



# **Graballo+ Integración de Portav a CRM**

Versión 2.0

## Índice

Índice .....	2
Introducción .....	3
<u>ANTECEDENTES GENERALES</u> .....	3
<u>ACERCA DE ESTE DOCUMENTO</u> .....	3
Descripción General .....	4
<u>CONCEPTOS GENERALES</u> .....	4
<u>GESTIÓN DE CLIENTES</u> .....	5
<u>REQUERIMIENTOS DE INTEGRACIÓN A CRM</u> .....	5
<i>Validación de usuarios</i> .....	5
<i>Asignación de número de caso (ticket)</i> .....	5
<i>Actualización archivo de audio</i> .....	5
<i>Actualización registro de operación</i> .....	5
<i>Estado de un caso</i> .....	6
Integración a CRM.....	7
<u>FORMATOS DE ARCHIVOS DE INTERACCIÓN</u> .....	7
<i>Validación de usuario</i> .....	7
<i>Asignación de ticket</i> .....	9
<i>Consulta del estado de una variable</i> .....	10
<i>Actualización de registro en base de dato</i> .....	11
<u>INTEGRACIÓN PASO A PASO</u> .....	13

## Introducción

Este documento contiene las especificaciones técnicas para integrar el portal de voz **Portav** de **Adelacu** con un sistema de control de gestión de requerimientos (CRM). La integración de **Portav** con el sistema SGI de ACGP componen el producto **Graballo+**.

### Antecedentes generales

La versión actual de Graballo+ se compone de la integración entre Portav y SGI, sin embargo, Portav se puede integrar a cualquier otro sistema de control de gestión.

### Acerca de este documento

Este documento está dirigido a ingenieros y personal técnico del área de la electrónica e informática.

## Descripción General

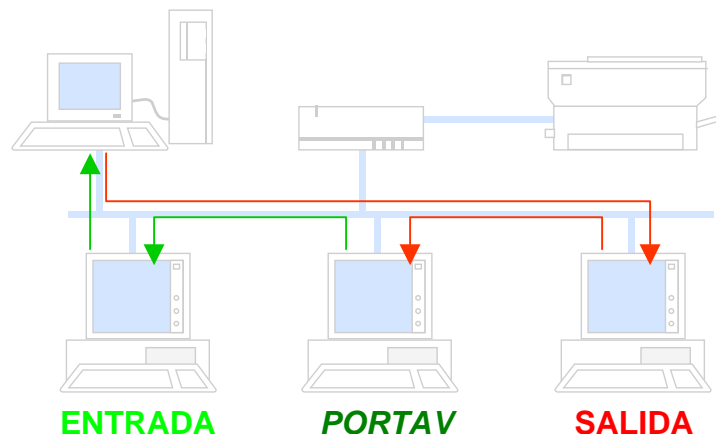
A continuación se indica la forma en que opera el sistema portal de Portav y su integración a otros sistemas externos.

### Conceptos generales

El portal de voz **Portav**, componente base de los productos de atención de clientes **Graballo+** y de evaluación de calidad de servicio **Enquesta**, contiene todos los elementos necesarios para responder una llamada telefónica y entregar información al usuario. La información puede estar almacenada dentro de **Portav** o consultada a sistemas externos.

Para las consultas e interacciones con sistemas externos se utiliza un sistema de solicitud y respuesta en base a archivos de textos, los que se manejan a través de un software independiente que se denomina **Junior** y conceptualmente realiza la misma tarea que un mensajero en una organización, recoge solicitudes y trae las respuestas.

A continuación se muestra un esquema de cómo se realiza la interacción con un sistema externo. Para representar la operación se han definido las carpetas (**ENTRADA**) y (**SALIDA**), aunque se muestran cada una en una máquina independiente, ambas carpetas pueden estar ubicadas en cualquier punto de la red que disponga de los permisos de acceso requeridos (lectura y escritura), incluso con un sistema operativo diferente a Windows.



1. Cada vez que se requiera realizar una consulta o una consulta, **Portav** dejará en **ENTRADA** un archivo con el requerimiento.
2. **Junior**, que se encuentra permanentemente en operación, detectará la existencia del archivo con el requerimiento y realizará la acción correspondiente. Una vez leído el requerimiento borrará el archivo.
3. Una vez obtenida la respuesta, **Junior** genera un archivo de salida en **SALIDA**.
4. Al estar disponible la respuesta, **Portav** lee el archivo respectivo y lo borra.

## Gestión de clientes

Para la atención de clientes, el portal de voz **Portav** puede entregar distintos tipos de información (valores numéricos, estados de variables, etc.), sin embargo, para la atención de un cliente se requiere que éste manifieste alguna inquietud. Para el caso de una atención telefónica se considera la grabación de un mensaje de voz, donde el cliente plantea sus requerimientos.

Una vez recibido el requerimiento, **Portav** informa al cliente un número único de atención, el que internamente se utiliza para darle atención y con el cual el cliente puede consultar posteriormente.

