UAI Centro

Trabajo de Campo

**Gestor de eventos**

Profesores: Battaglia, Nicolas; Gamboa, Leonel

Alumno: Pusderkis, Federico

Contenido

[Historial de Revisión 2](#_Toc1547426335)

[G00 Descripción global del producto 3](#_Toc1136671378)

[G01 Propósito 4](#_Toc2086850883)

[G02. Descripción Funcional del Producto y Alcance 4](#_Toc107836775)

[Requisitos funcionales 4](#_Toc375641675)

[G02RF01 Vender un evento 4](#_Toc897003467)

[G02RF02 Organizar evento 5](#_Toc1341261451)

[Alcance 5](#_Toc1572290869)

[G03 Definiciones, Acrónimos y Abreviaciones 6](#_Toc92443256)

[Definiciones 6](#_Toc1666553934)

[Acrónimos 6](#_Toc68114284)

[Abreviaciones 6](#_Toc1581519956)

[G04. Descripción de las personas participantes 6](#_Toc16604117)

[G05. Otros Requisitos del Producto Y documentación 7](#_Toc1357652499)

[Requisitos no funcionales de servicio 7](#_Toc785299630)

[Estándares aplicables 9](#_Toc2039463960)

[Entorno del Usuario a la aplicación 9](#_Toc2026917336)

[Suposiciones y dependencias 9](#_Toc1640467326)

[Otros Requisitos del Producto 9](#_Toc355755105)

[Requisitos de Sistema 9](#_Toc1900176706)

[Manual de usuario 10](#_Toc651342515)

[Guía de instalación 10](#_Toc736088640)

[G06 Diagrama de clases completo de todos los módulos implementados 10](#_Toc201031723)

[G07 Diagrama entidad relación de todas las entidades 11](#_Toc1302951235)

[T01. Arquitectura Base 12](#_Toc220620590)

[T01 Diagrama de componentes 12](#_Toc741208266)

[BE 13](#_Toc404453873)

[GUI 13](#_Toc442429405)

[BLL 13](#_Toc1046241992)

[DAL 13](#_Toc1019511638)

[SharedServices 13](#_Toc1938088192)

[T01 Esquema de persistencia 14](#_Toc1652634975)

[T01 Clase Database 14](#_Toc812025851)

[T01 Mapa tentativo de menús 15](#_Toc1285668073)

[T02 Gestión de Log In / Log Out 16](#_Toc793008491)

[T02CU01 Login 16](#_Toc1683360275)

[Diagrama de casos de uso 16](#_Toc1224532695)

[Descripción del caso de uso. 16](#_Toc296110903)

[Diagrama de secuencia 18](#_Toc211068082)

[T02CU03 Logout 19](#_Toc74980112)

[Descripción del caso de uso 19](#_Toc1477461312)

[Diagrama de caso de uso 20](#_Toc221817879)

[T02 Inicio del sistema 20](#_Toc76107103)

[T02 Diagrama de clases 22](#_Toc848758178)

[T02 Diagrama Entidad Relación 22](#_Toc254013656)

[T03 Gestión del encriptado 23](#_Toc2055703411)

[T03ENC01 Encriptar con hash unidireccional MD5 24](#_Toc18666098)

[Descripción del caso de uso 24](#_Toc1237035307)

[Diagrama de secuencia 25](#_Toc560906827)

[T03ENC02 Algoritmo bidireccional AES 25](#_Toc1040123323)

[Descripción del caso de uso 25](#_Toc1061359830)

[Diagrama de secuencia 26](#_Toc273273801)

[T03ENC03 Encriptar 26](#_Toc826724326)

[Descripción del caso de uso 26](#_Toc1467601498)

[Diagrama de secuencia 27](#_Toc369304424)

[T03ENC04 Desencriptar 27](#_Toc196894788)

[Descripción del caso de uso 27](#_Toc388285306)

[Diagrama de secuencia 28](#_Toc1765161685)

[T03 Diagrama de clases 28](#_Toc67197483)

[T04. Gestión de Perfiles de Usuario 29](#_Toc2127076877)

[T04PER Gestión de Permisos 29](#_Toc1815598024)

[T04PER01 Generar permiso + T04PER02 Generar patente 30](#_Toc925772515)

[T04PER03 Listar permisos 33](#_Toc1017255875)

[T04PER04 Listar Familias 36](#_Toc607204982)

[T04PER05 Configurar familia 39](#_Toc1515869795)

[T04PER06 Cargar familia 42](#_Toc206708154)

[T04USU Gestión de usuarios 45](#_Toc1474169724)

[T04USU01 Generar usuario 46](#_Toc1149430524)

[T04USU02 Listar usuario 49](#_Toc1221901098)

[T04USU03 Desbloquear usuario 49](#_Toc1634853065)

[T04USU04 Bloquear usuario 52](#_Toc738627366)

[T04USU05 Gestionar permiso de usuario 55](#_Toc1390643638)

[T04USU06 Agregar permiso de usuario 58](#_Toc373838246)

[T04USU06 Listar permisos de usuario 61](#_Toc426153018)

[T04USU10 Generar integridad de usuario 62](#_Toc204694176)

[T05 Gestión de múltiples idiomas 63](#_Toc201389407)

[T05.A Objetivo 63](#_Toc1907354325)

[T05.B Descripción detallada de cómo funciona 64](#_Toc1442087334)

[Diagrama de clases 68](#_Toc1338958708)

[Diagrama entidad relación 68](#_Toc1438992162)

[T06 Bitácora y control de cambios 69](#_Toc1935313971)

[T06A Bitácora 69](#_Toc124201925)

[Objetivo 69](#_Toc1061122098)

[Funcionamiento 69](#_Toc1373707197)

[T06BCAM Control de cambios 74](#_Toc1261862260)

[T06BCAM02 Restaurar versión anterior 77](#_Toc2133722956)

[T07. Gestión de Backup / Restore 80](#_Toc1529933468)

[T07BAC01 Generar backup 81](#_Toc1768521638)

[T07BAC02 Restaurar backup 84](#_Toc2131837839)

[T08. Gestión de Dígitos Verificadores 87](#_Toc190015193)

[Digito Verificador Horizontal (DVH) 88](#_Toc247048888)

[Digito Verificador Vertical (DVV) 89](#_Toc1503471694)

[T08DIGH01 Restaurar digito horizontal 90](#_Toc1944138799)

[Historial de revisiones 90](#_Toc145032153)

[T08DIGH01 Restaurar digito vertical 93](#_Toc1020502986)

[Historial de revisiones 93](#_Toc1599471485)

[N0 Procesos de Negocio 96](#_Toc644761292)

[N01 Especificación funcional por proceso de negocio 97](#_Toc13011395)

[N01PN01 Vender eventos privados 97](#_Toc664821658)

[N01PN01 Roles intervinientes 98](#_Toc721126071)

[N01PN01 Especificación funcional por proceso de negocio 98](#_Toc1057617351)

[N01PN01 Modelo conceptual 99](#_Toc1925684794)

[N01PN01 Diagrama de casos de uso 100](#_Toc762838202)

[PN02 Organizar evento 101](#_Toc1872216839)

[N01PN02 Roles intervinientes 102](#_Toc1011235275)

[N01PN02 Diagrama de ECS 103](#_Toc1984568975)

[N01PN02 Diagrama de proceso de negocio 104](#_Toc857915517)

[N01PN02 Diagrama de casos de uso 105](#_Toc368844568)

[N02 Casos de uso del proceso de negocio 106](#_Toc1141824781)

[PN01CU01 Buscar cliente 106](#_Toc1431640765)

[Historial de revisiones 106](#_Toc1092253712)

[PN01CU02 Registrar cliente 109](#_Toc460464834)

[Historial de revisiones 109](#_Toc1239263674)

[PN01CU03 Buscar evento 112](#_Toc641871164)

[Historial de revisiones 112](#_Toc1847800891)

[PN001CU04 Generar evento 114](#_Toc444016824)

[Historial de revisiones 115](#_Toc362557974)

[PN01CU05 Guardar evento 121](#_Toc520833473)

[PN01CU06 Home del vendedor 126](#_Toc318767080)

[Historial de revisiones 127](#_Toc1784974717)

[PN01CU07 Cargar evento 130](#_Toc1580022962)

[Historial de revisiones 130](#_Toc258898214)

[PN02CU01 Asignarse un evento 133](#_Toc737809148)

[Historial de revisiones 134](#_Toc782210582)

[PN02CU02 Registrar cronograma de actividades 137](#_Toc504804061)

[Historial de revisiones 138](#_Toc979050362)

[PN02CU03 Registrar participante 140](#_Toc190271042)

[Historial de revisiones 140](#_Toc953124168)

[PN02CU04 Confirmar participación de proveedor 144](#_Toc1511490089)

[Historial de revisiones 144](#_Toc283833130)

# Historial de Revisión

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Autor | Descripción |
| 05/06/2021 | 1.0 | Chamula | * Definir nombre del proceso de negocio * Rehacer el proceso de negocio * hacer casos de uso del proceso de negocio * Detallar roles intervinientes en el proceso de negocio * Explotar casos de uso (incluye diagrama de secuencia) * Armar un diagrama de clases * Armar DER |
| 4/4/2022 | 2.0 | Leonel | Redefinición del alcance: Se enforca el sistema en la venta de eventos masivos |
| 2/5/2022 | 2.0.1 | Leonel | Se confirma que el proceso de negocio puede ser sensillo, con 1 solo actor si es necesario. El foco tiene que estar puesto en los servicios que garantizan la robustes del sistema. |
|  |  |  |  |
|  |  |  |  |

# 

# G00 Descripción global del producto

La motivación del proyecto es generar un sistema informático que optimice la organización de eventos de gran escala.

El objetivo consta en tener mayor trazabilidad con los distintos elementos de la cadena de valor involucrados en la organización de un evento, a fin de cumplir con los objetivos centrales del negocio:

* Incrementar la cantidad de servicios adicionales contratados por evento
* Tener una mayor información sobre los proveedores que contribuya a mejorar la experiencia de los clientes.
* Conocer la agenda de los lugares online a la hora de cerrar una venta.
* Tener un histórico de los eventos realizados por cada cliente.

# G01 Propósito

Una cadena de salones con sede central en Puerto Madero, Ciudad Autónoma de Buenos Aires, Argentina, nos solicita la generación de un sistema que permita informatizar la organización de eventos, abarcando el proceso de reserva de evento y gestión de proveedores.

El proceso de vender un evento implica que un cliente se acerque a la empresa manifestando su necesidad, el vendedor lo guiara por el proceso de compra para la elección del salón y la contratación de servicios adicionales.

Una vez acordadas las características del evento, el cliente abona una seña y firma el contrato que da inicio a las actividades de organización.

# G02. Descripción Funcional del Producto y Alcance

En esta sección se pretende documentar los requisitos que el sistema debe abarcar para satisfacer la necesidad de negocio del usuario.

## Requisitos funcionales

### G02RF01 Vender un evento

El sistema debe facilitar las herramientas para llevar adelante la gestión de la venta de un evento, esto implica:

Cada vendedor debe tener la capacidad de:

* Registrar los datos de cada cliente que esté interesado en realizar un evento:  
  Identificador fiscal, Nombre y apellido de contacto, teléfono, correo electrónico.
* Consultar histórico de eventos organizados por el cliente en función de su identificador fiscal o correo electrónico.
* Generar eventos, para esto:
  + Detallar el tipo de evento privado, corporativo.
  + Categoría del evento (Presentación, Cumpleaños, Casamiento, Reunión corporativa, Otros)
  + Objetivo propuesto por el cliente en formato texto.
  + Consultar la disponibilidad de los distintos salones.
  + Conocer las características de cada salón: capacidad mínima de participantes, capacidad máxima, descripción, dirección, requisitos especiales para la reserva.
  + Agregar o quitar productos y/o servicios ofrecidos por proveedores con los que ya se tenga un acuerdo comercial preestablecido.
  + Detallar pedidos especiales de los clientes.
  + Registrar pagos realizados por el cliente en los eventos
  + Ofrecer descuentos en la contratación.
* Generar un imprimible del presupuesto donde figure todo lo ofrecido y las políticas de cancelación.

### G02RF02 Organizar evento

Cuando se confirma un evento comienza el proceso de organización, que consiste en planificar las distintas tareas relacionadas con la confección del evento, como pueden ser:

* Confirmar la reserva con los proveedores para asegurar la disponibilidad.
* Dimensionar los recursos humanos para atender al público.
* Diseñar los espacios donde se realizará el evento
* Mantener una comunicación agradable con el cliente a fin de que sume servicios al evento.

## Alcance

Es importante entender que el Sistema Gestor de Eventos no abarcara la totalidad de las actividades involucradas con la organización del evento, citando, por ejemplo:

* No será responsable de la facturación ni contabilidad de la empresa
* No realizara cobros, solamente se podrán registrar a modo informativo.
* La interacción con el sistema SOLAMENTE será con personal propio de la empresa, es decir que ni los proveedores ni los clientes podrán consultar o modificar la información del mismo.
* Las opiniones de los participantes serán recolectadas y cargadas al sistema por los usuarios del mismo.
* Si bien el sistema tendrá un registro, no contará con la gestión de la documentación por parte de los proveedores. Esta será llevada a cabo por fuera del sistema siendo responsabilidad del área de compras mantenerla actualizada.
* Queda completamente por fuera del alcance del sistema el pago de impuestos relacionados con los servicios ofrecidos, como por ejemplo el pago de regalías por la música de los eventos, cálculos de retenciones, IVA entre otros.
* La comunicación y coordinación con proveedores será llevada a cabo por el organizador por fuera del sistema.

# G03 Definiciones, Acrónimos y Abreviaciones

## Definiciones

**LUGAR:** Lugar donde se puede realizar los eventos.

**SERVICIO:** Es un servicio o producto que se pueda agregar a las reservas. Hace referencia a los servicios de valor agregado como la decoración.

**CLIENTE:** Es la persona que contrata el servicio.

**EVENTO:** Son el conjunto de salon + productos adicionales vendidos al cliente.

## Acrónimos

### 

### **Abreviaciones**

**SGE:** Sistema gestor de eventos

**PN:** Proceso de negocio

**CU:** Caso de uso

**LPDP:** Ley de protección de datos personales

# G04. Descripción de las personas participantes

Para un mejor entendimiento de los requisitos, se relevaron los distintos involucrados en el proceso de organización de eventos, resultando los siguientes particpantes.

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | **Responsabilidad** |
| Nicolas Galetti | Es el dueño del emprendimiento, quien puede cubre todas las necesidades y tomar decisiones de negocio. | Define procesos  Se encarga de conseguir personal calificado.  Asiste en las distintas actividades cuando es requerido.  Amplia la cartera de proveedores. |
| Ciro Borges | Es el encargado de la coordinación de los eventos. | Contacta a los proveedores para confirmar disponibilidad.  Asiste a los clientes en el proceso de realización del evento.  Dimensiona la cantidad de personas necesarias para cubrir el evento manteniendo la calidad. |
| Claudia Villanueva | Es la encargada de vender los eventos | Realiza la venta de eventos y los servicios involucrados.  Se encarga de mantener la comunicación con los clientes.  Solicita feedback a los clientes sobre los mismos. |

En la descripción anterior, se puede observar la existencia de diferentes responsabilidades, por consiguiente, roles a cubrir:

|  |  |  |
| --- | --- | --- |
| **Nombre** | Rol | Acceso |
| Nicolas Galetti | Administrador | TODOS |
|  |  |  |
| Claudia Villanueva | Ventas | GestionEventos |
| Ciro Borges | Compras | GestionProveedor  EndozarModificacion |

# G05. Otros Requisitos del Producto Y documentación

## Requisitos no funcionales de servicio

|  |
| --- |
| **RNF01 Login**  El sistema deberá permitir la identificación del usuario mediante su nombre de usuario y password, lo que permita identificar cada una de las acciones que realiza sobre el sistema.  El sistema deberá poder encriptar la contraseña de manera segura.  Por cada intento de inicio de sesión, el sistema guardara una entrada en la bitacora que permita realizar un diagnóstico de lo sucedido.  Se deberá bloquear el usuario si falla al escribir su contraseña 3 veces seguidas.  Cada inicio de sesión exitoso validará que el usuario cuente con información integra. En caso de no validar integridad, deberá ser desbloqueado por un administrador del sistema.  Tanto para usuarios nuevos, como para usuarios que se hayan reiniciado la contraseña, se deberá solicitar un cambio de contraseña obligatorio.  Se deberá permitir unicamente 1 sesión de usuario en simultaneo por cliente.  Como parte de la sesión de usuario, se debera traducir el sistema al idioma configurado por el usuario. |
| **RNF02 Logout**  Cuando un usuario finalice su labor en el sistema, deberá poder cerrar su sesión de usuario.  Se solicita mantener un registro de cuando la sesión finaliza.  Al finalizar la sesión se deben borrar los datos temporales, como son los permisos de usuario. |
| **RNF03 Encriptación** El sistema proveerá un mecanismo Bidireccional de encriptación / desencriptación de datos, que imposibilite la lectura de datos sensibles aun en caso de que por un error se filtrará la base de datos. |
| **RNF03 Digito verificador**  El sistema contará con un algoritmo propio de digito verificador, que será utilizado para mantener la integridad de la información sensible que haya sido persistida en la base de datos.  El digito verificador aplica tanto para validar la integridad vertical como la horizontal. |
| **RNF04 Gestión de perfiles de usuario**  El sistema deberá contar con un módulo de gestión de permisos por usuario que permita agregar permisos individuales o familias de permisos.  También se deberá poder dar de alta permisos y familias de permisos. |
| **RNF05 Múltiples idiomas**  El idioma deberá contar con la posibilidad de ser traducido en tiempo de ejecución a distintos idiomas.  Además, los administradores deberán poder crear, editar y eliminar traducciones. |
| **RNF06 Bitácora**  El sistema mantendrá una bitácora con los eventos que el programador considere relevantes para el diagnóstico de inconvenientes que pudieran surgir.  Cada ingreso en la bitácora tendrá un identificador de usuario, tipo de evento (info, error, debug) y descripción. Se deberá tener en cuenta que para eventos que ocurran antes de la identificación del cliente (login), la bitácora no contendrá la información del mismo. |
| **RNF07 Backup**  El sistema tendrá la capacidad de generar a demanda una copia de seguridad de los datos, persistiéndolos en un archivo “.bkp”.  Este archivo contendrá toda la información de la tablas del sistema, a excepción de la bitácora, que será opcional para el administrador incluirla como parte del backup.  Se deberá tener en cuenta que el sistema incluirá en el archivo información que permita una validación de integridad de los datos incluidos, a fin de poder determinar en el proceso de restauración la validez del mismo. |
| **RNF08 Restore**  El sistema tendrá la capacidad de restaurar sus datos partiendo de una copia de seguridad. Este proceso solo puede ser ejecutado por un usuario con rol de administrador.  Previo a la importación de la copia de seguridad, el sistema valida:   * Que no exista información en la base de datos. * Que el archivo de ".bkp” no este corrupto |
| **RNF09 Instalador**  El sistema contará con un instalador que realicé el setup inicial del sistema, instalando todos los componentes necesarios para que el mismo funcione, y un usuario “admin” para el primer acceso. |

## Estándares aplicables

* El logo del sistema estará visible en la parte superior de la barra lateral de menú, del lado derecho.
* En todo momento estará visible la barra de menús en la parte superior de la interfaz de usuario.

## Entorno del Usuario a la aplicación

En la versión inicial de esta aplicación, el usuario deberá contar con una PC con Windows 10 (o superior), donde pueda instalar la aplicación para comenzar a operar.

Es precondición que el usuario cuente con una base de datos SQL Server 2019 para utilizar la aplicación.

## Suposiciones y dependencias

Dadas las características de la versión inicial del sistema, se asume lo siguiente como condición mínima para el uso del sistema:

* Todos los usuarios contarán con acceso a internet.
* La conexión de internet permitirá establecer conexión con la base de datos centralizada del sistema.
* El sistema se ejecutará dentro del computador de cada usuario, en un entorno Windows.

