

PROYECTO DDSI

EQUIPO DE FÚTBOL

Amparo Almohalla Santiago

Pedro Gallego López

Antonio José Lara Peña

Hugo Antonio Teruel Muñoz

DESCRIPCIÓN

El presidente de un equipo de fútbol llamado “Los Serrano F.C”, nos ha solicitado un sistema para la gestión de diversas áreas relativas a la dirección del club. Se nos pide que el sistema sea capaz de administrar las fichas del equipo, los eventos deportivos y extra deportivos así como su organización, la gestión de los socios del club y su servicio de tienda y economía.

Así hemos decidido dividir el sistema en cuatro subsistemas que representan en su conjunto todas las necesidades del club y que se centran en áreas distintas. Estos subsistemas son equipo, eventos, abonados y tienda.

Al sistema solo podrán acceder ciertos empleados del club, que harán de nexos entre el sistema y los distintos usuarios que interactúen con el club. Así, siempre que el agente externo sea un usuario daremos por hecho que existe un empleado del club que haga el nexo que permita la operación en el sistema. Dicho esto, podemos diferenciar entre distintos tipos de agente externo:

- Empleado: trabajador del club.
- Usuario: cualquier persona que interactúa con el sistema.
- Cliente: tipo de usuario que realiza una compra.
- Abonado: tipo de cliente que posee un abono: socio del club.
- Proveedor: interactúa con la tienda para proporcionar los ítems de los distintos productos ofertados.

SUBSISTEMA DE EQUIPO

Hugo Antonio Teruel Muñoz

El subsistema de equipo se centrará principalmente en todo lo correspondiente a los componentes que forman parte del club, ya sean jugadores, plantilla del cuerpo técnico o empleados de cualquier otra área. Entre sus funciones podemos encontrar la del alta y baja de nuevos empleados, indicando sus datos personales, su salario y duración de contrato. Cuando se acaba un contrato que no ha sido renovado se finaliza el contrato. En el caso de, por ejemplo, un jugador se dé de baja antes de su contrato, ya sea por decisión del club o por pago de un fichaje, habrá de indicarse la cantidad que abona o recibe el club. También existe la posibilidad de renovar un contrato, indicando nuevas duraciones y salarios del contrato, incluso un posible nuevo puesto de trabajo. Otra funcionalidad será de la listar

todos los componentes registrados en el sistema. Finalmente, en este apartado del sistema también se hará un balance anual de las cuentas del equipo, incluyendo los beneficios de la tienda, entre otros.

SUBSISTEMA DE EVENTOS

Pedro Gallego López

En este subsistema nos vamos a centrar en la administración de los distintos eventos, tanto en el antes, durante y después del mismo. Es necesario asegurar que cualquier cliente puede comprar la entrada a un asiento y marcando ese asiento como no disponible. En un inicio, todos los asientos estarán disponibles exceptuando los asientos correspondientes a los abonados en los partidos de fútbol. Una funcionalidad necesaria es la consulta del calendario de partidos y del calendario de eventos extra deportivos para mostrar las fechas de cada evento que tendrá lugar en el estadio del club. Para llevar a cabo la gestión de eventos es necesario contactar con empresas externas contratadas por el club (servicio de seguridad, asistencia médica...), así podrá haber un servicio de mensajería de correo electrónico. En última instancia, hay que registrar el informe del evento en donde se tendrán en cuenta estadísticas como la asistencia y el listado de incidencias.

SUBSISTEMA DE ABONADOS

Antonio José Lara Peña

Este subsistema se encarga de gestionar toda interacción de los abonados. En primer lugar, para incluir un abonado nuevo es necesario darle de alta en la base de datos, para ello debe proporcionar sus datos personales tales como DNI, nombre, correo electrónico, asiento... Además este deberá de pagar una cuota de 150€ que será válida en la duración de un año. Respecto al asiento, el abonado propone uno y si no está ocupado éste se le queda asignado. Se deberá devolver un número de abonado que será único para cada uno.

Tras el periodo de un año habrá un mes en el que el abonado puede renovar y seguir siendo abonado siempre que se aporte de nuevo la cantidad de 150€. Si tras este mes el abonado no ha querido renovar automáticamente se dará de baja. Por lo que se dejará libre el asiento que tenía asignado y se restará en uno el número de abonados.

El abonado también puede darse de baja por su propia voluntad y así su asiento quedará libre.

Para poder comprobar que un asiento está disponible o no, se listan todos los asientos que ya estén ocupados.

El empleado debe poder ver los datos personales de cada abonado a los que accederá a través del número de abonado. Además también se podrá mostrar una lista de todos los abonados junto con su correspondiente número de abonado.

Por último, a cada abonado se le enviará noticias e información acerca de la actualidad del club, donde se incluyen próximos eventos y partidos.

SUBSISTEMA DE LA TIENDA

Amparo Almohalla Santiago

Este subsistema gestiona todo lo relativo a la tienda del club. Para ello, será muy importante el inventario, que no es más que una lista de todos los productos que se ofertan en la tienda junto con la cantidad disponible en el almacén de cada uno.

Hay que destacar la diferencia entre producto e ítem: nos referiremos a “producto” como cualquier “cosa” que se oferte en la tienda y que, por tanto, pueda ser comprada por un cliente. Por otro lado, los ítems son las unidades de las que se dispone de cada producto físicamente, en el almacén. Cada producto estará únicamente determinado por su identificador.

Habrá que asegurarse de que un empleado pueda consultar el inventario, y por tanto pueda saber cuándo es necesario realizar un pedido y evitar que no queden unidades de cualquier producto. Además, hay que tener en cuenta la posibilidad de que se oferten nuevos productos, y haya que registrarlos en el inventario.

Este subsistema se encargará también de la gestión de ventas, ya que cuando un cliente realiza una compra, habrá que modificar el inventario y registrar los nuevos ingresos del club.

Es el único sistema que interactúa con el proveedor, que será el encargado de proporcionar los productos solicitados y, por tanto, de dar respuesta a nuestros pedidos.

REQUISITOS

SUBSISTEMA DE EQUIPO

Requisitos Funcionales y Requisitos de datos

RF1.1: Dar de alta componente: Crear una ficha nueva de un componente del equipo previamente no registrado. Se proporcionan los datos personales del componente, así como la fecha de inicio y vencimiento del contrato.

- Entrada:
 - Agente externo: Empleado.
 - Acción: Solicitud de alta al sistema
 - Requisito de datos de entrada: RDE 1.1.1.
- RDE1.1.1 Datos personales del componente:
 - Puesto que va a ocupar en el club (Cadena de hasta 20 caracteres).
 - DNI(Cadena alfanumérica de 9 dígitos)
 - Nombre (Cadena de hasta 40 caracteres).
 - Teléfono (Cadena de hasta 20 caracteres).
 - Dirección de facturación (Cadena de hasta 40 caracteres).
 - Salario (Decimal de hasta 8 caracteres).
 - Cláusula de rescisión del contrato (Decimal de hasta 20 caracteres).
 - Cuenta bancaria (Cadena alfanumérica de hasta 30 caracteres).

- Fecha de inicio de contrato (Dato de tipo fecha).
 - Fecha fin de contrato (Dato de tipo fecha).
- BD: Requisito de datos de escritura RDW1.1.1
- RDW1.1.1. Se escribe la ficha del componente en la base de datos:
 - Puesto que ocupa en el club (Cadena de hasta 20 caracteres).
 - DNI(Cadena alfanumérica de 9 dígitos)
 - Nombre (Cadena de hasta 40 caracteres).
 - Teléfono (Cadena de hasta 20 caracteres).
 - Dirección de facturación(Cadena de hasta 40 caracteres).
 - Salario (Decimal de hasta 8 caracteres).
 - Cláusula de rescisión del contrato (Decimal de hasta 20 caracteres).
 -
 - Cuenta bancaria (Cadena alfanumérica de hasta 30 caracteres).
 - Fecha de inicio de contrato (Dato de tipo fecha).
 - Fecha fin de contrato (Dato de tipo fecha).
- Salida:
 - Agente externo:Empleado
 - Acción: Confirmación de alta.
 - Requisito de datos de salida: ninguno.

RF1.2: Finalización de contrato: Cuando no se renueva contrato o se finaliza antes de tiempo se da de baja como componente activo del club.

- Entrada:
 - Agente externo: Empleado.
 - Acción: Solicitud de baja de componente
 - Requisito de entrada: RDE 1.2.1.
- RDE 1.2.1: Se introduce el DNI, en caso de rescisión de contrato o venta, la cantidad que abona o recibe el club:
 - DNI (Cadena alfanumérica de 9 dígitos)
 - Cláusula de rescisión del contrato (Decimal de hasta 20 caracteres en negativo si es el club el que paga, y positivo si el club gana dinero)
- BD: Requisitos de datos de lectura RDR 1.2.1, requisito de datos de escritura RDW1.2.1 .
- RDR1.2. Se lee la ficha del componente del DNI indicado:
 - Puesto que ocupa en el club (Cadena de hasta 20 caracteres).
 - DNI(Cadena alfanumérica de 9 dígitos)
 - Nombre (Cadena de hasta 40 caracteres).
 - Teléfono (Cadena de hasta 20 caracteres).
 - Dirección de facturación (Cadena de hasta 40 caracteres).
 - Salario (Decimal de hasta 8 caracteres).
 - Cláusula de rescisión del contrato (Decimal de hasta 20 caracteres).

- Cuenta bancaria (Cadena alfanumérica de hasta 30 caracteres).
 - Fecha de inicio de contrato (Dato de tipo fecha).
 - Fecha fin de contrato (Dato de tipo fecha).
- RDW 1.2.1. Se introduce en BD la baja del componente, así como la cantidad que supone la baja (0 si es simplemente la no renovación del contrato).
- Salida:
 - Agente externo: Empleado del club.
 - Acción: Confirmación de baja del componente.
 - Requisito de datos de salida: ninguno.

RF1.3.Renovar contrato: Se introducen nuevas fechas de finalización y salario para renovar el contrato del componente.

- Entrada:
 - Agente externo: Empleado
 - Acción: Solicitud de renovación de contrato.
 - Requisito de datos de entrada: RDE 1.3.1.
- RDE1.3.1 DNI y nuevos datos del contrato, se compone de:
 - DNI (Cadena alfanumérica de 9 dígitos).
 - Posible nuevo puesto que va a ocupar en el club, si no, se mantiene el mismo puesto que mantenía en el pasado (Cadena de hasta 20 caracteres).
 - Nueva fecha de finalización de contrato (Dato de tipo fecha).
 - Nuevo salario, si lo tuviera (Decimal de hasta 8 caracteres).
- BD: Requisito de datos de lectura RDR1.3.1, requisito de datos de escritura RDW1.3.1.
- RDR1.3.1. Se utiliza el DNI para buscar su ficha en la BD:
 - Puesto que ocupa en el club (Cadena de hasta 20 caracteres).
 - DNI(Cadena alfanumérica de 9 dígitos)
 - Nombre (Cadena de hasta 40 caracteres).
 - Teléfono (Cadena de hasta 20 caracteres).
 - Dirección de facturación (Cadena de hasta 40 caracteres).
 - Salario (Decimal de hasta 8 caracteres).
 - Cláusula de rescisión del contrato (Decimal de hasta 20 caracteres).
 - Cuenta bancaria (Cadena alfanumérica de hasta 30 caracteres).
 - Fecha de inicio de contrato (Dato de tipo fecha).
 - Fecha fin de contrato (Dato de tipo fecha).
- RDW1.3.1. Se escriben en la ficha los nuevos datos asociados al nuevo contrato:
 - Puesto que ocupa en el club (Cadena de hasta 20 caracteres).
 - Salario (Decimal de hasta 8 caracteres).
 - Fecha fin de contrato (Dato de tipo fecha).

- Salida:
 - Agente externo: Empleado
 - Acción: Confirmación de renovación de contrato.
 - Requisito de datos de salida: ninguno

RF1.4. Listar componentes: Muestra un listado de componentes

- Entrada:
 - Agente externo: Empleado
 - Acción: Solicitud listado de componentes del club.
 - Requisito de datos de entrada: RDE1.4.1
- RDE1.4.1. Se introduce el tipo de componente que se quiere listar (jugadores, todos los empleados, entrenadores, etc...)
- BD: Requisito de datos de lectura: RDR1.4.1
- Salida:
 - Agente externo: Empleado de la dirección del club
 - Acción: Confirmación de resultado
 - Requisito de datos de salida: ninguno
- RDR1.4.1 Datos de los componentes indicados:
 - Se leen las tablas indicadas.

RF1.5 Balance de cuentas: Se leen en base de datos los datos asociados a economía de los distintos subsistemas para calcular un presupuesto global del club con todos ellos. Se leerán las cuentas desde el 1 de enero hasta el 31 de diciembre (balance anual).

- Entrada:
 - Agente externo: Empleado.
 - Acción: Balance de cuentas
 - Requisito de entrada: RDE1.5.1.
- BD: Requisito de datos de lectura RDR1.5.1 y requisito de datos de escritura RDW1.5.1.
- RDR1.5.1: Movimientos económicos del club:
 - Ingresos por número de abonados
 - Fondo económico de la tienda (decimal de hasta 15 caracteres)
 - Gastos por el pago del salario a los componentes
 - Beneficios de la venta de jugadores
 - Gastos por la compra de jugadores
 - Balance general del club del año anterior (cadena numérica decimal de hasta 25 caracteres).
- RDW1.5.1 Se utilizan los gastos e ingresos leídos y el último presupuesto calculado para proporcionar el nuevo presupuesto:
 - Fondo económico del club (Cadena numérica decimal de hasta 25 caracteres).

- Salida:
 - Agente externo: Empleado.
 - Acción: Confirmación de nuevo balance.
 - Requisito de datos de salida: ninguno.

SUBSISTEMA DE EVENTOS

Requisitos Funcionales y Requisitos de datos

RF2.1: Comprar entrada: Elimina la disponibilidad de compra del asiento registrándose como no disponible (un asiento va ligado como máximo a una única entrada, puede haber asientos sin entradas porque estarán reservados para los abonados). Para los partidos de fútbol, los abonados tendrán su asiento ya reservado (asiento registrado como no disponible).

- Entrada:
 - Agente externo: cliente
 - Acción: Compra de entrada
 - Requisito de datos de entrada: RDE2.1.1
- BD: Requisito de datos de escritura: RDR2.1.1, RDW2.1.1
- Salida:
 - Agente externo: cliente
 - Acción: Confirmación resultado
 - Requisito de datos de salida: ninguno
- RDE2.1.1 Datos de entrada de la compra de entrada.
 - Evento: Cadena de caracteres (40)
 - Asiento: Entero ({1,2,..., #(asientos)})
- RDR2.1.1 Lectura de disponibilidad del asiento.
 - Los mismos datos que RDE2.1.1
- RDW2.1.1 Cambiar disponibilidad del asiento.
 - Los mismos datos que RDE2.1.1

RF2.2: Consultar próximos partidos: Muestra el listado de próximos partidos.

- Entrada:
 - Agente externo: usuario
 - Acción: Solicitar listado del calendario de partidos
 - Requisito de datos de entrada: ninguno
- BD: Requisito de datos de lectura: RDR2.2.1
- Salida:
 - Agente externo: usuario
 - Acción: Confirmación de resultado
 - Requisito de datos de salida: RDS2.2.1
- RDR2.2.1 Datos de próximos partidos almacenados. Lista de:
 - Fecha: tipo date
 - Nombre equipo contrincante: Cadena de caracteres (40)

- RDS2.2.1 Listado de partidos almacenados.
 - Los mismos datos que RDR2.2.1

RF2.3: Consultar próximos eventos extradeportivos en el estadio: Muestra el listado de próximos eventos extradeportivos en el estadio (La Taberna de Los Serrano).

- Entrada:
 - Agente externo: usuario
 - Acción: Solicitar listado del calendario de eventos extradeportivos en el estadio
 - Requisito de datos de entrada: ninguno
- BD: Requisito de datos de lectura: RDR2.3.1
- Salida:
 - Agente externo: usuario
 - Acción: Confirmación de resultado
 - Requisito de datos de salida: RDS2.3.1
- RDR2.3.1 Datos de próximos eventos extradeportivos en el estadio almacenados.

