

Mi Gimnasio

Documentación Técnica

ACADEMIA SAPUI5 – SOFTTEK

TOMAS A. SANCHEZ

tosanchez@est.frba.utn.edu.ar

SEP. 24, 2020

Change Log

Autor	Descripción	Versión	Fecha
TOMAS A. SANCHEZ	RELEASE	RELEASE	24 SEP. 20

Contenido

Change Log	2
Introducción	4
Estructura del documento	4
Resumen	4
Detalle de funcionalidad	5
Descripción	5
Modo de uso	5
Estado Actual	6
Mejoras de implementación	6
Desarrollo	7
Estructura de la aplicación	7
Back-end	7
Front-End	8
Navegación	11
Diagrama de Navegación	11
Metodología	12
Conclusiones	13
Apéndice A	14
Acerca de este documento	15

Introducción

Estructura del documento

El documento comienza con una introducción donde, se aclarará la estructura a seguir, se procederá con un resumen, y luego se entrará en detalles. El lector conocerá el detalle funcional como así el un panorama del modo de uso del usuario. Se reflejará la situación actual y a su vez se brindarán ideas de mejora.

Encontrará los detalles técnicos en ambos niveles de desarrollo, *fronty back end*. Se ofrece una visualización de las vistas y navegación utilizada. Estos dos puntos están presentados con gráficos para una mejor transmisión del concepto.

También tendrá disponible una breve metodología de resolución, donde se exponen los obstáculos que fueron presentándose, junto con la solución que se les brindo. Sin embargo, no espere código *JavaScript* o *ABAP*.

A modo de cierre, una conclusión comentando las experiencias, incluyendo inconvenientes, expectativas, sugerencias, expuestas en un apéndice. Finalmente, encontrara los derechos de autor y distribución de este documento.

Resumen

En resumen, "Mi Gimnasio" nació como una aplicación desarrollada en ocho días como proyecto final de una academia *SAP UI5* brindada por *Softtek*. Actualmente permite gestionar visualizar y gestionar los distintos deportes brindados con los detalles de los clientes. Se utilizaron dos tablas, donde solo se implementaros los métodos *CRUD* en *ABAP*. Posee un gran alcance de ampliación sumándose rutinas, mantenimiento estadísticas de cuotas, etc. El principal inconveniente lo presentaron las tablas y se utilizo listas en su reemplazo, obteniendo un mejor resultado que el pensado. A continuación, se le presentara mas en detalle lo expuesto en este párrafo.

Detalle de funcionalidad

Descripción

Se trata de una aplicación SAP Fiori¹ que utiliza un servicio *oData*. Esta fue ideada para la autogestión de un *centro deportivo*. La interfaz está pensada para uso móvil-web, siendo la navegación simple e intuitiva. Dentro, el usuario podrá:

- Gestionar clientes.
 - o Añadir, Actualizar y Eliminar clientes.
- Gestionar de deportes.
 - o Suscripciones y rutinas.
- Controlar de cobros.
- Controlar el mantenimiento del equipamiento.
- Controlar las de ventas de insumos.

Modo de uso

Una vez dentro de “*Mi Gimnasio*”, se encontrará con un menú donde podrá visualizar los casos de uso antes nombrados.

Al ingresar a la sección *Clientes*, se ofrecerá el listado de estos; a su vez se permite navegar hacia el detalle de uno seleccionado, donde se cambiará a vista *Máster-Detail*². Allí se permite la edición de datos de cada cliente; si es que se habilita esa opción, caso contrario los campos permanecerán no editables; así como también la suscripción o baja de un deporte.

Cada caso de creación ya sea un nuevo cliente o un deporte, se da en otra nueva vista, donde se presentan campos para completar, se verifica que los datos sean correctos y se realiza la operación. De no poder completarse se informa que la/s *primary key* ya existe, cualquiera fuese el verdadero error.

Cabe mencionar en cuanto a la eliminación de un cliente, esta no es posible si primero no se eliminaron todos los deportes a los que estaba subscripto manualmente. Esto se da para asegurar que no queden deportes relacionados a un cliente inexistente.

