****

**ADMINISTRACIÓN HOTELERA**

**Sandra Gargiulo, Guillermo Obaya, Rodrigo Parada, Alejandro Pérez, Marcelo Wolf.**

**Cátedra Proyecto de Software**

**Licenciatura en Sistemas**

**2018**

*Administración Hotelera(AH)*

Sandra Gargiulo, Guillermo Obaya, Rodrigo Parada, Alejandro Pérez, Marcelo Wolf

Licenciatura en Sistemas – Departamento de Desarrollo Productivo y Tecnológico

Universidad Nacional de Lanús

Extracto- En el presente documento se incluyen las actividades concernientes al proyecto denominado “Administración Hotelera” (de ahora en más “AH”), perteneciente a la cátedra de Proyecto Software, de la carrera de Licenciatura en Sistemas de la Universidad Nacional de Lanús. El objetivo principal del proyecto es proveer de herramientas informáticas, mediante la creación de un software debidamente adaptado, a la empresa “Hotel de Cine Las Golondrinas”, sito en la localidad de Villa Gesell. El mismo tendrá como objetivo la asistencia en la gestión integral del hotel y sus reservas, control interno de la empresa y explotación de los datos de los clientes, para poder ofrecer promociones y realizar diversas acciones de marketing.

*Palabras Clave: AH, Administración Hotelera, Explotación de la información, manejo integral del sistema, gestión de reservas, gestión de promociones, sistema hotelero, Universidad Nacional de Lanús.*

1. Requisitos del sistema.
   1. *Introducción.*

El planteo del problema surge de la necesidad de reemplazarun sistema ya obsoleto, a partir de nuevas necesidades de la gerencia de la empresa que desea contratar los servicios de este grupo de trabajo.

El software AH aparece como una opción para reemplazar las funcionalidades que tenía el sistema anterior y mejorar mediante nuevas implementaciones y conceptos el uso de los datos que se obtienen de los clientes, para crear acciones de marketing y estrategias de promociones.

Las principales características con las que contará AH serán la de poder explotar los datos que se obtengan mediante formularios que llenarán los clientes registrados, la gestión integral del sistema y la posibilidad de generar reservas.

* 1. *Requisitos funcionales.*

Los requisitos funcionales del proyecto se resumen en los siguientes ítems, de acuerdo a las necesidades del usuario o administrador y cliente o usuario externo. Desde ahora llamados simplemente administrador y cliente (ambos son usuarios per se, pero de distinta índole).

* Permitir al administrador gestionar las reservas enviadas por el cliente desde la plataforma web.
* Permite al administrador gestionar las diversas tareas que se realizan en el hotel por parte del personal.
* Permitir al cliente programar su estadía con antelación, ya que contará con información de precios, promociones y productos.
* Permitirá al administrador usufructuar los datos de los clientes para generar promociones y estadísticas que ayudaran a tomar decisiones de mercado a corto, mediano y largo plazo.
  1. *Requisitos no funcionales.*

Los requisitos no funcionales (los que restringen la funcionalidad del sistema), se enumeran a continuación:

* El maquetado y Front End será realizado con CSS, Modelize y HTML, la codificación será en Python y el diseño de la base de datos con MySQL.
* Con XaMPP y Easy PHP se harán las conexiones remotas al servidor.
* La documentación, testeo y programas de estimación a utilizar serán mediante COCOMO II, Gantt Proyect y TestLink, además de otras herramientas informáticas (Word, Power Point, Excel, Visio)
* Integración interna e invisible al usuario (administrador y cliente). Sólo podrá interactuar con la parte visual, que será la intermediaria entre el usuario y las entradas que ingrese.
* La aplicación será multiplataforma (escritorio y móvil)
* La misma podrá ser modularmente ampliable y fácilmente modificable ante nuevos escenarios de requerimientos o actualizaciones.

1. Procesos de gestión del proyecto.

Se definen las estrategias, estimaciones y planes a seguir para el desarrollo total del proyecto. También se muestran los planes de estimaciones que se irán ajustando a medida que el trabajo vaya avanzando (los costos y los tiempos de cierre siempre son sobre supuestos, se irán presentando tablas nuevas con las estimaciones pertinentes al caso.)

*2.1 Inicio del proyecto*

El mismo arranca con la conformación del grupo de trabajo y queda definido en la etapa más temprana del comienzo de las clases. Gracias a la posibilidad que otorga la cátedra de traer un proyecto de la vida real, se optó por esta posibilidad, ya que algunos integrantes contaban con propuestas reales y rentables para desarrollar un producto con un destinatario determinado.

*2.2 Recursos humanos. Funciones. Metodología de trabajo.*

Se eligió para el desarrollo de AH una organización de tareas que, según Marilyn Mantei, se denomina como *descentralizado controlado,* con las siguientes características:

* Tiene un jefe definido que coordina tareas específicas y jefes secundarios que tienen responsabilidades sobre subtareas.
* La solución de problemas es una actividad del grupo, pero la implementación de soluciones se reparte entre los subgrupos por el jefe de equipo.
* La comunicación entre subgrupos e individuos es horizontal.
* También hay comunicación vertical a lo largo de la jerarquía de control.

La persona a cargo del proyecto, elegida por el equipo de trabajo, es Sandra Gargiulo, que se encarga de la planificación y de la coordinación del equipo a la hora de repartir las tareas, como así también asistir a alguno de los integrantes cuando surge algún problema o necesita ayuda de cualquier índole. El responsable de la documentación y de su entrega a término es Marcelo Wolf, quién irá armando el documento entregable, como así también harála realización de estimaciones para conocer los costos y los tiempos que llevará la realización de AH. El responsable de la parte de diseño de prototipos y frontend es Alejandro Pérez, quién diseñará, en conjunto con el resto de los integrantes, las plantillas visuales que serán la interacción entre el usuario final y la funcionalidad del sistema. El responsable de la codificación del sistema, modelización y armado de la base de datos es Rodrigo Parada, que será quien le dé funcionalidad al programa. Y quién responderá ante la responsabilidad de las distintas pruebas de integración y funcionamiento será Guillermo Obaya. Pese a tener tareas definidas de antemano, el equipo se encuentra capacitado para realizar todas las tareas concernientes al proyecto.

*2.2.1 Responsables de tareas del proyecto AH.*

A continuación se detalla mediante un análisis pormenorizado las tareas y los responsables a cargo de las mismas. La dirección y la documentación del proyecto son trazables a través del mismo, de principio a fin, y sus responsables quienes estarán al tanto de toda modificación o progreso en las distintas etapas de desarrollo en sí (Figura 1).