Lista de:

 - Fecha: tipo date
 - Nombre evento: Cadena de caracteres (40)
- RDS2.3.1 Listado de partidos almacenados.
 - Los mismos datos que RDR2.3.1

RF2.4: Contactar con empresa: servicio del subsistema para ponerse en contacto con las diferentes empresas que intervienen en la gestión de los eventos en el estadio (seguridad, asistencia sanitaria...)

- Entrada:
 - Agente externo: empleado
 - Acción: enviar mensaje.
 - Requisito de datos de entrada: RDE2.4.1
- BD: Requisito de datos de escritura: RDW2.4.1
- Salida:
 - Agente externo: empleado
 - Acción: Confirmación de envío
 - Requisito de datos de salida: ninguno
- RDE2.4.1 Destinatario.
 - Destinatario: Cadena de caracteres (40)
 - Asunto: Cadena de caracteres (50)
 - Cuerpo: Cadena de caracteres (10000)
- RDW2.4.1 Archivo de mensajes
 - Los mismos datos que RDE2.4.1
 - Fecha tipo date.

RF2.5: Registrar informe del evento: registra el informe del evento.

- Entrada:
 - Agente externo: empleado
 - Acción: registrar informe
 - Requisito de datos de entrada: RDE2.5.1
- BD: Requisito de datos de escritura: RDW2.5.1
- Salida:
 - Agente externo: empleado
 - Acción: Confirmación de registro
 - Requisito de datos de salida: ninguno
- RDE2.5.1 Entrada del informe:
 - Nombre evento: Cadena de caracteres (40)
 - Fecha: tipo date
 - Asistencia: entero ({0,1,...,#(asientos)})
 - Listado de incidencias: cadena de caracteres (10000)
- RDW2.5.1 Registro del informe
 - Los mismos datos que RDE2.5.1

SUBSISTEMA DE ABONADOS

Requisitos Funcionales y Requisitos de datos

RF3.1: Dar de alta abonado: Crear una ficha nueva de abonado no registrado actualmente en el sistema y que haya aportado la cantidad de 150€. El abonado proporciona al sistema un correo electrónico, nombre, apellidos, año de alta, dni, teléfono y asiento. Hay que asegurarse de que el asiento se encuentra disponible. El dni no puede estar repetido. Se almacena en la base de datos los datos personales del abonado y se devuelve en la salida el número de abonado correspondiente. El número de abonados aumenta en uno. Como salida se proporciona su número de abonado correspondiente (es único).

- RS: Un mismo asiento no puede estar ocupado por dos personas a la vez.
- Entrada:
 - Agente externo: Abonado.
 - Acción: Solicitud de alta en el sistema.
 - Requisito de datos de entrada: RDE3.1.1.
- RDE3.1.1 Datos de entrada de alta de abonado
 - Nombre: Cadena de caracteres (20)
 - Apellidos: Cadena de caracteres (40)
 - DNI: Cadena de caracteres(9)
 - Teléfono: Cadena de caracteres (20)
 - Correo electrónico: cadena de caracteres (40)
 - Año alta: cadena numérica (4)
 - Asiento: Entero ({1,2,..., #(asientos)})

- BD: Requisito de datos de escritura RDW3.1.1. Requisito de datos de lectura RDR3.1.1.
- Salida:
 - Agente externo: Abonado.
 - Acción: Confirmación de alta.
 - Requisito de dato de salida: RDS3.1.1.
- RDR3.1.1
 - Asientos ocupados
- RDW3.1.1 Datos personales almacenados de abonado y correspondiente número de abono.
 - Nombre: Cadena de caracteres (20)
 - Apellidos: Cadena de caracteres (40)
 - DNI: Cadena de caracteres(9)
 - Teléfono: Cadena de caracteres (20)
 - Correo electrónico: cadena de caracteres (40)
 - Año alta: Cadena numérica(4)
 - Asiento: Entero ({1,2,..., #(asientos)})
 - Número de abonado
- RDS3.1.1
 - Número de abonado

RF3.2: Dar de baja abonado: El sistema elimina la ficha del abonado cuyo número de abonado se indica en la entrada y deja libre su asiento correspondiente.

- Entrada:
 - Agente externo: Abonado.
 - Acción: Solicitud de baja en el sistema.
 - Requisito de datos de entrada: RDE3.2.1.
- RDE3.2.1 Datos de entrada de baja de abonado
 - Número de abonado .
- BD: Requisito de datos de lectura RDR3.2.1. Requisito de datos de escritura RDW3.2.1, RDW3.2.2.
- RDR3.2.1 Datos del abonado:
 - Número de abonado.
- RDW3.2.1 Se eliminan los datos personales del abonado:
 - Nombre: Cadena de caracteres (20)
 - Apellidos: Cadena de caracteres (40)
 - DNI: Cadena de caracteres(9)
 - Teléfono: Cadena de caracteres (20)
 - Correo electrónico: cadena de caracteres (40)
 - Año alta: Cadena numérica(4)
- RDW3.2.2 Se deja libre el asiento
 - Asiento: Entero ({1,2,..., #(asientos)})
- Salida:
 - Agente externo: abonado.
 - Acción: confirmación resultado.
 - Requisito de datos de salida: ninguno.

RF3.3: Mostrar información de un abonado: El usuario quiere obtener información sobre el historial de un abonado cuyo número de abonado coincide con el proporcionado al sistema.

- Entrada:
 - Agente externo: empleado.
 - Acción: Solicitar información sobre el historial de un abonado.
 - Requisito de datos de entrada RDE3.3.1.
- RDE3.3.1 Número de abonado.
- BD: Requisito de datos de lectura RDR3.3.1.
- RDR3.3.1 Datos del abonado.
 - Número de abonado: Cadena numérica de hasta 10 caracteres
 - Nombre: Cadena de caracteres (20)
 - Apellidos: Cadena de caracteres (40)
 - DNI: Cadena de caracteres(9)
 - Teléfono: Cadena de caracteres (20)
 - Correo electrónico: cadena de caracteres (40)
 - Año alta: Cadena numérica(4)
 - Asiento: Entero ({1,2,..., #(asientos)})
- Salida:
 - Agente externo: empleado.
 - Acción: confirmación resultado.
 - Requisito de datos de salida: RDS3.3.1.
- RDS3.3.1 Datos del abonado almacenado
 - Nombre: Cadena de caracteres (20)
 - Apellidos: Cadena de caracteres (40)
 - DNI: Cadena de caracteres(9)
 - Teléfono: Cadena de caracteres (20)
 - Correo electrónico: cadena de caracteres (40)
 - Año alta: Cadena numérica(4)
 - Asiento: Entero ({1,2,..., #(asientos)})

RF3.4: Mostrar listado de abonados: El usuario quiere obtener una lista con todos los nombres y apellidos de todos los abonados incluyendo el número de abonado.

- Entrada:
 - Agente externo: empleado.
 - Acción: solicitar listado.
 - Requisito de datos de entrada: ninguno.
- BD: Requisito de datos de lectura RDR3.4.1.
- RDR3.4.1:
 - Nombre: Cadena de caracteres (20)
 - Apellidos: Cadena de caracteres (40)
 - Número de abonado
- Salida:
 - Agente externo: empleado.
 - Acción: confirmación resultado.
 - Requisito de datos de salida: RDS3.4.1

- RDS3.4.1: Listado de todos los registros, cada uno de ellos con los mismos datos de RDR3.4.1.

RF3.5: Envío de noticias: El sistema envía noticias con periodicidad de un mes al abonado, estas noticias incluyen información sobre la actualidad del club.

- Entrada:
 - Agente externo: empleado.
 - Acción: envío de noticias.
 - Requisitos de entrada: RDE3.5.1
- RDE3.5.1:
 - Número de abonado
 - Correo electrónico: cadena de caracteres (40)
- BD: Requisito de datos de lectura RDR3.5.1 y RDR3.5.2.
- RDR3.5.1: Los mismos datos que en RDR2.2.1.
 - Fecha(prox partidos): tipo date
 - Nombre equipo contrincante: Cadena de caracteres (40)
- RDR3.5.2: Los mismos datos que en RDR2.3.1.
 - Fecha(prox eventos): tipo date
 - Nombre evento: Cadena de caracteres(40)
- Salida:
 - Agente externo: abonado.
 - Acción: confirmación resultado.
 - Requisito de datos de salida: RDS3.5.1.
- RDS3.5.1. Los mismos datos que RDR3.5.1 y RDR3.5.2.
 - Fecha(prox partidos): tipo date
 - Nombre equipo contrincante: Cadena de caracteres (40)
 - Fecha(prox eventos): tipo date
 - Nombre evento: Cadena de caracteres(40)

RF3.6: Renovar abonado: Siempre que el abonado (previamente registrado en el sistema como tal) aporte 150€ en el mes indicado, se le renovará y será durante una temporada más, abonado. Si el abonado no indica un asiento se le asignará el mismo que tenía la temporada anterior. Por otra parte si el abonado no renueva dentro del período indicado, se le dará de baja automáticamente dejando libre su asiento.

- Entrada:
 - Agente externo: abonado.
 - Acción: renovar ficha de abonado.
 - Requisitos de entrada: RDE3.6.1.
- RDE3.6.1:
 - Número de abonado
 - Asiento: Entero ({1,2,..., #(asientos)})
- BD: Requisito de datos de lectura RDR3.6.1. Requisito de datos de escritura RDW3.6.1.
- RDR3.6.1:
 - Asientos ocupados
- RDW3.6.1:
 - Asiento: Entero ({1,2,..., #(asientos)})

- Salida:
 - Agente externo: abonado.
 - Acción: confirmación resultado.
 - Requisito de datos de salida: ninguno.

RF3.7: Mostrar asientos ocupados: Se listan todos los asientos que estén asociados a algún abonado.

- Entrada:
 - Agente externo: empleado.
 - Acción: mostrar asientos ocupados.
 - Requisitos de entrada: ninguno.
- BD: Requisito de datos de lectura RDR3.7.1.
- RDR3.7.1 asientos correspondientes de los abonados. Lista de:
 - Asientos Entero ({1,2,..., #(asientos)})
- Salida:
 - Agente externo: empleado.
 - Acción: confirmación resultado.
 - Requisito de datos de salida: RDS3.7.1.
- RDS3.7.1 los mismos datos que en RDR3.7.1:
 - Asientos Entero ({1,2,..., #(asientos)}).

SUBSISTEMA DE TIENDA

RF4.1: Vender ítems: se realiza la venta de uno o varios ítems. Cada venta se describe mediante un identificador de producto y cantidad del producto. No es necesario que el comprador sea abonado para realizar la compra, trataremos con clientes. Antes de la confirmación de venta, habrá que comprobar si el producto se encuentra en stock, y si hay unidades suficientes (si no es así, habrá que realizar un pedido). Tras realizar una venta será necesario modificar el inventario. En concreto, habrá que modificar la cantidad de ítems almacenada de cada producto vendido, restando tantas unidades como haya comprado el cliente. Evidentemente también aumentará el fondo económico de la tienda.

- Entrada:
 - Agente externo: Cliente.
 - Acción: Comprar productos.
 - Requisito de datos de entrada: RDE4.1.1
- RDE4.1.1: Descripción de la compra.
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad (número natural)
- BD: Requisitos de datos de lectura RDR4.1.1 y de escritura RDW4.1.1, RDW4.1.2
- RDR4.1.1: Descripción del producto.
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad (número natural)

- Precio (decimal positivo)
- RDW4.1.1: Confirmación de la compra.
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad (número natural)
- Salida:
 - Agente externo: Cliente.
 - Acción: Confirmación de compra.
 - Requisito de dato de salida: ninguno.

RF4.2: Comprar ítems (realizar pedido): se realiza un pedido de aquellos productos que más se vendan, que estén a punto de agotarse o que no queden en stock. Para realizar un único pedido común tendremos que realizar una lista de los diferentes productos que queremos pedir, donde cada pedido se describe mediante un identificador de producto y cantidad del producto. Tras realizar la compra habrá que modificar el inventario, aumentando la cantidad de cada producto pedido tantas unidades como se hayan comprado. Además, disminuirá el fondo económico de la tienda.

- RS: no se pueden realizar dos pedidos idénticos (mismos productos y misma cantidad de cada uno de ellos) antes de recibir el primero.
- Entrada:
 - Agente externo: Empleado.
 - Acción: Realizar un pedido.
 - Requisito de datos de entrada: RDE4.2.1
- RDE4.2.1: Descripción de la compra. Lista de:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad del producto (número natural)
- BD: Requisito de datos de lectura RDR4.2.1 y requisito de datos de escritura RDW4.2.1.
- RDR4.2.1: Descripción de los productos. Para cada producto:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad (número natural)
 - Precio (decimal positivo)
- RDW4.2.1: Confirmación de la compra. Lista de:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad del producto (número natural)
- Salida:
 - Agente externo: Empleado.
 - Acción: Confirmación de compra.

- Requisito de dato de salida: ninguno.

RF4.3: Dar de alta un producto: incluir en el inventario un nuevo producto (asegurarse de que el nuevo identificador no coincida con ninguno de los ya registrados previamente).

- Entrada:
 - Agente externo: Empleado.
 - Acción: Dar de alta producto.
 - Requisito de datos de entrada: RDE4.3.1
- RDE4.3.1: Descripción del producto:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad (número natural)
 - Precio (decimal positivo)
- BD: Requisito de datos de escritura RDW4.3.1
- RDW4.3.1: Descripción del producto:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad (número natural)
 - Precio (decimal positivo)
- Salida:
 - Agente externo: Empleado.
 - Acción: Confirmación de nuevo producto registrado en el inventario.
 - Requisito de dato de salida: ninguno.

RF4.4: Dar de baja un producto: eliminar del inventario los datos almacenados acerca de un producto (porque no se va a ofertar más), cuyo identificador se proporciona como entrada.

- Entrada:
 - Agente externo: Empleado.
 - Acción: Dar de baja un producto.
 - Requisito de datos de entrada: identificador de producto.
- RDE4.4.1: Identificador del producto:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
- BD: Requisito de datos de lectura RDR4.4.1.
- RDR4.4.1: Descripción del producto:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad (número natural)

- Precio (número positivo)
- Salida:
 - Agente externo: Empleado.
 - Acción: Confirmación de producto eliminado del inventario.
 - Requisito de dato de salida: ninguno.

RF4.5: Mostrar inventario: consultar el inventario almacenado. El inventario es una relación de todos los productos que oferta la tienda del club, junto con la cantidad almacenada en stock de cada producto.

- Entrada:
 - Agente externo: Empleado.
 - Acción: Mostrar inventario.
 - Requisito de datos de entrada: ninguno
- BD: Requisito de datos de lectura RDR4.5.1.
- RDR4.5.1: Inventario. Lista de:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad (número natural)
- RDS4.5.1: Inventario. Lista de:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad (número natural)
- Salida:
 - Agente externo: Empleado.
 - Acción: Mostrar inventario.
 - Requisito de dato de salida: RDS4.5.1.

RF4.6: Recibir pedido: el proveedor da respuesta a un pedido realizado anteriormente, y por tanto recibimos nuevos ítems de los productos ofertados.

- Entrada:
 - Agente externo: Proveedor.
 - Acción: Recibir un pedido.
 - Requisito de datos de entrada: RDE4.2.1
- RDE4.2.1: Descripción de la compra. Lista de:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad del producto (número natural)
- BD: Requisito de datos de escritura RDW4.2.1.

- RDW4.2.1: Confirmación de la compra. Lista de:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad del producto (número natural)
- Salida:
 - Agente externo: Empleado.
 - Acción: Confirmación de compra.
 - Requisito de dato de salida: ninguno.

RESTRICCIONES SEMÁNTICAS

RS3.1: Un mismo asiento no puede estar ocupado por dos personas a la vez.

RS4.6: No se pueden realizar dos pedidos idénticos hasta que no se haya recibido el primero.