## Otros Requisitos del Producto

Se especifican los requerimientos mínimos y productos de software a tener previamente instalados para que se ejecute el sistema gestor de eventos en el manual de Instalación del sistema.

## Requisitos de Sistema

El Sistema operativo debe ser Windows 10 Home o Profesional (32 o 64 bits) o Superior.

Deberá tener las últimas actualizaciones instaladas

Deberá poseer Net Core 3.1 o Superior

El ordenador principal debe contar con la capacidad de ejecutar SQL Server 2019

Verificar la sesión del servidor SQL→ validación de usuarios del SQL debe ser por Username y por sesión de Windows.

Se deben poder realizar conexiones a servidores remotos

## Manual de usuario

El sistema contará con un manual de usuario que describa las funcionalidades del mismo.

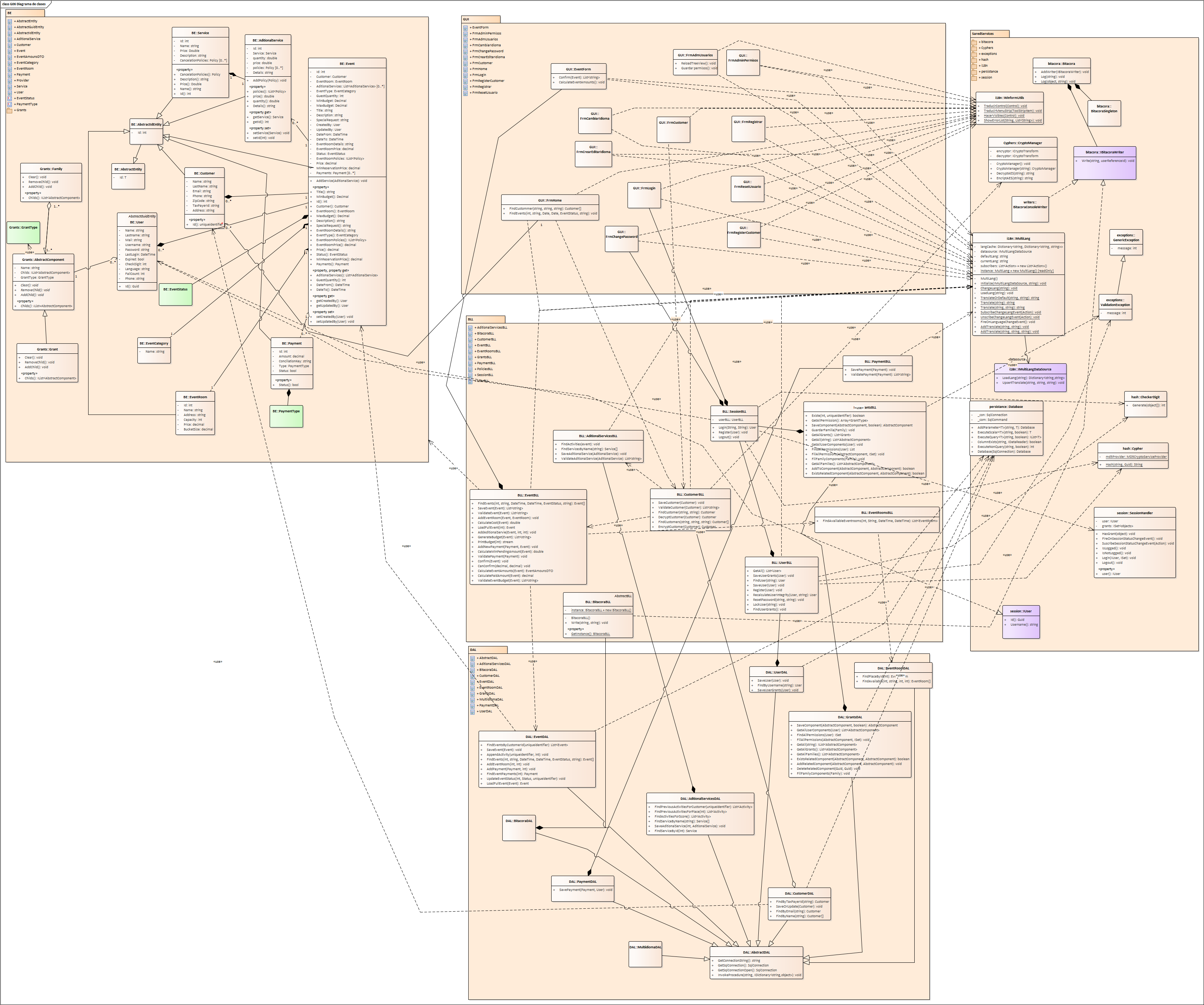
El mismo se podrá acceder desde el disco de instalación, la carpeta dentro del menú inicio o bien desde el menú de ayuda dentro de la aplicación

## Guía de instalación

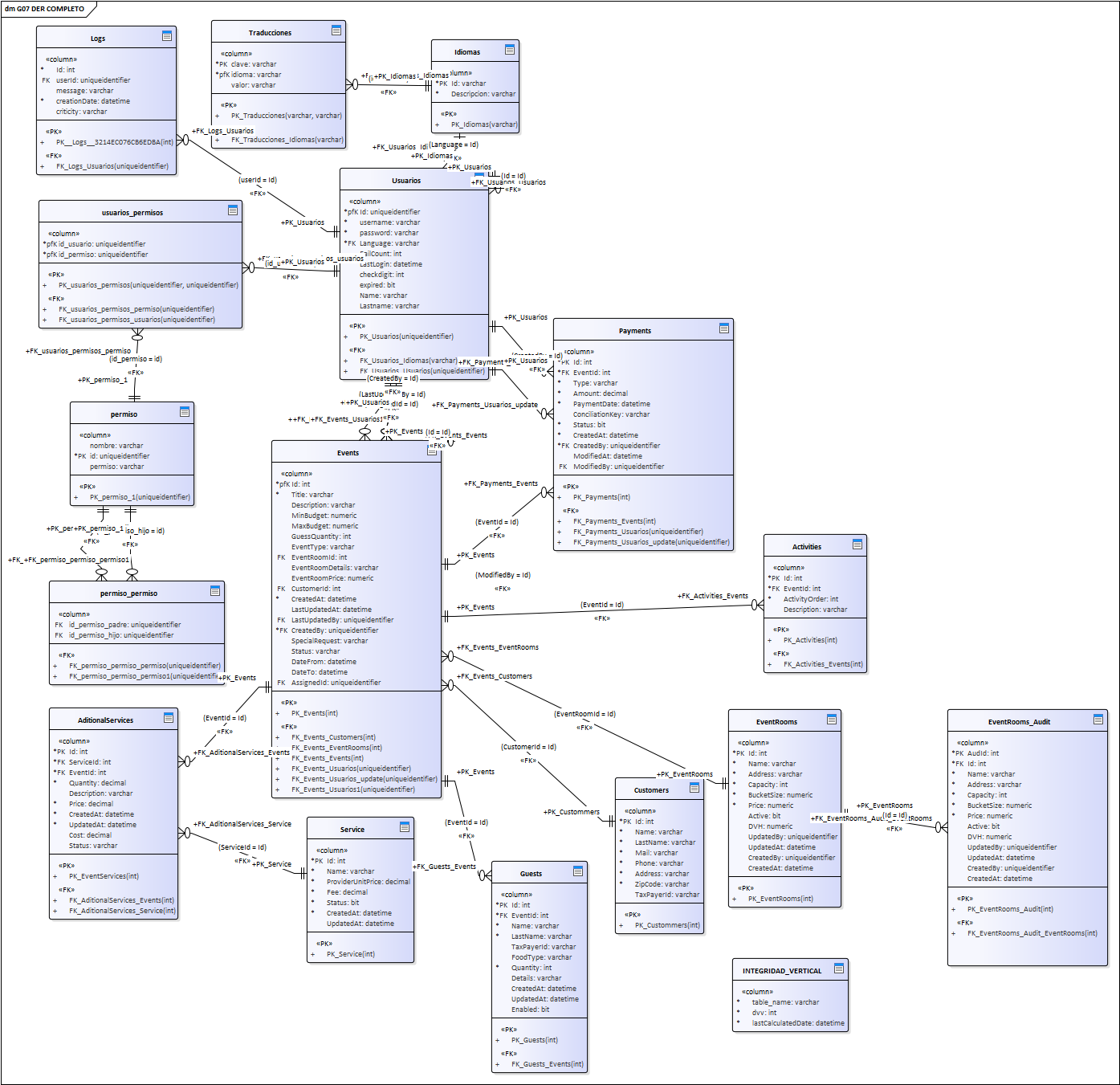
Como parte de la solución productiva se incluye una guía de instalación y configuración del sistema.

Esta misma guia se incluye dentro del manual del sistema accesible en todo momento.

# G06 Diagrama de clases completo de todos los módulos implementados



# G07 Diagrama entidad relación de todas las entidades

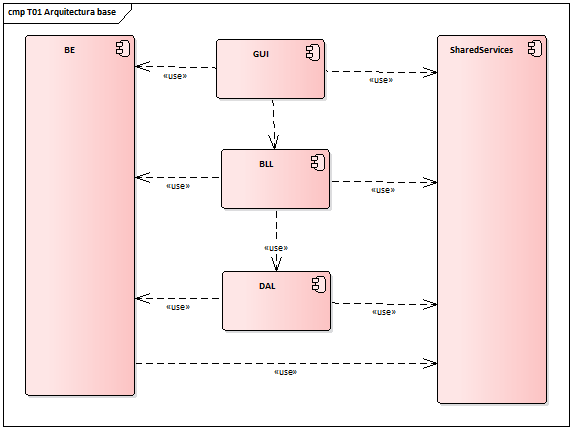


# T01. Arquitectura Base

## 

## T01 Diagrama de componentes

La arquitectura base del sistema estará dividida en 5 capas graficadas a continuación:



### BE

Contendrá todas las entidades del negocio que viajarán de forma transversal a todas las capas del sistema.

El módulo BE usa SharedServices ya que algunas entidades implementan las interfaces, por ejemplo, la BE Usuario implementa la IUsuario de SharedServices, que a posterior se utiliza dentro de la BLL para el inicio de sesión.

### GUI

Contendrá todas las pantallas de usuario.

El usuario solo podrá interactuar con las interfaces provistas en esta capa.

### BLL

Capa encargada de aplicar la lógica de negocio sobre las entidades recibidas de la GUI.

Entre sus responsabilidades se encuentra realizar las validaciones necesarias por cada caso de uso.

### DAL

Capa de acceso a datos, cuya única responsabilidad es manejar la interacción entre el sistema y la base de datos.

### SharedServices

Contiene funcionalidades que pueden ser reutilizadas por cualquier otro sistema.

Un ejemplo es la bitácora, que podrá ser utilizada por cualquier sistema para guardar los sucesos claves que ocurran.

Este módulo se describirá más adelante en este mismo documento.

## T01 Esquema de persistencia

La persistencia se dará respetando la arquitectura del sistema, pasando por cada capa y su responsabilidad.

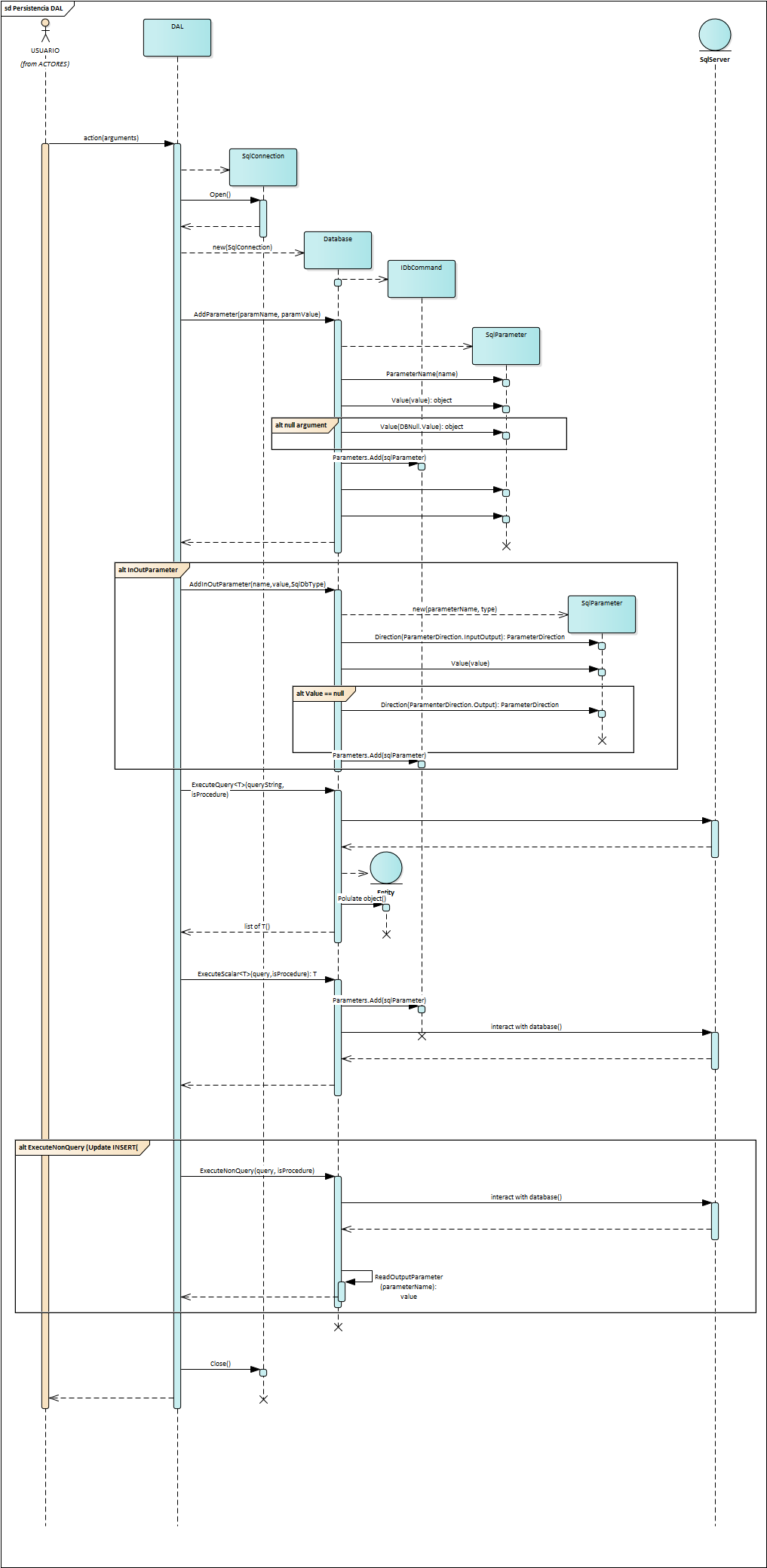
El usuario interactúa con la interfaz gráfica cargando los datos requeridos.

La capa BLL se encargará de validar los datos y en caso de ser válidos, invocar la capa DAL.

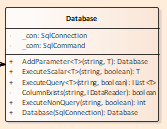
La capa DAL es responsable de gestionar la sesión e interactuar con la clase Database, indicando las queries sql, stored procedures y parámetros a utilizar.

La clase Database se encargará del mapeo de objetos e interacción con la base de datos.

Adjunto diagrama de secuencia mostrando un flujo genérico.



### T01 Clase Database



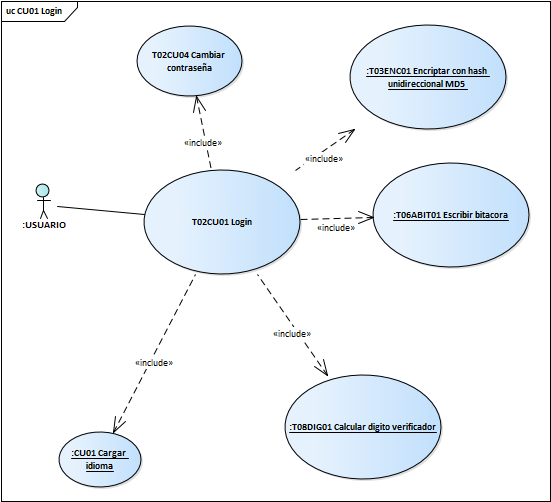
## T01 Mapa tentativo de menús

* Usuario
  + Logout
  + Administrar permisos
  + Administrar usuarios
* Proveedores
* Idioma
  + Cambiar idioma

# T02 Gestión de Log In / Log Out

## T02CU01 Login

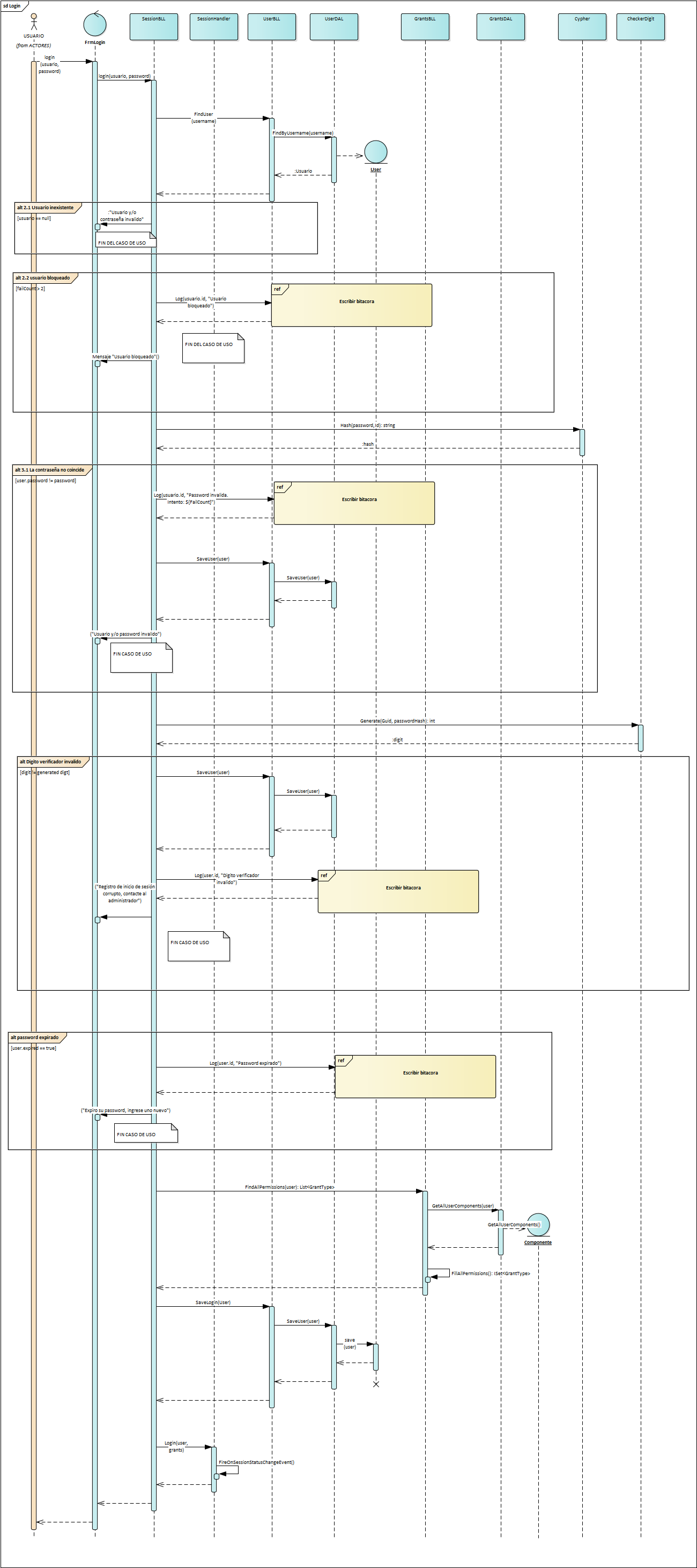
### Diagrama de casos de uso



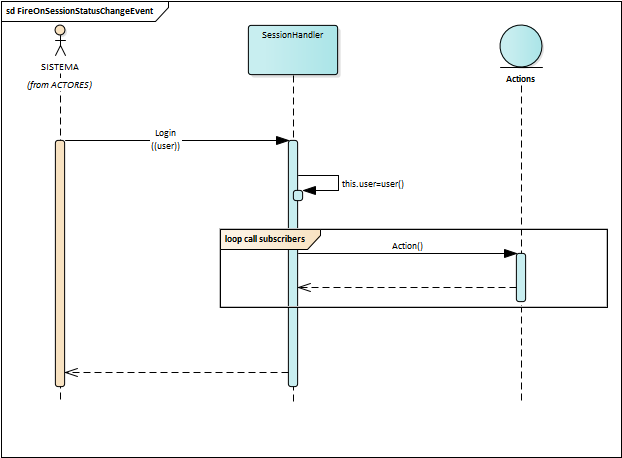
### Descripción del caso de uso.