*Fig1: cuadro con las tareas a realizar, su*

*tiempo estimado de realización y los*

*responsables de llevarlas a cabo.*

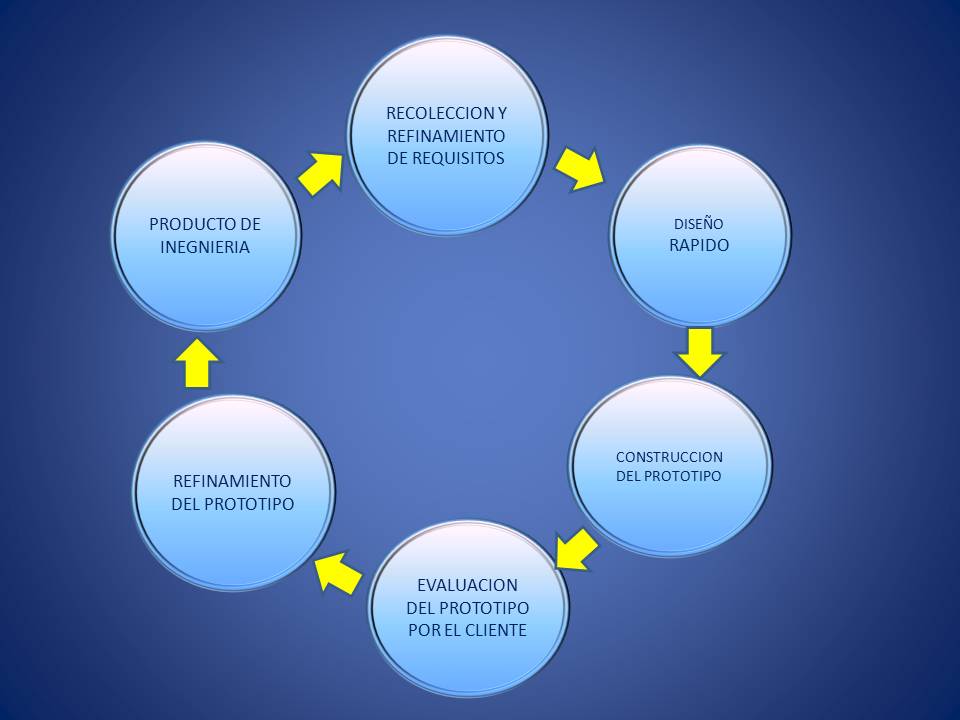
*2.3Modelo de ciclo de vida a utilizar en el proyecto.*

El modelo de ciclo de vida a utilizar en el proyecto es el del prototipado evolutivo (ver figura 2). El mismo se adapta a las necesidades de optimización de los tiempos, y por otro lado, pueden aparecer nuevos datos a tener en cuenta a la hora de ajustar los requisitos iniciales. En el prototipado evolutivo hay una ventaja sobre otros modelos de ciclo de vida a considerar:

• Mediante gráficos se pueden obtener datos de primera mano de parte del administrador que quizás no supo o no pudo explicar debidamente cuando se lo entrevistó.

• Hacia donde apuntar, tanto en el diseño como en la funcionalidad.

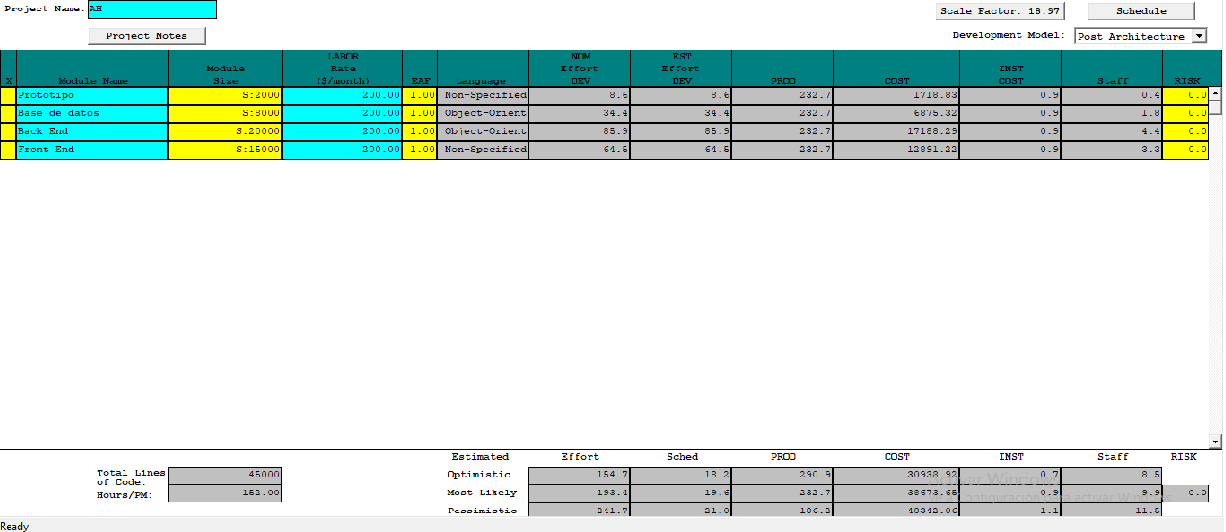
• Permite, en el mejor de los casos, adaptar el prototipo al funcionamiento final, y así ahorrar tiempo en el diseño gráfico del mismo, ya que un prototipo de calidad posiblemente pueda ser parte del producto terminado.



*Figura 2: Ciclo de vida de prototipado evolutivo.*

*2.4 Costos estimados calculados con COCOMO II*

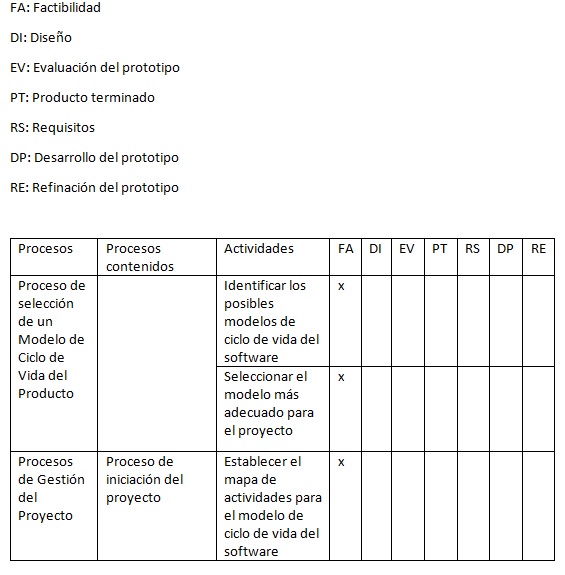
Se estimó esfuerzo y costos mediante el método de COCOMO II. (Fig 3)