¹ SAP Fiori is the design language that brings great user experiences to enterprise applications. Based on user roles and business processes, SAP Fiori simplifies doing business.

² A master–detail interface displays a master list and the details for the currently selected item.

Estado Actual

A la última versión indicada en el [Change Log](#), se encuentre implementada:

- Back-end:
 - Tabla Clientes
 - Tabla Deportes
- Home menú
- Gestión de Clientes:
 - Listado
 - Anadir
 - Máster-Detail
 - Actualizar y Eliminar.
 - Suscripción y baja de Deportes.
- Visualización de listado de suscripciones a Deportes
- Recurso no disponible.

Lo no implementado redirige a no disponible.

Mejoras de implementación

Se presenta el siguiente listado como propuesta de desarrollo:

- Back-end:
 - Tabla Clientes x Deportes³
 - Tabla de Cuotas
 - Tabla de Profesores
- Gestión de Clientes
 - Máster-Detail
 - Redireccionamiento de urls: arreglar el patrón.
- Gestión de Deportes
 - Filtrado de deportes
 - Actualización de precio del deporte

Si bien fue ideado bajo estas expectativas, el tiempo de desarrollo no alcanza para lograr al menos su funcionalidad. Estas resultan interesantes y podrían lograr completar la funcionalidad de autogestión

Lo primordial sería evitar el hard-coding de ciertos parámetros, para lograr un dinamismo automático, suprimir la creación de deportes, ya que estos contienen información repetida.

³ La idea es una tabla con los identificadores del cliente y del deporte, para evitar la repetición de deportes.

Desarrollo

Estructura de la aplicación

Back-end

Se desarrollo únicamente las operaciones básicas CRUD⁴ en ABAP

Entidad relación

Es un modelo simple, una relación uno a muchos



Figure 1 - Entity Relation Diagram

Detalles Técnicos

SISTEMA	
AEX	

ORDEN DE TRANSPORTE	
AEXK90042	TOSA8 - ACADEMIA

PAQUETE	
Z001	

TABLAS	
ZTCA_GYM_CLTS	ZTCA_GYM_DEP

TIPOS DE DATOS		
ZEDCA_DNI	ZEDCA_GDEP	ZEDCA_GHOR

⁴ Create, Update, Read and Delete

DOMINIOS	
ZEDCA_GDEPS	ZEDCA_GHORS

ODATA SERVICE	
ZOS_CA_TOSA8_ACADEMIA	

Front-End

Los siguientes esquemas son conceptuales, bien no representan la interfaz real.

Interfaz de Usuario

Comenzamos en un home-page, donde se dispondrán las distintas operaciones.

