

## Enunciado

### RETO 2

En el reto 2, el equipo desplegará nuevos servicios y además creará un front-end el cual se ejecutará desde el navegador web para probar el acceso y la funcionalidad. Este front-end será la base de la funcionalidad de futuros retos y de la aplicación que da solución al proyecto planteado.

Se deben implementar los accesos a las tablas Palco, Categoría y a las nuevas tablas Cliente y Mensaje. En estas tablas almacenaremos el cliente cuando se registra y los mensajes que los clientes dejen respecto a los Palcos.

Al finalizar el reto, podremos registrar clientes, visualizar el perfil del cliente, visualizar la lista de los palcos, visualizar el detalle del palco, escribir mensajes y ver los mensajes escritos.

### Creación de servicios:

- 1) Se debe crear un servicio que mediante una petición GET entregue la información de un palco en particular, para esto, se agregará a la URL una barra "/" y seguido a esto, el identificador del palco.
- 2) Se deben crear servicios GET, POST, PUT, DELETE al igual que en el reto 1, pero esta vez interactuar con las tablas Cliente y Mensaje. También se debe crear.
- 3) Se debe crear un servicio que mediante una petición GET entregue la información de un cliente en particular, de la misma manera que se hizo en el punto 1.

La tabla Cliente se debería ver así:

quema

ADMIN

Nombre

CLIENT

Columnas

Clave primaria

Claves Únicas

Índices

Claves Ajenas

Restricciones de Tabla

Comentarios

General

Restricciones

Columna de Identidad

	Nombre	Tipo de dato	Clave primaria
1	ID	NUMBER	<input checked="" type="checkbox"/>
2	NAME	VARCHAR (4000)	<input type="checkbox"/>
3	EMAIL	VARCHAR2 (20)	<input type="checkbox"/>
4	AGE	NUMBER	<input type="checkbox"/>

La tabla Mensaje se debería ver así:

Esquema

ADMIN

Nombre

MESSAGE

Columnas

Clave primaria

Claves Únicas

Índices

Claves Ajenas

Restricciones de Tabla

Comentarios

General

Restricciones

Columna de Identidad

	Nombre	Tipo de dato	Clave primaria
1	ID	NUMBER	<input checked="" type="checkbox"/>
2	MESSAGETEXT	VARCHAR2 (4000)	<input type="checkbox"/>

## Creación de Frontend.

Haciendo uso de lo aprendido sobre JavaScript y HTML, se deben crear los formularios para:

- Creación de Palcos.
- Registro de Clientes.
- Registro de mensajes.

Así mismo, se debe poder visualizar una lista de los elementos registrados en las tablas de:

- Palcos.
- Clientes.
- Mensajes.

Una vez hecha la lista, cada elemento debe tener en el título un enlace para poder visualizar en una sección "detalle" los detalles de cada:

- Palco.
- Cliente.
- Mensaje.

Al poder ver el detalle de cada uno, se podrá editar la información que contiene (excepto el id, este no se deberá poder editar).

Es necesario también crear un botón para eliminar cada palco y mensaje.

Dificultad

Instrucciones

Instrucciones para la calificación automática

Para que nuestra prueba sea exitosa, nos aseguraremos de nombrar el módulo y la plantilla de la misma manera, haciendo que la url del módulo sea client/ las plantillas serán client/ y client/:id. Esta segunda plantilla será utilizada para acceder a la información detallada de cada client.

Para el caso de Mensajes, tendremos la misma estructura donde el módulo deberá llamarse a través de la url message/ y las plantillas serán message/ y message/:id.

Las url son sensibles a mayúsculas, esto significa que la palabra CLIENT es diferente a la palabra client.

Es importante crear un archivo con las consultas sql que utilicemos en cada servicio, este servicio lo subiremos como evidencia en el momento de hacer la prueba automática, de otro modo NO SE REALIZARÁN las pruebas y no se podrá evaluar.

En la URL que la plataforma oraclecloudapps.com nos entrega, identificaremos como BASE\_URL justo aquella parte que va hasta .com. Así en el ejemplo anterior, el BASE\_URL correspondería a https://xxxxxxx.oraclecloudapps.com. Para acceder a los servicios de Mensaje (por ejemplo), Mastertech automáticamente completará la url con /ords/admin/message/message. Mientras el módulo y la plantilla tengan por url message, la URL estará correcta.

Instrucciones para realizar Prueba AUTOMÁTICA

En este caso, no se ejecutará el código directamente sino que se evaluarán los resultados obtenidos al realizar peticiones HTTP al API construida y que actualmente debe estar desplegada en Oracle Cloud. Las peticiones que se realizarán serán de tipo GET, POST, PUT y DELETE Para realizar las pruebas, las tablas de la base datos deben estar vacías y el valor

Entrada	Cada grupo debe proveer URL proporcionada por APEX
Salida	Corresponde a los mensajes JSON que retorna cada petición. MasterTech hará la comparación de forma automática.

A continuación se presenta un ejemplo del proceso de evaluación. La prueba estará formada por 3 pasos y tendrá los siguientes parámetros.

- Tipo:** describe el tipo de petición HTTP
- Endpoint:** Base URL/enlace de acceso al servicio a evaluar
- Body:** JSON de entrada para la petición HTTP
- Salida:** JSON de salida de la petición HTTP

Recomendaciones generales

- Antes de enviar la solución del reto, por favor tenga en cuenta los siguientes aspectos:
- Verifique que la aplicación esté desplegada correctamente y sea accesible a través de la URL
  - Verifique que todos los endpoint funcionen. Recuerde que puede probarlos a través de PostMan
  - Las tablas de la base de datos deben estar vacías y recién creadas para que el identificador generado automáticamente en cada tabla esté en 0.
  - Lo que debe proveer a MasterTech para las pruebas es la URL de su API y subir un archivo con las sentencias SQL de cada servicio.
  - Se debe utilizar una URL distinta por estudiante. Se validará que las urls utilizadas no sean utilizadas por estudiantes diferentes para evitar acciones fraudulentas.