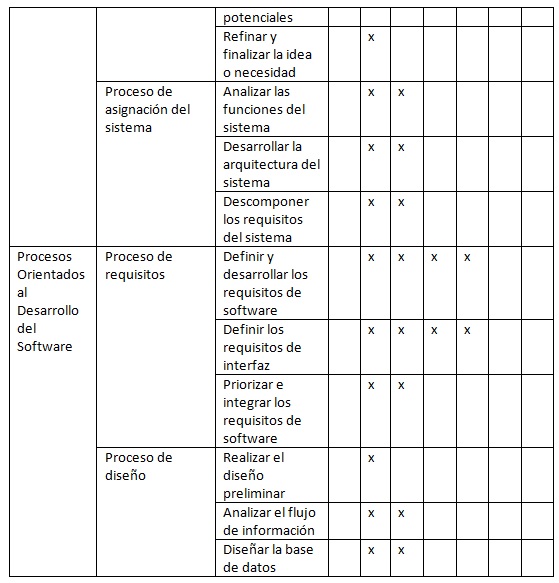
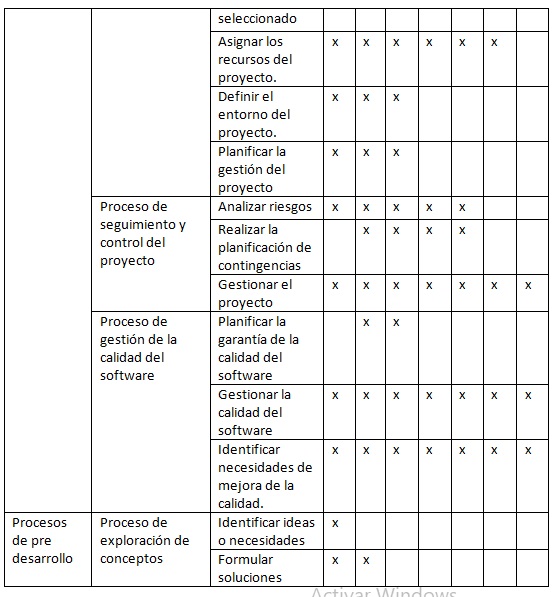


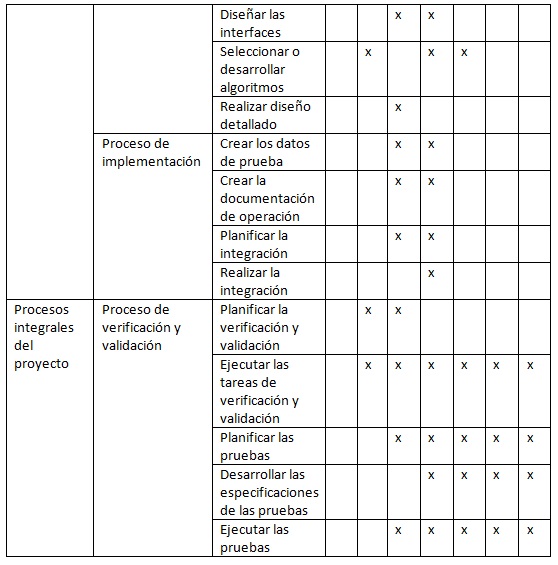
*Fig 3: estimación de recursos mediante COCOMO II*

*2.5 Mapa de actividades.*

Detallado en (Fig.4)







*Fig 4: Mapa de actividades.*

*2.6 Diagrama de Gantt*

A continuación el diagrama de Gantt con la línea de tiempo tentativa, la división de tareas y la trazabilidad de algunas de ellas a lo largo de la duración del ciclo de vida (Fig. 5). La fecha estipulada para el comienzo del proyecto es el día 7/08/18, y la finalización estaría estimada para el día 6/11/18.

*Fig 5: Diagrama de Gantt con la línea base, las tareas, los integrantes, los días tentativos de duración y las fechas estipuladas.*

*2.7 Análisis de riesgos y plan de contingencias.*

Se analizaron los siguientes riesgos a tener en cuenta a la hora de llevar adelante AH, y su plan de contingencias para salvar aquellos riesgos potenciales:

* Desconocimiento previo de algunas de las herramientas de programación, diseño y documentación, resuelto con la averiguación del manejo de las mismas en parelelo al desarrollo del proyecto.
* Posible baja de algún integrante del grupo, por eso se decidió utilizar la estrategia de liderazgo propuesta, ya que al ser trazables las tareas a realizar y hacer partícipes necesarios e importantes a todos los integrantes, se pueden retomar las tareas del desertor con naturalidad y con un mínimo de retraso, pero con una natural sobrecarga de trabajo.
* Cambios bruscos en los requisitos del cliente, riesgo cubierto con buenos prototipos y adecuados a las necesidades detectadas.
* Problemas de codificación y funcionales. Para ello se decidió hacer un testeo permanente de las funcionalidades y los requisitos de diseño para evitar errores a futuro, más difíciles de atacar y detectar.

*2.8 plan de garantía de calidad de software*

Para lograr el objetivo planteado en base a los requisitos presentados, se utilizará el patrón de diseño MVC (modelo vista controlador), ya que el producto a realizar está orientado a satisfacer las necesidades de la interacción de un usuario con la interfaz gráfica, tanto del lado del administrador como del cliente, y que los mismos actores no se preocupen de la lógica de negocio y la explotación de los datos en el programa a construir.

1. Procesos de Predesarrollo.

*3.1 Exploración de conceptos.*

* *Situación actual*
* *Dominio del problema*
* *Informe preliminar de necesidades*
* *Soluciones alternativas*
* *Soluciones recomendadas.*

*3.1.1 Situación actual.*

Los programas de gestión hotelera están muy difundidos en el mundo actual, y no se necesita más que un poco de curiosidad para buscar opciones de todo tipo. En el caso de AH se priorizó la exigencia del cliente y se decidió concentrar esfuerzos en cumplir con los requisitos, sin agregar funcionalidades que alargaran los tiempos de realización y que serían superfluos a la hora de validar su funcionamiento. Con AH se logró abstraer dichas necesidades y entregar un software potente y actualizado para los tiempos que corren.

*3.1.2 Dominio del problema*

En AH se busca abordar una problemática ya conocida, pero no por eso sin matices por descubrir. La gestión de negocios tiene por objetivo la organización de la empresa en todos sus aspectos, pero fundamentalmente la orientación al marketing está muy presente, y eso fue en lo que nuestro cliente puso mucho énfasis.Con esta herramienta el administrador del sistema va a poder tener una llegada más fluida con el cliente, haciéndole llegar distintas promociones y ofertas mediante el mailing “personalizado”, que se diseñará como una función clave en el sistema; a su vez podrá gestionar los datos obtenidos para poder realizar planes de negocios y planificaciones en base a estadísticas rápidas y amenas.

De parte del cliente habrá una interfaz gráfica interactiva, llamativa, amigable y viva en colores, para que de primera impresión tenga un contacto preliminar satisfactorio con la empresa a través de su página web.

*3.1.3 Informe preliminar de necesidades.*

Las necesidades detectadas para la creación del producto fueron las siguientes:

* Generación de promociones y utilización de diferentes estrategias de mercadeo electrónico (manejos teóricos de big data, por ejemplo)
* Interfaz gráfica dinámica y moderna, acorde a los tiempos que corren.
* Operatividad del lado administrador.
* Producto moderno, actualizable y de fácil mantenimiento.