|  |  |
| --- | --- |
| **Identificación** | T02CU01 Login |
| **Descripción** | El usuario se identifica en el sistema con su usuario y contraseña para **comenzar a operar.**  Cuando el usuario inicia sesión se valida la integridad de sus datos, calculando un digito verificador en función de su contraseña y la fecha último login.  El usuario contara de 3 intentos fallidos para iniciar sesión, si supera los 3 intentos el sistema lo bloqueara dejando un registro en la bitácora.  Al iniciar sesión se cargan todos los permisos del usuario, así como también el idioma por defecto de la interfaz de usuario. |
| **Precondiciones** | No hay ninguna sesión iniciada previamente. |
| **Actor principal** | TODOS |
| **Puntos de inclusión** |  |
| **Escenario principal** | **Flujo alternativo** |
| 1. El usuario ingresa su usuario y contraseña, luego presiona el botón “Ingresar”. |  |
| 1. El sistema busca al usuario en la base de datos y lo recupera. Luego verifica que no se encuentre bloqueado. | 2.1 El usuario no existe en la base de datos, el sistema informa “Usuario y/o contraseña invalido”.  2.2.1 El usuario está bloqueado (intentos > 2). El sistema muestra un mensaje por pantalla “Usuario bloqueado, intente recuperar contraseña o póngase en contacto con el administrador”. Se registra en la bitácora (T06ABIT01) el mensaje “Intento de inicio de sesión con usuario bloqueado”. **Fin caso de uso.** |
| 1. El sistema encripta la contraseña ingresada por el usuario (T03CU01) y la compara con la de la base de datos. | 3.1 la contraseña no coincide, el sistema registra en la bitácora el intento de inicio de sesión fallido, suma 1 al contador de intentos, y notifica al usuario un mensaje genérico “Usuario y/o contraseña invalido”. **Se finaliza el caso de uso.** |
| 1. El sistema valida la integridad de la información del usuario calculando un digito verificador (T08DIG01) | 4.1 El sistema no valida integridad, informa al usuario “Registro de inicio de sesión corrupto, contacte al administrador”. Se registra en la bitácora el mensaje “Intento login fallido, no valida integridad” (T06ABIT01). **Se finaliza el CU.** |
| 1. El sistema valida que el password no este expirada. | 5.1 La password está expirada, el sistema redirige al usuario al caso de uso “T02CU04 Cambiar contraseña”. **Se finaliza el caso de uso.** |
| 1. El sistema actualiza la fecha de ultimo login, el contador de intentos en “0”, inicializa la sesión y genera un registro de bitácora (T06ABIT01) con el texto “Inicio de sesión correcto”, traduce la interfaz al idioma por defecto del usuario (T05CU01), carga sus permisos en la sesión y lo dirige a la pantalla principal. |  |
| FIN DEL CASO DE USO | |

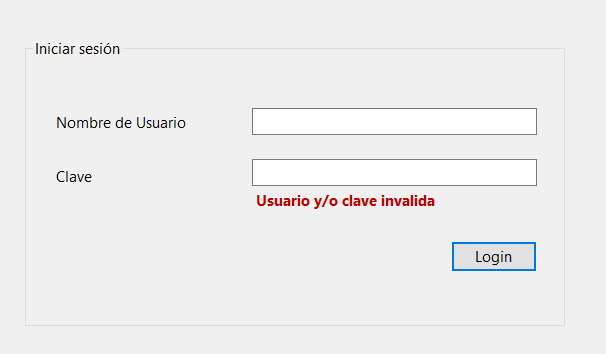
### Diagrama de secuencia



Detalle de la función FireOnSessionStatusChangeEvent



GUI

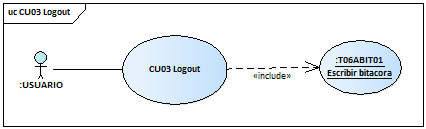


## T02CU03 Logout

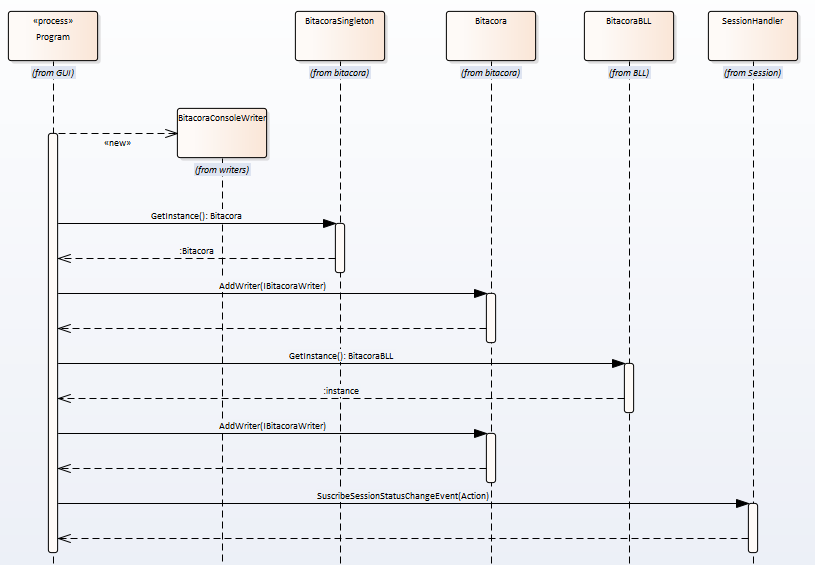
### Descripción del caso de uso

|  |  |
| --- | --- |
| **Identificación** | T02CU03 Logout |
| **Descripción** | El usuario finaliza su sesión de trabajo. |
| **Precondiciones** | No hay ninguna sesión iniciada previamente. |
| **Actor principal** | TODOS |
| **Puntos de inclusión** |  |
| **Escenario principal** | **Flujo alternativo** |
| 1. El usuario ingresa al menú de usuario, y pulsa el botón “Cerrar sesión” |  |
| 1. El sistema registra el logout en la bitácora T06ABIT01 y redirige al usuario a la pantalla de login. |  |
| FIN DEL CASO DE USO | |

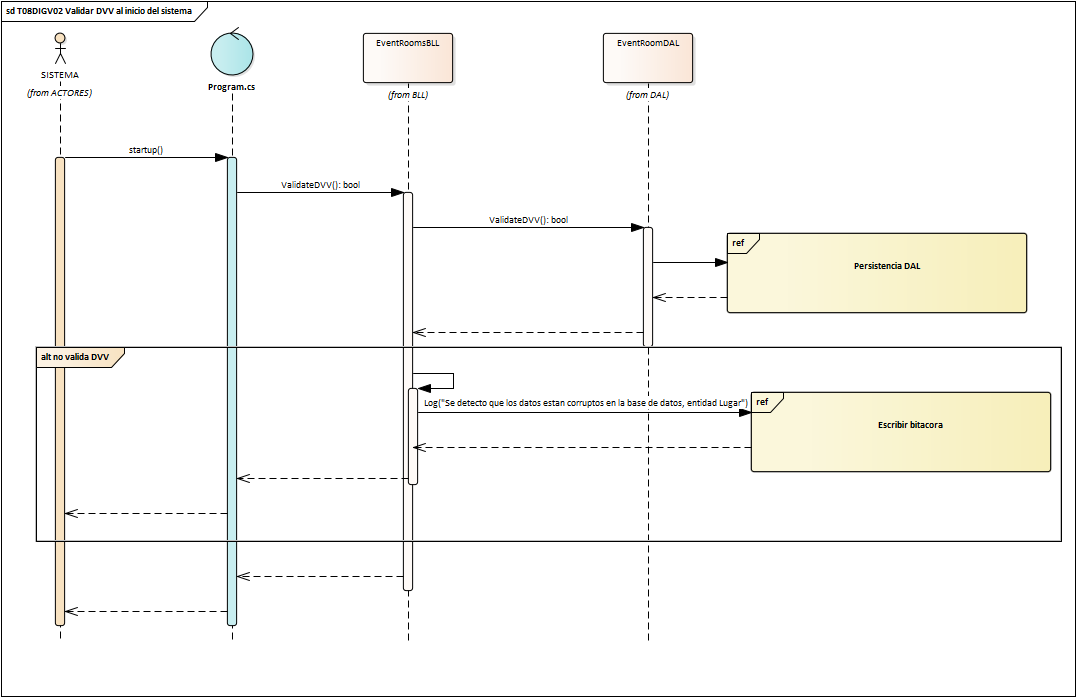
### Diagrama de caso de uso



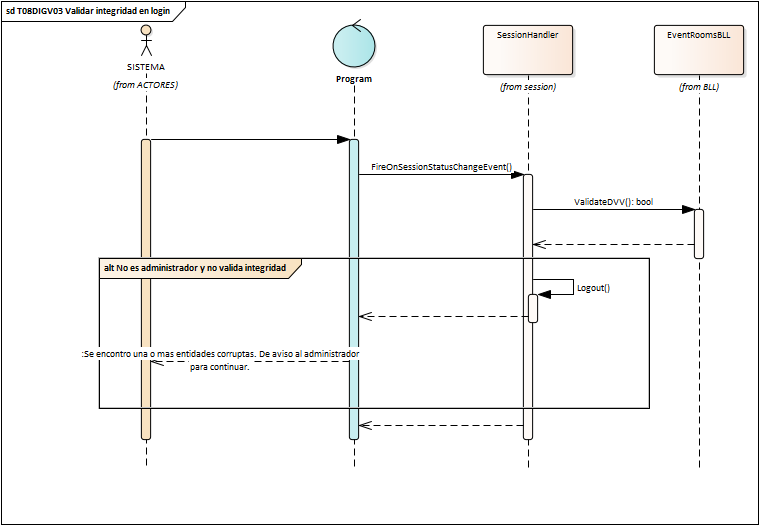
## T02 Inicio del sistema



Validación del DVV:



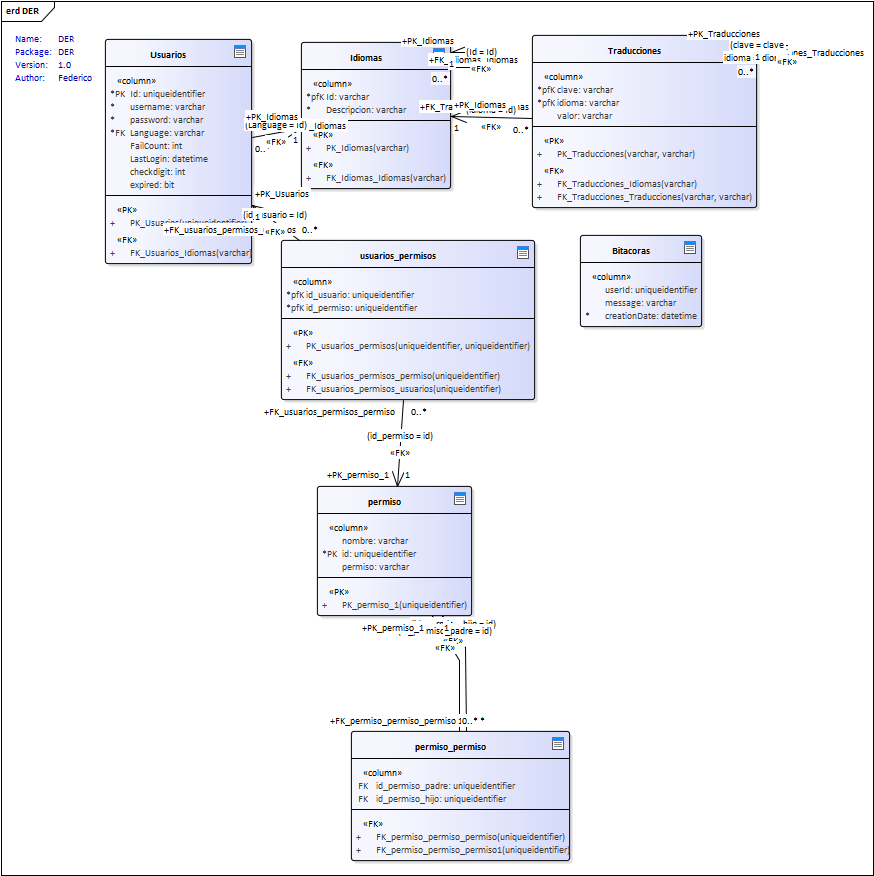
Validaciones en el LOGIN



## T02 Diagrama de clases



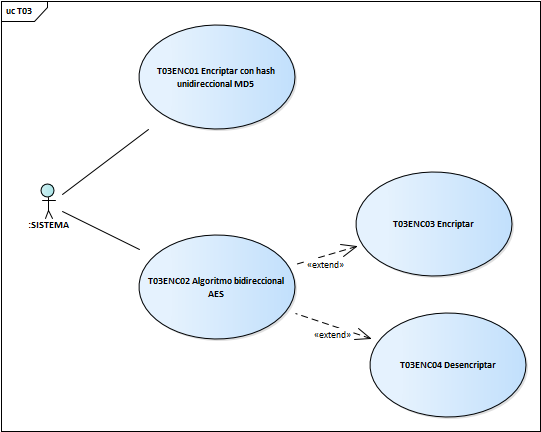
## T02 Diagrama Entidad Relación



# T03 Gestión del encriptado

La gestión del encriptado tiene por objetivo principal proteger los datos sensibles del sistema, para que estos no puedan ser fácilmente accedidos por usuarios maliciosos.

Hay que tener presente que esta funcionalidad es un caso de uso del sistema, y que como tal no serán accedidos por los usuarios de este, sino que en su lugar serán invocados por casos de uso del negocio.

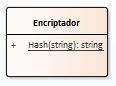


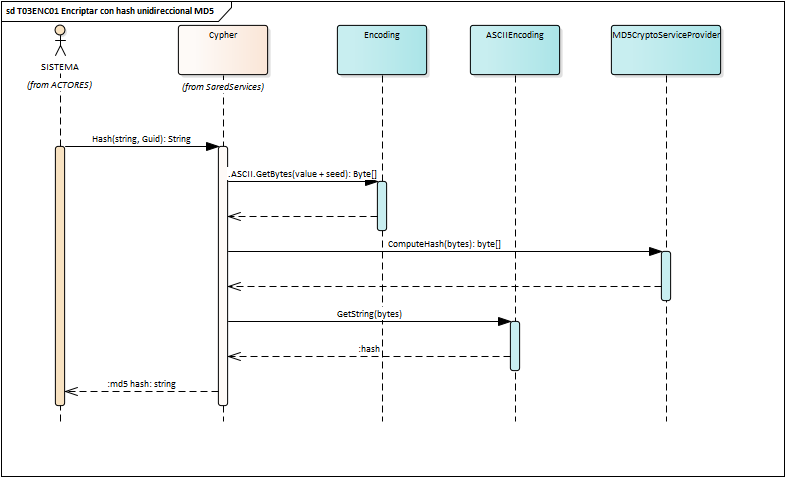
## T03ENC01 Encriptar con hash unidireccional MD5

### Descripción del caso de uso

|  |  |
| --- | --- |
| **Identificación** | T03ENC01 Encriptar con hash unidireccional MD5 |
| **Descripción** | Consiste en una función de hash unidireccional que ante una entrada genera un único e irrepetible hash.  Se lo describe como unidireccional ya que no es posible partiendo de un hash, obtener el input utilizado para generarlo. |
| **Precondiciones** |  |
| **Actor principal** | TODOS |
| **Puntos de inclusión** |  |
| **Escenario principal** | **Flujo alternativo** |
| 1. El sistema toma los parámetros de texto a procesar, lo convierte en una cadena de bytes, genera un hash MD5 utilizando el módulo MD5CryptoServiceProvider provisto por C# y lo vuelve a transformar en una cadena de texto. |  |
|  |  |
| FIN DEL CASO DE USO | |

### Diagrama de secuencia





## T03ENC02 Algoritmo bidireccional AES

Se describe en mecanismo de cifrado AES como bidireccional ya que permite, partiendo de una entrada, obtener un código de encriptado único.

A su vez, este código puede ser desencriptado volviendo al valor original.