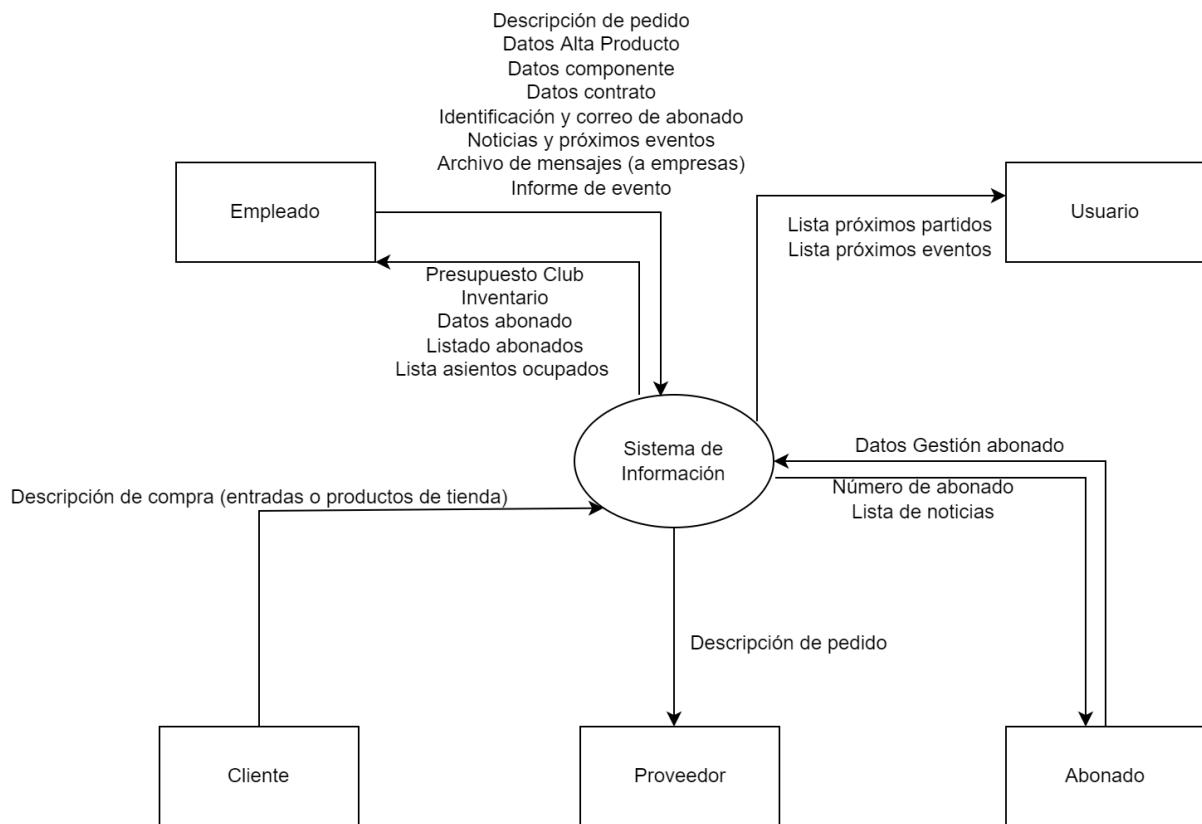
Cambios realizados desde la entrega de la Práctica 1:

1. Dentro de los requisitos del subsistema de abonados, había una incongruencia entre los requisitos “Dar de alta abonado” (3.1) y “Dar de baja abonado” (3.2): dentro del requisito “Dar de alta”, se especificaba que el abonado no podía haber sido registrado previamente, mientras que en “Dar de baja” se decidió que se eliminaría la ficha del abonado. En consecuencia, el abonado que se dé de baja no podría volver a darse de alta, ya que previamente había estado registrado, y al darse de baja se eliminaría su ficha, por lo que no podría hacer uso del requisito “Renovar abonado” (3.6). Resolvemos el problema indicando en “Dar de alta” que el abonado no puede estar registrado en el sistema como tal a la hora de darse de alta (sí que puede haber estado registrado como abonado tiempo atrás y después haberse dado de baja, pero no puede estar registrado como abonado a la hora de hacer la inscripción).
2. En el requisito “Dar de baja producto” (4.4) hay que proporcionar como entrada el identificador del producto, para saber a qué producto nos referimos.
3. Falta un requisito semántico: no se pueden realizar dos pedidos idénticos antes de recibir uno de ellos.
4. El requisito “Gestionar economía” (1.5) está mal planteado. Para empezar cambiamos el nombre a “Balance de cuentas”, ya que “Gestionar economía” es demasiado general. El problema básicamente es que habíamos utilizado requisitos de lectura y de escritura para decidir cómo se lleva a cabo el balance de cuentas, cuando la cuestión es solamente describir el problema, no cómo resolverlo. Resumimos todos los requisitos de lectura que habíamos puesto en uno sólo, donde sólo se indican los datos que hay que reunir para calcular el balance (no especificamos cómo calcular dichos datos). Además indicamos que el balance se realizará una vez al año.

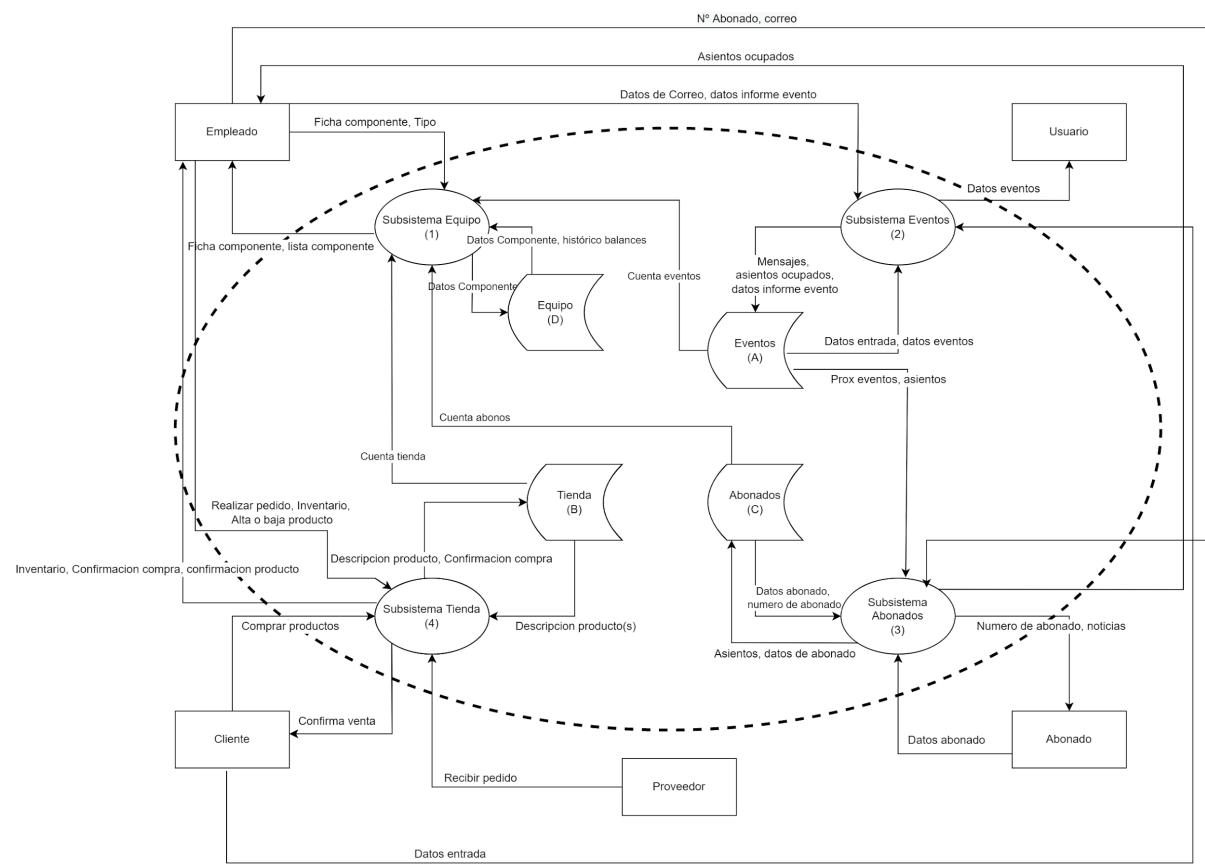
5. En la descripción del subsistema de Equipo, faltaba indicar que existía el requisito “Listar componentes” (1.4).