*3.1.4 Soluciones alternativas*

Al tener claro el ciclo de vida y considerarlo acorde al proyecto, no se tuvieron en cuenta soluciones alternativas con respecto a ese ítem. Sí se evalúan opciones en torno al framework a utilizar con Python, actualmente se utiliza Geeny, pero también se puede utilizar Django o PyDev de Eclipse.

*3.1.5 Soluciones recomendadas.*

Como se mencionó en el plan de garantía de calidad, el modelo que mejor se adapta al proyecto es el del MVC, ya que es clave la interacción del usuario/administrador o cliente con la interfaz gráfica y su funcionalidad, sin importar el proceso sino las salidas.

El paradigma adecuado a utilizar es el de Objetos, ya que al hacer modular la programación facilita a futuro el mantenimiento y las pruebas de caja blanca para todo el equipo de testing, en cualquier momento del proceso en el que se encuentre el producto, además de ser un paradigma conocido a la perfección por todo el equipo de trabajo.

1. Procesos de desarrollo

En esta sección se describirá el proceso de requisitos,el contexto en el que el software deberá operar y el proceso de diseño y codificación para realizar las tareas solicitadas por el usuario (discriminado por Cliente y Administrador). En las figuras 6 y 7 veremos los casos de uso del cliente y del administrador, respectivamente.

*4.1 Proceso de requisitos:*

La aplicación deberá permitir al Administrador:

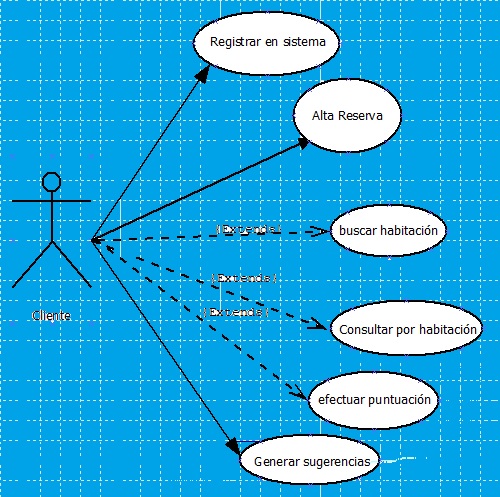
* Alta, baja y modificación de las reservas hechas por el cliente.
* Ingresar y salir del sistema.
* Alta, baja y modificación de usuarios que ingresen al sistema, con sus correspondientes permisos (usuarios que se presuponen empleados de la empresa que también puedan manejar el sistema, no clientes)
* Alta, baja y modificación de promociones.
* Realizar check-in, check-out y checklist de las habitaciones.
* Gestionar la base de datos, con el objetivo de realizar la explotación de la información y usarlos para estrategias de negocio.
* Alta, baja y modificación de las habitaciones.

La aplicación deberá permitir al cliente:

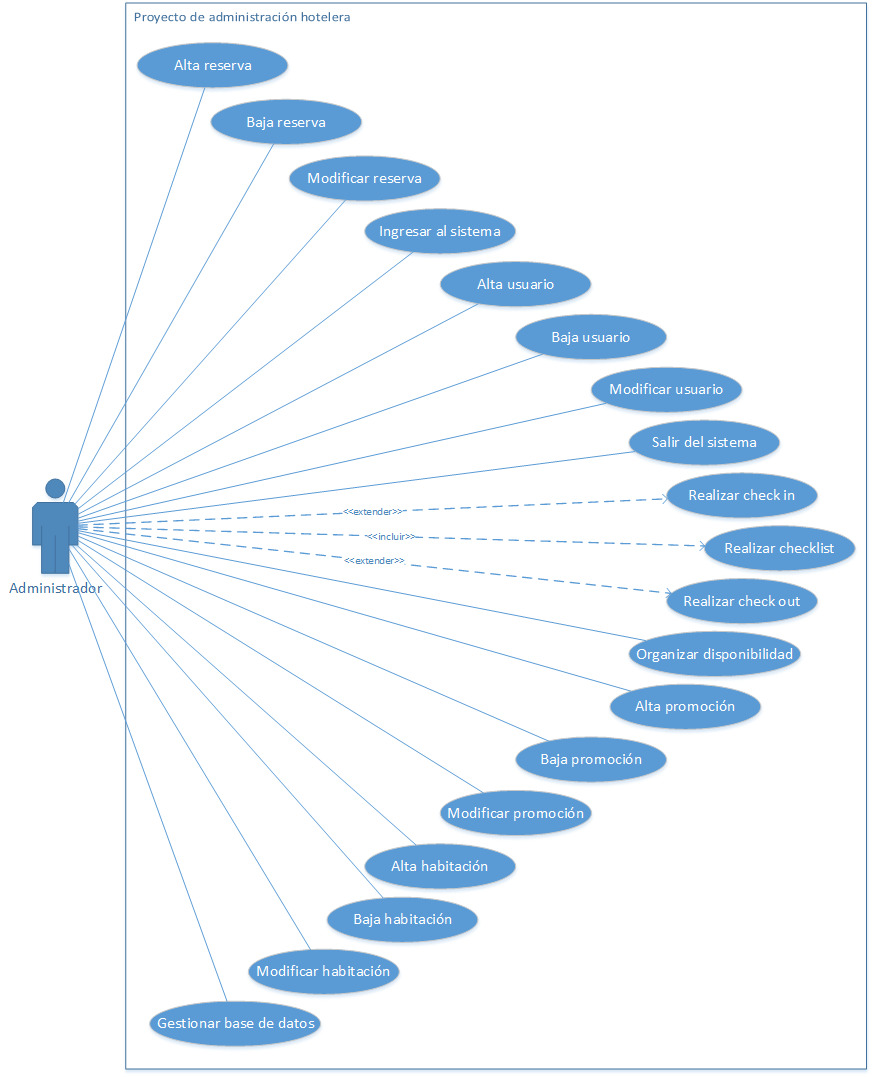
* Registrarse al sistema.
* Generar una reserva, como así también cancelarla o modificarla.
* Búsqueda de habitación.
* Consulta de estado de habitación y reserva.
* Realizar sugerencias.
* Calificar el servicio del hotel en la plataforma.

*4.2 requisitos en negativo*

* Los clientes no podrán explorar los datos de reservas que no sean las propias.
* Un cliente no podrá hacer gestiones si no cumple con el paso de registro.
* Un usuario sin permisos determinados estará limitado a manejar el sistema de manera sesgada, bajo la determinación del administrador.
* El sistema no eliminará los datos de los clientes o usuarios que se den de baja en la aplicación.
* El sistema no dejará registrar a clientes o usuarios que ya están registrados en la plataforma.
* El sistema no permitirá que el cliente desde la web pueda cancelar o modificar la reserva, sólo lo podrá hacer, vía aviso previo, el administrador desde la parte de manejo de negocio de AH.