Vista principal

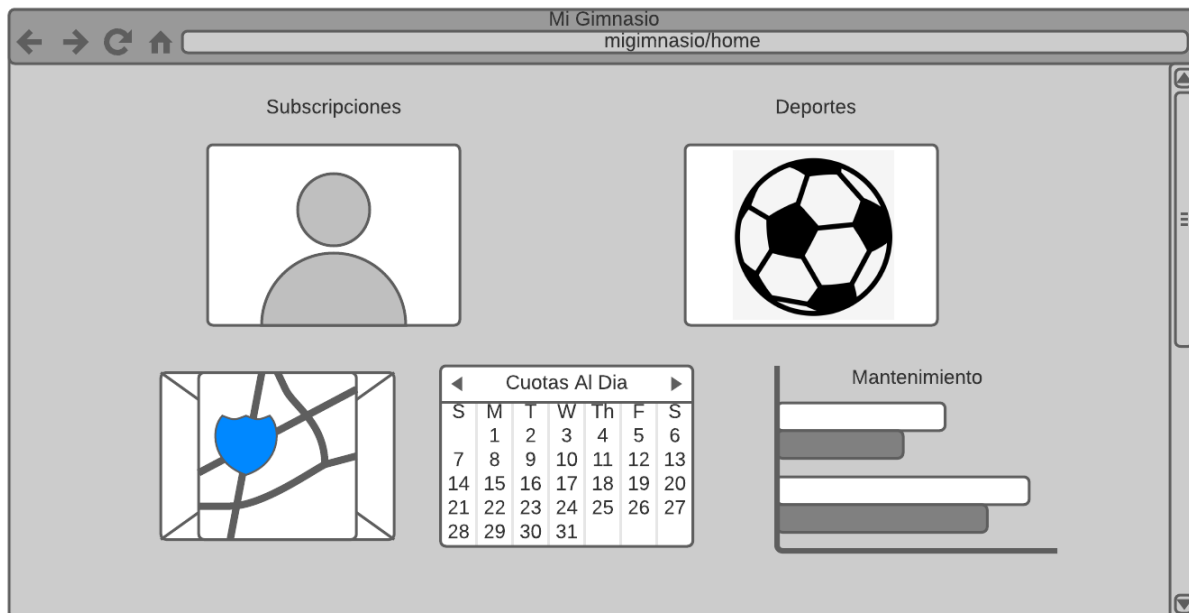


Figure 2 - Home Page

Si se desea realizar la operación de gestionar las subscripciones

Gestor de Clientes

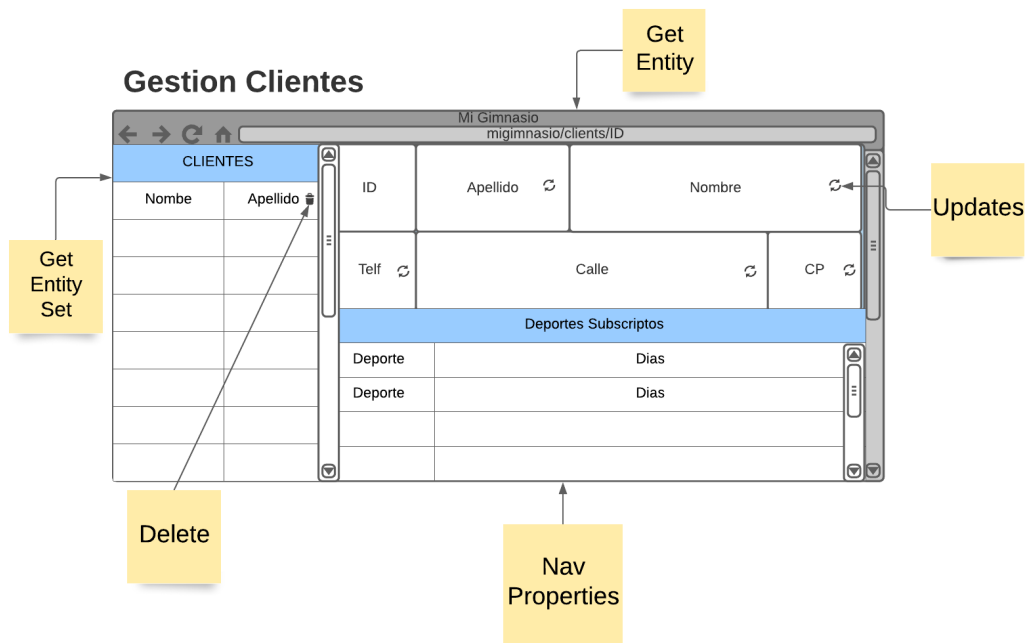


Figure 3 - Client Manager

Añadir cliente

The screenshot shows the 'Nuevo Cliente' form in the application. The title bar indicates the application name 'Mi Gimnasio' and the current path 'migimnasio/clients/new'. The form is divided into two main sections. The left section, titled 'Nuevo Cliente', contains a list of input fields for 'ID', 'Nombre', 'Apellido', 'Telefono', 'CP', and 'Calle'. The right section contains a list of input fields for 'INPUT'. At the bottom of the form, there are two buttons: 'CANCELAR' and 'CREAR'.

Figure 4 - Client Constructor

Para el caso de un nuevo deporte, restringimos el deporte a un id, por lo que no se podrá crear un deporte que no tenga cliente, recordando el diagrama de entidad relación.