PRÁCTICA 2 DDSI

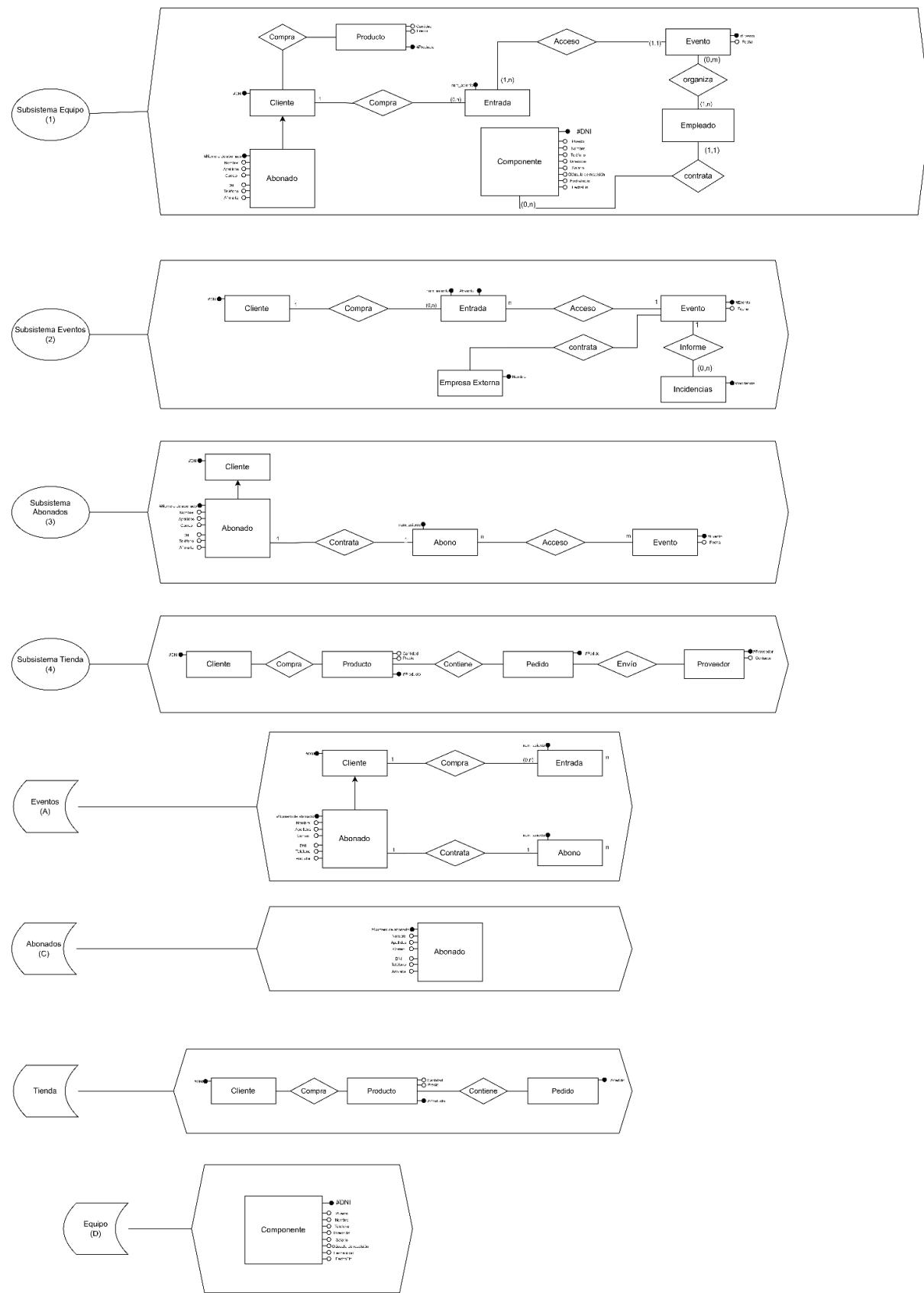
ESQUEMA CAJA NEGRA



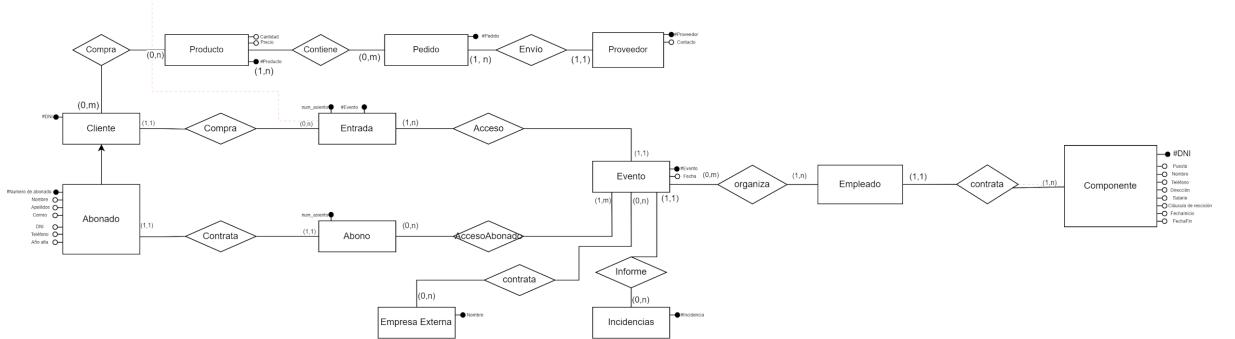
DFD ARMAZÓN O DFD0



ESQUEMAS EXTERNOS DEL ARMAZÓN

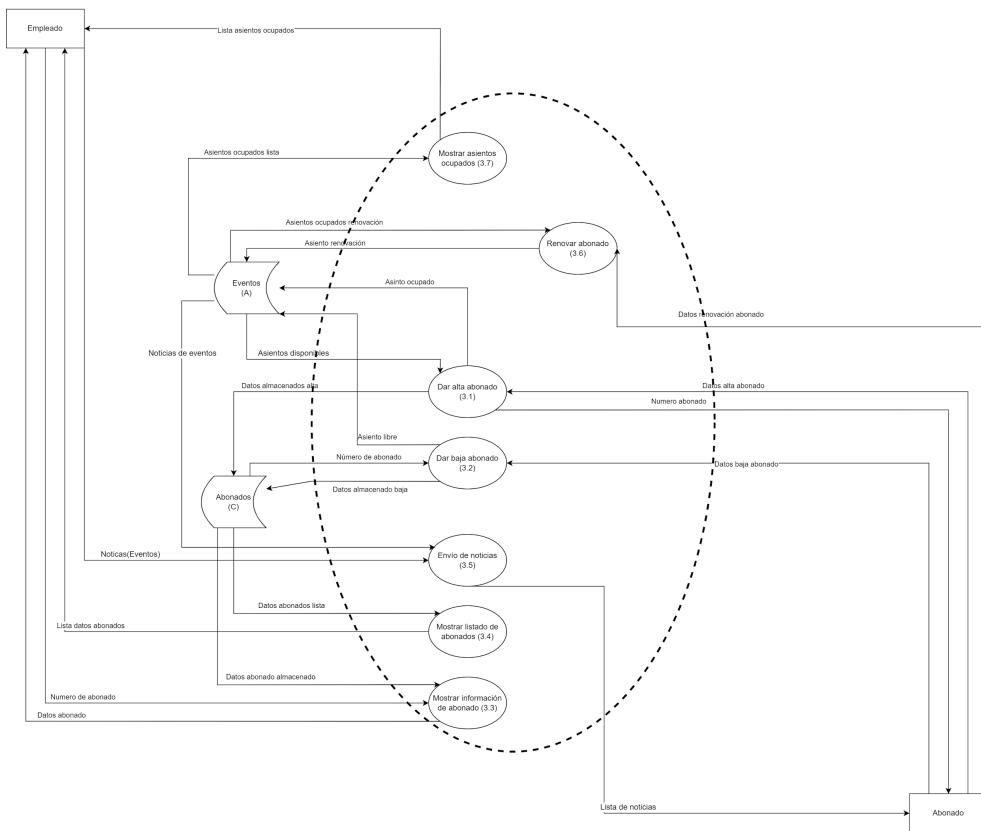


ER ARMAZÓN

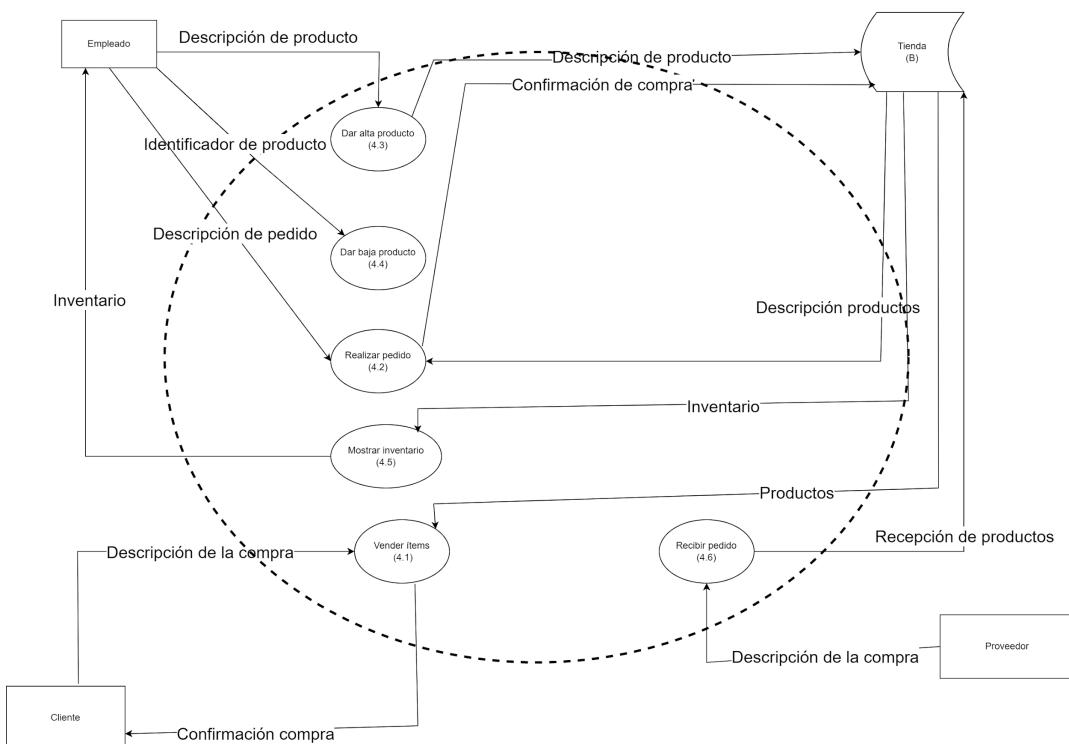


PRIMEROS REFINAMIENTOS PARCIALES

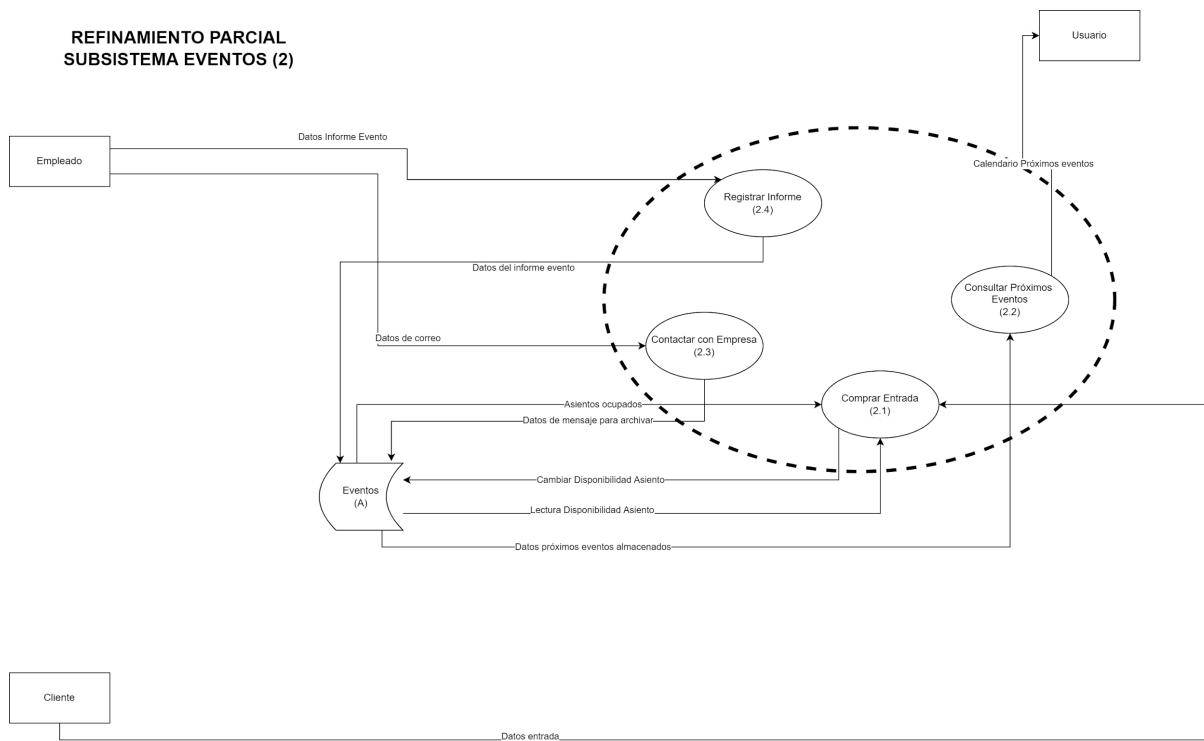
REFINAMIENTO PARCIAL SUBSISTEMA ABONADOS (3)



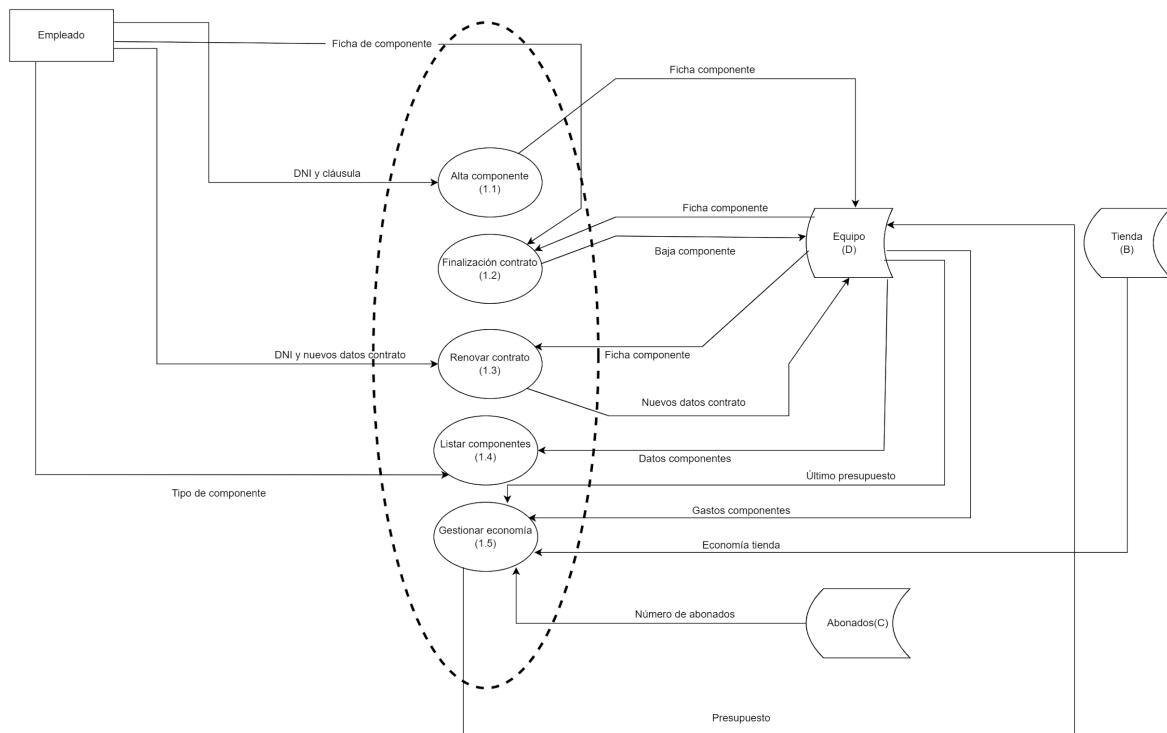
REFINAMIENTO PARCIAL SUBSISTEMA TIENDA (4)



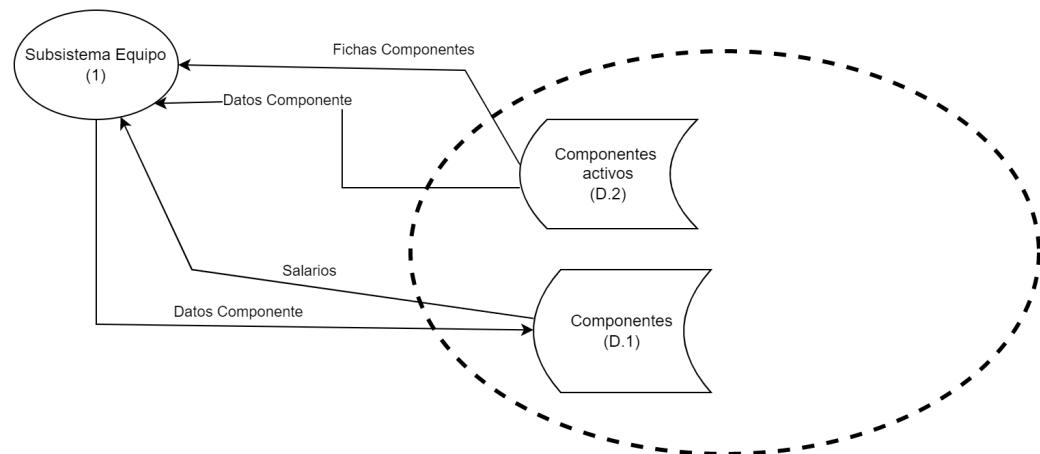
**REFINAMIENTO PARCIAL
SUBSISTEMA EVENTOS (2)**



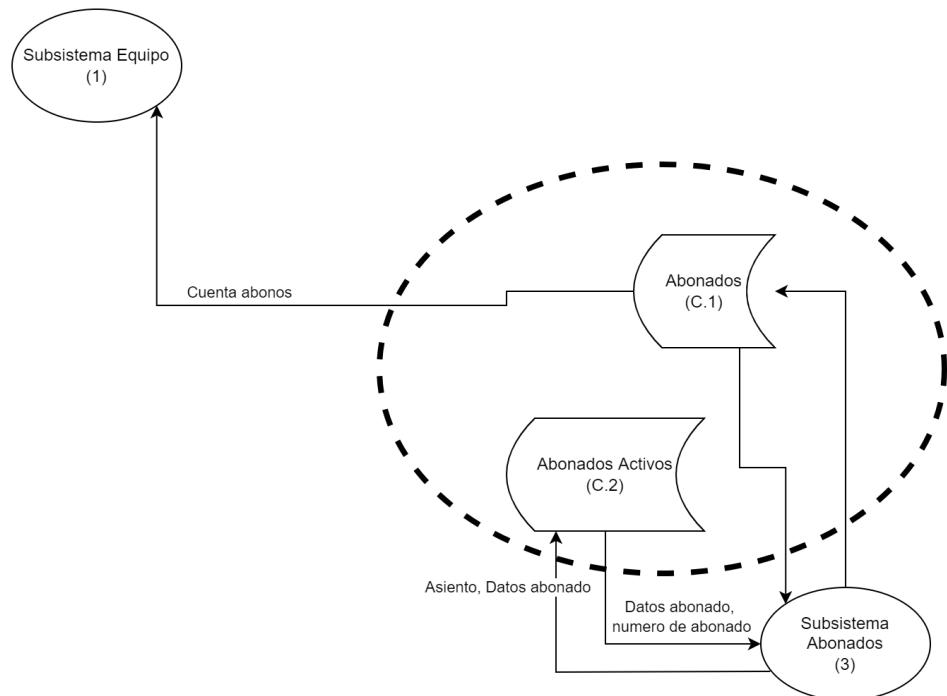
**REFINAMIENTO PARCIAL
SUBSISTEMA EQUIPO (1)**



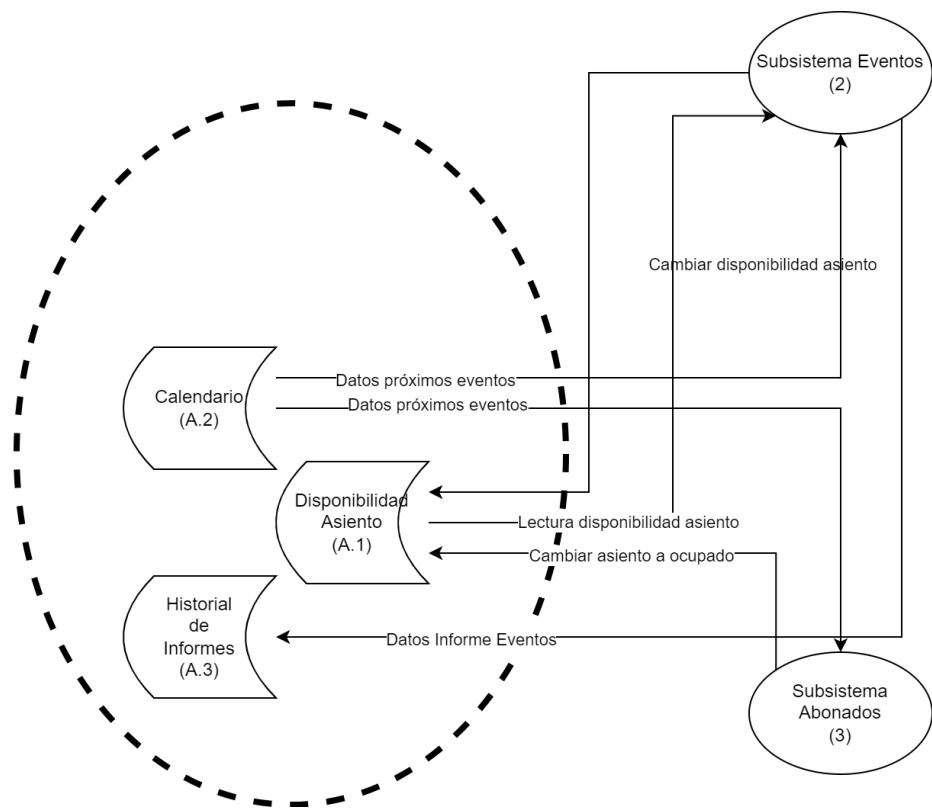
REFINAMIENTO PARCIAL ALMACÉN EQUIPO (D)