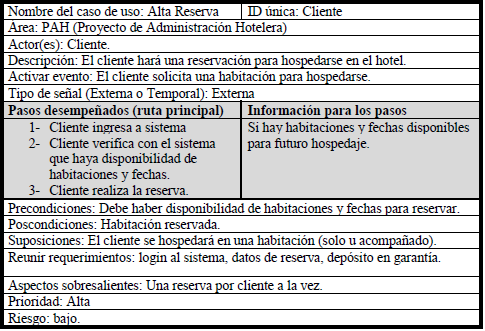
*Fig.6: caso de uso del cliente.*

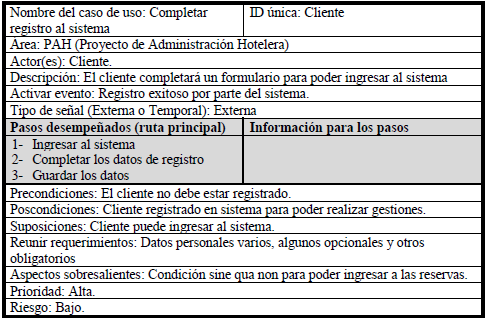
**

*Fig7:Caso de uso de administrador.*

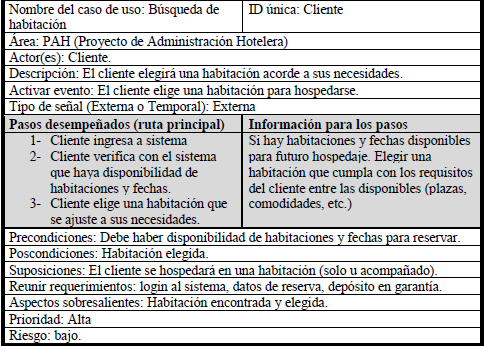
*4.2 Escenarios.*

Los escenarios correspondientes a cliente y administrador son la descripción pormenorizada de cada caso de uso. Cada escenario deberá estar identificado y representará un comportamiento deseado por el sistema ante la interacción de cada actor con el mismo.

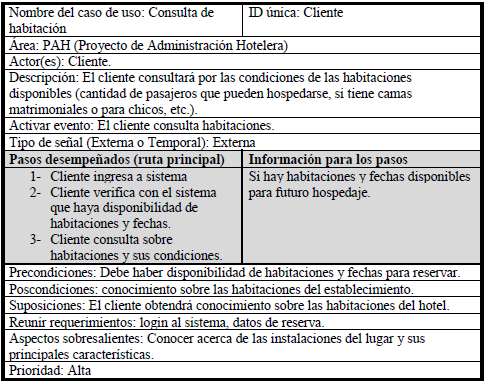
* Fig 8: escenario de alta reserva de cliente.*

**

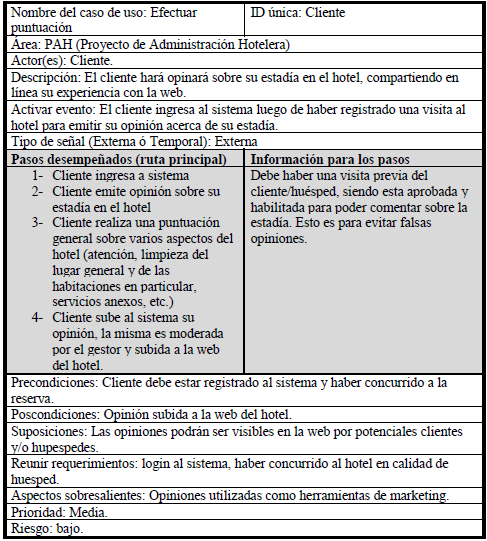
*Fig 11: escenario de completar registro cliente*

**

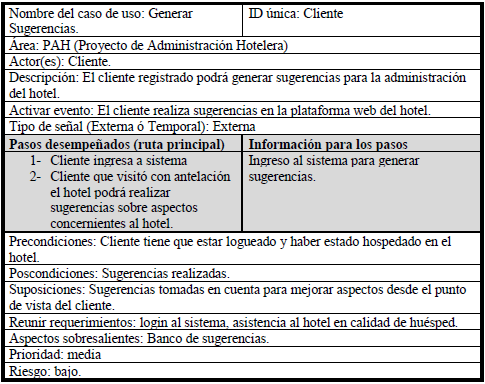
*Fig12:Escenario de búsqueda de habitación cliente*

**

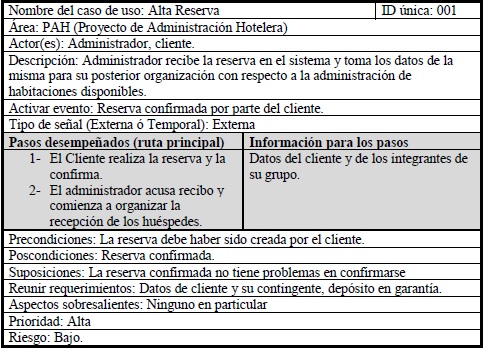
*Fig 13: Escenario de consulta de habitación cliente.*

**

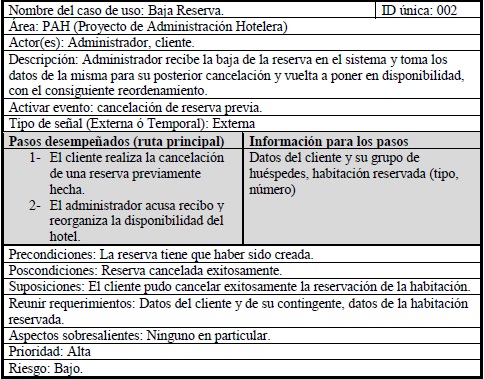
*Fig 14: Escenario de efectuar puntuación cliente*

**

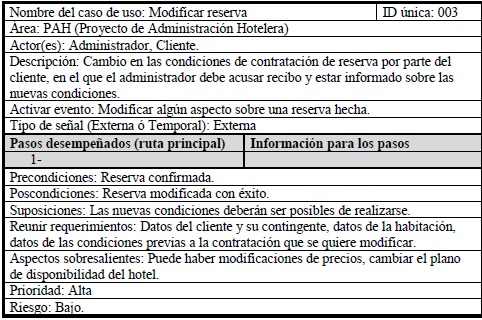
*Fig 15: Escenario de generar sugerencias cliente.*

**

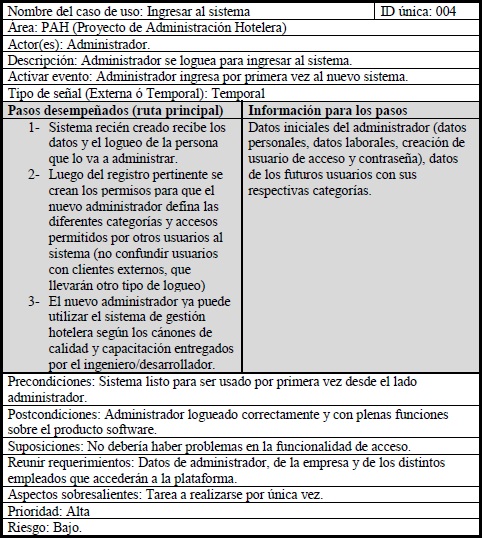
*Fig.16: Escenario “Alta Resetva” de actor administrador.*