AES Consiste en dos valores claves para su funcionamiento:

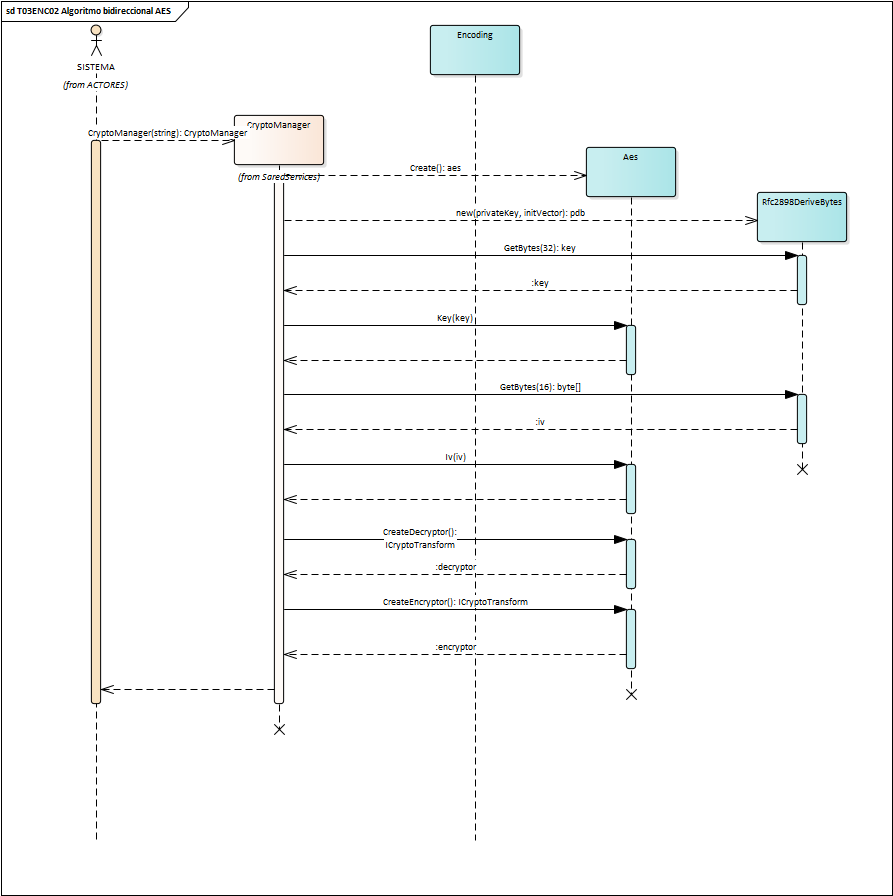
* clave privada: es un código conocido únicamente por nuestra aplicación.
* IV: El vector de inicialización (IV por sus siglas en inglés)

Para más detalles sobre este algoritmo se puede consultar el siguiente enlace: <https://datatracker.ietf.org/doc/html/rfc3826>

### Descripción del caso de uso

|  |  |
| --- | --- |
| **Identificación** | T03ENC02 Algoritmo bidireccional AES |
| **Descripción** | Inicializa el módulo de encripción bidireccional. |
| **Precondiciones** |  |
| **Actor principal** | TODOS |
| **Puntos de extensión** | 1.1 Encriptar, 1.2 Desencriptar |
| **Escenario principal** | **Flujo alternativo** |
| 1. El sistema toma la clave privada e inicializa el módulo de encripción. |  |
|  |  |
| FIN DEL CASO DE USO | |

### Diagrama de secuencia

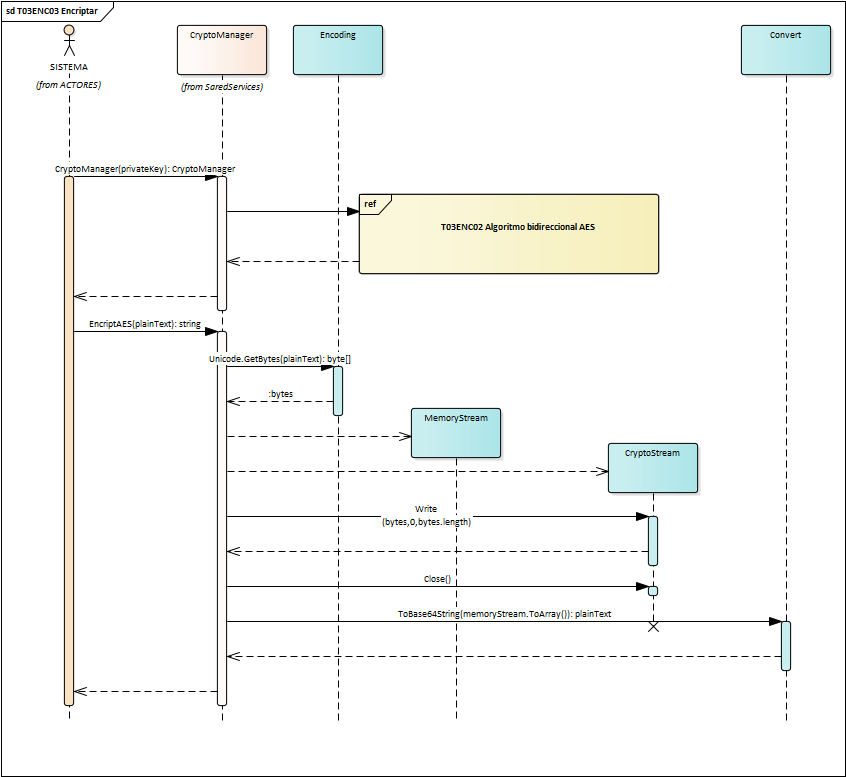


## T03ENC03 Encriptar

### Descripción del caso de uso

|  |  |
| --- | --- |
| **Identificación** | T03ENC03 Encriptar |
| **Descripción** | Encripta la información utilizando AES. |
| **Precondiciones** | Debe haberse inicializado el módulo (T03ENC02) |
| **Actor principal** | TODOS |
| **Puntos de extensión** |  |
| **Escenario principal** | **Flujo alternativo** |
| 1. El sistema transforma el texto plano en una cadena de bytes, los encripta y lo vuelve a transformar en una cadena de texto plano (encriptada). |  |
|  |  |
| FIN DEL CASO DE USO | |

### Diagrama de secuencia

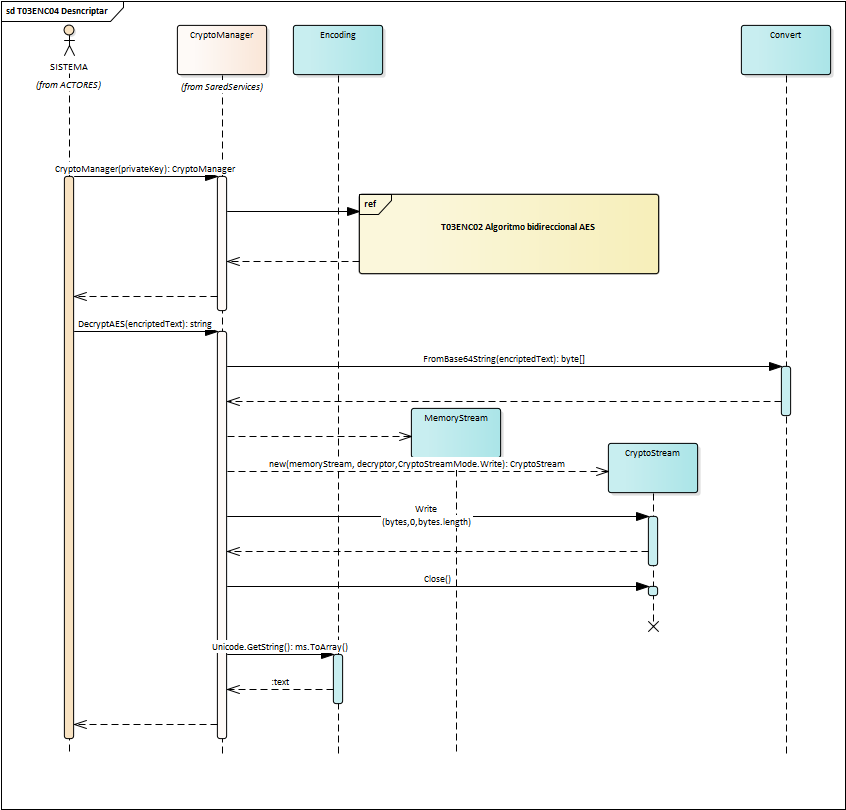


## T03ENC04 Desencriptar

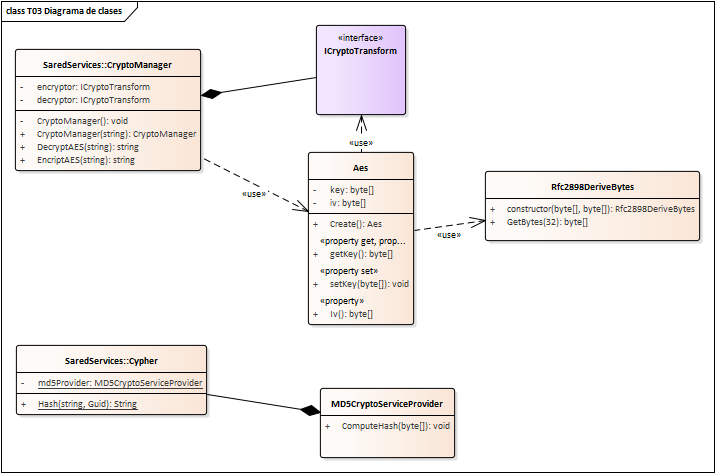
### Descripción del caso de uso

|  |  |
| --- | --- |
| **Identificación** | T03ENC04 Desencriptar |
| **Descripción** | Encripta la información utilizando AES. |
| **Precondiciones** | Debe haberse inicializado el módulo (T03ENC02) |
| **Actor principal** | TODOS |
| **Puntos de extensión** |  |
| **Escenario principal** | **Flujo alternativo** |
| 1. El sistema transforma el texto encriptado en una cadena de bytes, los desencripta y lo vuelve a transformar en una cadena de texto plano (desencriptada). |  |
|  |  |
| FIN DEL CASO DE USO | |

### Diagrama de secuencia



## T03 Diagrama de clases



# T04. Gestión de Perfiles de Usuario

Cada usuario podrá tener un rol distinto dentro del sistema, lo que implica que cada usuario tendrá distintas responsabilidades.

Para contribuir con una mejor integridad de las operaciones, es que se permitirá a los usuarios únicamente realizar las acciones que estos requieran realizar.

Esa es la razón por la cual se agrega el módulo de Gestión de Permisos y el módulo de Gestión de Usuarios al sistema, descritos a continuación.

## T04PER Gestión de Permisos

**Patente**: Es un permiso individual que permite realizar una acción concreta, ejemplo: acceder al módulo de gestión de permisos.

Las patentes cobran importancia ya que son las únicas que otorgan un permiso específico al usuario mediante su “Tipo de permiso”.

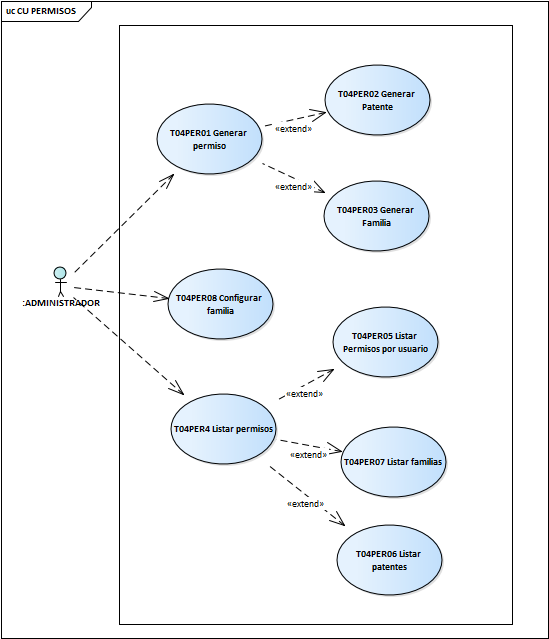
Los tipos de permiso serán los que finalmente controlan que el usuario pueda realizar o no la acción dentro de la aplicación, y puede haber más de una patente asociada al tipo de permiso.

**Familia**: Es un conjunto de patentes.

Esto es muy útil, por ejemplo, para asignar un grupo de patente a lugares que tengan más de un administrador.

Las familias podrán estar asociadas con otras familias, en cuyo caso tendrán todas las patentes que estas tengan relacionadas. Es importante tener en cuenta que el hecho de quitar una patente de una familia no implica reducir los permisos que tenga un usuario, ya que esta patente podría estar asociada a otra familia.

A continuación, se anexa el diagrama de casos de uso para la gestión de permisos



### T04PER01 Generar permiso + T04PER02 Generar patente

|  |  |
| --- | --- |
| **Nombre** | T04PER01 Generar permiso |
| **Objetivo** | Generar un permiso nuevo |
| **Precondiciones** | El usuario cuenta con rol de administrador para acceder al modulo |
| **Postcondiciones** | Hay un nuevo permiso generado |
| **Eventos disparadores del caso de uso** | El administrador desea dar de alta un nuevo permiso |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El administrador ingresa el nombre del permiso a agregar y presiona el boton “Agregar” |  |
| 1. El sistema valida que no exista otro permiso con ese mismo nombre. | 2.1 Existe otro permiso con el mismo nombre, el sistema advierte al usuario con un mensaje de error. Finaliza el caso de uso. |
| 1. El sistema guarda la nueva patente | 3. 1 El sistema guarda la familia. |
| 1. Se invoca el caso de uso “T06BIT01 Escribir bitacora” con el mensaje: Se agrego el permiso ${nombre} con exito”. |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### T04PER03 Listar permisos

|  |  |
| --- | --- |
| **Nombre** | T04PER03 Listar permisos |
| **Objetivo** | Listar los permisos disponibles |
| **Precondiciones** | El usuario cuenta con rol de administrador para acceder al modulo |
| **Postcondiciones** | Se visualiza una lista de permisos |
| **Eventos disparadores del caso de uso** | Se necesitan ver permisos |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El administrador ingresa los parametros de busqueda como tipo de permiso, permiso principal , usuario |  |
| 1. El sistema busca y lista los permisos. | . |
|  |  |
|  |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### T04PER04 Listar Familias

|  |  |
| --- | --- |
| **Nombre** | T04PER04 Listar familias |
| **Objetivo** | Listar las familias de permisos disponibles |
| **Precondiciones** | El usuario cuenta con rol de administrador para acceder al modulo |
| **Postcondiciones** | Se visualiza una lista de permisos |
| **Eventos disparadores del caso de uso** | Se necesitan ver permisos |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El administrador ingresa al módulo e inicia la carga de familias |  |
| 1. El sistema busca y lista las familias. | . |
|  |  |
|  |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### T04PER05 Configurar familia

|  |  |
| --- | --- |
| **Nombre** | T04PER05 Configurar familia |
| **Objetivo** | El administrador desea configurar una familia de permisos existentes |
| **Precondiciones** | El usuario cuenta con rol de administrador para acceder al modulo, se invocaron los casos de uso T04PER03 y T04PER04 |
| **Postcondiciones** | Se relacionan los permisos a la familia a configurar |
| **Eventos disparadores del caso de uso** | El administrador desea configurar la familia |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El administrador selecciona la familia y presiona el botón “Configurar” |  |
| 1. El sistema carga la familia y su arbol de dependencias invocando el caso de uso PER04CU06 |  |
| 1. El administrador selecciona una familia o permiso y presiona el botón de agregar | 3.A El administrador presiona el botón eliminar. |
| 1. El sistema agrega la familia peremiso a la familia (T06ABIT01) y actualiza el arbol de permisos (T04PER06). | 4.A El sistema elimina el permiso, escribe en la bitacora que se elimino el permiso (T06ABIT01) y actualiza el arbol de permisos (T04PER06). |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### T04PER06 Cargar familia

|  |  |
| --- | --- |
| **Nombre** | T04PER06 Cargar familia |
| **Objetivo** | El administrador desea configurar una familia de permisos existentes |
| **Precondiciones** | No hay |
| **Postcondiciones** | Se cuenta con una familia completamente instanciada. |
| **Eventos disparadores del caso de uso** | Se necesita obtener el arbol de permisos completos. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El sistema recupera de la base de datos todos los permisos relacionados con la familia solicitada. |  |
| **FIN DEL CASO DE USO** | |

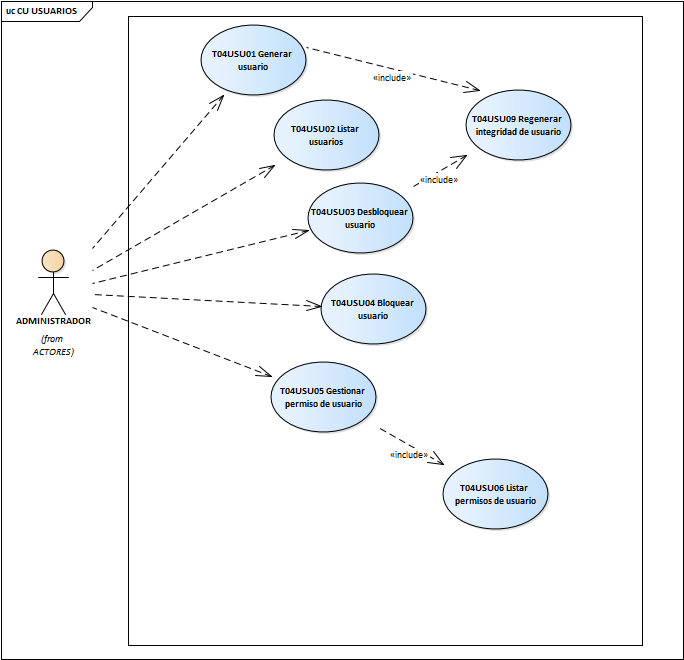
|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

## T04USU Gestión de usuarios

Los administradores del sistema tienen la necesidad de realizar gestiones relacionadas con los usuarios, como, por ejemplo:

* ABMC de usuarios
* Desbloquear / bloquear usuarios
* Gestion de permisos

Se adjunta el diagrama de casos de uso:



### T04USU01 Generar usuario

|  |  |
| --- | --- |
| **Nombre** | T04USU01 Generar usuario |
| **Objetivo** | Registrar un nuevo usuario al sistema |
| **Precondiciones** | El usuario tiene rol de administrador. |
| **Postcondiciones** | Un usuario dado de alta. |
| **Eventos disparadores del caso de uso** | El administrador presiona el boton “Alta” de la pantalla de gestión de usuarios. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El usuario administrador ingresa: nombre de usuario, contraseña, email, telefono y pulsa el botón registrar. |  |
| 1. El sistema valida que el usuario haya cargado todos los campos. | 2. 1 El sistema notifica al usuario que le falto cargar un campo, especificando cual. Fin del caso de uso. |
| 1. El sistema valida que no exista otro usuario con el mismo nombre de usuario cargado. | 3.1 El sistema notiffica al usuario que ya existe otro usuario con ese nombre de usuario. |
| 1. El sistema cifra el password (T03ENC01), configura en 0 la cantidad de intentos de inicio de sesión fallidos, calcula el digito verificador (T08DIG01) y guarda el usuario. |  |
| 1. El sistema registra en la bitacora el alta del usuario y notifica al usuario del alta exitosa. |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### T04USU02 Listar usuario

Se omite intencionalmente ya que carece de complejidad.

### T04USU03 Desbloquear usuario

|  |  |
| --- | --- |
| **Nombre** | T04USU03 Desbloquear usuario |
| **Objetivo** | Desbloquear un usuario |
| **Precondiciones** | El usuario tiene rol de administrador. |
| **Postcondiciones** | El usuario queda desbloqueado con la password expirada. |
| **Eventos disparadores del caso de uso** | El administrador presiona el botón “Desbloquear” de la pantalla de gestión de usuarios. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El usuario administrador selecciona el usuario y pulsa desbloquear. |  |
| 1. El sistema le solicita un nuevo password. |  |
| 1. El usuario administrador coloca un nuevo password y presiona desbloquear. | 3.1 El sistema identifica que no se ingreso un password. Notifica al usuario del dato obligatorio. Fin del caso de uso. |
| 1. El sistema valida que se haya ingresado un password. Cifra el password (T03ENC01), configura en 0 la cantidad de intentos de inicio de sesión fallidos, calcula el digito verificador (T08DIG01), activa el flag de password expirada y guarda el usuario. |  |
| 1. El sistema registra en la bitácora el desbloqueo de usuario y notifica al usuario. |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### T04USU04 Bloquear usuario

|  |  |
| --- | --- |
| **Nombre** | T04USU04 Bloquear usuario |
| **Objetivo** | Bloquear un usuario para que no pueda iniciar sesión. |
| **Precondiciones** | El usuario tiene rol de administrador. |
| **Postcondiciones** | El usuario queda bloqueado. |
| **Eventos disparadores del caso de uso** | El administrador presiona el botón “Desbloquear” de la pantalla de gestión de usuarios. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El usuario administrador selecciona el usuario y pulsa bloquear. |  |
| 1. El sistema consulta al usuario si está seguro de querer bloquear el usuario. |  |
| 1. El usuario administrador confirma que desea bloquear el usuario. |  |
| 1. El sistema bloquea el usuario marcando el contador de fallos en 10 y registra en la bitácora que se bloqueó por un administrador. |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### T04USU05 Gestionar permiso de usuario

|  |  |
| --- | --- |
| **Nombre** | T04USU05 Gestionar permiso de usuario |
| **Objetivo** | Asociar los permisos a los usuarios. |
| **Precondiciones** | El usuario tiene rol de administrador. Existen permisos cargados con anterioridad. |
| **Postcondiciones** | El usuario queda con el permiso relacionado. |
| **Eventos disparadores del caso de uso** |  |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El administrador selecciona el usuario a configurar y presiona el botón “Configurar” |  |
| 1. El sistema carga los permisos del usuario (T04USU08) |  |
| 1. El usuario selecciona un permiso de la lista y presiona “Agregar” | 3.1 El usuario selecciona un permiso y presiona “Quitar”. |
| 1. El sistema agrega el permiso al arbol de permisos del usuario. | 4.1 el sistema borra el permiso del árbol de permisos del usuario. |
| 1. El usuario presiona el botón guardar configuración. |  |
| 1. El sistema borra los permisos anteriores relacionados con el usuario y vuelve a guardar el arbol de permisos. Se invoca al caso de uso T06ABIT01 con el mensaje “Se modificaron los permisos del usuario ${username}”. |  |
|  |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### T04USU06 Agregar permiso de usuario

|  |  |
| --- | --- |
| **Nombre** | T04USU06 Agregar permiso de usuario |
| **Objetivo** | Asociar los permisos a los usuarios. |
| **Precondiciones** | El usuario tiene rol de administrador. |
| **Postcondiciones** | El usuario queda con el permiso relacionado. |
| **Eventos disparadores del caso de uso** | El usuario solicita al administrador que le asocie el permiso. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El sistema asocia |  |
| 1. El sistema carga los permisos del usuario (T04USU08) |  |
| 1. El usuario selecciona un permiso de la lista y presiona “Agregar” (T04USU06) | 3.1 El usuario selecciona un permiso de la lista y presiona “Eliminar” (T04USU07) |
| 1. El sistema vuelve a cargar los permisos del usuario actualizados. |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### T04USU06 Listar permisos de usuario

Se omite por la simplicidad.