Suscripción a deporte

MI Gimnasio
migimnasio/clients/ID/new

Nuevo Deporte

ID

Deporte

Profe

Costo

Costo

CANCELAR CREAR

INPUT (BLOCKED)

Opción 1
Opción ...
Opción n

INPUT

INPUT

Opción 1
Opción ...
Opción n

Figure 5 - Sport Membership

Gestor de Deportes

Deportes

MI Gimnasio
migimnasio/sports

Deportes	Tenis	Costo	Cliente
Futbol	Profe	Dias	Cliente
Tenis	Profe	Dias	Cliente
Natacion	Profe	Dias	Cliente
Gym	Profe	Dias	Cliente
Basket	Profe	Dias	Cliente
Hockey	Profe	Dias	Cliente

Update All

Get Entity Set Filtered

Delete Entity

Figure 6 - Sport Manager

Navegación

Mi gimnasio cuenta con siete vistas, y una mas no visible al usuario, se muestra a continuación un esquema

Diagrama de Navegación

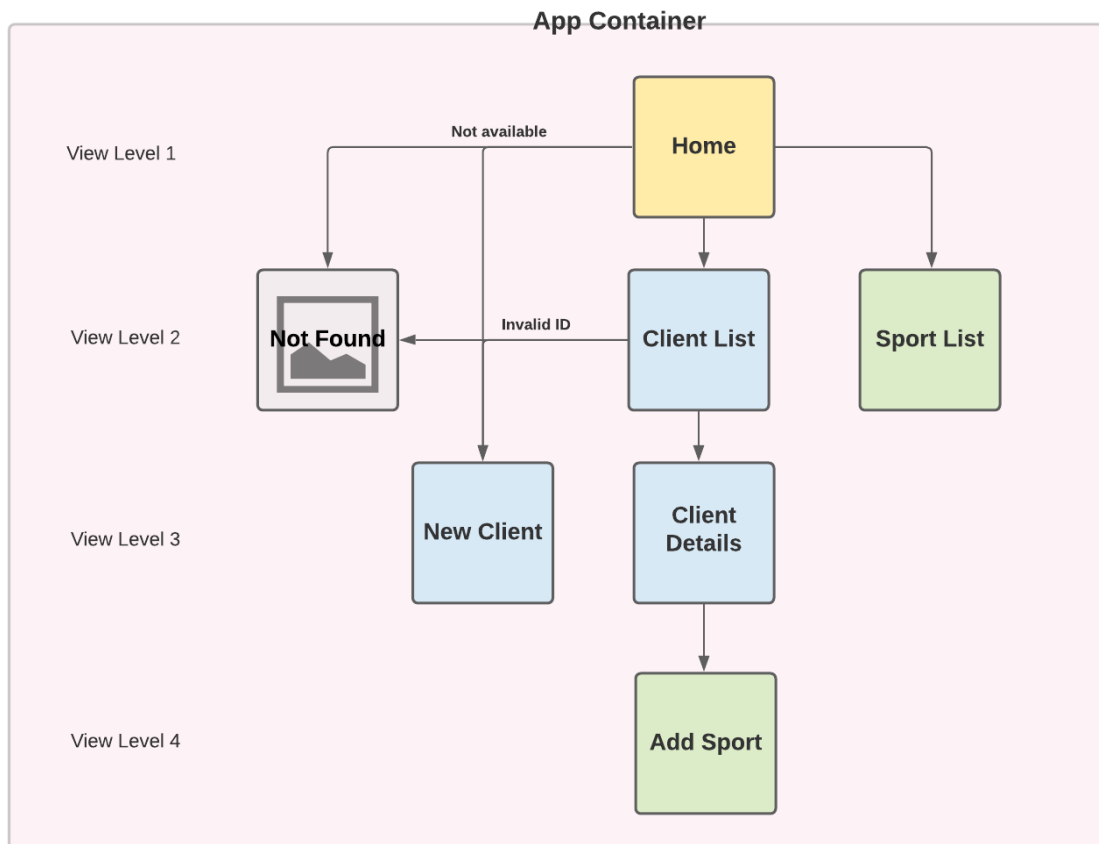


Figure 7 - Navigation Diagram

Metodología

Durante la etapa de boceto y planificación, Mi gimnasio sufrió grandes cambios.