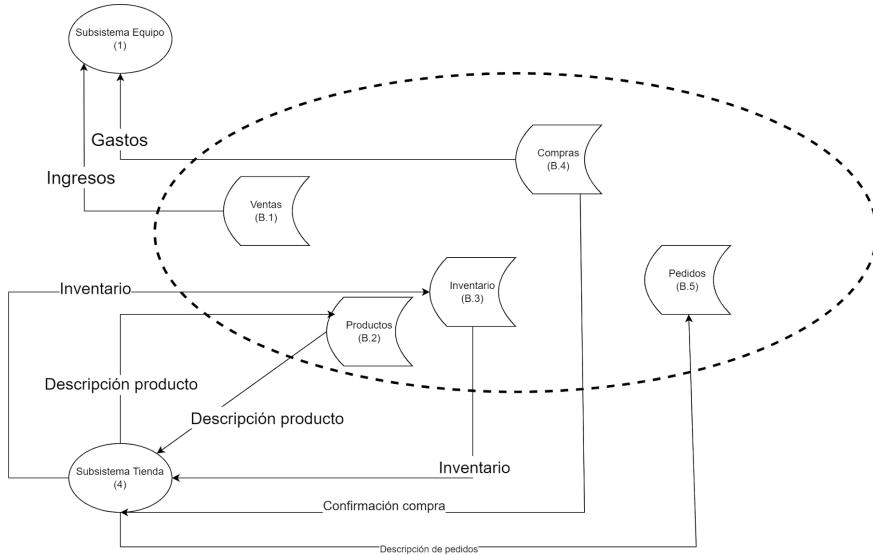
REFINAMIENTO PARCIAL ALMACÉN ABONADOS (C)



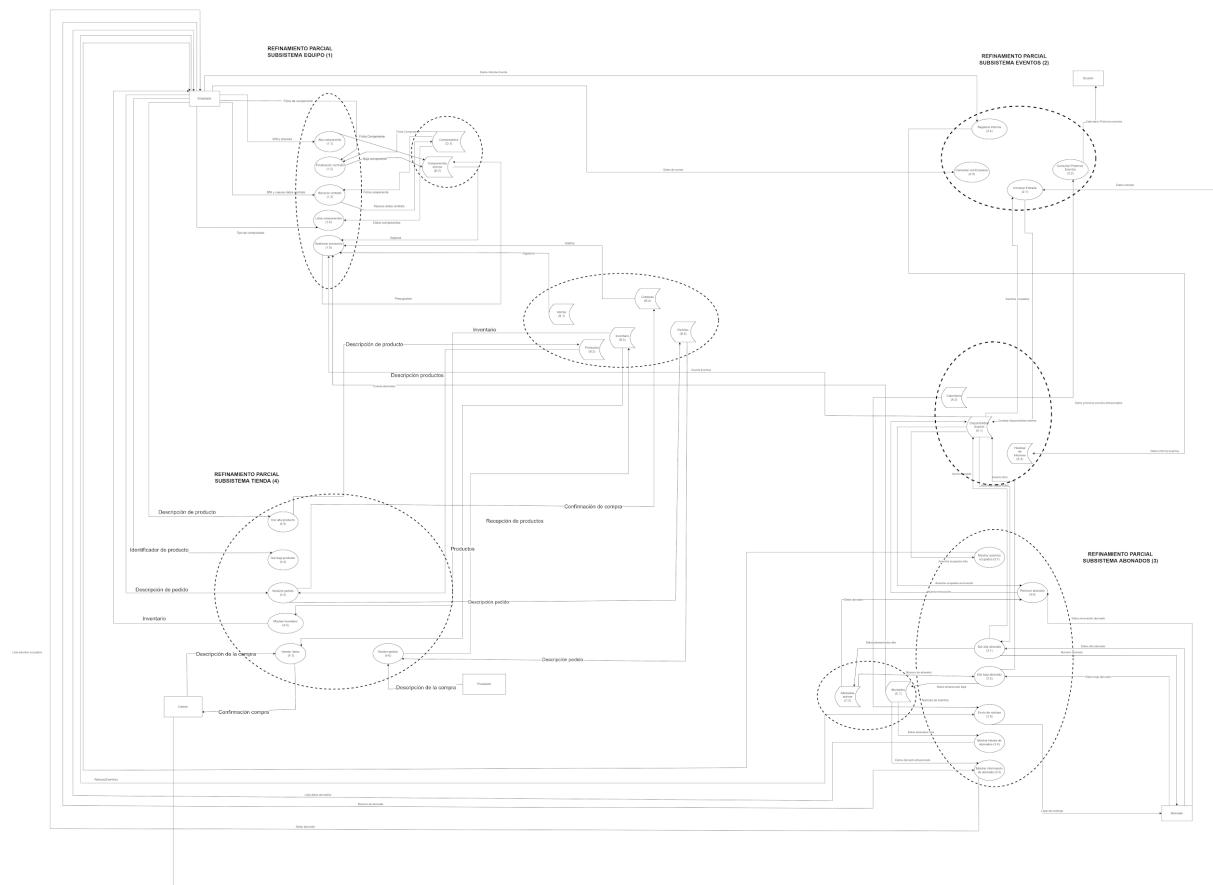
REFINAMIENTO PARCIAL ALMACÉN EVENTOS (A)



REFINAMIENTO PARCIAL ALMACÉN TIENDA (B)

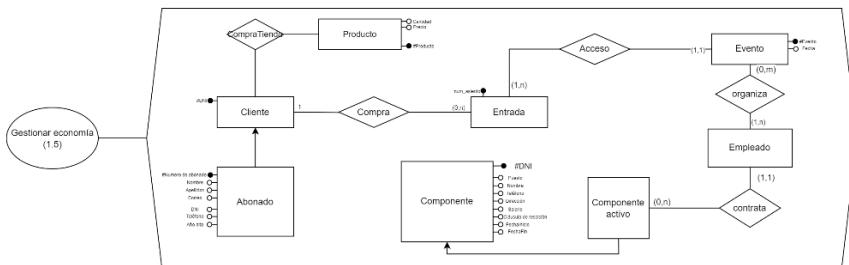
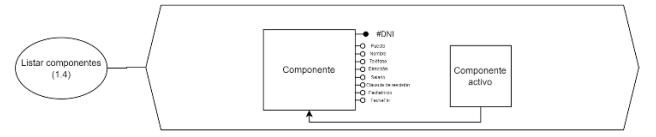
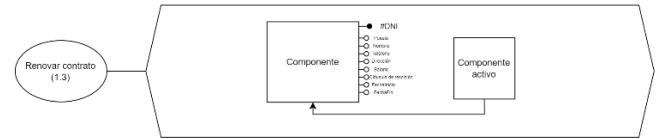
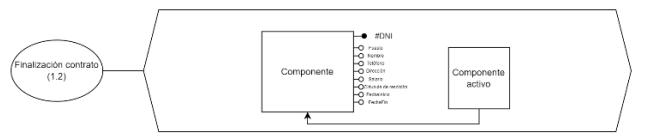
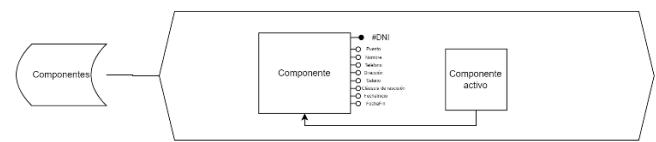
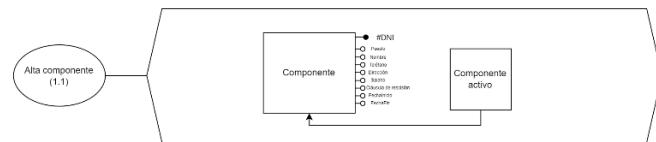


DFD1 COMPLETO

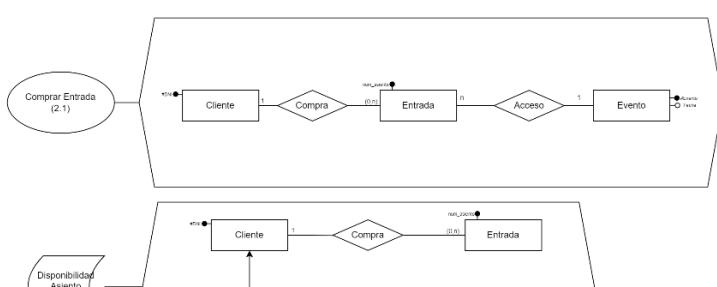


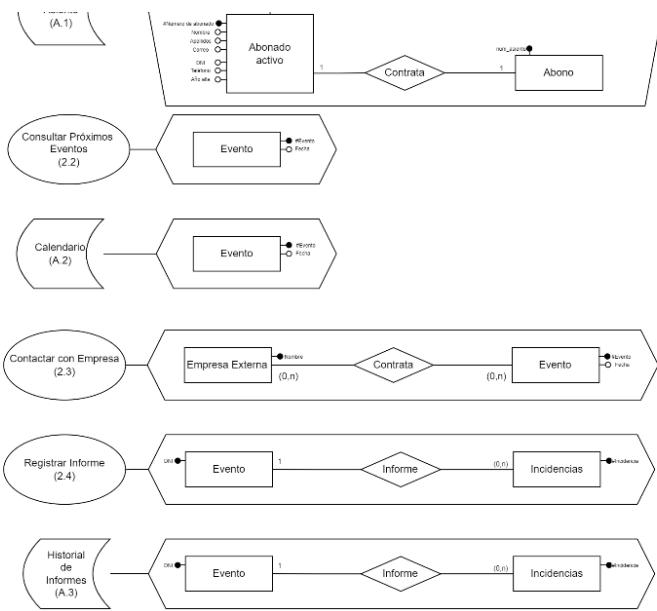
ESQUEMAS EXTERNOS DFD1

SUBSISTEMA EQUIPO (1)

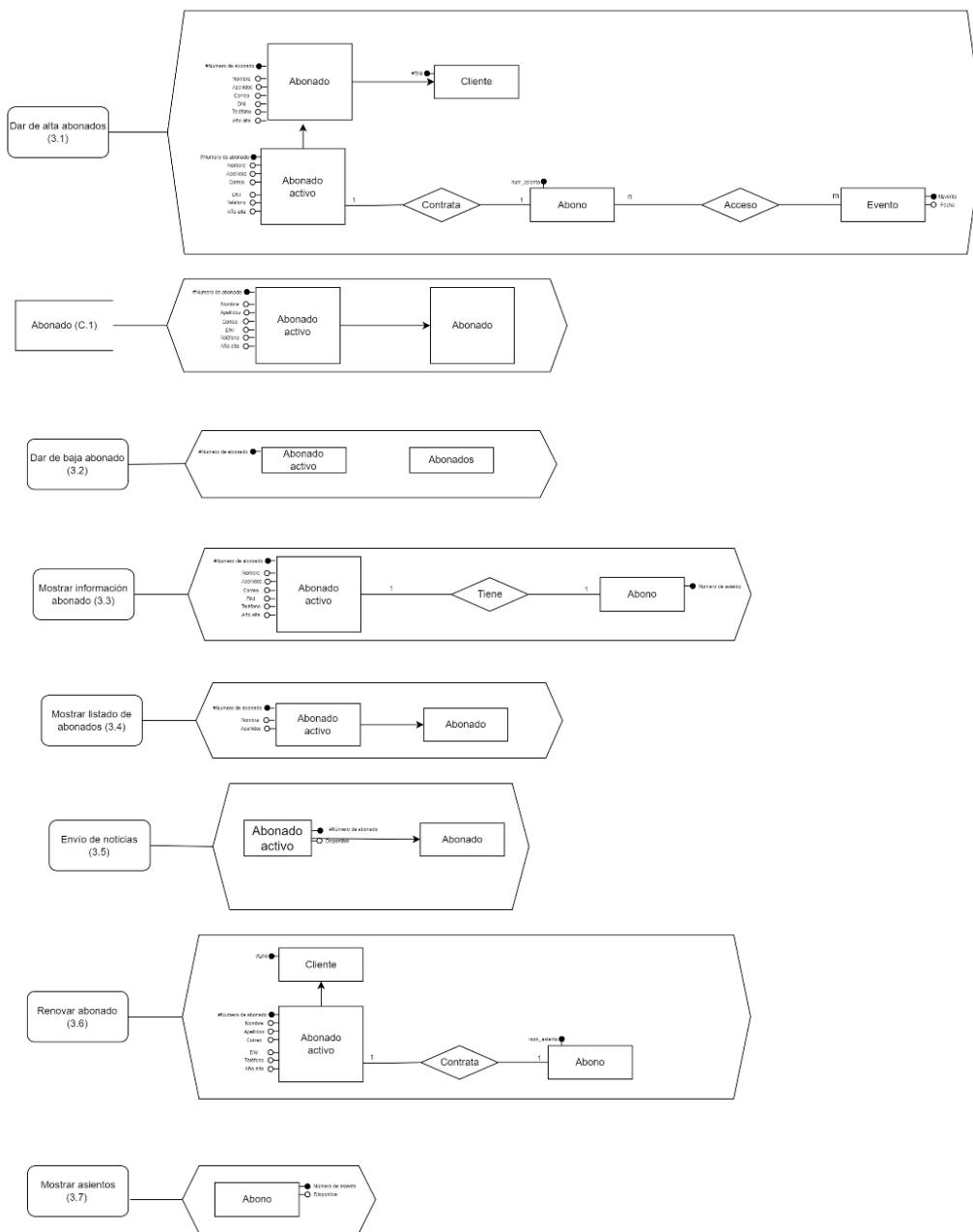


SUBSISTEMA EVENTOS (2)

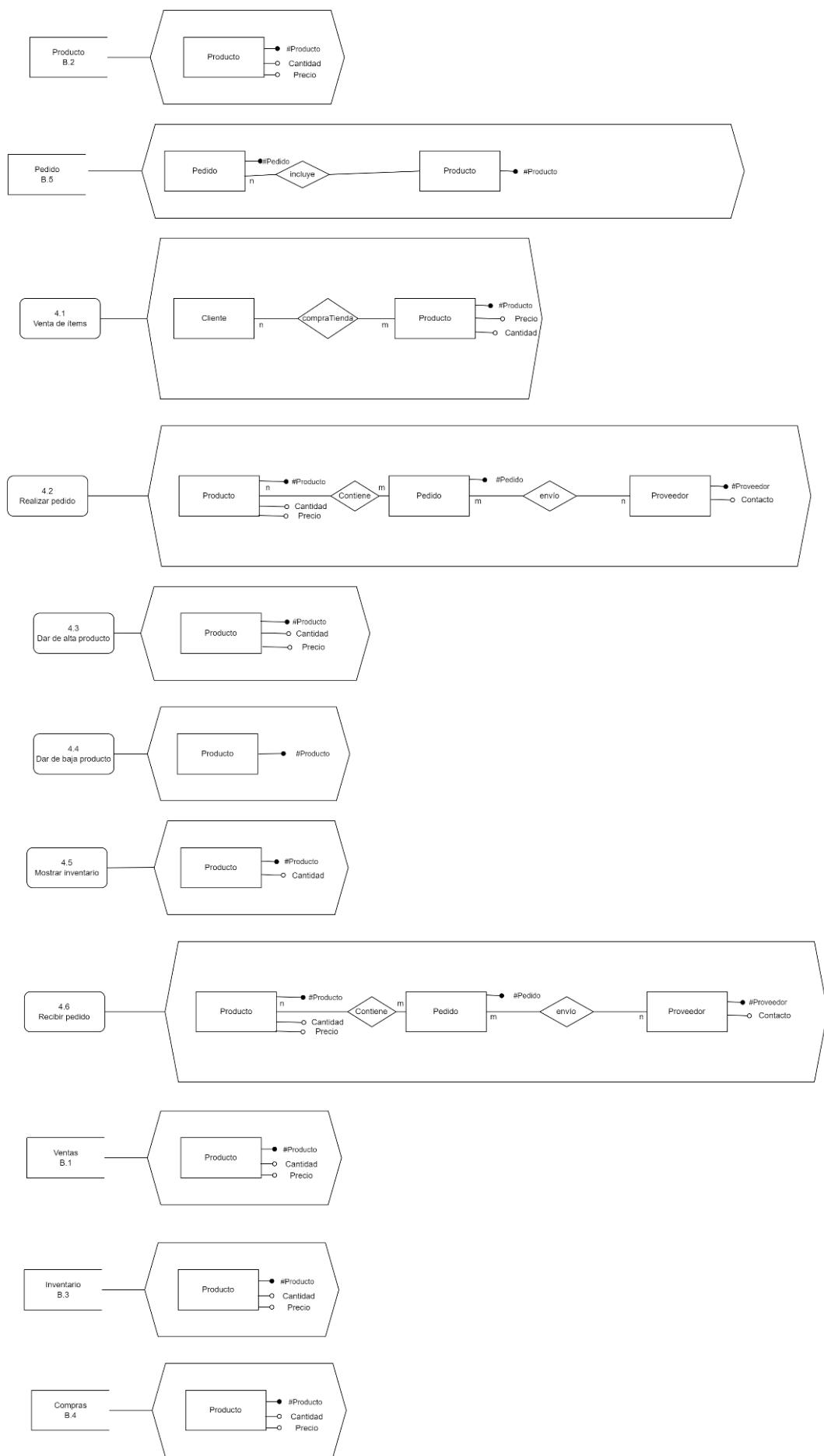




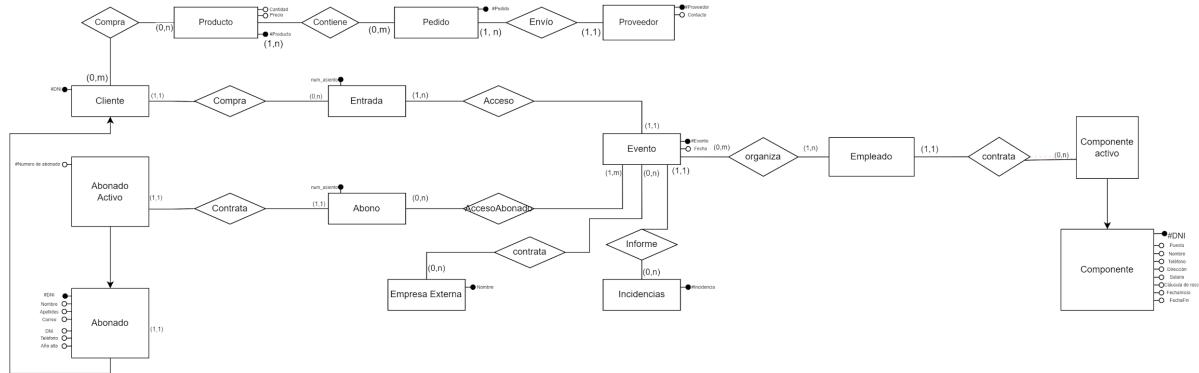
SUBSISTEMA ABONADOS (3)