### T04USU10 Generar integridad de usuario

|  |  |
| --- | --- |
| **Nombre** | T04USU10 Generar integridad de usuario |
| **Objetivo** | Configurar el usuario de forma que valide su integridad y pueda operar. |
| **Precondiciones** | Es invocado por otro caso de uso. |
| **Postcondiciones** | El usuario queda configurado con la integridad correcta.. |
| **Eventos disparadores del caso de uso** | Se necesita resetear la integridad del usuario.. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** | No aplica. |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El sistema toma el usuario, genera el hash de la contraseña, calcula el digito verificador y setea el contador de fallos en 0. |  |
|  |  |
|  |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** | No Aplica. |

# T05 Gestión de múltiples idiomas

## T05.A Objetivo

El sistema gestor de eventos podrá ser utilizado en distintos países, con lo que es importante contar con la posibilidad de traducir su interfaz de usuario acorde a las necesidades de sus usuarios.

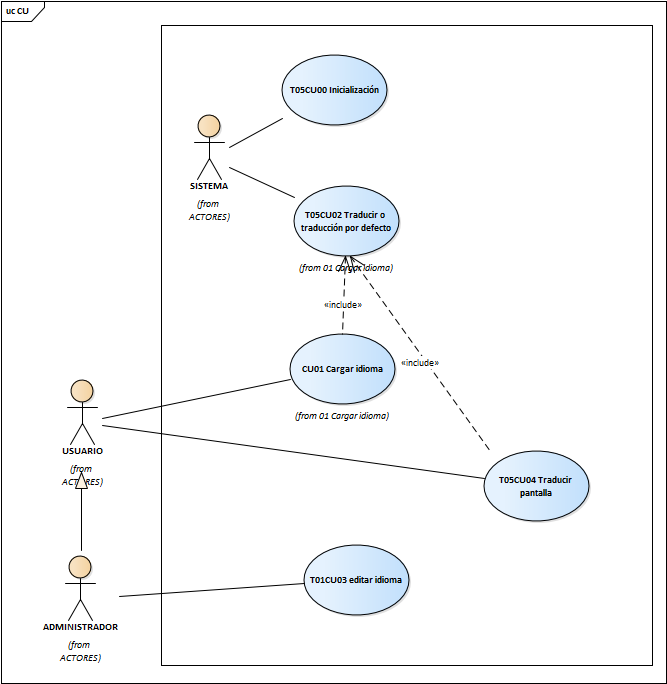
La traducción de las interfaces será dinámica, entendiendo que el usuario puede elegir cambiar de idioma en cualquier momento y esto no debe impedir continuar o finalizar un flujo.

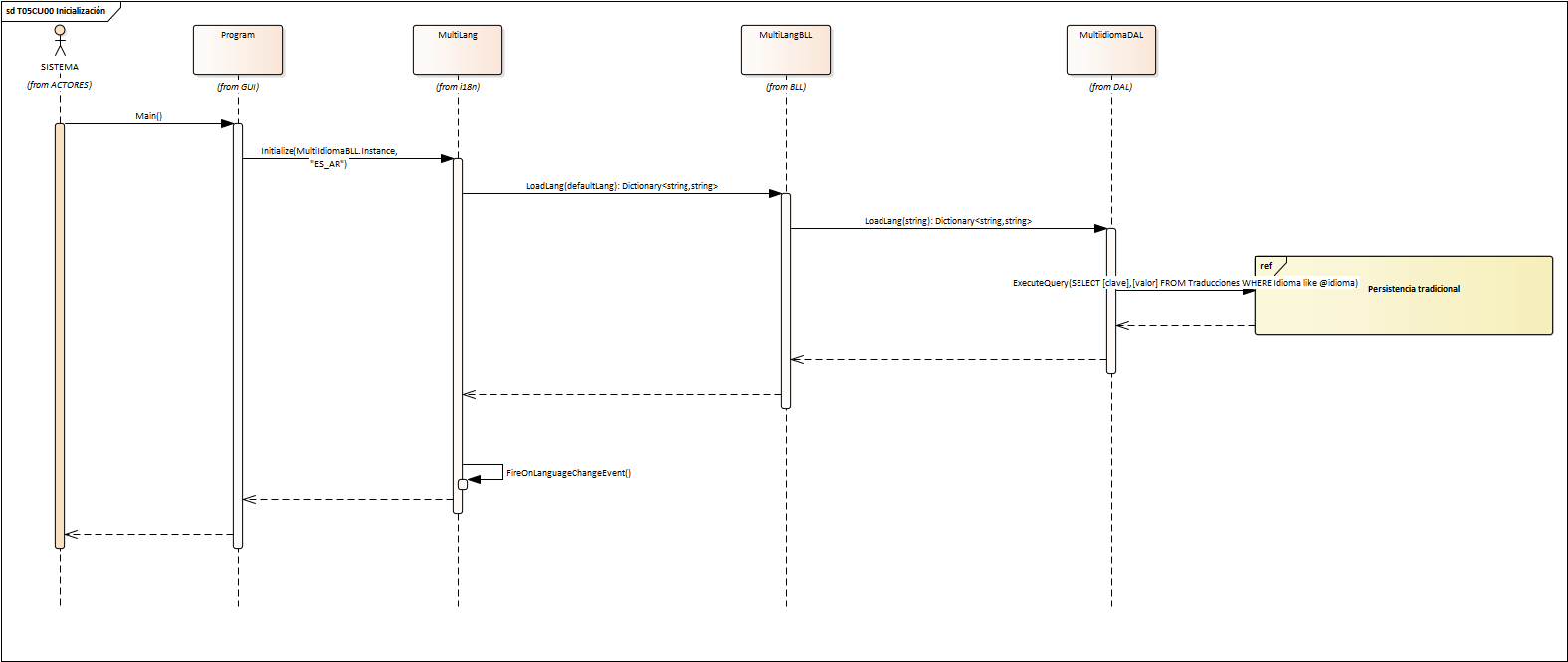
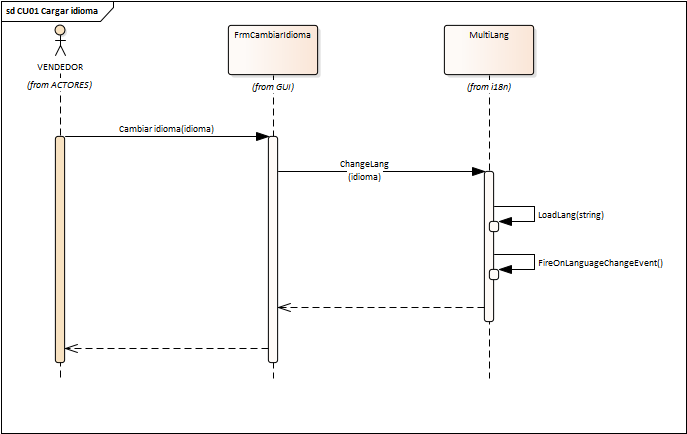
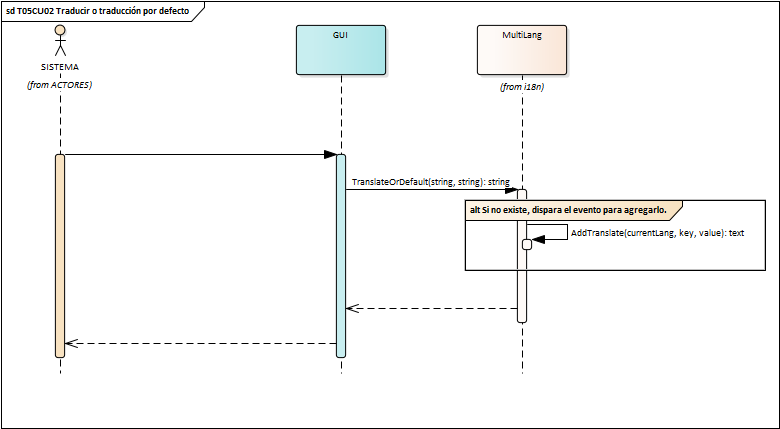
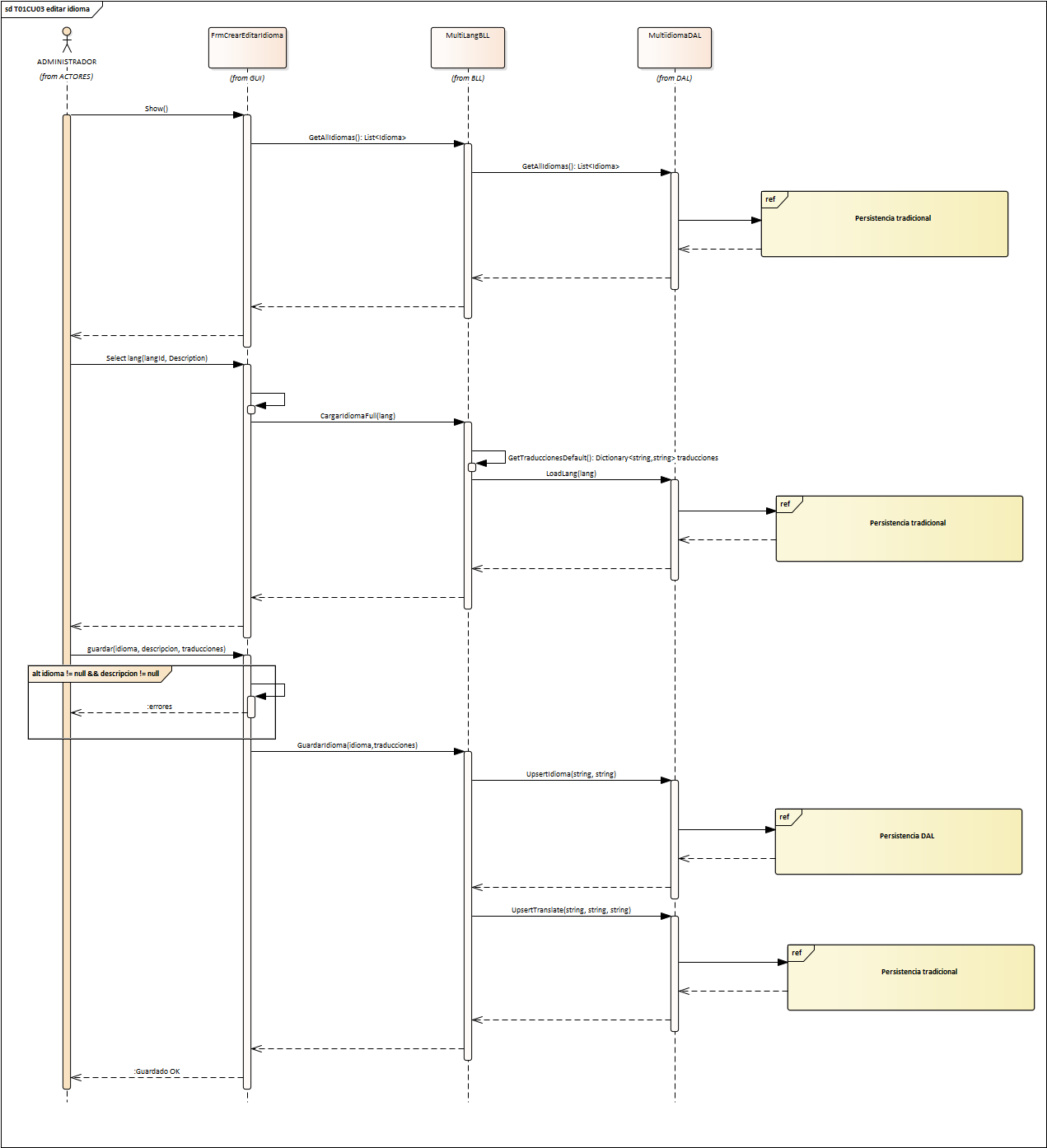
## T05.B Descripción detallada de cómo funciona

El modulo de MultiIdioma permite traducir cualquier texto de la interfaz de usuario utilizando un listado del tipo “Clave-Valor” para las traduccciones.

Este listado es almacenado en una cache de la clase Singleton **MultiLang**, lo que permite que las traducciones estén accesibles en todo momento.

A continuación se describen los distintos procesos relacionados, apalancandonos en un diagrama de casos de uso:

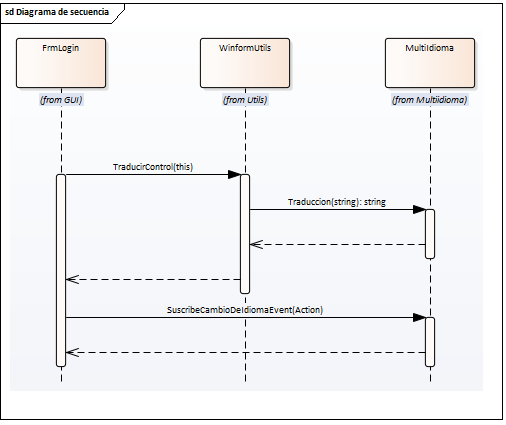


* T05CU00 Inicialización:  
  Se carga el lenguaje por defecto y se deja disponible para el resto de la ejecución.  
  
* T05CU01 Cargar idioma  
  
* T05CU02 Traducción o traducción por defecto  
  
* T05CU03 Editar idioma  
  Edita / crea el idioma  
  
* T05CU04 Traducir formulario

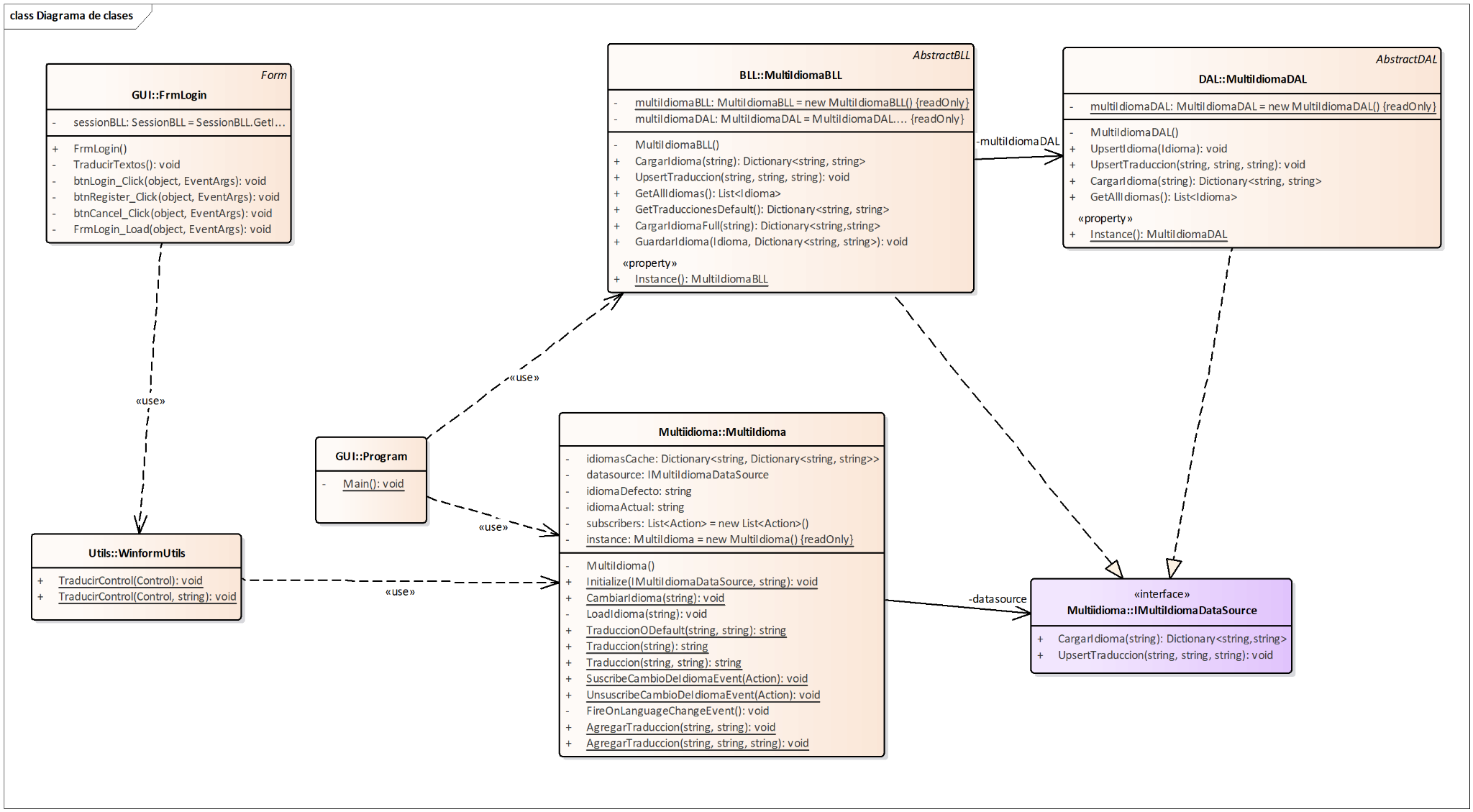
Para este caso de uso se utiliza el patron observer, suscribiendo todos los formularios al evento “SuscribeCambioDeIdiomaEvent”.

Se recomienda la utilización de “WinformUtils”, clase encargada de hacer una busqueda recursiva en todos los controles de la pantalla, recuperar la “key” de la property “Tag” de los controles de windows forms y traducir automaticamente las leyendas.

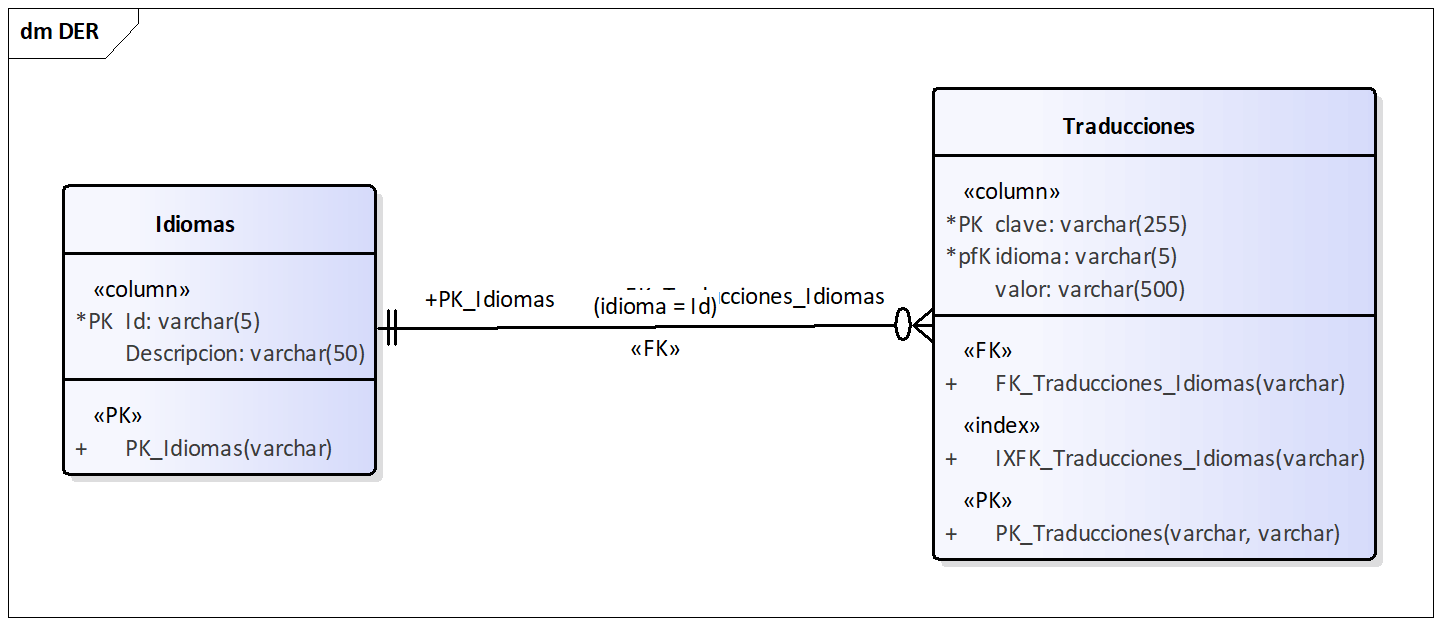
En caso de no encontrar traducción, se dispara el evento que la agrega.



