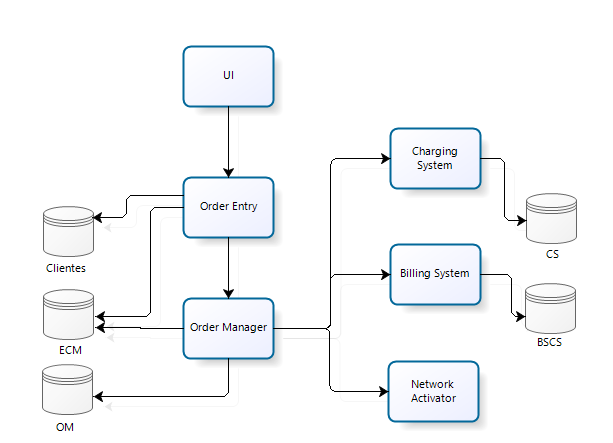
Proyecto “Cobras”

Introducción

El objetivo del proyecto es crear un conjunto de sistemas que permitan simular un flujo simple de creación de una orden de venta para luego crear una orden de activación en sistema de charging, billing y activación en la red.

Nuestro grupo eligió el escenario de una venta de línea nueva (Prepago)

Diseño de alto nivel



En la UI, el usuario tendrá la posibilidad de:

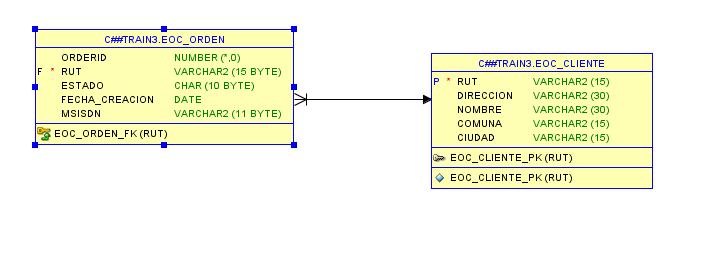
* Buscar clientes existentes
* Crear clientes
* Comprar un plan de prepago
* Crear una orden

Cada una de los sistemas mencionados en el diagrama será un proyecto independiente en Velocity y se comunicarán a través de Web Services.

Order Entry

El Order Entry disponibilizará una GUI donde se podrá dar de alta clientes, así como seleccionar un cliente para activarle un plan. Este plan se seleccionará desde el ECM y posteriormente la orden se enviará al Order Manager.

El modelo de datos preliminar es el siguiente:



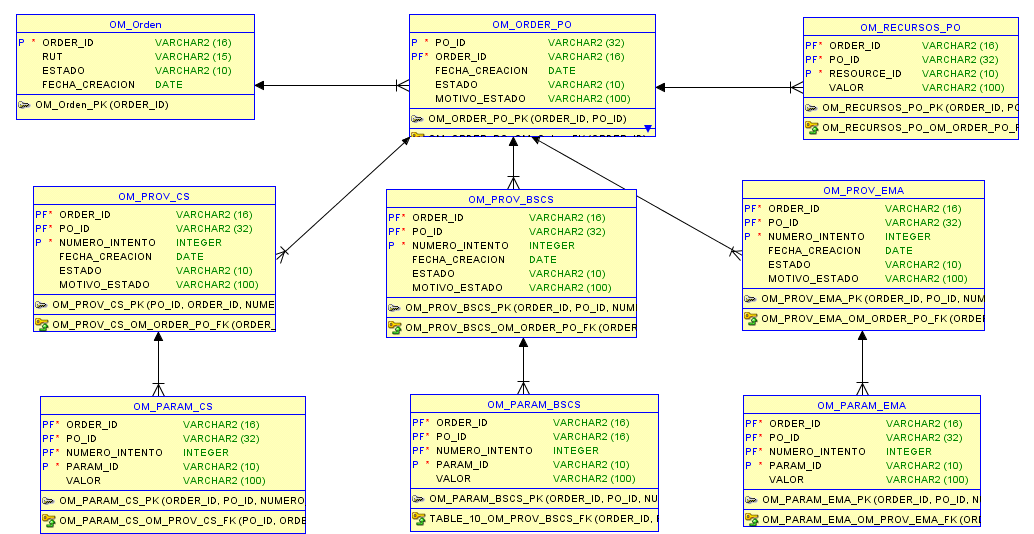
Este modelo no incluye las tablas que corresponden al ECM, pero si se deberán agragar campos a la tabla EOC\_ORDEN con datos referentes a la PO obtenida desde el ECM.

Order Manager

El Order Manager será el encargado de recibir una petición de orquestación de una orden. Recibirá la identificación del cliente y del plan seleccionado, y con esa información consultará al catálogo los parámetros para la activación en CS, BSCS y EMA.

El OM almacenará el estado de la provisión en cada sistema en la medida que va avanzando en el flujo de provisión, para tener un registro de lo que pasó con la orden.

El modelo de datos preliminar es el siguiente:



Operaciones

**OM\_PROVISIONAR\_ORDEN**

Será una operación Request-Response que recibirá el número de orden, el cliente y las PO que se deben provisionar. Retornará un ack. La respuesta del resultado de la provisión será notificada mediante una operación Notification.

Parámetros de Entrada:

* ORDER\_ID, *VARCHAR(16):* Identificador de la Orden
* RUT, VARCHAR(15): Rut del cliente
* PO\_IDs, VARCHAR(32): Lista de POs asociadas a la orden. Los parámetros que OM debe enviar a cada uno de los sistemas, los rescatará desde ECM dentro del flujo de provisión.