La primera idea fue una Tabla que mostrara todos los datos del cliente, con navegación a los detalles de cada uno. El uso de navegación por tabla solo trajo problemas, principalmente con `getBinding()`, `getBindingContext()`, donde no se supo resolver y se optó por la sustitución a Listas.

Descartada la lista, se paso a un modelo inspirado en el template de Máster-Detail, los inconvenientes se presentaron debido al mal uso del container `<splitApp />` que no permitía agregar vistas nuevas con otro tipo de container app. Esto se solucionó cambiándolo a `<App />`.

Este modelo fue avanzando hasta que comenzaron los inconvenientes con la sección de detalles, los cuales estaban planificados en modo tabla. Se soluciono pasando a contenedores de texto, luego se amplio en contenedores de tipo `<input />`, que por razones estéticas seria cambiado a `<inputItemList />`.

Nuevamente las vistas causaron problemas, al intentar aplicar la nueva vista para “No encontrado”, donde sumado al agregado de más contenedores para mostrar los detalles, recordemos la [Figura 3 - Gestor de clientes](#), lo cual ocasiono un cambio de proyecto y se implemento el contenedor App hasta ahora vigente.

Una vez más, las tablas vuelven a romper la aplicación, en este caso se trato de la tabla dedicada al *navigation property*⁵, donde se repitió el problema de obtener el contexto del objeto en cuestión a la hora de implementar el eliminado. La solución volvió a recaer en las listas, donde se configuraron para soportar el modo *delete*.

```
var oList = oEvent.getSource(),
    oArray = oList.getModel("oneClientSports").getData(), //Array with all listed sports
    oIndex = oList._oItemNavigation.iFocusedIndex, // Index of item selected
    oSport= oArray[oIndex]; // Sport fetched
```

Se considera importante destacar el uso de un *BaseController*, del cual todos los demás *controllers* heredaran sus métodos. En este se define algo esencial para la navegación que es el botón de atrás, y de no haber redirige hacia home. Otro punto sobre este tema de navegación es el acceso a la *url* de un cliente, al caso donde se quiera agregar un deporte a un cliente inexistente o visualizar detalles de este. Se verifica que sea correcto de no encontrarse el Id solicitado navega hacia *No encontrado*.

⁵ A navigation property is an optional property on an entity type that allows for navigation from one end of an association to the other end.

Conclusiones

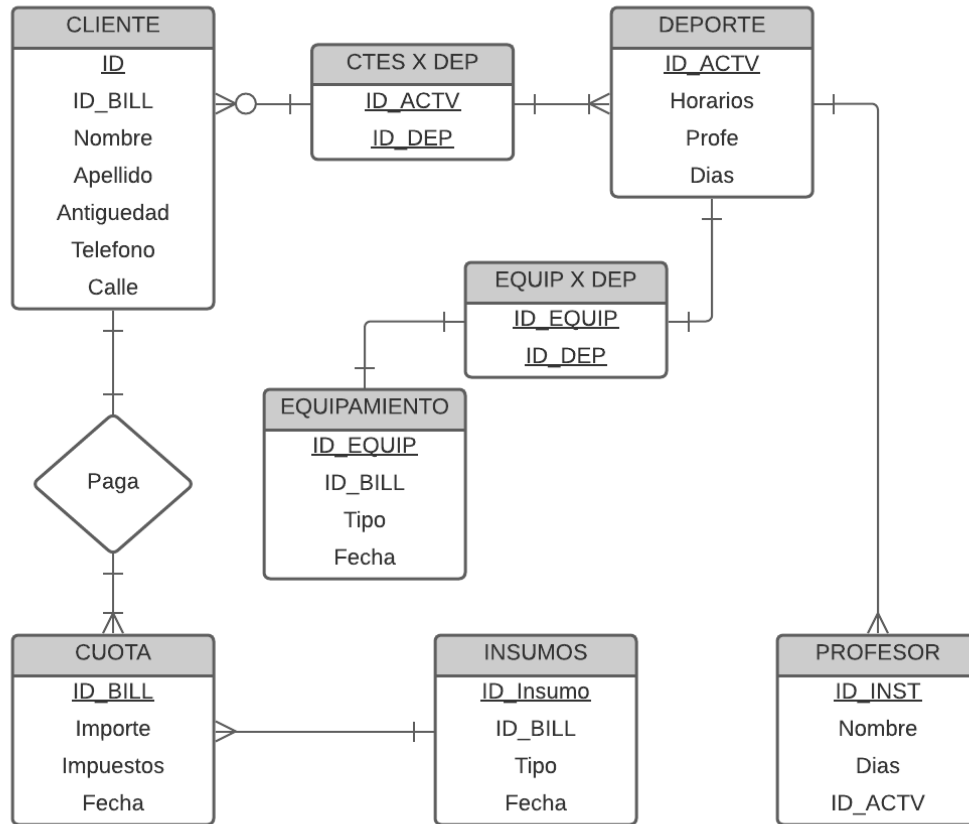
A pesar de ser pocos días de desarrollo, se logró una aplicación funcional, con una interfaz intuitiva. Aun así, apenas cubre una parte del objetivo de desarrollo.