## Diagrama de clases



## Diagrama entidad relación



# T06 Bitácora y control de cambios

## T06A Bitácora

### Objetivo

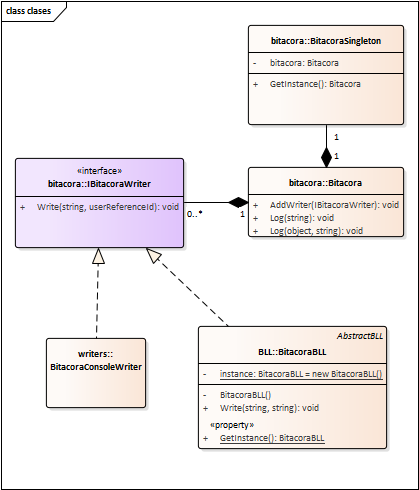
El objetivo de la bitacora es tener una trazabilidad de las acciones que se van realizando en el sistema.

Esta trazabilidad permitirá comprender el comportamiento de los usuarios dentro del sistema, ya sea para diagnosticar errores o bien para entender como utilizaron el sistema.

### Funcionamiento

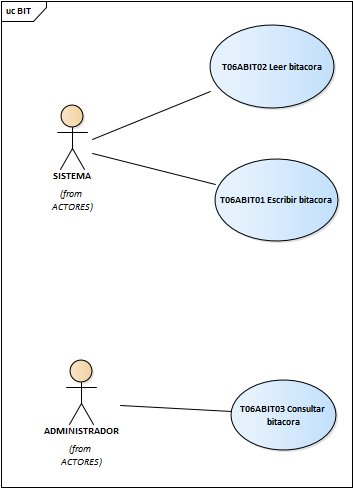
La bitácora es permite guardar los eventos generados en el sistema en distintos repositorios, siendo de esta manera agnóstica del sistema donde se utilicé.

Para comprender el funcionamiento, se expone el diagrama de clases a continuación:



En el diagrama de clases se puede apreciar a nivel conceptual el funcionamiento de la bitácora, implementando los patrones de diseño Singleton y PubSubs.

* Singleton: se eligió este patrón ya que solo debe existir una bitácora activa por instancia de la aplicación activa.  
  Entre las ventajas que se pueden apreciar, es que la bitácora se subscribe al inicio de sesión, donde se indica el identificador del usuarios.
* PubSubs: Facilita imprimir la bitácora en distintos destinos, por ejemplo: un archivo de texto, la consola de un IDE de desarrollo, una base de datos.



Solo se realizará la plantilla del caso de uso de consultar la bitácora, ya que es la única que cuenta con una pantalla.

|  |  |
| --- | --- |
| **Nombre** | T06BITA03 Consultar bitácora |
| **Objetivo** | Consultar los sucesos ocurridos en el sistema. |
| **Precondiciones** | El usuario es ADMINISTRADOR |
| **Postcondiciones** | Se visualizan los datos de la bitacora. |
| **Eventos disparadores del caso de uso** | El administrador desea entender que sucedió. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El ADMINISTRADOR carga los criterios de búsqueda y pulsa consultar. Criterios: nombre de usuario, fecha desde, fecha hasta |  |
| 1. El sistema recupera las trazas relacionadas con los criterios de busqueda y las muestra en una grilla |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

Adicionalmente, se documenta el diagrama de secuencia del caso de uso de sistema “T06ABIT01 Escribir en bitacora” ya que es referenciado desde multiples casos de uso para escribir mensajes en la bitacora:

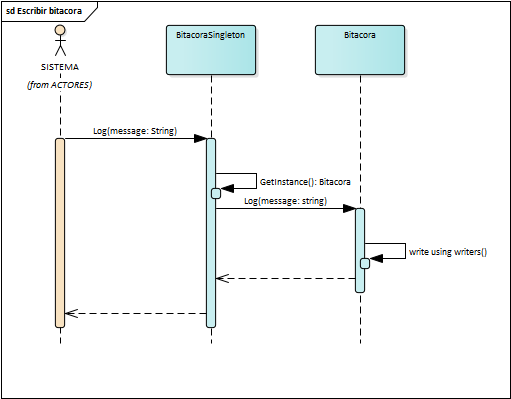
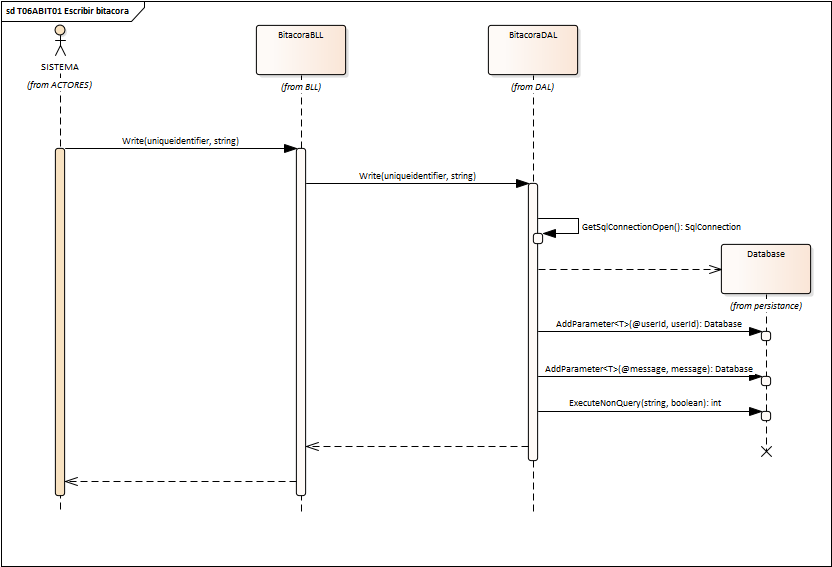


Diagrama de secuencia del Writter “BitacoraBLL”

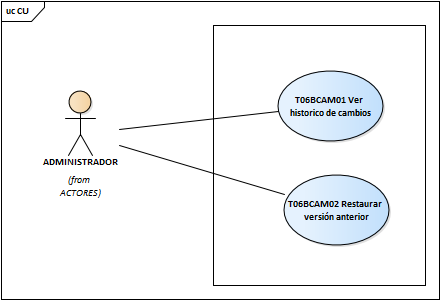


## T06BCAM Control de cambios

El control de cambios tiene por objetivo tener la trazabilidad de todas las modificaciones sufridas por una entidad del negocio, permitiendo de esta manera restaurarla a un estado anterior en caso de ser necesario.

Esto solo podrá ser efectuado por un administrador del sistema.

A continuación, se presenta el diagrama de casos de uso.



|  |  |
| --- | --- |
| **Nombre** | T06BCAM01 Ver historico de cambios |
| **Objetivo** | Ver el historico de cambios de la entidad Lugares |
| **Precondiciones** | El usuario es ADMINISTRADOR |
| **Postcondiciones** | Se visualizan los cambios realizados. |
| **Eventos disparadores del caso de uso** | El administrador selecciona un lugar y presiona el botón “historial” |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El ADMINISTRADOR presiona el botón “Historial” de la pantalla de Lugares |  |
| 1. El sistema recupera todas las versiones de ese lugar y las exibe por pantalla. |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### T06BCAM02 Restaurar versión anterior

|  |  |
| --- | --- |
| **Nombre** | T06BCAM02 Restaurar version anterior |
| **Objetivo** | Restaurar una version anterior |
| **Precondiciones** | El usuario es ADMINISTRADOR |
| **Postcondiciones** | Se crea una nueva versión a partir de la versión seleccionada. |
| **Eventos disparadores del caso de uso** | El administrador selecciona una versión y presiona “Restaurar” |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

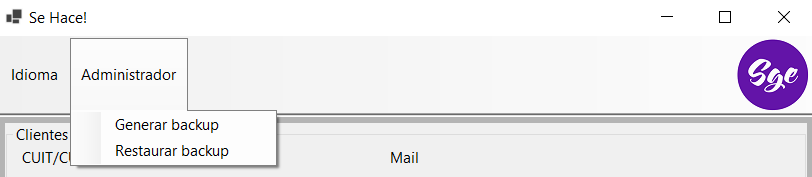
|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El ADMINISTRADOR restaurar de una versión que valida integridad. |  |
| 1. El sistema invoca el caso de uso “T08DIGH01 Restarurar digito horizontal“ dese la función de guardado del eventRoom. Invoca el CU BITA01 Escribir bitacora con el texto "Se restauro la veresión " + this.current.CreatedAt + " - " + this.current.UpdatedBy + " del eventRoom " + this.current.Id Luego informa al usuario que se restauro la versión correctamente y cierra la ventana de historial. |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

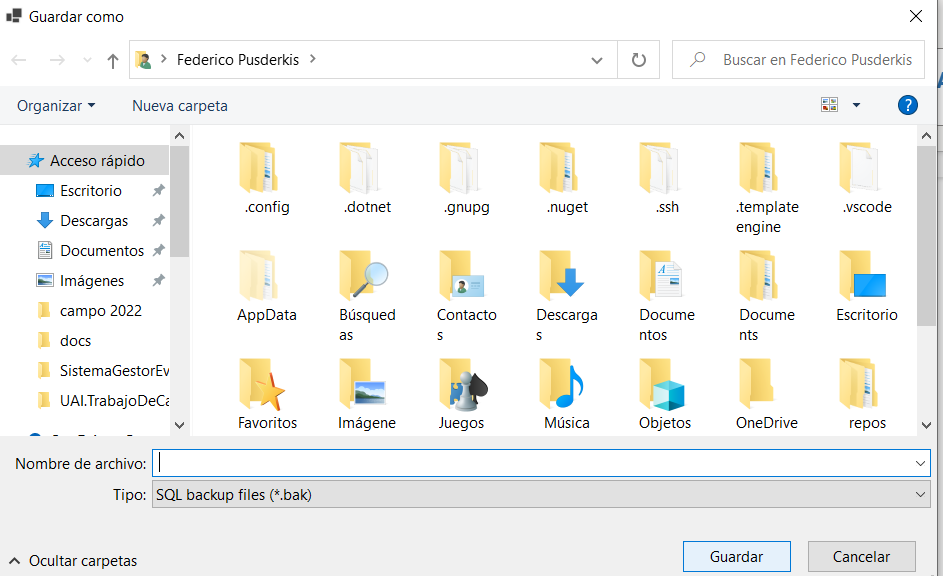
# T07. Gestión de Backup / Restore

El sistema permite la generación de Backups y Restores de la base de datos, con el objetivo de preservar la información almacenada.

Ambas operaciones solo son posibles teniendo el rol ADMINISTRADOR y se pueden acceder desde el menu “Administrador” --> “Generar Backup” / “Restaurar backup”



En ambos casos, es posible buscar el archivo dentro de la estructura de directorios del equipo local.



A continuación, se presenta el diagrama de casos de uso:



### T07BAC01 Generar backup

|  |  |
| --- | --- |
| **Nombre** | T07BAC01 Generar backup |
| **Objetivo** | Crea una copia de seguridad de la base de datos. |
| **Precondiciones** | El usuario es ADMINISTRADOR |
| **Postcondiciones** | Se genera un backup de la base de datos. |
| **Eventos disparadores del caso de uso** | El administrador presiona el botón “Generar backup” |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El ADMINISTRADOR presiona el boton “Generar backup”. |  |
| 1. El sistema le muestra la pantalla de guardar archivo del sistema operativo. |  |
| 1. El ADMINISTRADOR selecciona el lugar donde guardar el backup y un nombre. |  |
| 1. El sistema invoca el CU T06ABIT01 con el mensaje “se procede a realizar un backup en el directorio ${directorio}”. Luego invoca la generación de un backup a la base de datos. Finalizado el mismo, muestra por pantalla un mensaje informativo de que el backup se realizo correctamente. |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** | N/A |
| **Prototipo de interfaz de usuario** |  |

### T07BAC02 Restaurar backup

|  |  |
| --- | --- |
| **Nombre** | T07BAC01 restaurar backup |
| **Objetivo** | restaura una copia de seguridad de la base de datos. |
| **Precondiciones** | El usuario es ADMINISTRADOR |
| **Postcondiciones** | El sistema vuelve a una versión anterior de la base de datos. |
| **Eventos disparadores del caso de uso** | El administrador presiona el botón “restaurar backup” |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

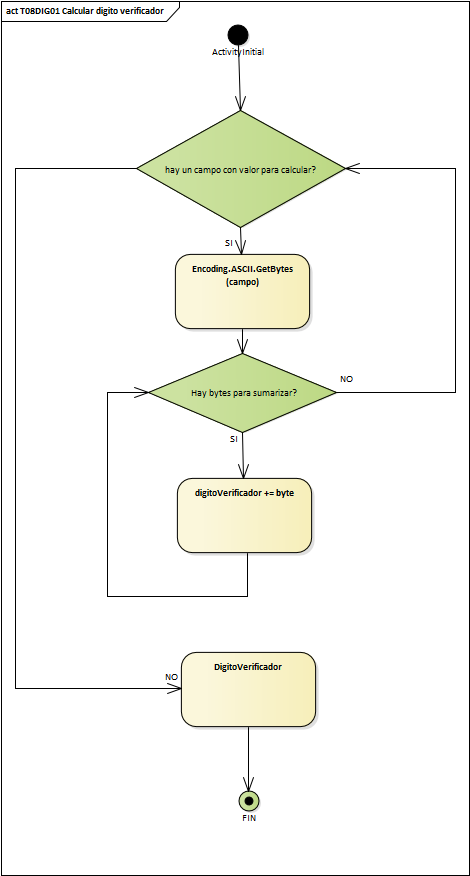
|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El ADMINISTRADOR presiona el boton “restaurar backup”. |  |
| 1. El sistema le muestra la pantalla de abrir archivo del sistema operativo. |  |
| 1. El ADMINISTRADOR selecciona el backup a restaurar y pulsa abrir. |  |
| 1. El sistema restaura el backup. Se invoca el CU T06ABIT01 con el mensaje “Backup restaurado con éxito: ${directorio}”. Se muestra un mensaje al usuario de restauración exitosa y se procede a cerrar el sistema. |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** | N/A |
| **Prototipo de interfaz de usuario** |  |

# T08. Gestión de Dígitos Verificadores

La función de los dígitos verificadores es la de permitir comprobar la integridad de los datos almacenados en la base de datos, a fin de poder detectar si estos fueron modificados por fuera del sistema.

Para el cálculo del digito verificador, se toma un conjunto de campos de un registro, se lo transforma en el tipo de dato string, obtenemos los bytes y los vamos sumando.

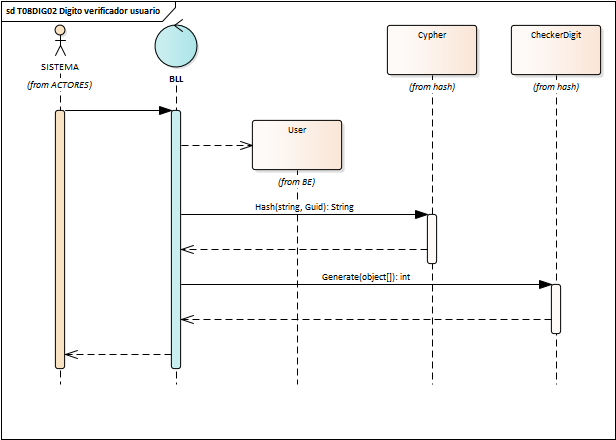


## **Digito Verificador Horizontal (DVH)**

El digito verificador horizontal se apoya en el algoritmo de creación de digito verificador para el control de integridad de las interfaces clave del sistema, se calcula cada vez que se guaran estas entidades almacenándose en la base de datos.

A su vez, se valida el digito verificador al cargar una entidad, en caso de ser incorrecto, se informa un error por pantalla al usuario indicando que el registro este corrupto y que deberá validar con un administrador lo sucedido.

Como ejemplo, se puede observar el cálculo de DVH de la entidad Usuario



Para el caso puntual de la entidad usuario, cuando se detecta que el registro esta corrupto es necesario contactar a un administrador para que realicé una evaluación de impacto y, de considerarlo necesario, desbloquear las credenciales del mismo.

Este proceso se puede ver con más detalle en la descripción del CU “[T04USU03 Desbloquear usuario](#_T04USU03_Desbloquear_usuario)“

## Digito Verificador Vertical (DVV)

El digito verificador vertical es sumamente útil para asegurar la integridad de entidades que no deban ser creadas manualmente, por ejemplo, la entidad “pagos”.

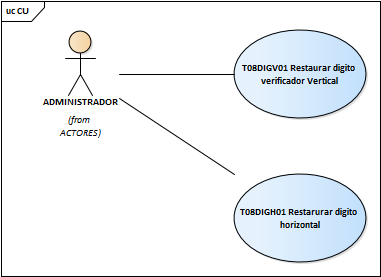
A nivel producto, permitir cargar manualmente pagos en la base de datos, puede entenderse como un intento de fraude ya que se podrían dar por saldadas deudas que siguen pendientes.

Por razones de performance, el DVV se volverá a calcular cada vez que un pago sea registrado, sumando la información del nuevo pago, logrando de esta manera mantenerlo actualizado.

Fórmula para el digito verificador vertical:

Una vez calculado el digito verificador vertical, se guarda en la tabla “INTEGRIDAD\_VERTICAL”.

A continuación, se documenta el diagrama de secuencia y su especificación:



### T08DIGH01 Restaurar digito horizontal

#### Historial de revisiones

|  |  |  |
| --- | --- | --- |
| Autor | Fecha | Comentarios |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| **Nombre** | T08DIGH01 Restaurar digito horizontal |
| **Objetivo** | Permite editar o crear un lugar, lo que trae por consecuencia la regeneración del DVH. |
| **Precondiciones** | El usuario inicio sesión en el sistema y tiene el rol ORGANIZADOR. Existe el evento creado y confirmado. |
| **Postcondiciones** | Se guarda un lugar con el DVH y el DVV correctos. |
| **Eventos disparadores del caso de uso** | Se intenta guardar un lugar |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El ORGANIZADOR ingresa todos los datos de la entidad EventRoom y presiona guardar. |  |
| 1. El SISTEMA valida: - Se ingreso un nombre - Se ingreso un precio - Se ingreso una capacidad - Se ingreso un tiempo por turno | 2.1 No supero las validaciones. Se invoca el caso de uso T06ABIT01 con el mensaje “Se intento guardar un lugar, pero fallo por errores de validación.” Se notifica al usuario los errores de validación. FIN DEL CASO DE USO |
| 1. El sistema valida la integridad verticual, calcula el DVH y guarda el EventRoom. | 3.1 Valido previamente la integridad vertical. El sistema recalcula la integridad verticual para que sea consistente con la última modificación. |
| 1. Notifica al usuario de que se guardo correctamente. Invoca el CU BIT06A01 Escribir bitácora con el mensaje “Se guardo el lugar id: " + eventRoom.Id” |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### T08DIGH01 Restaurar digito vertical

#### Historial de revisiones

|  |  |  |
| --- | --- | --- |
| Autor | Fecha | Comentarios |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| **Nombre** | T08DIGV01 Restaurar digito vertical |
| **Objetivo** | Restaurar la integridad en el sistema para continuar operando. |
| **Precondiciones** | El usuario inicio sesión en el sistema y tiene el rol ADMINISTRADOR. Existe el evento creado y confirmado. |
| **Postcondiciones** | Se genera un DVV correcto. |
| **Eventos disparadores del caso de uso** | El administrador presiona el botón “Regenerar DVV” en la pantalla de lugares |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El ADMINISTRADOR presiona el botón generar DVV |  |
| 1. El sistema recupera el listado completo de lugares, valida que todos cumplan con la integridad horizontal. | 2.1 Hay lugares que no validan integridad. El sistema informa al usuario el listado de lugares que no validan integridad. Fin del caso de uso. |
| 1. El sistema recalcula el DVV y lo guarda en la tabla de control “INTEGRIDAD\_VERTICAL”. Se invoca el CU T06BITA01 Escribir bitacora con el menssaje: "Se regenero el DVV de la entidad lugares" |  |
|  |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

# N0 Procesos de Negocio

Un proceso de negocio es una secuencia de actividades necesarias en una organización para cumplir con su objetivo comercial.

