en él, una ventana se abre permitiendo al usuario final elegir cualquiera de los temas de la lista, incluyendo el que creamos en el paso 1.

Práctica 14. Implementación de la seguridad en la App

La seguridad es una parte fundamental en la mayoría de aplicaciones, salvo para las que tengan un carácter público. Por tanto, es importante asegurarse de que los usuarios acceden con unas credenciales válidas, habitualmente usuario y contraseña (autenticación) y una vez dentro de la app, verificar si el usuario tiene privilegios para acceder a determinadas partes de la aplicación (autorización)

Creación y uso de un esquema de Autorización

Para este epígrafe, crearemos un esquema de autorización que solo quienes pertenecen a Team Members puedan acceder a la app Demo Projects.

1. Navegamos a Demo Projects, y accedemos a Shared Components > Authorization Schemes
2. Escogemos crear un Authorization Scheme desde cero (from Scratch) con las siguientes características:
 - a. Nombre: Team Members
 - b. Schema Type: Exists SQL Query
 - c. Para la consulta, inserta el siguiente código:

```
select 1
from demo_proj_team_members
where upper(username) = upper(:APP_USER)
```

- d. Error message: You must be a team member to use this application.
3. Una vez hemos creado el esquema de autenticación, volvemos a Shared Components para navegar hasta Security Attributes.



4. En la pestaña de Autorización, seleccionamos el esquema Team Members que acabamos de crear y guardamos los cambios.
5. Por último, para verificar el correcto funcionamiento de nuestro esquema de autorización, vamos a Administration > Manage Users and Groups y creamos un nuevo usuario.
6. Cuando intentemos acceder a Demo Projects con este nuevo usuario, observaremos un mensaje de error que deniega el acceso por no cumplir los requisitos del esquema.

Creación y uso de un esquema de Autenticación

En este punto, crearemos un esquema de autenticación para la aplicación Budget App llamado Open Door Credentials.

1. En Budget App, navegamos a Shared Components > Authentication Schemes.
2. Creamos un nuevo esquema basado en uno ya pre-configurado, Based on a pre-configured scheme from the gallery, con los siguientes parámetros:
 - a. Name: Open Door Credentials
 - b. Scheme Type: Open Door Credentials.
 - c. Al crear el tema, se establece como tema por defecto.
3. Si ejecutamos ahora la app, vemos que ha cambiado el modo de login, de forma que solo habría que introducir nuestro nombre de usuario para acceder.
4. Para revertir el cambio y volver al esquema de autenticación anterior, navegamos de nuevo a Shared Components > Authentication Schemes y seleccionamos la fila Application Express Authentication para acceder al detalle. Una vez dentro, pulsamos el botón 'Make Current Scheme' y al volver a la página de login veremos que hemos vuelto al esquema de autenticación anterior.

Controlando el acceso de usuarios utilizando Administración de Control de Acceso

Para esta práctica, vamos a crear una página de control de acceso y habilitaremos a la app para restringir dicho acceso.

1. Para poder verificar esta funcionalidad, vamos a necesitar distintos tipos de usuarios. Para eso, necesitamos 3 usuarios de APEX, uno de cada tipo: Workspace Administrator, Developer y End User.
2. Una vez creados, Creamos una página del tipo Access Control dentro de Budget App, creando una nueva entrada en el menú de navegación con el nombre **Access Control**.
3. Al ejecutar la página, vemos una región donde se puede establecer el modo del control de acceso, y una lista para ese fin. En la lista, incluimos los usuarios que hemos creado antes, otorgando privilegios de Administrator (WS Admin), Edit (Dev) o View (End User). Aplicamos los cambios y marcamos opción 'Restricted Access' en el modo de aplicación.

Ya hemos creado un control de acceso, y ahora vamos a aprender como de sencillo puede resultar hacer uso del mismo.