SUBSISTEMA TIENDA (4)

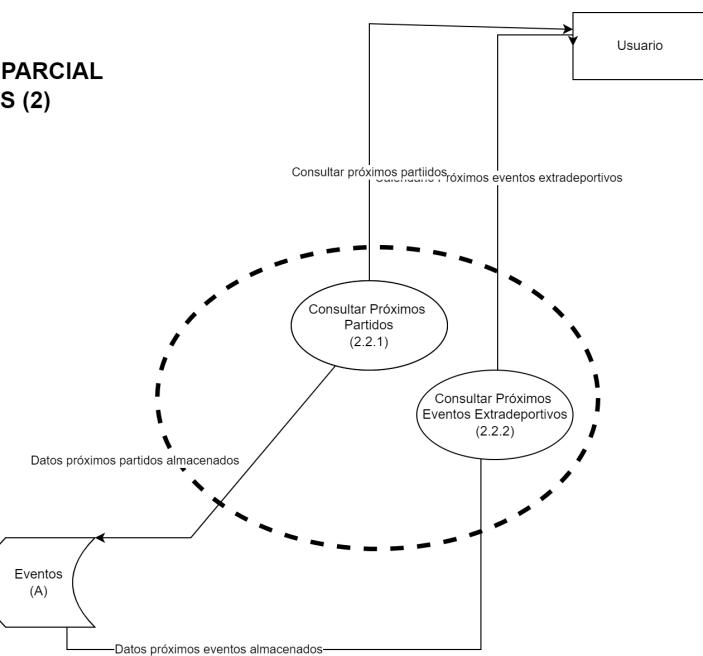


ER1

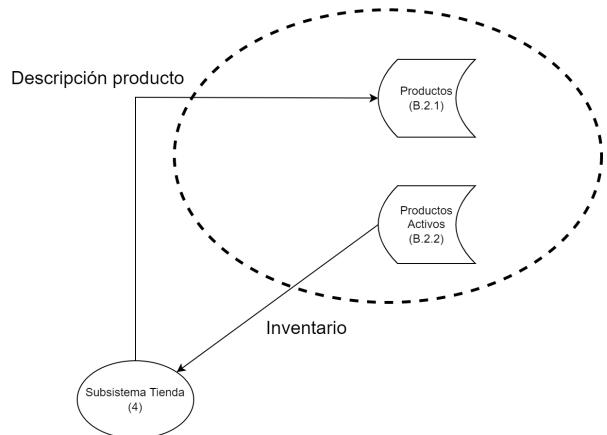


SEGUNDOS REFINAMIENTOS PARCIALES

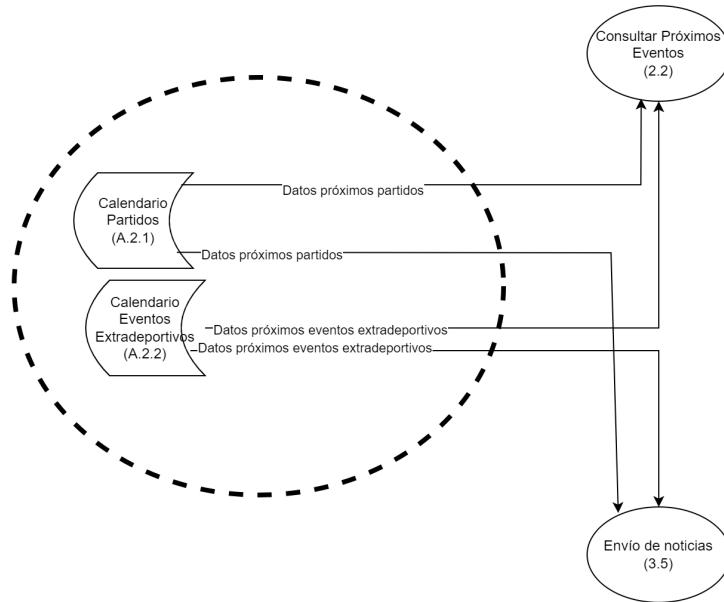
SEGUNDO REFINAMIENTO PARCIAL SUBSISTEMA EVENTOS (2)



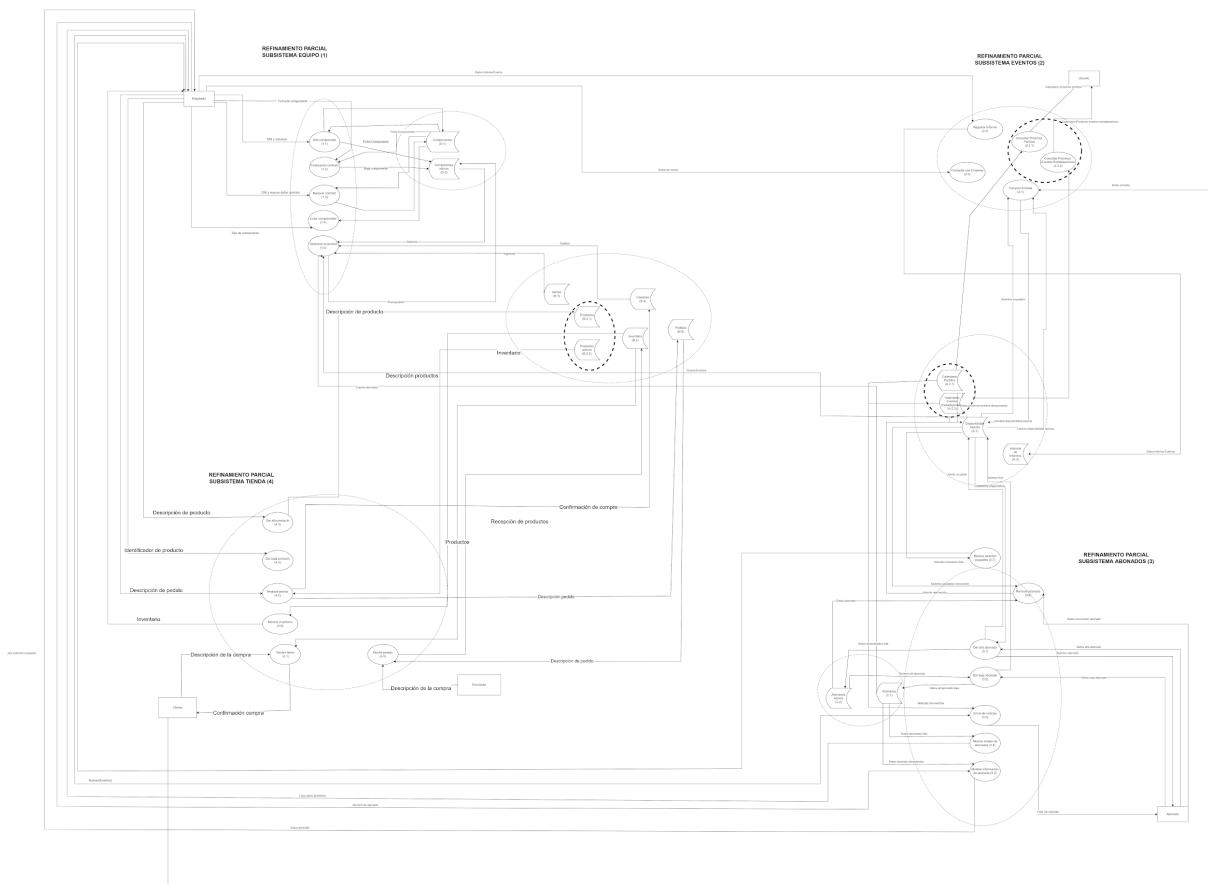
SEGUNDO REFINAMIENTO PARCIAL ALMACÉN TIENDA (B)



REFINAMIENTO PARCIAL ALMACÉN EVENTOS (A)

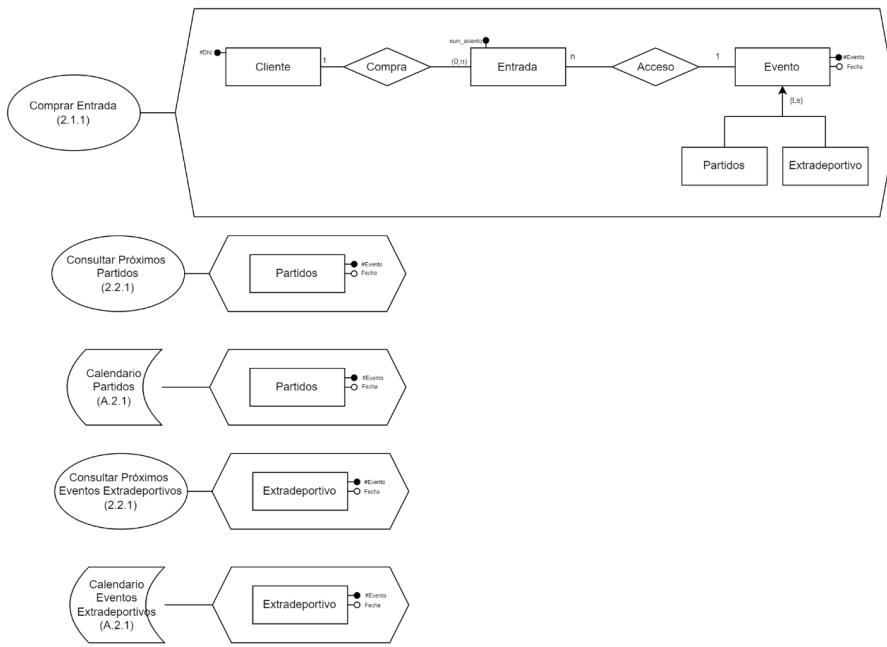


DFD2 COMPLETO

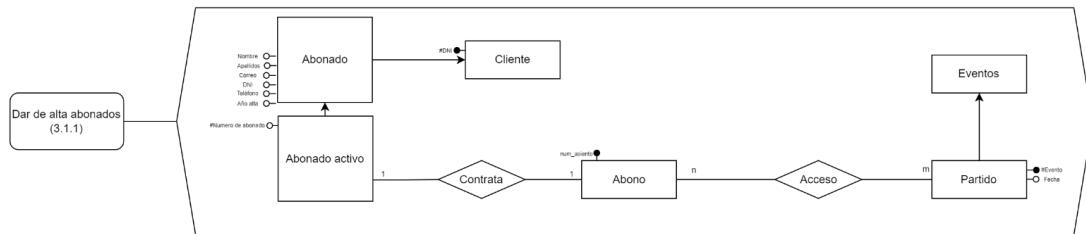


ESQUEMAS EXTERNOS DFD2

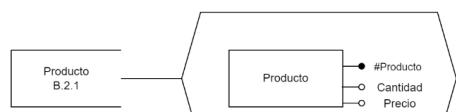
SUBSISTEMA EVENTOS (2)

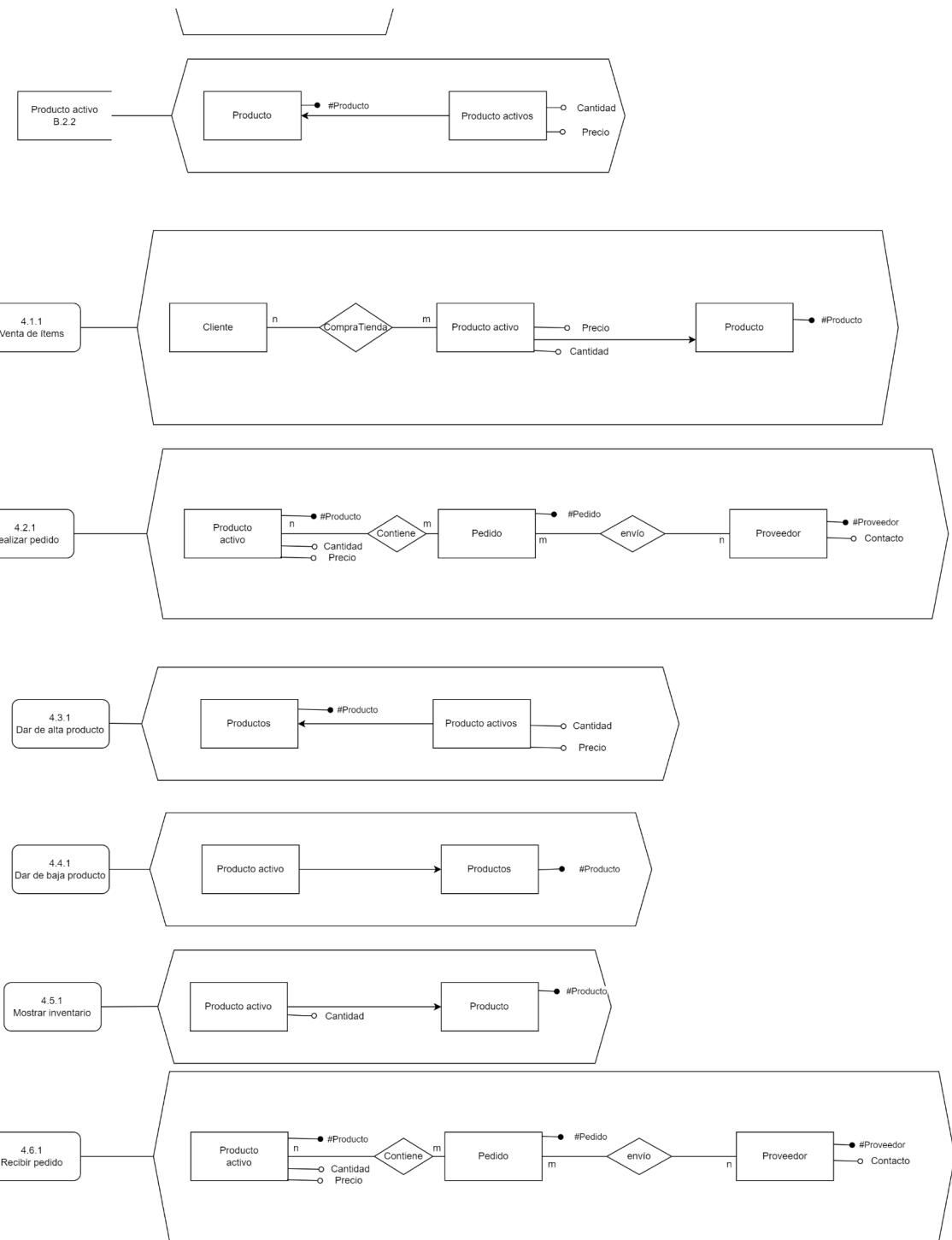


SUBSISTEMA ABONADOS (3)