Casos de prueba		
Paso 1	Tipo	GET
	Endpoint	BASE_URL/ords/admin/client/client
	Body	
	Salida	{ "items": [], "hasMore": false, "limit": 25, "offset": 0, "count": 0, "links": [{"rel": "self", "href": "BASE_URL/ords/admin/client/client"}, {"rel": "describedby", "href": "BASE_URL/ords/admin/metadata-catalog/client/item"}, {"rel": "first", "href": "BASE_URL/ords/admin/client/client"}]}
Paso 2.a	Tipo	POST
	Endpoint	BASE_URL/ords/admin/client/client
	Body	{ "id": 1, "name": "Juan", "email": "juan@correo.co", "age": 33 }
	Salida	
Paso 2.b	Tipo	GET

	<b>Endpoint</b>	BASE_URL/ords/admin/client/client	
	<b>Body</b>		
	<b>Salida</b>	{ "items": [{"id":1, "name": "Juan", "email": "juan@correo.co", "age": 33}], "hasMore": false, "limit": 25, "offset": 0, "count": 1, "links": [{"rel": "self", "href": "BASE_URL/ords/admin/client/client"}, {"rel": "edit", "href": "BASE_URL/ords/admin/client/client"}, {"rel": "describedby", "href": "BASE_URL/ords/admin/metadata-catalog/client/item"}, {"rel": "first", "href": "BASE_URL/ords/admin/client/client"}]}	
<b>Paso 3.a</b>	<b>Tipo</b>	PUT	
	<b>Endpoint</b>	BASE_URL/ords/admin/client/client	
	<b>Body</b>	{ "id":1, "name": "Alberto", "email": "alberto@correo.co", "age": 62 }	
	<b>Salida</b>		
<b>Paso 3.b</b>	<b>Tipo</b>	GET	
	<b>Endpoint</b>	BASE_URL/ords/admin/client/client	
	<b>Body</b>		
	<b>Salida</b>	{ "items": [{"id":1, "name": "Alberto", "email": "alberto@correo.co", "age": 62}], "hasMore": false, "limit": 25, "offset": 0, "count": 1, "links": [{"rel": "self", "href": "BASE_URL/ords/admin/client/client"}, {"rel": "edit", "href": "BASE_URL/ords/admin/client/client"}, {"rel": "describedby", "href": "BASE_URL/ords/admin/metadata-catalog/client/item"}, {"rel": "first", "href": "BASE_URL/ords/admin/client/client"}]}	
<b>Paso 4.a</b>	<b>Tipo</b>	DELETE	
	<b>Endpoint</b>	BASE_URL/ords/admin/client/client	
	<b>Body</b>	{ "id":1 }	
	<b>Salida</b>		
<b>Paso 4.b</b>	<b>Tipo</b>	GET	
	<b>Endpoint</b>	BASE_URL/ords/admin/client/client	
	<b>Body</b>		
	<b>Salida</b>	{ "items": [], "hasMore": false, "limit": 25, "offset": 0, "count": 0, "links": [{"rel": "self", "href": "BASE_URL/ords/admin/client/client"}, {"rel": "edit", "href": "BASE_URL/ords/admin/client/client"}, {"rel": "describedby", "href": "BASE_URL/ords/admin/metadata-catalog/client/item"}, {"rel": "first", "href": "BASE_URL/ords/admin/client/client"}]}	
<b>Paso 5.a</b>	<b>Tipo</b>	POST	
	<b>Endpoint</b>	BASE_URL/ords/admin/message/message	
	<b>Body</b>	{ "id":1, "messagetext": "Este es un mensaje" }	
	<b>Salida</b>		
<b>Paso 5.b</b>	<b>Tipo</b>	GET	
	<b>Endpoint</b>	BASE_URL/ords/admin/message/message	
	<b>Body</b>		
	<b>Salida</b>	{ "items": [{"id":1, "messagetext": "Este es un mensaje"}], "hasMore": false, "limit": 25, "offset": 0, "count": 1, "links": [{"rel": "self", "href": "BASE_URL/ords/admin/message/message"}, {"rel": "edit", "href": "BASE_URL/ords/admin/message/message"}, {"rel": "describedby", "href": "BASE_URL/ords/admin/metadata-catalog/client/item"}, {"rel": "first", "href": "BASE_URL/ords/admin/message/message"}]}	
<b>Paso 6.a</b>	<b>Tipo</b>	DELETE	
	<b>Endpoint</b>	BASE_URL/ords/admin/message/message	
	<b>Body</b>	{ "id":1 }	
	<b>Salida</b>		
<b>Paso 6.b</b>	<b>Tipo</b>	GET	
	<b>Endpoint</b>	BASE_URL/ords/admin/message/message	
	<b>Body</b>		
	<b>Salida</b>	{ "items": [], "hasMore": false, "limit": 25, "offset": 0, "count": 0, "links": [{"rel": "self", "href": "BASE_URL/ords/admin/message/message"}, {"rel": "edit", "href": "BASE_URL/ords/admin/message/message"}, {"rel": "describedby", "href": "BASE_URL/ords/admin/metadata-catalog/client/item"}, {"rel": "first", "href": "BASE_URL/ords/admin/message/message"}]}	