1. Vamos a Página 1, y en la región Project Budget, seleccionamos Atributos. Buscamos Authorization Scheme en el editor de propiedades y seleccionamos **access control-edit**.
2. Como solo queremos que el botón CREATE aparezca solo para quien tenga privilegios de edición o de administración, en el editor de propiedades buscamos Security y seleccionamos **access control-edit** como Authorization Schema.
3. Hasta ahora, hemos protegido que los usuarios que no tengan privilegios puedan editar un interactive report o usando el botón, pero estableciendo la url correcta en la barra del navegador podrían acceder a esta página, por lo que restringimos su acceso del mismo modo. Para ello, navegamos a página 2 y restringimos en acceso en la sección de seguridad, asignando el esquema de autorización **access control-edit**.
4. Ejecutamos la aplicación y comprobamos que nuestras restricciones son correctas si entramos con el usuario que solo cuenta con permisos de visualización.



Práctica 15. Añadiendo otras páginas a la App

En APEX existen otro tipo de páginas que podemos crear, además de las que hemos observado anteriormente. Mejoraremos

Creación y personalización de un Calendario

1. En App Builder, accedemos a Demo Projects, y creamos una nueva página de tipo Calendar, utilizando Breadcrumb y asignando una nueva entrada de navegación.
2. Elegimos la tabla DEMO_PROJ_TASKS, seleccionamos las columnas NAME, START_DATE y END_DATE y creamos este tipo de página.
3. Editamos ahora la apariencia del calendario, cambiando el template con Header: **Hidden but accesible** y con Style: **Remove UI Interface**. Podemos, igualmente, modificar la Lista de Navegación para actualizar el icono que se muestra en la barra de navegación, usando **fa-calendar**.
4. Volvemos a la página 10 para editar el calendario, navegamos hasta Atributos de la región Calendar.
5. Buscamos la sección Create Link, para definir la página 9 como creación de elementos, limpiando caché en la misma página.
6. Del mismo modo, usamos la página 9 para el enlace View / Edit Link, pasando el ítem **P9_ID** con valor **&ID.**, limpiando nuevamente cache.
7. Para habilitar Arrastrar y Soltar en el calendario, debemos hacer algunos cambios:
 - a. Cambiamos la Primary Key Column por ROWID
 - b. Habilitamos Drag and Drop
 - c. Incluimos el siguiente código en el lugar requerido.

```
begin
update DEMO_PROJ_TASKS
set start_date = to_date(:APEX$NEW_START_DATE,
'YYYYMMDDHH24MISS'),
end_date = to_date(:APEX$NEW_END_DATE, 'YYYYMMDDHH24MISS')
where ROWID = :APEX$PK_VALUE;
end;
```

8. En este punto, si ejecutamos el calendario podemos probar esta funcionalidad.

Añadiendo un Gráfico de Tareas de Proyecto

1. En primer lugar, creamos una página en blanco con el nombre Reports, que utilice Breadcrumb y Entrada en la lista de navegación (podemos editar la lista para asignarle el icono **fa-bar-chart**).
2. Creamos una nueva página de tipo Chart, seleccionando un gráfico de tipo Bar, con el nombre Tasks Review, que use Breadcrumb y con la página Reports como Parent Entry.
3. Creamos una nueva entrada en el menú de navegación, con Reports como Parent Navigation Menu Entry.
4. El gráfico utilizará la siguiente consulta como fuente para el tipo SQL Query:



```

select p.id
, p.name as label
, (select count('x') from demo_proj_tasks t
where p.id = t.project_id
and nvl(t.is_complete_yn,'N') = 'Y'
) value
, 'Completed Tasks' series
, p.created
from demo_projects p
UNION ALL
select p.id
, p.name as label
, (select count('x') from demo_proj_tasks t
where p.id = t.project_id
and nvl(t.is_complete_yn,'N') = 'N'
) value
, 'Incompleted Tasks' series
, p.created
from demo_projects p
order by 5

```

5. Mapeamos las columnas seleccionando los siguientes atributos:
 - a. Orientation: Horizontal
 - b. Label Column: LABEL
 - c. Value Column: VALUE
6. Una vez creado el informe, editamos sus atributos para que tenga la apariencia que buscamos. Cambiamos en Attributes que el gráfico pueda ser apilado (Stack: Yes).
7. Después, pinchamos sobre Series 1 y seleccionamos **SERIES** como Series Name.
8. Seleccionamos el eje X y cambiamos el título por Project.
9. Seleccionamos el eje Y y cambiamos el título por Tasks.