**

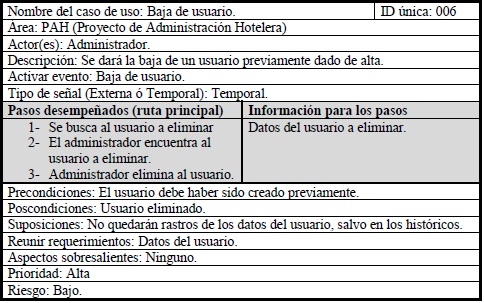
*Fig 17: Escenario “Baja Reserva” del actor administrador*

**

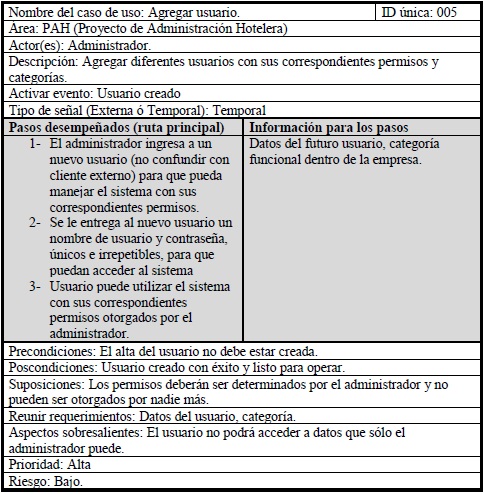
*Fig 18: Escenario ”modificar reserva” de actor administrador.*

**

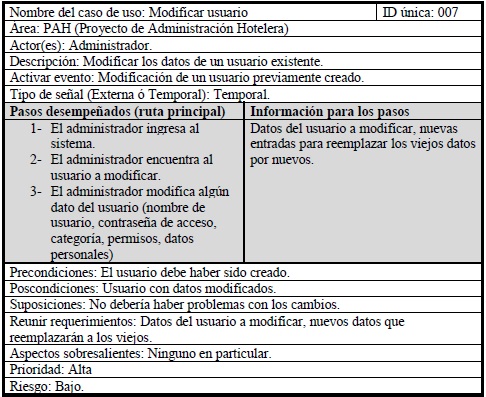
*Fig 19: Escenario “Ingresar al sistema” de actor administrador.*

**

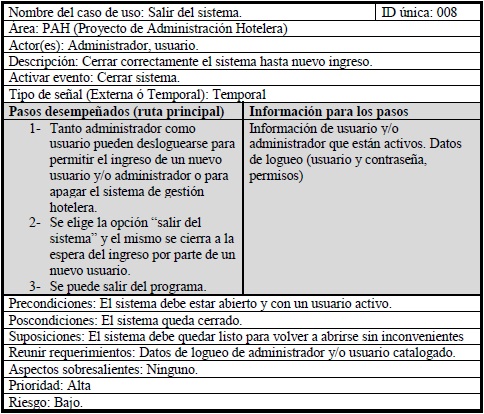
*Fig 20: Escenario “Baja de usuario” de actor administrador.*

**

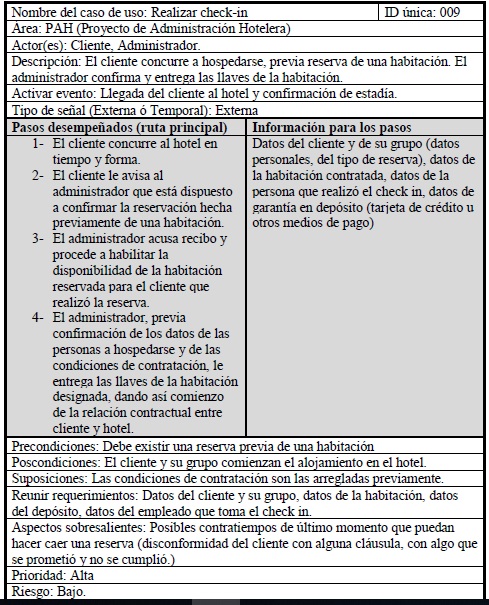
*Fig 21: Escenario “Agregar usuario” de actor administrador.*

**

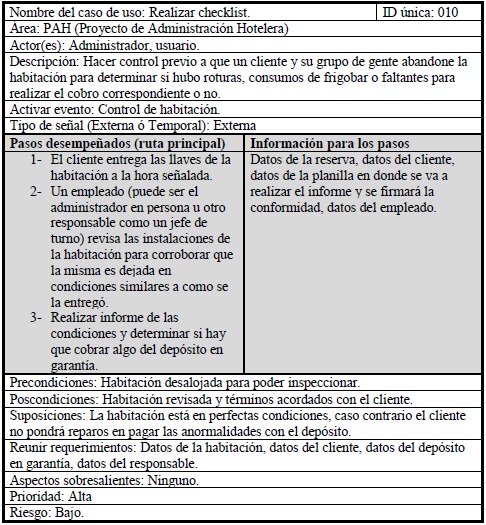
*Fig 21: Escenario “modificar usuario” de actor administrador.*

**

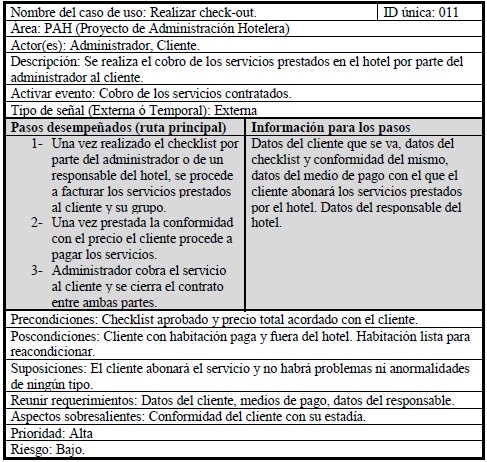
*Fig 22: Escenario “Salir del sistema” del actor administrador.*

**

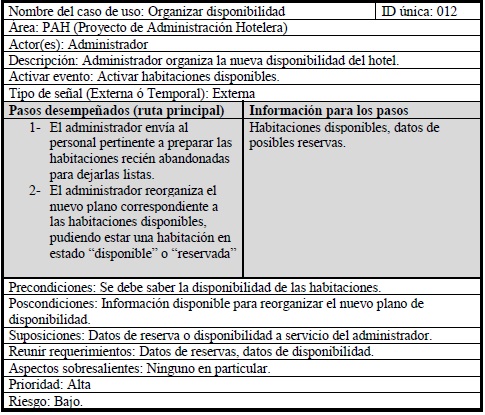
*Fig 22: Escenario “realizar check-in” de actor administrador.*