## Requerimientos de integración a CRM

Para la integración a un sistema externo de CRM, como es el caso de SGI, **Portav** necesita realizar algunas tareas básicas, tal como se indica a continuación.

### Validación de usuarios

Adicionalmente, antes de grabar un mensaje con un requerimiento, el cliente puede ingresar un identificador que permita aceptar o rechazar el ingreso y adicionar información que el propio cliente ingrese, como por ejemplo el número de RUT, número de factura, número de cliente, etc.. Los clientes se pueden validar localmente en base a la información contenida en un archivo de texto, o en línea a través de la consulta en un sistema de base de datos.

### Asignación de número de caso (ticket)

Aunque **Portav** genera internamente un número único (ticket) por cada llamada, para la integración a un sistema externo se requiere que la numeración coincida. Para tal efecto, el número del caso (ticket) con que se identifica la grabación del cliente debe ser asignado por el sistema de CRM externo. Por este motivo, al comenzar la operación de grabación, **Portav** realiza un requerimiento al CRM para la asignación del número.

### Actualización archivo de audio

En la operación de **Portav** con SGI, el archivo de audio con la grabación se almacena en el sistema de archivos del equipo y una variable de operación define su acceso desde la plataforma web. Para la operación con otros sistemas CRM, el archivo de audio se puede traspasar vía FTP e ingresar a la base de datos mediante sentencias SQL o procedimientos almacenados.

### Actualización registro de operación

Una vez grabado el mensaje del cliente e informado el número de atención (ticket), se debe actualizar en la base de datos la información del nuevo requerimiento (ticket). La información disponible para ingresar a la base de datos incluye la identificación de cliente, el UserID o RUT si se realizó alguna validación, y las opciones seleccionadas durante la navegación por el portal de voz **Portav**.

### Estado de un caso

Una vez que un cliente ha dejado registrado un caso y dispone de un ticket, puede hacerle un seguimiento a través de una consulta telefónica, para ello debe solamente ingresar el número de ticket.

La respuesta puede ser el estado en que se encuentra el caso (aceptado, rechazado, etc.), o bien la lectura de una información. Para esto último se utiliza un software de síntesis de texto a voz (TTS), con lo que se puede leer un texto completo.

Mayor información de TTS se encuentra en [http://www.adelacu.com/pags/ficha\\_tts.html](http://www.adelacu.com/pags/ficha_tts.html).

## Integración a CRM

A continuación se indican los aspectos técnicos de la operación entre **Portav** y el CRM.

### Formatos de archivos de interacción

Cada consulta que se realiza entre **Portav** y el CRM se realiza sobre la base de comunicación de Junior. Esto se realiza en base a requerimientos formulados en archivos de textos. Los detalles se encuentran en el manual de definición de Junior *Especificaciones Junior.pdf*.

Si el CRM soporta interacción con comandos SQL, se puede utilizar el software Junior-BDD, en tal caso sólo se requiere configurar. Mayor información se encuentra en el manual *junior\_bdd\_manual.pdf*.

### Validación de usuario

A continuación se indica un ejemplo con un archivo de configuración para la validación de un usuario en una operación de **Portav**. El archivo de configuración **usr.xml** tiene el siguiente contenido:

```
<consulta>
    <Dsn name="Adelacu" usr="sap" psw="user_sap" />
    <Respta name="inf" />
    <query>
        Valida_Cliente'${1}'
    </query>
    <respuesta>
        ${Estado_Cliente}
    </respuesta>
</consulta>
```

Condiciones establecidas en el archivo de configuración:

- Archivo de salida: **nnnnnn.inf**, donde nnnnnn corresponde al nombre del archivo con extensión **usr** usado en la consulta.
- La respuesta corresponde al contenido de la variable **Estado\_Cliente** definida en el procedimiento almacenado de consulta y se ubica en la primera fila del archivo con extensión .inf. Si contiene el texto "VOID" el cliente no está registrado en la base de datos, de lo contrario tendrá la clave indicada en dicha línea, incluso el string vacío si el cliente existe, pero no tiene clave asignada.
- La consulta se realizará con el procedimiento almacenado **Valida\_Cliente**, con el parámetro indicado en la primera línea del archivo **nnnnnn.usr**.

Archivo de solicitud: **nnnnnn.usr**

778889992  
PORTAV BDD

Archivo Respuesta: **nnnnnn.inf**, cliente existe, su clave secreta es 1234

1234  
PORTAV BDD

Archivo Respuesta: **nnnnnn.inf**, cliente existe, no tiene clave secreta

PORTAV BDD

Archivo Respuesta: **nnnnnn.inf**, cliente no existe

VOID  
PORTAV BDD



### Asignación de ticket

A continuación se indica un ejemplo con un archivo de configuración para la solicitud de un ticket de atención en una operación de **Portav**. El archivo de configuración **sgi.xml** tiene el siguiente contenido:

```
<consulta>
  <Dsn name="Adelacu" usr="sap" psw="user_sap" />
  <Respta name="asg" />
  <query>
    Get_Ticket'${1}'
  </query>
  <respuesta>
    ${Ticket_Number}
  </respuesta>
</consulta>
```

Condiciones establecidas en el archivo de configuración:

- Archivo de salida: **nnnnnn.asg**, donde nnnnnn corresponde al nombre del archivo con extensión **sgi** usado en la consulta.
- La respuesta corresponde al contenido de la variable **Ticket\_Number** definida en el procedimiento almacenado de consulta y se ubica en la primera fila del archivo con extensión .asg.