Añadiendo un Gráfico de Hitos de Proyecto

1. Ahora vamos a crear este gráfico creando una página como copia de la que acabamos de crear. Para ello, pulsamos el botón correspondiente.
2. En el asistente, seleccionamos la página de origen, damos el nombre a esta página, llamándola **Milestones Review**, y seleccionamos todos los parámetros correspondientes para lograr una navegación similar a la que creamos antes, asignando el nombre Project Milestones a la región que copiamos.
3. Cuando se haya creado, navegamos hasta la consulta de Series 1 y la cambiamos por el siguiente código .SQL:



```

select p.id
, p.name as label
, (select count('x') from
demo_proj_milestones m
where p.id = m.project_id
and m.due_date < sysdate
) value
, 'Past Milestones' series
, p.created
from demo_projects p
UNION ALL
select p.id
, p.name as label
, (select count('x') from
demo_proj_milestones m
where p.id = m.project_id
and m.due_date >= sysdate
) value
, 'Future Milestones' series
, p.created
from demo_projects p
order by 5

```

4. Cambiamos la apariencia del gráfico de Horizontal a Vertical.
5. Podemos alterar el nombre de los ejes X e Y a Milestone y Projects respectivamente.

Añadiendo un Gráfico de Lideres de Proyecto

1. Volvemos a crear una página de tipo gráfico creando una página como copia de la que acabamos de crear. Para ello, pulsamos el botón correspondiente.
2. En el asistente, seleccionamos la página de origen, damos el nombre a esta página, llamándola **Team Members Review**, y seleccionamos todos los parámetros correspondientes para lograr una navegación similar a la que creamos antes, asignando el nombre Project Leads a la región que copiamos.
3. Cuando se haya creado, navegamos hasta la consulta de Series 1 y la cambiamos por el siguiente código .SQL:



```

select p.id
, (select nvl(tm.full_name, tm.username)
  from demo_proj_team_members tm
 where tm.id = p.project_lead
)
|| ' [||| p.name |||]' as label
, (select count('x') from demo_proj_tasks
  t
 where p.id = t.project_id
 and nvl(t.is_complete_yn, 'N') = 'Y'
) value
, 'Completed Tasks' series
, p.created
from demo_projects p
UNION ALL
select p.id
, (select nvl(tm.full_name, tm.username)
  from demo_proj_team_members tm
 where tm.id = p.project_lead
)
|| ' [||| p.name |||]' as label
, (select count('x') from demo_proj_tasks
  t
 where p.id = t.project_id
 and nvl(t.is_complete_yn, 'N') = 'N'
) value
, 'Incompleted Tasks' series
, p.created
from demo_projects p
order by 5

```

4. Podemos alterar el nombre de los ejes X e Y a ‘ningún título’ y Tasks respectivamente.

Añadiendo un Árbol de Proyectos

1. Para añadir un árbol de proyectos, vamos a añadir en primer lugar una página en blanco con nombre Project Tree, que tenga Breadcrumb y una entrada de navegación anidadas con Reports.
2. Dentro de la nueva página, añadimos una nueva región con los siguientes parámetros:
 - a. Title: Project Tree
 - b. Type: Tree
 - c. Copiamos el código SQL incluido en el fichero Project_Tree.sql en la consulta.
3. Navegamos ahora hasta Appearance > Template Options Button y asignamos los siguientes parámetros:
 - a. General: Remove Body Padding
 - b. Header: Hidden but accesible
 - c. Style: Remove UI Decoration.
4. Por último, en property editor navegamos hasta Attributes > Settings y asignamos Database Column en el campo Tooltip. Guardamos y vemos el resultado.
5. Si queremos, podemos manipular la navegación de la lista Menu de Navegación.