**

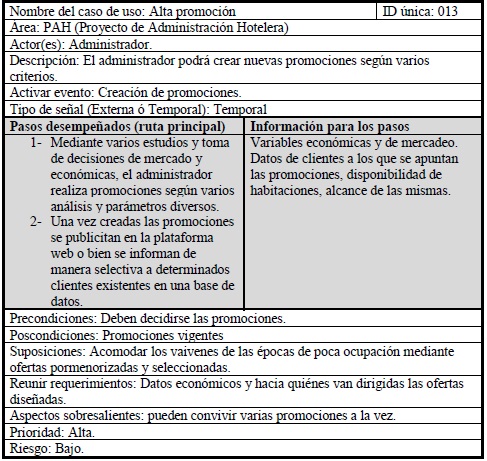
*Fig 23: Escenario “Realizar checklist” de actor administrador.*

**

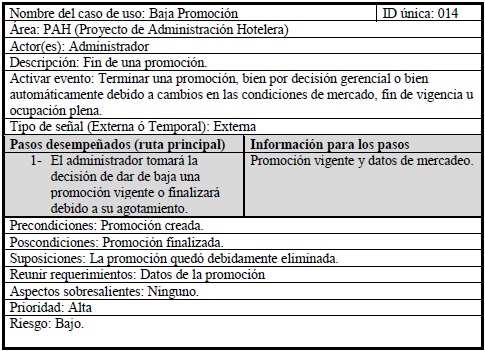
*Fig 24: Escenario “Realizar check-out” de actor administrador.*

**

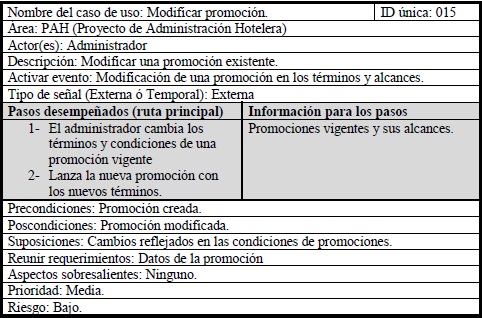
*Fig 25: Escenario “organizar disponibilidad” de actor administrador.*

**

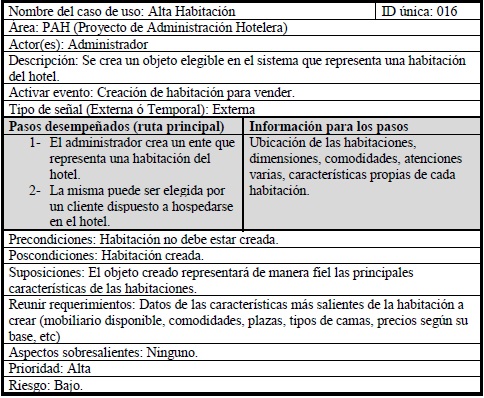
*Fig 26: Escenario “Alta promoción” de actor administrador.*

**

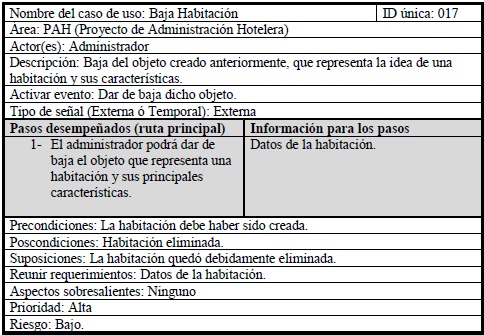
*Fig 27: Escenario “Baja promoción” de actor administrador.*

**

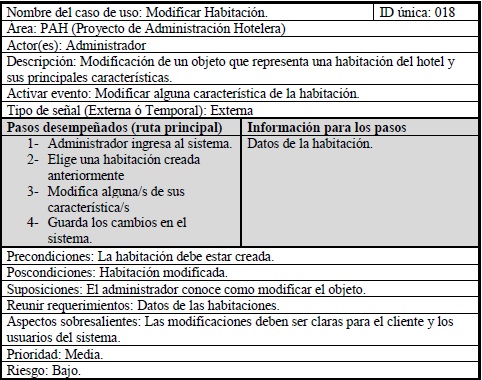
*Fig 28: Escenario “Modificar promoción” de actor administrador.*

**

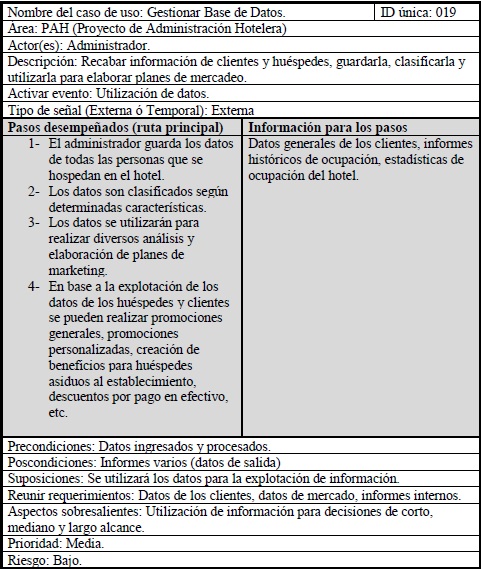
*Fig 29: Escenario “Alta Habitación” de actor administrador.*