Parámetros de Salida

* Ack
* Estado, VARCHAR(10)
* Descripción, VARCHAR(20)

Parámetros del método Notification

* Response
  + OrderId
  + Rut
  + EstadoOrden
  + MSISDN
  + POIdBasica
  + EstadoPOBasica
  + MotivoEstadoPOBasica
  + POIdAdicional
  + EstadoPOAdicional
  + MotivoEstadoPOAdicional

Errores:

1: La orden ingresada ya existe

2: Error al obtener un nuevo MSISDN

**OM\_CONSULTAR\_ESTADO\_ORDEN**

Permite consultar el estado de la Orden, retorna el estado de la provisión en cada sistema.

Parámetros de Entrada

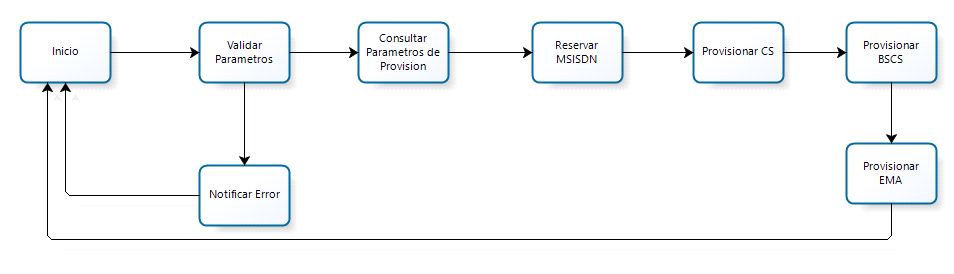
* ORDER\_ID, VARCHAR(16)

Parámetros de Salida

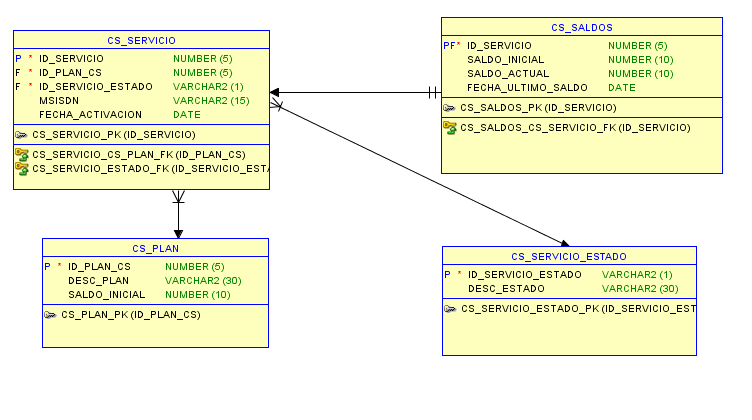
* ORDER\_ID, VARCHAR(16)
* FECHA\_CREACION, DATETIME
* ESTADO\_ORDEN, VARCHAR(10)
* MOTIVO\_ESTADO\_ORDEN, VARCHAR(100)
* PO\_ID, VARCHAR(32)
* ESTADO\_PO, VARCHAR(10)
* MOTIVO\_ESTADO\_PO, VARCHAR(100)
* ESTADO\_BSCS, VARCHAR(10)
* MOTIVO\_ESTADO\_BSCS, VARCHAR(100)
* ESTADO\_CS, VARCHAR(10)
* MOTIVO\_ESTADO\_CS, VARCHAR(100)
* ESTADO\_EMA, VARCHAR(10)
* MOTIVO\_ESTADO\_EMA, VARCHAR(100)

Flujo de provision

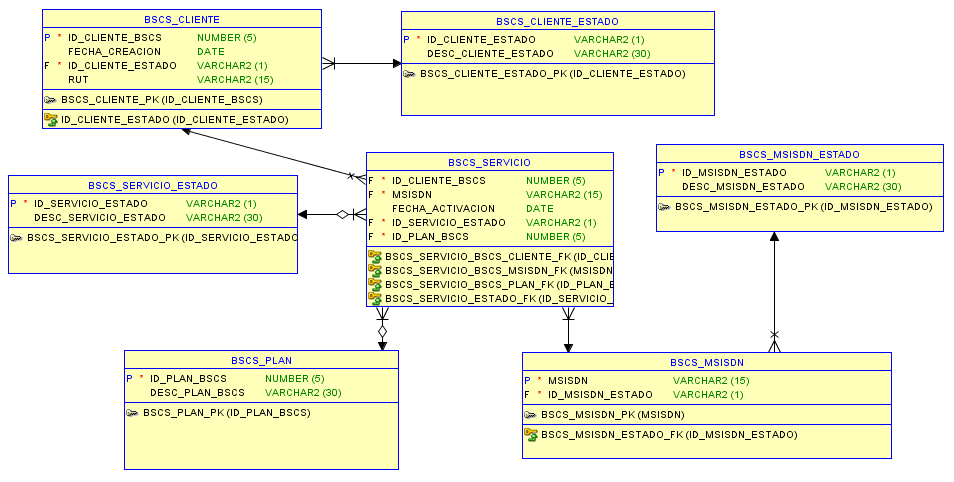
Preliminarmente, el proceso de provisión debería estar representado por un proceso de usuario, similar a mostrado en el siguiente diagrama:



Charging System



Billing System



Network Activator