Práctica 16. Creación y uso de Acciones Dinámicas y Plug-ins

Las acciones dinámicas son una herramienta realmente útil cuando desarrollamos aplicaciones con APEX. Del mismo modo, los plug-ins nos permiten extender funcionalidades o añadir otras nuevas con componentes ya creados.

Acciones dinámicas en Demo Projects

1. En Demo Projects, creamos una Lista de valores dinámica, de nombre STATUS con la siguiente consulta:

```
select description d, cd r
from demo_proj_status
order by display_order
```

2. Si navegamos hasta un proyecto con estado In Progress o Assigned y editamos sus detalles, vemos que la fecha de completado aparece independientemente del estado. Vamos a crear una acción dinámica que solo muestre el elemento si el estado es **Completed**.
3. Navegamos hasta la lista de valores P5_STATUS_CD, le asignamos en primer lugar la lista que acabamos de crear como fuente.
4. Pulsamos con el botón derecho sobre P5_STATUS_CD y seleccionamos la opción de crear una Acción Dinámica, con las siguientes características:
 - a. Name: Show Completed Date
 - b. Client-side Condition: Item = Value / P5_STATUS_CD / COMPLETED
5. Dentro de las opciones de la acción dinámica, creamos una acción de True que sea Show Item(s) P5_COMPLETED_DATE y después generamos la opción opuesta.
6. Guardamos y navegamos hasta el diálogo para ver el funcionamiento.

Editamos ahora la página 9 del ejemplo, Tasks.

1. Cambiamos la etiqueta de Breadcrumb a **Maintain Task**.
2. Ahora, añadimos una acción dinámica para establecer el valor de Due Date cuando se cambie un Milestone.
3. Añadimos una acción dinámica sobre el elemento P9_MILESTONE_ID:
 - a. Name: Get Due Date.
 - b. True action: Set Value, incluyendo el siguiente código SQL:

```
select due_date
from demo_proj_milestones
where id = :P9_MILESTONE_ID
```

- c. Page Items to submit: P9_MILESTONE_ID
 - d. Affected Elements: Item(s) P9_MILESTONE_DUE_DATE
4. Guardamos y navegamos a una tarea con Milestones para ver el efecto.

Vamos a editar ahora la página de Project Tree, pagina 16 de la app de ejemplo. Vamos a añadir dos botones, para extender y colapsar el árbol.

1. En la región Breadcrumb, creamos un botón:
 - a. Name: CONTRACT_ALL
 - b. Label: Collapse All
 - c. Position: Edit



- d. Behavior > Action: Defined by Dynamic Action
 - e. En template, Use Template Defaults.
2. Duplicamos el element, con el nombre EXPAND_ALL y la etiqueta Expand All, desmarcando la opción de Use Template Defaults en Template.
3. Creamos las acciones dinámicas de cada uno de los botones:
 - a. CONTRACT_ALL: Collapse Tree / Region / Project Tree
 - b. EXPAND_ALL: Expand Tree / Region / Project Tree
4. Guardamos y vemos el resultado.

Creando y usando un plug-in

Vamos a utilizar un plug-in incluido en una de las apps de muestra, Sample Charts. Para ello, podemos importarlo directamente si lo tenemos como fichero .SQL, o una app en el WS ya cuenta con el plug-in que necesitamos instalado, solo tendríamos que realizar la copia del mismo en nuestra aplicación. En nuestro caso, el plug-in está incluido en la carpeta de ficheros, de modo que realizaremos una importación del mismo.

1. Usamos el botón de importar, seleccionamos el fichero y lo instalamos.
2. Editamos página 1, e incluimos la región con el Title: Summary y el siguiente código:
3. Editamos las opciones de Template y eliminamos Body Padding, dejando el Header Hidden but accesible.
4. En Attributes, elegimos:
 - a. Top Label: LABEL
 - b. Value: VALUE
 - c. Target: URL - &LINK.