En los próximos apartados se documentarán los procesos de negocio incluidos en el Sistema Gestor de eventos de forma tal que pueda llevarse a cabo la implementación del sistema brindando claridad a todos los involucrados sobre “qué” y “cómo” se resuelve la necesidad de gestionar eventos.

Tener en cuenta que se dividirá en dos bloques:

\* N01: Especificación general del proceso de negocio

\* N02: Especificación detallada de los casos de uso del sistema.

# N01 Especificación funcional por proceso de negocio

*Se identifican los requisitos funcionales de valor para el sistema y se realiza el modelo de Casos de Uso de Negocio para formalizar, identificando en este punto todos los roles intervinientes.*

*Por cada proceso de negocio se detallará:*

*A. Identificación de Roles intervinientes*

*B. Descripción funcional del proceso: Entrada / Comportamiento / Salida: Se formaliza el RF que describe al Caso de Uso de Negocio.*

*C. Diagrama de Proceso*

*D. Modelo Conceptual: Se realiza el modelo conceptual o de dominio por medio de un diagrama de clases. Las entidades principales son los flujos de información que surgen dentro del DA.*

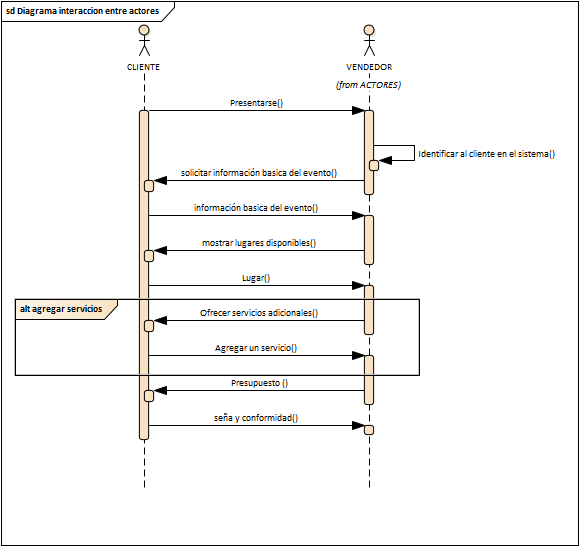
## N01PN01 Vender eventos privados

1. El vendedor consulta en el sistema si el cliente se encuentra dado de alta por eventos o presupuestos anteriores buscando por el identificador fiscal.
2. Si no encuentra al cliente, el vendedor procede a registrarlo dejando asentada la información mínima necesaria:  
   Nombre, Apellido, Identificador fiscal (CUIL), teléfono de contacto, mail de contacto, CUIT (Opcional, para casos donde vengan representado una empresa), Razón Social.
3. El vendedor solicita al cliente la información básica para comenzar a gestionar el evento:
   1. Objetivo descrito por el cliente.
   2. Presupuesto estimado (mínimo / máximo)
   3. Cantidad de invitados
   4. Categoría de evento (Cumpleaños, Casamiento, evento corporativo, presentación de producto, Otros).
   5. Fecha tentativa de realización (desde, hasta)
   6. Pedidos especiales.
4. El vendedor le muestra el catálogo impreso de salones al cliente para seleccionar uno. Una vez seleccionado agrega el lugar al evento.
5. El vendedor ofrece servicios adicionales al cliente entre los que tiene en su maestro de proveedores, por cada servicio adicional el vendedor especifica la cantidad de unidades contratadas y si hubiera alguna consideración adicional.
6. El vendedor le genera un presupuesto completo al cliente.
7. El cliente revisa el presupuesto, si está de acuerdo abona la seña (correspondiente al 50% del valor total), firma el contrato de contratación y el vendedor le da curso al resto de las tareas de organización del evento.

### N01PN01 Roles intervinientes

En el proceso de venta solamente hay un actor / rol: el VENDEDOR

Para modelar con mayor claridad el proceso de negocio, se incluirá el cliente con las interacciones con el vendedor.



**VENDEDOR:** Es quien cuenta con habilidades blandas para realizar una venta, y el encargado de cargar el presupuesto inicial en el sistema con todo lo requerido para la organización del evento.

**CLIENTE:** El cliente es quien interactúa con el vendedor brindándole información para la generación del evento.

Se debe tener en cuenta que el cliente no interactúa directamente con el sistema, sino que lo hace a través del vendedor.

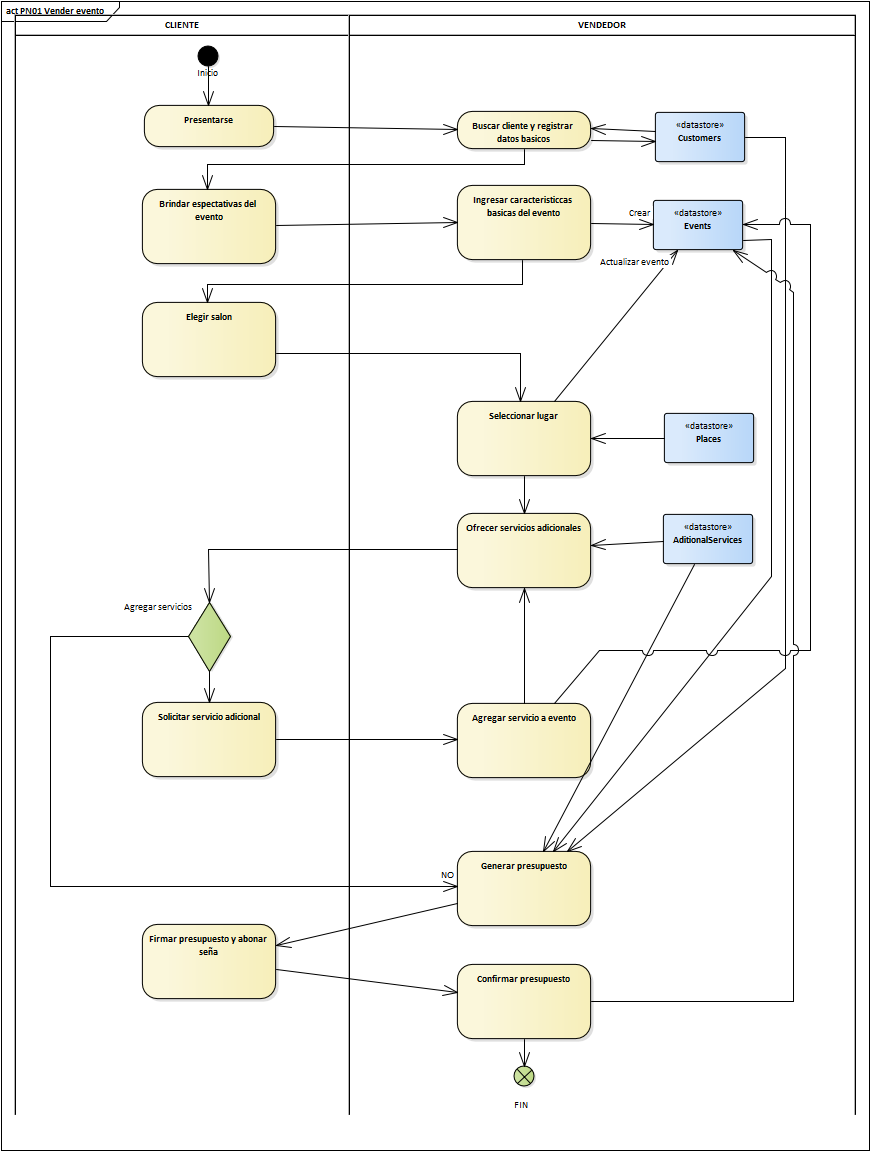
### N01PN01 Diagrama de ECS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENTRADA** |  | **COMPORTAMIENTO** |  | **SALIDA** |
| Cargar información basica del evento (titulo, descripción, pedidos especiales, cant. invitados, fecha, lugar, servicios) |  | El sistema toma la información, la almacena en la base de datos y genera un evento presupuestado |  | Evento presupuestado ok |
| Registrar pago en el evento |  | El sistema registra el pago, descuenta del saldo pendiente de pago. Si cubrio el minimo permite confirmar el evento. |  | Incrementa el monto abonado por el cliente. |
| Confirmar evento |  | Se confirma el evento y queda pendiente de gestionarse. |  | Evento confirmado |
| Registro del cliente |  | Se registran los datos del cliente |  | Registro guardado ok. |

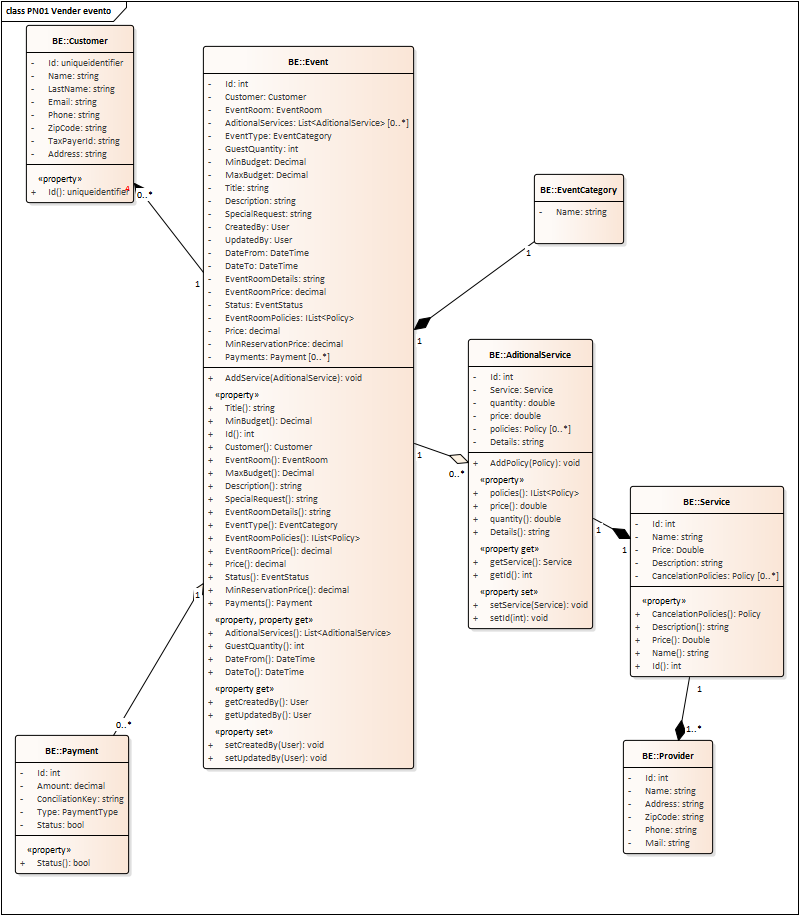
### N01PN01 Especificación funcional por proceso de negocio

Se recuerda que el cliente es un actor EXTERNO que no interactúa con el sistema.

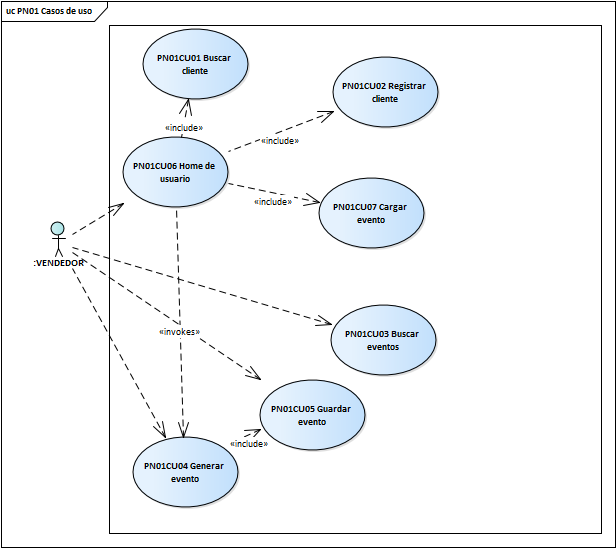
Se incluye dentro del diagrama de proceso para aportar mayor claridad al mismo.



### N01PN01 Modelo conceptual



### N01PN01 Diagrama de casos de uso



## PN02 Organizar evento

Una correcta organización de eventos permite asegurar la calidad de los entregables, teniendo distintas actividades a realizar.

Para garantizar la calidad es que se debe ser muy meticuloso y ordenado coordinando cada uno de los involucrados. Este proceso es ejecutado por un ORGANIZADOR.

1. El ORGANIZADOR es el encargado de realizar una entrevista profunda con el cliente, para relevar al detalle las particularidades del evento que no hayan sido correctamente relevadas en la venta del mismo.

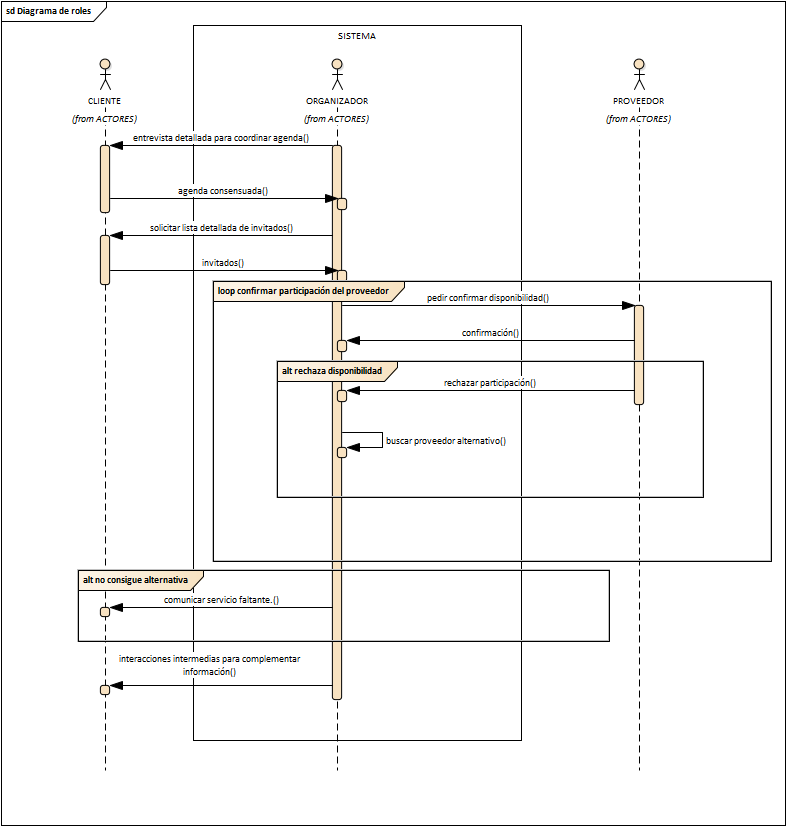
De la interacción con el cliente es deseable obtener:

* Listado de invitados, por cada invitado se debe contar con el Nombre, apellido, identificador fiscal, correo electrónico.
* Orden de ejecución de las actividades del evento. Por ejemplo:  
  *Se citan los invitados media hora antes del evento y se los hace aguardar afuera. Luego se les permite el ingreso a las 21 hs hasta las 22* *sirviéndoles bebida y un* *tentempié.*  
  *A las 22 aparece el presentador del evento convocando a los invitados y brindando un discurso que introduzca al CEO de la compañía.*  
  *Se estima que la presentación durará hasta las 22:45 y luego se dara lugar a un intervalo de preguntas y respuestas organizado por el cliente.*  
  *A las 23 se realizará un show musical hasta las 24 hs*  
  *Luego del show, se abren las mesas y retiran las sillas para invitar a los participantes a bailar. Se habilita la barra libre y se retiran los mozos.*  
  *A las 2:00 hs se comienza la invitación a retirarse a los invitados y se da por finalizado el evento.*

1. Por cada servicio contratado, el organizador se pone en contacto con los proveedores contratados para confirmar la reserva. En caso de que el proveedor rechace su participación, el ORGANIZADOR buscara un proveedor alternativo.  
   Si no consigue un proveedor para satisfacer la necesidad del cliente, este le comunicará que no podrá disponer del servicio.  
   Tener en cuenta que la única acción tipificada y diferencial de cada interacción con el proveedor será cuando confirma participación.  
   En caso de que el precio se incremente significativamente y supere la comisión estipulada para el servicio, el organizador se contactara con el cliente para ofrecerle el cambio del proveedor con el nuevo precio.
2. El ORGANIZADOR mantendrá comunicación constante con el cliente para afinar detalles dejando registro de cada interacción.

### N01PN02 Roles intervinientes

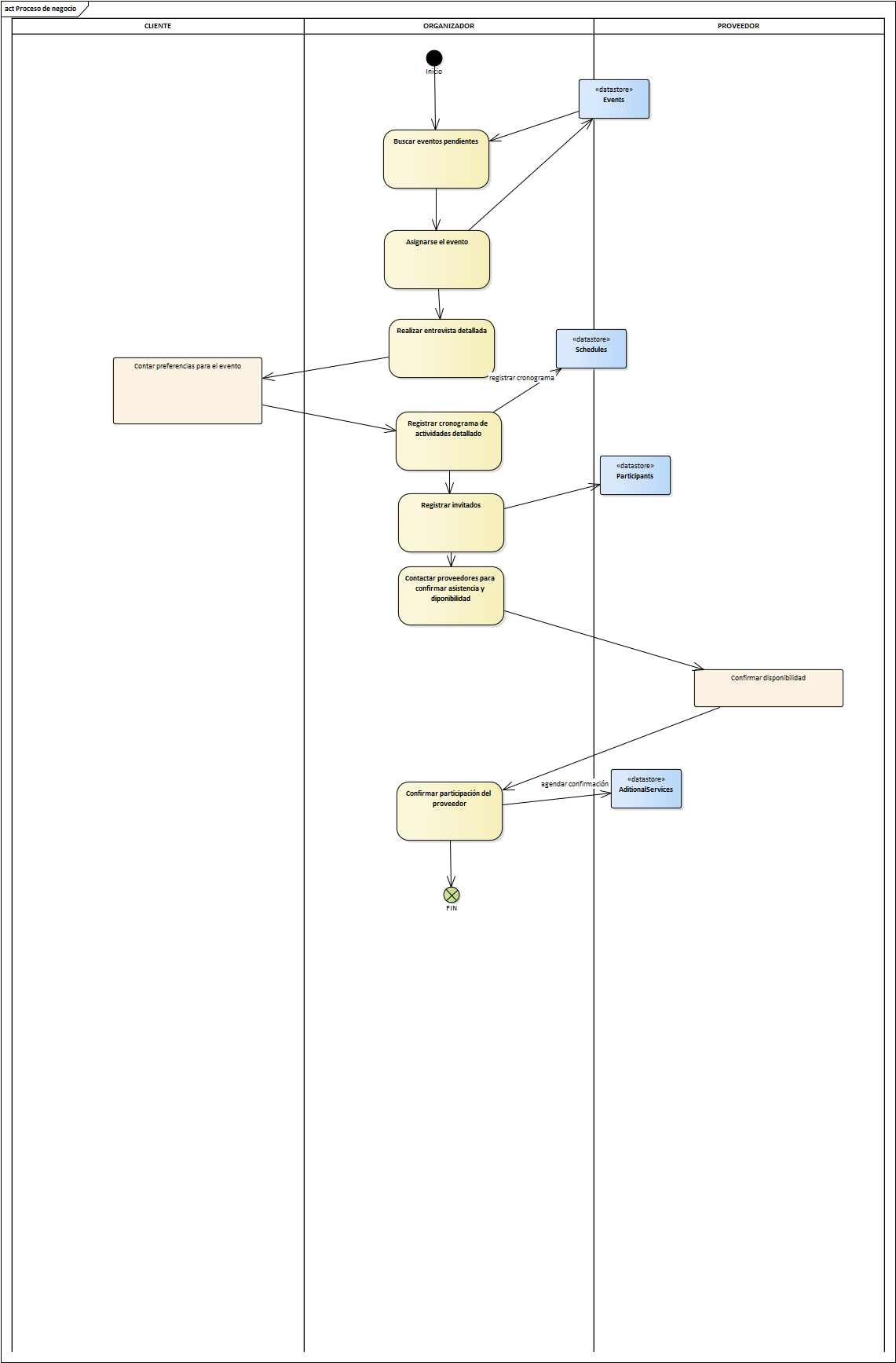
Tener en cuenta que el único actor que interactúa con el sistema es el ORGANIZADOR, el resto de los actores son incluidos con la finalizad alcanzar una mayor comprensión del proceso de negocio.