Archivo de solicitud: **nnnnnn.sgi**

```
nnnnnn
PORTAV BDD
```

Archivo Respuesta: **nnnnnn.asg**. El ticket asignado es el 12345678

```
12345678
PORTAV BDD
```

### Consulta del estado de una variable

A continuación se indica un ejemplo con un archivo de configuración para una consulta sobre el estado de una variable en una operación de **Portav**. Si el archivo tiene el nombre **r07.xml**, esto significa que **Junior-BDD** habrá leído los parámetros de operación en el archivo **r07.xml**.

El archivo de configuración **r07.xml** tiene el siguiente contenido:

```
<consulta>
  <Dsn name="Adelacu" usr="sap" psw="user_sap" />
  <Respta name="ans" />
  <query>
    Consulta_Estado_Ticket'${1}'
  </query>
  <respuesta>
    ${Estado_Ticket}
  </respuesta>
</consulta>
```

Condiciones establecidas en el archivo de configuración:

- Archivo de salida: **nnnnnn.ans**, donde nnnnnn corresponde al nombre del archivo con extensión **r07** usado en la consulta.
- Manejo de errores: Como no existe el identificador **<Error>** ni los identificadores de cada tipo de error en particular (**<Connect>**, **<SQL>** y **<Datos>**), cuando no exista comunicación con la base de datos, se produzca un error SQL o no exista información sobre el parámetro nnnnnn, la respuesta será 1, 2 y 3 respectivamente.
- La consulta se realizará con el parámetro indicado en la primera línea del archivo **nnnnnn.r07**.
- Como respuesta se entrega el valor de la variable **respuesta** de la base de datos.

Archivo de solicitud: **nnnnnn.r07**

```
555777
PORTAV BDD
```

Archivo Respuesta: **nnnnnn.ans**. El estado actual corresponde al número 5, por lo que se informará el mensaje correspondiente.

```
5
PORTAV BDD
```

### Actualización de registro en base de dato

A continuación se indica un ejemplo de la actualización de un ticket. Los datos con que operará **Junior-BDD** están especificados en el archivo **uph.xml**, el que tiene el siguiente contenido:

```
<consulta>
  <Dsn name="Adelacu" usr="sap" psw="user_sap" />
  <Respta name="aup" />
  <query>
    ADD_Ticket '${1}', '${2}', '${3}', '${4}'
I  </query>
  <respuesta>
    '$1'
  </respuesta>
</consulta>
```

Condiciones establecidas en el archivo de configuración:

- Parámetros de entrada, ubicados en las 4 primeras líneas del archivo **nnnnnn.uph**.
  - Número de ticket.
  - Identificador del cliente, por ejemplo RUT. Si el cliente no se autenticó se utiliza un identificador por defecto.
  - Identificador del producto, GRABALLO para mensajes de voz, FAXIL para fax.
  - Información de la llamada, string de hasta 256 caracteres, los retornos de carro se identifican con los caracteres \n.
- Archivo de salida: **nnnnnn.aup**, donde nnnnnn corresponde al nombre del archivo con extensión **p2w** usado en la consulta. La respuesta que se entrega es el valor de '\$1', que será 3 si es que no hay errores con la base de datos.

Archivo de solicitud: **nnnnnn.uph**

```
12345678
778889992
GRABALLO
Mensaje de Inicio\nReclamos\n
PORTAV BDD
```

Archivo Respuesta: **nnnnnn.aup**. Los datos se actualizaron en la base de datos

3 PORTAV BDD
-----------------

## Integración paso a paso

A continuación se indican los pasos que se deben realizar para integrar **Portav** a un CRM.

1. Definición del servicio a entregar. Se deben definir las funcionalidades, capacidades y restricciones que tendrá el sistema desde una perspectiva estratégica comercial. Esto incluye desde el número de acceso del sistema, las opciones de entrega de información y los niveles de servicio a entregar.
2. Diseño del software que responderá las consultas y requerimientos de **Portav**. Si el CRM acepta consultas SQL, no es necesario diseñar este software, se puede utilizar el software **Junior-BDD** de Adelacu.
3. Diseño de consultas SQL o procedimientos almacenados para la integración de los sistemas.
  - Validación de usuarios
  - Asignación de ticket
  - Actualización de registro en base de datos
  - Consulta de estado de ticket
4. Las pruebas de integración se pueden realizar simulando la operación de **Portav** mediante la creación de archivos de texto.

Sociedad Comercial Adelacu Limitada  
[www.adelacu.com](http://www.adelacu.com)