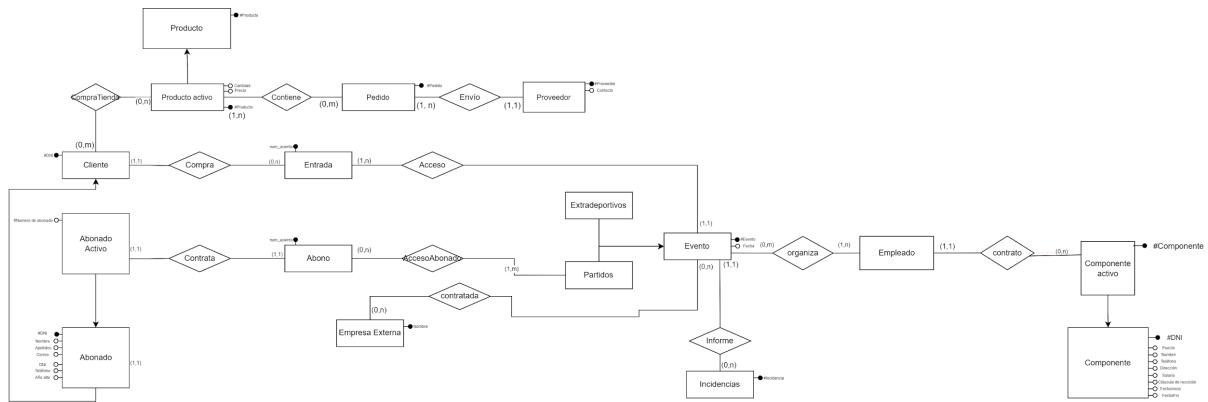


SUBSISTEMA TIENDA (4)

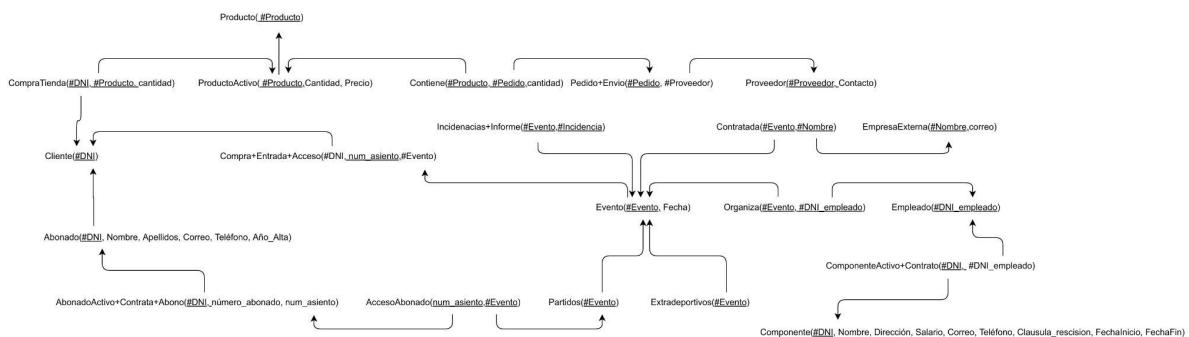




ER2



PASO A TABLAS



El conjunto de tablas obtenidas se ha conseguido a través de la fusión de tablas correspondiente. El proceso de normalización se ha resumido en comprobar que todas las tablas están en Forma normal de Boyce-Codd: viendo que están en FNBC tendríamos que están ya normalizadas de por sí.

Para ello hemos visto que para toda dependencia no trivial se cumple que el atributo alpha (del que dependen los demás) es una superclave. Esto nos ha llevado a comprobar la propiedad de Unicidad para cada instancia de la relación. Efectivamente, en cada tabla, si dos tuplas de dicha instancia coinciden en los atributos de la clave primaria entonces coinciden en el resto de atributos de la relación.

Vemos el proceso de deducción para cada tabla:

Producto: Claves candidatas `{#Producto}`. Clave primaria `{#Producto}`. Está en forma normal de Boyce-Codd porque solo tiene un atributo luego para toda dependencia no trivial `#Producto` es superclave.

ProductoActivo: Claves candidatas {#Producto}. Clave primaria {#Producto}. Las dependencias funcionales son : #Producto -> Cantidad,Precio. Está en FNBC (que implica 3FN y 2FN) porque #Producto es superclave para toda dependencia no trivial. Como todas las dependencias funcionales dependen de #Producto que es nuestra única clave candidata, está en FNBC.

CompraTienda: Claves candidatas {#DNI, #Producto}. Clave primaria {#DNI, #Producto}. Al no tener más atributos solo existen dependencias funcionales triviales. Está en FNBC (que implica 3FN y 2FN) porque #DNI, #Producto es superclave para toda dependencia no trivial ya que no hay dependencias no triviales.

Contiene: Claves candidatas {#Producto, #Pedido}. Clave primaria {#Producto, #Pedido}. Al no tener más atributos solo existen dependencias funcionales triviales. Está en FNBC (que implica 3FN y 2FN) porque #Producto, #Pedido es superclave para toda dependencia no trivial ya que no hay dependencias no triviales.

Pedido+Envio: Claves candidatas {#Pedido}. Clave primaria {#Pedido}. Las dependencias funcionales son : #Pedido -> #Proveedor. Está en FNBC (que implica 3FN y 2FN) porque #Pedido es superclave para toda dependencia no trivial. Como todas las dependencias funcionales dependen de #Pedido que es nuestra única clave candidata, está en FNBC.

Proveedor: Claves candidatas {#Proveedor}. Clave primaria {#Proveedor}. Las dependencias funcionales son : #Proveedor -> Contacto. Está en FNBC (que implica 3FN y 2FN) porque #Proveedor es superclave para toda dependencia no trivial. Como todas las dependencias funcionales dependen de #Proveedor que es nuestra única clave candidata, está en FNBC.

Cliente: Claves candidatas {#DNI}. Clave primaria {#DNI}. Está en forma normal de Boyce-Codd porque solo tiene un atributo luego para toda dependencia no trivial #DNI es superclave.

Compra+Entrada+Acceso: Claves candidatas: {#DNI}, {num_asiento}. Clave primaria {num_asiento}. Las dependencias funcionales son #DNI -> num_asiento y num_asiento->#DNI. Está en FNBC (que implica 3FN y 2FN) porque ambas claves candidatas son superclave para toda dependencia no trivial. Luego está en FNBC

Incidencias+Informe: Claves candidatas: {#Evento}, {#incidencia}. Clave primaria {#incidencia}. Las dependencias funcionales son #Evento -> #incidencia y #incidencia->#Evento. Está en FNBC (que implica 3FN y 2FN) porque ambas claves candidatas son superclave para toda dependencia no trivial. Luego está en FNBC

Contratada: Claves candidatas {#Evento, #Nombre}. Clave primaria {#Evento, #Nombre}. Al no tener más atributos solo existen dependencias funcionales triviales. Está en FNBC (que implica 3FN y 2FN) porque #Evento, #Nombre es superclave para toda dependencia no trivial ya que no hay dependencias no triviales.

EmpresaExterna: Claves candidatas $\{\#Nombre\}$. Clave primaria $\{\#Nombre\}$. Está en forma normal de Boyce-Codd porque solo tiene un atributo luego para toda dependencia no trivial $\#Nombre$ es superclave.

Evento: Claves candidatas $\{\#Evento\}$. Clave primaria $\{\#Evento\}$. Las dependencias funcionales son : $\#Evento \rightarrow \text{Fecha}$. Está en FNBC (que implica 3FN y 2FN) porque $\#Evento$ es superclave para toda dependencia no trivial. Como todas las dependencias funcionales dependen de $\#Evento$ que es nuestra única clave candidata, está en FNBC.

Partidos: Claves candidatas $\{\#Evento\}$. Clave primaria $\{\#Evento\}$. Está en forma normal de Boyce-Codd porque solo tiene un atributo luego para toda dependencia no trivial $\#Evento$ es superclave.

ExtradeporTivos: Claves candidatas $\{\#Evento\}$. Clave primaria $\{\#Evento\}$. Está en forma normal de Boyce-Codd porque solo tiene un atributo luego para toda dependencia no trivial $\#Evento$ es superclave.

Organiza: Claves candidatas $\{\#Evento, \#DNI_Empleado\}$. Clave primaria $\{\#Evento, \#DNI_Empleado\}$. Al no tener más atributos solo existen dependencias funcionales triviales. Está en FNBC (que implica 3FN y 2FN) porque $\#Evento, \#DNI_Empleado$ es superclave para toda dependencia no trivial ya que no hay dependencias no triviales.

Empleado: Claves candidatas $\{\#DNI_Empleado\}$. Clave primaria $\{\#DNI_Empleado\}$. Está en forma normal de Boyce-Codd porque solo tiene un atributo luego para toda dependencia no trivial $\#DNI_Empleado$ es superclave.

ComponenteActivo+Contrato: Claves candidatas $\{\#DNI\}$. Clave primaria $\{\#DNI\}$. Las dependencias funcionales son : $\#DNI \rightarrow \text{DNI_Empleado}$. Está en FNBC (que implica 3FN y 2FN) porque $\#DNI$ es superclave para toda dependencia no trivial. Como todas las dependencias funcionales dependen de $\#DNI$ que es nuestra única clave candidata, está en FNBC.

Componente: Claves candidatas $\{\#DNI\}$. Clave primaria $\{\#DNI\}$. Las dependencias funcionales son : $\#DNI \rightarrow \text{Nombre, Dirección, Salario, Correo, Teléfono, Clausula_rescisión, FechaInicio, FechaFin}$. Está en FNBC (que implica 3FN y 2FN) porque $\#DNI$ es superclave para toda dependencia no trivial. Como todas las dependencias funcionales dependen de $\#DNI$ que es nuestra única clave candidata, está en FNBC.

Abonados: Claves candidatas $\{\#DNI\}$. Clave primaria $\{\#DNI\}$. Las dependencias funcionales son : $\#DNI \rightarrow \text{Nombre, Apellidos, Correo, Teléfono, Año_alta}$. Está en FNBC (que implica 3FN y 2FN) porque $\#DNI$ es superclave para toda dependencia no trivial. Como todas las dependencias funcionales dependen de $\#DNI$ que es nuestra única clave candidata, está en FNBC.

AbonadoActivo+Contrata+Abono: Claves candidatas: $\{\#DNI\}, \{\text{número_abonado}\}$. Clave primaria $\{\#DNI\}$. Las dependencias funcionales son $\#DNI \rightarrow \text{número_abonado, num_asiento}$ y $\#\text{número_abonado} \rightarrow \#\text{DNI, num_asiento}$. Está en FNBC (que implica 3FN y 2FN) porque

ambas claves candidatas son superclave para toda dependencia no trivial. Luego está en FNBC

AccesoAbonado: Claves candidatas $\{\#Evento, num_asiento\}$. Clave primaria $\{\#Evento, num_asiento\}$. Al no tener más atributos solo existen dependencias funcionales triviales. Está en FNBC (que implica 3FN y 2FN) porque $\#Evento, num_asiento$ es superclave para toda dependencia no trivial ya que no hay dependencias no triviales.

PRÁCTICA 3

En este último trabajo de nuestro proyecto tenemos que poner en práctica el diseño realizado durante las prácticas anteriores e implementar todo el sistema en un programa.

Lenguaje utilizado

En nuestro caso hemos decidido utilizar Java para implementar la aplicación y Oracle SQL como SGBD. Esto se debe a que los conocimientos en el resto de trabajos de la asignatura nos han hecho aprender que un lenguaje relacional es lo más indicado y eficiente para la implementación de nuestro sistema. Además, SQL lo utilizamos con más soltura al haberlo trabajado en otras asignaturas anteriormente. Con respecto a la elección de Java como lenguaje de programación, creemos que es el que más se adecúa al programa a desarrollar, además de ser uno de los que mejor integran las bases de datos de tipo SQL. Ahora, vamos a proceder a lo importante.

Además, hemos implementado una interfaz gráfica sencilla con lo aprendido en el Seminario 2:

ABONADOS			
Cdni	Nombre	Apellidos	
7777777A	Amparo	Melano	
7777777B	Pedro	Picapiedra	
7777777C	Antonio	Vignas	
7777777D	Hugo	Lara	

COMPONENTES			
Cdni	Nombre	Direccion	Salario
76544313W	Tomate Tomatito...	Calle La Amarg...	1200
73133133A	Hugo Antonio Te...	Calle Hornillo d...	2000
17122639D	Antonio José Lara	Calle Emperatri...	1500
75334239C	Amparo Almohalla	Plazita	2000

COMPONENTE SEARCH		
Productoactivo		
Search	Cproducto	1001
Cproducto	Cantidad	Precio
Select	15	11,95

Creación de tablas SQL de nuestro Sistema

Estas son las sentencias SQL que hemos implementado en el código de nuestra aplicación para la creación de las tablas:

```
CREATE TABLE Producto(CProducto NUMBER CONSTRAINT CProducto_no_nulo NOT NULL CONSTRAINT CProducto_clave_primaria PRIMARY KEY);
```

```

CREATE TABLE Proveedor(CProveedor NUMBER CONSTRAINT Cproveedor_no_nulo
NOT NULL CONSTRAINT Cproveedor_clave_primaria PRIMARY KEY,Contacto
VARCHAR2(40));

CREATE TABLE Cliente( CDNI VARCHAR2(9) CONSTRAINT Cdni_no_nulo NOT
NULL CONSTRAINT Cdni_clave_primaria PRIMARY KEY);

CREATE TABLE EmpresaExterna(CNombre VARCHAR2(40) CONSTRAINT
Cnombre_no_nulo NOT NULL CONSTRAINT Cnombre_clave_primaria PRIMARY KEY,
Correo VARCHAR2(40) CONSTRAINT correo_empresa_no_nulo NOT NULL);

CREATE TABLE Empleado( CDNI_empleado VARCHAR2(9) CONSTRAINT
Cdniempleado_clave_primaria PRIMARY KEY );

CREATE TABLE Componente(
    CDNI VARCHAR2(9) CONSTRAINT Cdni_componente_no_nulo NOT NULL
CONSTRAINT Cdni_componente_clave_primaria PRIMARY KEY,
    Nombre VARCHAR2(40) CONSTRAINT nombre_no_nulo NOT NULL,
    Direccion VARCHAR2(40) CONSTRAINT direccion_no_nulo NOT NULL,
    Salario NUMBER CONSTRAINT salario_no_nulo NOT NULL,
    Correo VARCHAR2(40) CONSTRAINT correo_no_nulo NOT NULL,
    Telefono VARCHAR2(9) CONSTRAINT telefono_no_nulo NOT NULL,
    Clausula_rescision NUMBER CONSTRAINT clausula_no_nulo NOT NULL,
    FechaInicio DATE, FechaFin DATE
);

CREATE TABLE ProductoActivo(
    CProducto NUMBER CONSTRAINT Cproducto_clave_externa REFERENCES
Producto(CProducto) CONSTRAINT Cproducto_act_clave_primaria PRIMARY
KEY,
    Cantidad NUMBER CONSTRAINT cantidad_produto_no_nulo NOT NULL,
    Precio NUMBER CONSTRAINT precio_produto_no_nulo NOT NULL);

CREATE TABLE CompraTienda(
    CDNI VARCHAR2(9) CONSTRAINT Cdni_clave_externa REFERENCES
Cliente(CDNI),
    CProducto NUMBER CONSTRAINT Cproducto_CT_clave_externa REFERENCES
Producto(CProducto),
    Cantidad NUMBER,
    CONSTRAINT clave_primaria_compraTienda PRIMARY KEY
(CDNI,CProducto))
;

CREATE TABLE Pedido_Envio(
    CPedido NUMBER CONSTRAINT Cpedido_no_nulo NOT
NULL CONSTRAINT Cpedido_clave_primaria PRIMARY KEY,
    CProveedor NUMBER CONSTRAINT
Cproveedor_clave_externa REFERENCES Proveedor(CProveedor));

CREATE TABLE Contiene(