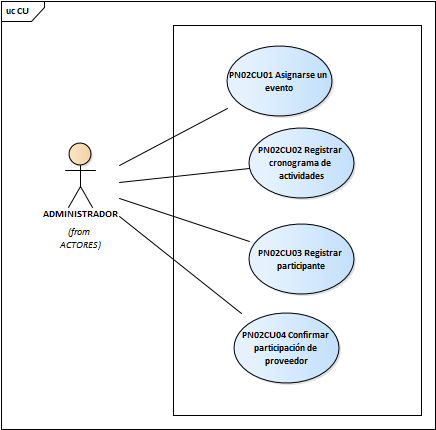
### N01PN02 Diagrama de ECS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENTRADA** |  | **COMPORTAMIENTO** |  | **SALIDA** |
| Características entrevistas de entendimiento |  | Almacenamiento de cronograma de actividades |  | Cronograma de actividades |
| Lista de invitados |  | Registro de invitados |  | Lista de invitados habilitados a participar |
| Confirmación de proveedor |  | Cambiar estado de servicio a CONFIRMADO |  | Se notifica proveedor confirmado |
| Proveedor de reemplazo |  | Cancelar la contratación con el proveedor y registrar una nueva en estado CONFIRMADO |  | Se notifica proveedor confirmado |
| Cancelación servicio |  | Se cancela la contratación dejando saldo a favor en el evento para el cliente. |  | Servicio cancelado |
| Interacción intermedia con el cliente |  | Se registra un resumen de lo conversado. |  | Registro guardado ok. |

### N01PN02 Diagrama de proceso de negocio



### N01PN02 Diagrama de casos de uso



## N02 Casos de uso del proceso de negocio

### PN01CU01 Buscar cliente

#### Historial de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Historial de revisión | | | |
| Versión | Fecha | Nombre | Comentarios |
| 1.0 |  |  | Versión inicial |
|  |  |  |  |

|  |  |
| --- | --- |
| **Nombre** | PN01CU01 Buscar cliente |
| **Objetivo** | Buscar los clientes |
| **Precondiciones** | El usuario inicio sesión |
| **Postcondiciones** | Se visualiza en una grilla los clientes encontrados. |
| **Eventos disparadores del caso de uso** | El usuario desea buscar los eventos. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El VENDEDOR ingresa los parámetros de búsqueda (CUIT,Nombre, Mail) y pulsa el botón buscar. |  |
| 1. El sistema busca en la base de datos por CUIT y devuelve los resultados desencriptados. | 2.1. No se informó el cuit, pero si el Mail. El sistema busca por mail y devuelve los resultados desencriptados  2.2 No se informó cuit ni mail. El sistema busca por nombre y apellido y devuelve los resultados desencriptados. |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### PN01CU02 Registrar cliente

#### Historial de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Historial de revisión | | | |
| Versión | Fecha | Nombre | Comentarios |
| 1.0 |  |  | Versión inicial |
|  |  |  |  |

|  |  |
| --- | --- |
| **Nombre** | PN01CU02 Registrar cliente |
| **Objetivo** | Registrar un nuevo cliente al sistema |
| **Precondiciones** | El usuario inicio sesión |
| **Postcondiciones** | Se visualiza en una grilla los distintos eventos encontrados. |
| **Eventos disparadores del caso de uso** | El usuario desea buscar los eventos. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El VENDEDOR ingresa los datos del cliente (CUIT/CUIL, mail, dirección, teléfono, nombre, apellido) y pulsa el botón guardar. |  |
| 1. El sistema valida que todos los campos estén completos | 2.1 hay campos incompletos, el sistema devuelve un error mencionando los campos vacíos. Fin del caso de uso. |
| 1. El sistema encripta el CUIL/CUIT y busca que no exista un cliente distinto registrado con ese CUIT/CUIL. | 3.1 Existe un cliente, el sistema muestra un error al usuario. Fin del caso de uso. |
| 1. El sistema encripta el MAIL y valida si existe un cliente con ese mismo correo. | 4.1 Existe un cliente. El sistema muestra un mensaje de error. Finaliza el caso de uso. |
| 1. El sistema encripta el resto de la información sensible y persiste en la base de datos. Se escribe en la bitácora que se guardó el cliente (T06BITA01). Se notifica al usuario de la operación exitosa. |  |
|  |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### PN01CU03 Buscar evento

#### Historial de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Historial de revisión | | | |
| Versión | Fecha | Nombre | Comentarios |
| 1.0 |  |  | Versión inicial |
|  |  |  |  |

|  |  |
| --- | --- |
| **Nombre** | PN01CU03 Buscar eventos |
| **Objetivo** | Buscar los eventos registrados |
| **Precondiciones** | El usuario inicio sesión. Opcional: Se invoco el caso de uso PN01CU01 y se seleccionó un cliente. |
| **Postcondiciones** | Se visualiza en una grilla los distintos eventos encontrados. |
| **Eventos disparadores del caso de uso** | El usuario desea buscar los eventos. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El VENDEDOR completa los criterios de búsqueda que desee, así como también selecciona un cliente y pulsa el botón buscar. |  |
| 1. El sistema busca en la base de datos los eventos relacionados y los lista por proximidad a la fecha de realización y estado. |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### PN001CU04 Generar evento

#### Historial de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Historial de revisión | | | |
| Versión | Fecha | Nombre | Comentarios |
| 1.0 |  |  | Versión inicial |
|  |  |  |  |

|  |  |
| --- | --- |
| **Nombre** | CU04 Generar evento |
| **Objetivo** | Pretende generar un evento completo listo para ser organizado. |
| **Precondiciones** | Ya se encuentra el cliente registrado en el sistema. |
| **Postcondiciones** | Se genera un evento en estado en estado SOLICITADO. |
| **Eventos disparadores del caso de uso** | El vendedor busca iniciar un nuevo evento. |
| **Puntos de extensión y condiciones** | no aplica |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El VENDEDOR carga la información básica del evento: titulo, descripción, presupuesto mínimo / máximo, fecha y hora desde/hasta, tipo de evento, cantidad de invitados y pedidos especiales. |  |
| 1. El VENDEDOR ingresa el código de lugar / nombre y presiona buscar. |  |
| 1. El sistema busca los lugares con disponibilidad utilizando la fecha de inicio y fin y los lista. |  |
| 1. El vendedor selecciona un lugar y presiona SELECCIONAR. |  |
| 1. El sistema agrega el lugar al evento y recalcula los montos. |  |
| 1. El VENDEDOR escribe el nombre del servicio y presiona buscar. |  |
| 1. El sistema lista los servicios. |  |
| 1. El VENDEDOR selecciona el servicio a agregar y la cantidad de unidades a agregar. Luego pulsa el botón Agregar. El sistema recalcula los montos. |  |
| 1. El VENDEDOR selecciona “Presupuestar” |  |
| 1. El sistema valida que la información es correcta verificando:  - Titulo != null  - Cantidad de invitados > 0  - Presupuesto Max >= presupuesto min > 0  - Fecha inicio > hoy  - Feca Fin < Fecha Inicio  - Tipo de evento seleccionado  - Que haya seleccionado un lugar. Luego guarda el evento con estado BUDGET y habilita la pestaña de “Pagos” | 10.1 Hay datos inválidos. Se notifica al usuario con un mensaje por pantalla. |
| 1. El VENDEDOR informa por cada pago el metodo de pago (CASH, TARJETA DE CREDITO, DEBITO, CRYPTO), monto, clave de conciliación y fecha de pago. |  |
| 1. El sistema valida que todos los campos tengan datos y registra el pago. Actualiza los montos del evento. | 12.1 Hay datos inválidos / faltantes: el sistema informa por un mensaje de error para que sean corregidos. |
|  |  |
|  |  |
| 1. El vendedor presiona el botón “Aprobar y confirmar”. |  |
| 1. El sistema valida que haya abonado al menos el %50 del presupuesto y guarda el evento en estado “Confirmado”. Escribe en la bitacora. | 14.1 No abono la seña minima: se notifica que debe abonar la seña minima. |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

#### 

### PN01CU05 Guardar evento

|  |  |  |  |
| --- | --- | --- | --- |
| Historial de revisión | | | |
| Versión | Fecha | Nombre | Comentarios |
| 1.0 |  |  | Versión inicial |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
| **Nombre** | PN01CU05 Guardar evento |
| **Objetivo** | Guarda el evento completo en la base de datos. |
| **Precondiciones** | Ninguna |
| **Postcondiciones** | El evento queda guardado en la base de datos. Si es la primera vez, se genera el número de evento. |
| **Eventos disparadores del caso de uso** | El vendedor desea guardar el evento, o bien un flujo requiere guardarlo. |
| **Puntos de extensión y condiciones** | no aplica |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El usuario pulsa la opción “GUARDAR” |  |
| 1. El sistema valida el evento completo, si cumple las condiciones lo guarda. Condiciones: 1. Si el evento no está guardado previamente, valida que la fecha sea posterior. 2. La fecha desde debe ser anterior a la fecha hasta y ambas deben ser informadas. 3. El presupuesto máximo no debe ser menor a 0. 4. El presupuesto mínimo debe ser mayor que el máximo. 5. Se debe indicar la tipificación del evento. Por cada servicio adicional: 1. Cantidad mayor o igual a 1 2. El precio no debe ser menor al costo del proveedor. | 2.1 Hay errores de validación, el sistema los traduce utilizando el módulo de multi idioma y lista los errores por pantalla para que el usuario los corrija. Fin del caso de uso. |
| 1. El sistema persiste el evento en la base de datos y notifica al usuario que se guardó correctamente. |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### 

### 

### PN01CU06 Home del vendedor

#### Historial de revisiones

|  |  |  |
| --- | --- | --- |
| Autor | Fecha | Comentarios |
|  |  |  |
|  |  |  |

#### 

|  |  |
| --- | --- |
| **Nombre** | PN01CU06 Home del vendedor |
| **Objetivo** | Brindar una pantalla central para que el vendedor pueda buscar clientes, darlos de alta, buscar eventos, editarlos / crearlos. |
| **Precondiciones** | El vendedor inicio sesión en el sistema. |
| **Postcondiciones** | El vendedor pudo operar. |
| **Eventos disparadores del caso de uso** | El vendedor desea atender un cliente |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El VENDEDOR ingresa los criterios de busqueda del cliente y presiona buscar. |  |
| 1. Se invoca el CU “PN01CU01 Buscar cliente” |  |
| 1. El vendedor selecciona el cliente y pulsa buscar. |  |
| 1. Se invoca el CU “PN01CU03 buscar eventos” |  |
| 1. El vendedor presiona “Nuevo” | 5.1 No hay ningun cliente seleccionado. El sistema advierte al usuario que no hay cliente seleccionado y solicita buscar / seleccionar uno. **Fin del caso de uso.**  5.2 El vendedor selecciona un evento y pulsa el botón “Abrir”. Se invoca el CU “PN01CU07 Cargar evento completo” y se deriva al cliente al CU PN01CU04 Generar evento. Fin del caso de uso. Se registra en bitacora. |
| 1. El sistema crea un evento nuevo e invoca el CU “PN01CU04 Generar evento”. Se registra en la bitacora. |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### PN01CU07 Cargar evento

#### Historial de revisiones

|  |  |  |
| --- | --- | --- |
| Autor | Fecha | Comentarios |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| **Nombre** | PN01CU07 Cargar evento |
| **Objetivo** | Cargar evento completo. |
| **Precondiciones** | El vendedor inicio sesión en el sistema. Existe el evento creado |
| **Postcondiciones** | El evento se carga completo. |
| **Eventos disparadores del caso de uso** | El vendedor desea editar/visualizar un evento. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El VENDEDOR selecciona un evento y presiona abrir |  |
| 1. El sistema carga el evento junto con todas sus dependencias: \* Servicios adicionales \* Lugar \* Detalles \* Cliente relacionado \* Usuario que lo creo y el último que lo modifico. |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### PN02CU01 Asignarse un evento

#### Historial de revisiones

|  |  |  |
| --- | --- | --- |
| Autor | Fecha | Comentarios |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| **Nombre** | PN02CU01 Asignarse un evento |
| **Objetivo** | Buscar eventos sin atención, o bien que tenga asignado el usuario para darle seguimiento. |
| **Precondiciones** | El vendedor inicio sesión en el sistema. Existe el evento creado y confirmado. |
| **Postcondiciones** | Se muestra la lista de eventos |
| **Eventos disparadores del caso de uso** | El usuario ORGANIZADOR busca un evento nuevo para organizar. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El ORGANIZADOR selecciona un evento y pulsa el botón asignar. | 1.1 El usuario busca un evento que ya esta asignado y pulsa el botón “Desasignar”  1.2 El sistema borra la asignación del usuario anterior. Se invoca el CU “T06BITA01 Escribir en bitacora” con el mensaje "Se desasigno el usuario {selectedEvent.Assigned} al evento {selectedEvent.Id}". El sistema muestra un mensaje confirmado la operación. Continua en el paso 3. |
| 1. El sistema asigna el usuario logueado al evento. Se invoca el CU T06BTA01 Escribir bitacora con el mensaje “$"Se asigno el usuario al evento {selectedEvent.Id}". El sistema muestra un mensaje confirmando la operación. |  |
| 1. Se invoca el caso de uso PN01CU03 Buscar eventos |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### PN02CU02 Registrar cronograma de actividades

#### Historial de revisiones

|  |  |  |
| --- | --- | --- |
| Autor | Fecha | Comentarios |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| **Nombre** | PN02CU02 Registrar cronograma de actividades |
| **Objetivo** | Permitirle al usuario ORGANIZADOR dejar un registro escrito de como se ejecutará el evento. |
| **Precondiciones** | El vendedor inicio sesión en el sistema. Existe el evento creado y confirmado. Se cargo el evento invocando el CU “PN01CU07 Cargar evento” |
| **Postcondiciones** | Se guarda un registro en formato texto indicando el orden de ejecución de cada actividad en el evento. |
| **Eventos disparadores del caso de uso** | El usuario ORGANIZADOR registra la cronología acordada con el cliente. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El ORGANIZADOR El organizador ingresa las actividades a realizarse en el evento y pulsa el bot{on Guardar. |  |
| 1. El sistema guarda las actividades, por cada actividad genera un registro en bitacora invocando el CU T06BITA01 con el mensaje $"Se guarda la actividad {activity.Id}-{activity.Description} asociada al evento {eventId}". Una vez finalizado el guardado, registra en bitacora que se guardo ok con el mensaje "Se guardo la cronología para el evento {eventId}" y notifica al usuario del guardado ok. |  |
|  |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### PN02CU03 Registrar participante

#### Historial de revisiones

|  |  |  |
| --- | --- | --- |
| Autor | Fecha | Comentarios |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| **Nombre** | PN02CU03 Registrar participante |
| **Objetivo** | Permitirle al usuario ORGANIZADOR dejar un registro escrito de la lista de invitados. |
| **Precondiciones** | El usuario inicio sesión en el sistema y tiene el rol ORGANIZADOR. Existe el evento creado y confirmado. Se cargo el evento invocando el CU “PN01CU07 Cargar evento”. |
| **Postcondiciones** | Se guarda un registro de los participantes del evento. |
| **Eventos disparadores del caso de uso** | El usuario ORGANIZADOR desea registrar los invitados. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El ORGANIZADOR ingresa los datos del participante y pulsa agregar.  **DATOS:** CUIT, Nombre, apellido, tipo comida (indistinto, vegetariano, vegano, celiaco), cantidad de acompañantes y detalles adicionales (por ejemplo: “viene con chicos menores de 2 años”) |  |
| 1. El sistema valida no exceder cantidad de participantes contratados ni la capacidad máxima del lugar. Agrega el participante a la lista de invitados. Se Invoca el CU T06BITA01 con el mensaje “Se guardo el participante para el evento ${eventId}”. | 2.1 Excede la capacidad contratada  2.1.1 El sistema advierte al usuario que podrían aplicar cargos adicionales una vez finalizado el evento.  2.1.2 El usuario acepta la advertencia.  2.1.3 El sistema agrega el invitado a la lista. Se Invoca el CU T06BITA01 con el mensaje “Se guardo el participante ADICIONIAL para el evento ${eventId}”.  2.2 Excede la capacidad del salón  2.2.1 El sistema indica que no es posible exceder la capacidad del salón. Se Invoca el CU T06BITA01 con el mensaje “No se pudo agregar el participante al evento ${EventId}: capacidad de ${roomCapacity} participantes excedida”. |
|  |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |

### PN02CU04 Confirmar participación de proveedor

#### Historial de revisiones

|  |  |  |
| --- | --- | --- |
| Autor | Fecha | Comentarios |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| **Nombre** | PN02CU04 Confirmar participación de proveedor |
| **Objetivo** | Permitirle al usuario ORGANIZADOR dejar un registro escrito de la lista de invitados. |
| **Precondiciones** | El usuario inicio sesión en el sistema y tiene el rol ORGANIZADOR. Existe el evento creado y confirmado. Se cargo el evento invocando el CU “PN01CU07 Cargar evento”.  El evento tiene servicios vinculados. |
| **Postcondiciones** | Se guarda un registro de los participantes del evento. |
| **Eventos disparadores del caso de uso** | El usuario ORGANIZADOR desea registrar los invitados. |
| **Puntos de extensión y condiciones** |  |
| **Gráfico del CU** |  |

|  |  |
| --- | --- |
| **Camino principal** | **Flujo alternativo** |
| 1. El ORGANIZADOR ingresa el precio acordado con el proveedor al momento de confirmar, edita los comentarios y presiona “Confirmar”. |  |
| 1. El SISTEMA valida el precio acorado con el proveedor, si es inferior al acordado con el cliente se confirma el servicio. Se invoca el CU BIT=6A01 Escribir Bitácora con el mensaje “"Se confirmo el servicio ${nombre} para el evento {eventId}" | 2.1 El precio excede el acordado. El sistema notifica un error al usuario.  Se invoca el CU de uso BIT06A01 con el mensaje "No se confirmó el servicio ${nombre} para el evento {eventId} ya que excede el monto abonado por el cliente." |
|  |  |
| **FIN DEL CASO DE USO** | |

|  |  |
| --- | --- |
| **Diagrama de clases** |  |
| **Diagrama de secuencia** |  |
| **DER** |  |
| **Prototipo de interfaz de usuario** |  |