Práctica 17. Exportando la App

Importando una aplicación

En esta primera parte, vamos a importar una aplicación ya disponible como fichero .SQL.

1. Dentro de App Builder, seleccionamos la opción Import.
2. En el asistente, escogemos la primera opción (Database Application, Page or Component Export) y buscamos el fichero correspondiente.
3. Completamos el asistente y ya podemos ejecutar la aplicación.

Nota: si utilizamos el mismo scheme, puede fallar la instalación de los Objetos de Base de Datos y el correspondiente relleno de Datos que se incluye con fichero .SQL.

Migrando una aplicación entre entornos

En esta segunda parte, vamos a aprender como exportar una aplicación para poder instalarla en cualquier otro lugar como vimos en la primera parte, así como la forma en que podemos exportar los objetos de base de datos y los datos almacenados (si fuese necesario).

1. Navegamos dentro de la app que vamos a exportar.
2. Seleccionamos la opción de Exportación, y completamos el asistente. Es importante tener en cuenta la exportación de informes privados (si existen) al seleccionar las preferencias de exportación.
3. Tras completar el asistente, se descarga al directorio local un fichero con extensión .SQL (y es conveniente renombrarlo en lugar de dejar el ID de aplicación que tiene como nombre por defecto).

Para exportar ahora los objetos de base de datos y los datos asociados, podemos hacerlo igualmente usando herramientas de APEX.

1. Navegamos a SQL Developer > Utilities. Entramos en la opción Generate DDL.
2. En el asistente que se abre, podemos seleccionar toda la información relevante que necesitemos en el script.
 - a. Generamos un fichero SQL



- b. Seleccionamos el schema. Después escogemos Generar un Fichero para guardarlo como Script.
 - c. Seleccionamos Tables > En el siguiente paso, escogemos las tablas que queremos meter en el script. En este ejemplo, seleccionamos las siguientes tablas:
 - i. DEMO_PROJECTS
 - ii. DEMO_PROJ_COMMENTS
 - iii. DEMO_PROJ_CONSTRAINTS
 - iv. DEMO_PROJ_MILESTONES
 - v. DEMO_PROJ_STATUS
 - vi. DEMO_PROJ_TASKS
 - vii. DEMO_PROJ_TASK_LINKS
 - viii. DEMO_PROJ_TASK_TODOS
 - ix. DEMO_PROJ_TEAM_MEMBERS
 - d. Asignamos un nombre al script (Create Demo Project Tables, por ejemplo) y creamos el script.
3. Repetimos el proceso, creando un script para los Triggers, llamando al fichero Create Demo Project Triggers:
- a. biu_demo_proj_status
 - b. biu_demo_proj_team_members
 - c. biu_demo_projects
 - d. biu_demo_proj_milestones
 - e. biu_demo_proj_tasks
 - f. biu_demo_proj_task_todos
 - g. biu_demo_proj_task_links
 - h. biu_demo_proj_comments

En este punto, hemos generado dos scripts, conteniendo las tablas necesarias para correr la aplicación y los triggers asociados a las mismas. Nos faltarían los datos contenidos en la aplicación, si estos también deben ser exportados.

1. Navegamos a SQL Workshop > Utilities > Data Workshop
2. Escogemos la opción Data Unload > to XML
3. Dentro del correspondiente asistente, seleccionamos la tabla y las columnas que queremos exportar, y tras asignar un nombre con significado coherente, tenemos nuestro fichero con los datos que queríamos.

A la hora de importar tanto la aplicación como los objetos de base de datos y datos en sí, podemos proceder como ya hemos visto en otras prácticas con anterioridad.

Créditos

Este documento ha sido generado por Juan López García para Novoshore Europe SL.

Bibliografía:

- Oracle University, APEX Curriculum con licencia Creative Commons Attribution 4.0 International Public License y Universal Permissive License (UPL) Version 1.0
- Oracle APEX Cookbook – Marcel van der Plas, Michel van Zoest
- Expert Oracle Application Express – John Scott, Dietmar Aust – Martin D’Souza & otros
- Apex.world
- Otros