**

*Fig 30: Escenario “Baja habitación” de actor administrador.*

**

*Fig 31: Escenario “Modificar habitación” de actor administrador.*

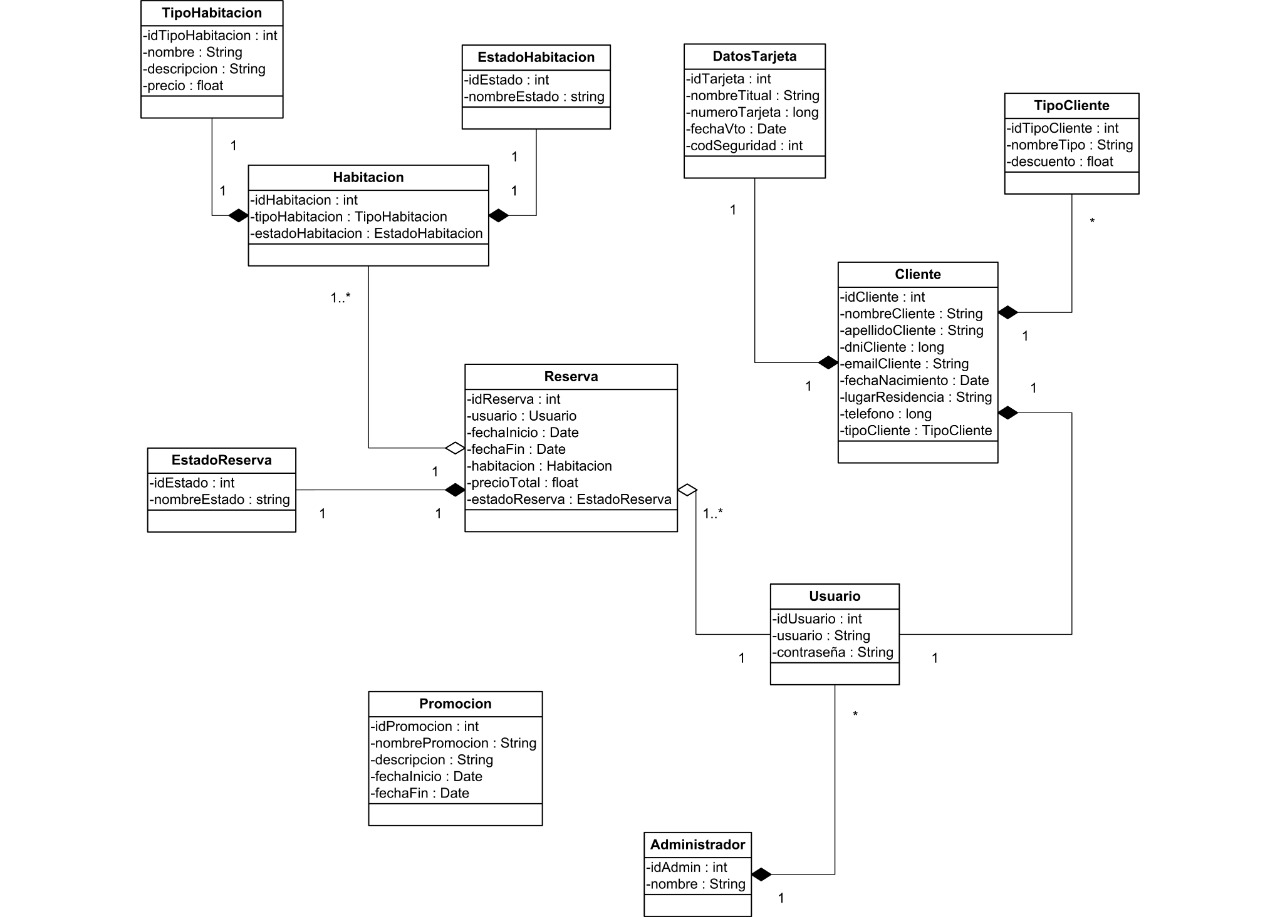
**

*Fig 32: Escenario “Gestionar base de datos” de actor administrador.*

*4.3 Diagrama de clases:*

Mediante el siguiente diagrama UML se muestrala estructura estática del programa, las relaciones entre clases, sus atributos y métodos.

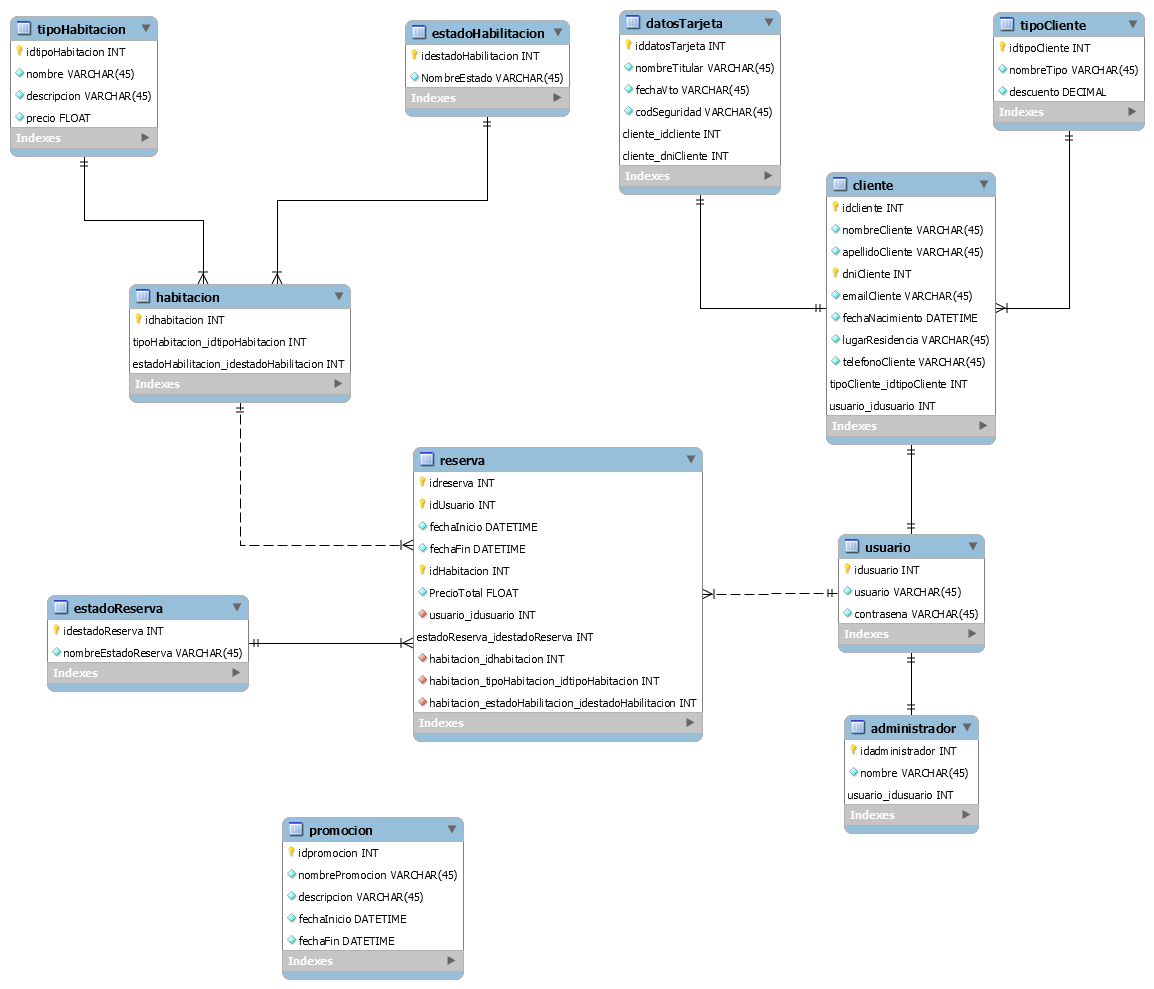
El diagrama fue diseñado en Visio y nos sirve para representar gráficamente su estructura base.

**

*Fig 32: Diagrama de clases de AH.*

*4.4 Desarrollo de la base de datos:*

EL diagrama de entidad-relación pensado para el desarrollo de la base de datos del sistema fue diseñada en MySql, y lo podemos ver en la figura 33.



1. Procesos de prueba:

Si bien los casos de prueba son aún prematuros, se pensó como una buena práctica aplicas pruebas unitarias al maquetado para corroborar que cumpla con las condiciones mínimas de requisitos para presentar ante el cliente.