```

```

CProducto NUMBER CONSTRAINT Cproducto_con_clave_externa REFERENCES
ProductoActivo(CProducto),
CPedido NUMBER CONSTRAINT Cpedido_clave_externa REFERENCES
Pedido_Envio(CPedido),
Cantidad NUMBER,
CONSTRAINT clave_primaria_contiene PRIMARY KEY (CProducto,CPedido));

CREATE TABLE Evento(CEvento NUMBER CONSTRAINT Cevento_no_nulo NOT NULL
CONSTRAINT Cevento_clave_primaria PRIMARY KEY, fecha DATE);

CREATE TABLE Compra_Entrada_Acceso(
CDNI VARCHAR2(9) CONSTRAINT
Cdni_ent_acc_clave_extern REFERENCES Cliente(CDNI),
CEvento NUMBER CONSTRAINT Cevento_no_nul NOT
NULL CONSTRAINT Cevento_ent_acc_clave_e REFERENCES Evento(CEvento),
num_asiento NUMBER CONSTRAINT
num_asiento_no_nul NOT NULL,
CONSTRAINT asiento_cevento_clave_primer
PRIMARY KEY (num_asiento, CEvento));

CREATE TABLE Incidencias_Informe(
CEvento NUMBER CONSTRAINT
Cevento_clave_externa REFERENCES Evento(CEvento),
CIcidencia VARCHAR2(100) CONSTRAINT
Cincidencia_no_nulo NOT NULL CONSTRAINT Cincidencia_clave_primaria
PRIMARY KEY);

CREATE TABLE Contratada(
CEvento NUMBER CONSTRAINT Cevento_contr_clave_externa REFERENCES
Evento(CEvento),
CNombre VARCHAR2(40) CONSTRAINT Cnombre_clave_externa REFERENCES
EmpresaExterna(CNombre),
CONSTRAINT clave_primaria_contratada PRIMARY KEY (CEvento,CNombre)
);

CREATE TABLE Organiza(
CEvento CONSTRAINT Cevento_clave_externa_Evento REFERENCES
Evento(CEvento),
CDNI_empleado CONSTRAINT Cdniempleado_Emp_clave_externa REFERENCES
Empleado(CDNI_empleado),
CONSTRAINT clave_primaria_Organiza PRIMARY KEY
(CEvento,CDNI_empleado)
);

CREATE TABLE Abonado(
CDNI VARCHAR2(9) CONSTRAINT Cdni_ab_no_nulo NOT NULL CONSTRAINT
Cdni_ab_clave_primaria PRIMARY KEY REFERENCES Cliente(CDNI),
Nombre VARCHAR2(40) CONSTRAINT nombre_ab_no_nulo NOT NULL,
Apellidos VARCHAR2(40) CONSTRAINT direccion_ab_no_nulo NOT NULL,
Correo VARCHAR2(40) CONSTRAINT correo_ab_no_nulo NOT NULL,
Telefono VARCHAR2(9) CONSTRAINT telefono_ab_no_nulo NOT NULL,

```

```

    Anyo_Alta NUMBER CONSTRAINT num_abonado_no_nulo NOT NULL
);

CREATE TABLE ComponenteActivo_Contrato(
    CDNI VARCHAR2(9) CONSTRAINT Cdniempl_act_clave_primaria PRIMARY KEY
CONSTRAINT Cdni_comp_clave_externa REFERENCES Componente(CDNI),
    CDNI_empleado VARCHAR2(9) CONSTRAINT Cdniempleado_no_nulo NOT NULL
REFERENCES Empleado(CDNI_empleado)
);

CREATE TABLE AbonadoActivo_Contrata_Abono(
    CDNI VARCHAR2(9) CONSTRAINT
Cdni_ab_act_no_nulo NOT NULL CONSTRAINT Cdni_ab_act_clave_primaria
PRIMARY KEY REFERENCES Abonado(CDNI),
    num_asiento NUMBER CONSTRAINT
num_asiento_ab_no_nulo NOT NULL UNIQUE,
    numero_abonado NUMBER CONSTRAINT
num_abonado_ab_no_nulo NOT NULL );
;

CREATE TABLE Extradeportivo(CEvento NUMBER CONSTRAINT
Cevento_ext_clave_primaria PRIMARY KEY CONSTRAINT
Cevento_ext_clave_externa REFERENCES Evento(CEvento));

CREATE TABLE Partido(CEvento NUMBER CONSTRAINT
Cevento_pa_clave_primaria PRIMARY KEY CONSTRAINT
Cevento_pa_clave_externa REFERENCES Evento(CEvento));

CREATE TABLE AccesoAbonado(
    CEvento CONSTRAINT Cevento_acc_clave_externa REFERENCES
Partido(CEvento),
    num_asiento NUMBER CONSTRAINT num_asiento_acc_no_nulo NOT NULL
REFERENCES AbonadoActivo_Contrata_Abono(num_asiento),
    CONSTRAINT clave_primaria_AccesoAbonado PRIMARY KEY
(CEvento,num_asiento)
);
;
```

Así mismo, para trabajar con información sobre cada tabla, añadimos inicialmente algunas tuplas a cada una de ellas:

```

INSERT INTO Componente VALUES ('76544313W', 'Tomate Tomatito López',
'Calle La Amargura 23', 1200, 'pedrogallego@correo.ugr.es','666622554',
7000000, TO_DATE('01/01/2022', 'dd/mm/yyyy'), TO_DATE('26/01/2023',
'dd/mm/yyyy'));
INSERT INTO Componente VALUES ('73133133A', 'Hugo Antonio Teruel',
'Calle Hornillo de Cartuja 11', 2000,
'hugovignas@gmail.com','66804746', 50000, TO_DATE('05/03/2022',
'dd/mm/yyyy'), TO_DATE('30/05/2023', 'dd/mm/yyyy'));
INSERT INTO Componente VALUES ('17122639D', 'Antonio José Lara', 'Calle
Emperatriz Eugenia', 1500, 'antoniojoselp@gmail.com','66444554', 30000,
```

```

TO_DATE('01/11/2022',           'dd/mm/yyyy'),          TO_DATE('30/10/2023',
'dd/mm/yyyy'));  

INSERT INTO Componente VALUES ('75334239C', 'Amparo Almohalla',
'Plazita', 2000, 'amparoalmohalla@gmail.com', '66534554', 30500,
TO_DATE('06/05/2022',           'dd/mm/yyyy'),          TO_DATE('30/04/2023',
'dd/mm/yyyy'));  
  

INSERT INTO Producto VALUES(1000);  

INSERT INTO Producto VALUES(1001);  

INSERT INTO Producto VALUES(1002);  

INSERT INTO Producto VALUES(1003);  
  

INSERT INTO Proveedor VALUES(800, 'proveedorserrano@gmail.com');  

INSERT INTO Proveedor VALUES(801, 'camisetas@gmail.com');  

INSERT INTO Proveedor VALUES(802, 'balonesManolo@gmail.com');  
  

INSERT INTO Cliente VALUES('86543423W');  

INSERT INTO Cliente VALUES('42343423L');  

INSERT INTO Cliente VALUES('99744323Q');  

INSERT INTO Cliente VALUES('77777777A');  

INSERT INTO Cliente VALUES('77777777B');  

INSERT INTO Cliente VALUES('77777777C');  

INSERT INTO Cliente VALUES('77777777D');  
  

INSERT      INTO      EmpresaExterna      VALUES ('Transportes
Paqui', 'paquiempresa@gmail.com');
INSERT      INTO      EmpresaExterna      VALUES ('Seguridad      Paco
SL.', 'pacosecurity@hotmail.com');
INSERT      INTO      EmpresaExterna      VALUES ('Refrigerios
Martinez', 'noreplay@gmail.com');  
  

INSERT INTO Empleado VALUES('76543123W');  

INSERT INTO Empleado VALUES('76544223W');  

INSERT INTO Empleado VALUES('76544434L');  
  

INSERT INTO ProductoActivo VALUES(1000, 20, 15.0);  

INSERT INTO ProductoActivo VALUES(1001, 30, 19.95);  
  

INSERT INTO CompraTienda VALUES('77777777A', 1000, 3);
INSERT INTO CompraTienda VALUES('77777777B', 1001, 10);
INSERT INTO CompraTienda VALUES('77777777C', 1001, 5);
INSERT INTO CompraTienda VALUES('77777777C', 1000, 2);

```

```

INSERT INTO Pedido_Envio VALUES(700, 800 );
INSERT INTO Pedido_Envio VALUES(701, 800 );
INSERT INTO Pedido_Envio VALUES(702, 801 );

INSERT INTO Contiene VALUES(1000,700,35);
INSERT INTO Contiene VALUES(1000,701,3);
INSERT INTO Contiene VALUES(1001,702,5);

INSERT INTO Evento VALUES (500, TO_DATE('26/01/2022', 'dd/mm/yyyy'));
INSERT INTO Evento VALUES (501, TO_DATE('29/01/2022', 'dd/mm/yyyy'));
INSERT INTO Evento VALUES (502, TO_DATE('03/02/2022', 'dd/mm/yyyy'));

INSERT INTO Compra_Entrada_Acceso VALUES ('86543423W', 500, 2003);
INSERT INTO Compra_Entrada_Acceso VALUES ('42343423L', 500, 19532);
INSERT INTO Compra_Entrada_Acceso VALUES ('99744323Q', 501, 10000);

INSERT INTO Incidencias_Informe VALUES (500, 'Ha vomitao un chiquillo');
INSERT INTO Incidencias_Informe VALUES (500, 'Ha habido un incendio');
INSERT INTO Incidencias_Informe VALUES (502, 'Se ha muerto un jugador');

INSERT INTO Contratada VALUES (500, 'Transportes Paqui');
INSERT INTO Contratada VALUES (500, 'Seguridad Paco SL.');
INSERT INTO Contratada VALUES (501, 'Seguridad Paco SL.');

INSERT INTO Organiza VALUES (500, '76543123W');
INSERT INTO Organiza VALUES (502, '76544223W');
INSERT INTO Organiza VALUES (501, '76543123W');

INSERT INTO Abonado VALUES ('77777777A', 'Amparo', 'Melano',
'e.amparoalmo@go.ugr.es', '668822554', 1976);
INSERT INTO Abonado VALUES ('77777777B', 'Pedro', 'Picapiedra',
'e.pedrogallego@go.ugr.es', '732822554', 1998);
INSERT INTO Abonado VALUES ('77777777C', 'Antonio', 'Vignas',
'e.antoniojoselp@go.ugr.es', '687362374', 1977);
INSERT INTO Abonado VALUES ('77777777D', 'Hugo', 'Lara',
'e.hugovignas@go.ugr.es', '623737282', 2000);

INSERT INTO ComponenteActivo_Contrato VALUES('76544313W', '76543123W');
INSERT INTO ComponenteActivo_Contrato VALUES('73133133A', '76544223W');
INSERT INTO ComponenteActivo_Contrato VALUES('75334239C', '76543123W');

```

```

INSERT INTO AbonadoActivo_Contrata_Abono VALUES ('42343423L', 21021,
1);
INSERT INTO AbonadoActivo_Contrata_Abono VALUES ('99744323Q', 17532,
2);

INSERT INTO Extradeportivo VALUES (501);

INSERT INTO Partido VALUES (502);
INSERT INTO Partido VALUES (500);

INSERT INTO AccesoAbonado VALUES (502, 17532);

```

Transacciones

Para las transacciones, hemos seleccionado las siguientes:

SUBSISTEMA DE EQUIPO

- Una transacción para la inserción de uno o más componentes a la vez.
- Una transacción para la baja de un único componente, es decir cuando se le acaba el contrato.
- Para la renovación de un contrato una para cada componente.

SUBSISTEMA DE EVENTOS

- Una transacción para comprar una o más entradas.
- Una transacción cada vez que se realice un registro de incidencia.

SUBSISTEMA DE ABONADOS

- Una transacción para la inserción de uno o más abonados a la vez.
- Una transacción para la baja de un único abonado, es decir cuando se acaba su suscripción.

SUBSISTEMA TIENDA

- Una transacción para cada cliente al vender (el sistema) ítems.
- Una transacción para cada pedido.
- Una transacción para el alta de uno o más productos a la vez.
- Una transacción para la baja de un único producto.
- Una transacción para acusar cada recibo de pedidos.

Disparadores

Hemos implementado los siguientes disparadores:

-No deja asignar un asiento a un nuevo abonado que ya haya sido reservado por otro abonado.

```
CREATE OR REPLACE TRIGGER insertar_abonado
    BEFORE
    INSERT ON AbonadoActivo_Contrata_Abono
FOR EACH ROW
DECLARE asiento_ocupado INTEGER;
    BEGIN SELECT COUNT(*) INTO asiento_ocupado FROM
AbonadoActivo_Contrata_Abono WHERE num_asiento= :new.num_asiento;
IF(asiento_ocupado!=0) THEN RAISE_APPLICATION_ERROR(-20001,'No se
puede asignar dicho asiento al abonado.');
END IF;
END;
```

-No se podrá realizar otro pedido hasta que se haya recibido el último pedido realizado.

```
CREATE OR REPLACE TRIGGER nuevo_pedido
    BEFORE
    INSERT ON Pedido_Envio
FOR EACH ROW
DECLARE encontrado INTEGER;
    BEGIN SELECT COUNT(*) INTO encontrado FROM Pedido_Envio
WHERE CProveedor = :new.CProveedor; IF(encontrado!=0) THEN
RAISE_APPLICATION_ERROR(-20002,'No se puede realizar otro pedido
a dicho proveedor hasta que llegue el pedido actual.');
END IF;
END;
```

-Cuando se inserta una tupla en CompraTienda, se actualiza la nueva cantidad en el stock del producto que se ha vendido.

```
CREATE OR REPLACE TRIGGER actualizarcantidadVentaItems
    AFTER
    INSERT ON CompraTienda
BEGIN
    UPDATE ProductoActivo SET Cantidad=Cantidad-:new.Cantidad WHERE
CProducto=:new.CProducto;
END;
/
```

-Antes de hacer una compra en la tienda, se añade el DNI a la tabla de clientes.

```
CREATE OR REPLACE TRIGGER anadir_cliente_compratienda
    BEFORE
    INSERT ON CompraTienda
FOR EACH ROW
BEGIN
    INSERT INTO Cliente VALUES (:new.CDNI);
```

```

EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('INSERCIÓN FALLIDA.1');

END;
/

```

-Antes de hacer una compra de una entrada, se añade el DNI a la tabla de clientes.

```

CREATE OR REPLACE TRIGGER anadir_cliente_compraentrada
    BEFORE
    INSERT ON Compra_Estrada_Acceso
    FOR EACH ROW
BEGIN
    INSERT INTO Cliente VALUES(:new.CDNI);
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('INSERCIÓN FALLIDA.2');
END;
/

```

-Antes de hacer una compra de un abono, se inserta el abonado como cliente.

```

CREATE OR REPLACE TRIGGER anadir_cliente_abonado
    BEFORE
    INSERT ON Abonado
    FOR EACH ROW
BEGIN
    INSERT INTO Cliente VALUES(:new.CDNI);
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('INSERCIÓN FALLIDA.3');
END;
/

```

-Antes de dar de alta un producto (insertar en productoactivo), se inserta en producto.

```

CREATE OR REPLACE TRIGGER anadir_producto
    BEFORE
    INSERT ON ProductoActivo
    FOR EACH ROW
BEGIN
    INSERT INTO Producto VALUES(:new.CProducto);
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Ya era un producto (no activo)');
END;
/

```