Los principales problemas fueron ocasionados por el uso de tablas, sin conocer bien su funcionamiento. Las listas surgieron como un muy buen reemplazo, y además proporcionaron más estética a la aplicación.

Debe mencionarse el detalle de atención a descuidos del usuario, que represento la mayor parte del desarrollo de scripts. Esto se puede ver en prohibir fijar

Hay muchas cosas por mejorar, tanto los algoritmos de *JavaScript* como la organización del *oData*, donde se puede ampliar el mundo y el alcance de la aplicación, tal como fue ideada. Como sugerencia se encuentra adjunto en un apéndice un nuevo diagrama de entidad relación, que bien podrían corregir y brindar facilidades al modelo, como también añadir funciones interesantes.

Apéndice A



La división en clientes por deporte permitiría tener una tabla de deportes, donde únicamente se detallan los deportes, evitando poseer en la base de datos varios mal llamados deportes con la misma actividad.

También se propone expandir tomando en cuenta las cuotas a pagar, para llevar un control de la recaudación, y quien debe ponerse al día con los pagos. Reflejando también ganancias tenemos la venta de insumos. Las pérdidas se podrían visualizar en gastos por mantenimiento de equipamiento, cuando se deberían actualizar.

Acerca de este documento

Copyright ©2020 por Tomás Agustín Sánchez

Distribuido bajo una licencia Creative Commons Atribución/Reconocimiento - NoComercial - CompartirIgual 4.0 Internacional (BY-NC-SA). Para más información acerca de los permisos y restricciones de estas considere visitar: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

La información contenida en este documento fue desarrollada persiguiendo fines académicos, a lo largo de una capacitación en SAP UI5 brindada por Softtek Argentina, dictado por Lucas Uccello y asistido por Diego Gagliardi, con duración de dos semanas, siendo el despliegue de una aplicación, y su documentación correspondiente la última instancia de evaluación. Para saber más acerca de Softtek visite: <https://www.softtek.com/>

Para más información sobre el autor visite: <https://tomasanchez.github.io/about>

Fuentes de información:

SAP SE, SAPUI5 APPLICATION PROGRAMMING INTERFACE.

<https://sapui5.hana.ondemand.com/sdk/>

SAP SE, SAPUI5 Documentation – Navigation and Routing Guide.

<https://sapui5.hana.ondemand.com/1.36.6/docs/guide/1b6dcd39a6a74f528b27ddb22f15af0d.html>

SAP SE, SAPUI5 Samples – Input List Item sample.

<https://sapui5.hana.ondemand.com/sdk/#/entity/sap.m.InputListItem/sample/sap.m.sample.InputListItem>

SAP SE, SAPUI5 API – Message Box API.

<https://sapui5.hana.ondemand.com/sdk/#/api/sap.m.MessageBox>

SAP SE, SAPUI5 API – Input API.

<https://sapui5.hana.ondemand.com/sdk/#/api/sap.m.Input>