



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escuela Técnica Superior de Ingeniería Informática
Universitat Politècnica de València

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

Proyecto Final de Carrera

Ingeniero Técnico en Informática de Gestión

Autor: Miguel Monzó Pastor

Director: Lluís Felip Hurtado Oliver

Valencia, 23 de septiembre de 2012

Resumen

El proyecto consiste en una aplicación sencilla, y de fácil manejo, para la reserva de habitaciones de hotel.

Las operaciones que puede realizar el usuario son:

- ✓ Creación / modificación /eliminación de reservas
- ✓ Visualización de información de hoteles y tipos de habitación por hotel
- ✓ Alta / baja de usuarios en el sistema de reservas.
- ✓ Imprimir una reserva

La interfaz gráfica se ha realizado en Visual Basic .NET 2008 con el entorno integrado de desarrollo (IDE) Visual Studio .NET 2008 y el motor de Base de Datos es SQL server 2008 R2 Express.

Tabla de contenidos

| | |
|--|-----|
| 1. INTRODUCCIÓN | 3 |
| 1.1 El proyecto software..... | 3 |
| 1.2 El proceso software..... | 7 |
| 2. PROYECTO EN VISUAL BASIC .NET : RESERVAS DE HOTEL 1.0 | 9 |
| 2.1 Introducción teórica: Definición de proyecto..... | 9 |
| 2.2 Planificación del proyecto con Microsoft Project 2007..... | 10 |
| 2.3 Análisis..... | 12 |
| 2.3.1 Especificación de requerimientos..... | 12 |
| 2.3.2 Modelado orientado a objetos..... | 13 |
| 2.3.2.1 Casos de Uso..... | 13 |
| 2.3.2.2 Diagramas de Secuencia..... | 20 |
| 2.3.2.3 Diagrama estático de clases..... | 34 |
| 2.4 Diseño..... | 3 |
| 2.4.1 Diseño de la Base de Datos..... | 35 |
| 2.4.1.1 Entradas y etapas del diseño lógico..... | 35 |
| 2.4.1.2 Modelo conceptual de los datos: Modelo Entidad/Relación..... | 36 |
| 2.4.1.3 Modelo lógico estándar: Modelo Relacional..... | 41 |
| 2.4.1.4 Modelo lógico específico del SGBD: SQL Server 2008 R2..... | 44 |
| 2.4.2 Construcción y uso de la interfaz de usuario..... | 54 |
| 2.4.2.1 Acceder al sistema y darse de alta en el sistema..... | 54 |
| 2.4.2.2 Consultar información del hotel y hacer una reserva..... | 59 |
| 2.4.2.3 Imprimir una reserva..... | 79 |
| 2.4.2.4 Consultar información del hotel y habitación de una reserva..... | 80 |
| 2.4.2.5 Modificar una reserva..... | 83 |
| 2.4.2.6 Borrar (Cancelar) una reserva..... | 88 |
| 2.5 Implementación..... | 91 |
| 2.5.1 Instalación y configuración del SGBD SQL Server 2008 R2..... | 91 |
| 2.5.2 Codificación del programa (listado completo y comentado)..... | 128 |
| 2.6 Pruebas..... | 266 |
| 2.7 Instalación..... | 266 |



1. INTRODUCCIÓN

1.1 El proyecto software

Introducción teórica

Concepto de software

Pressman (2006): "El software se forma con 1) las instrucciones (programas de computador) que al ejecutarse proporcionan características, funciones y el grado de desempeño deseados; 2) las estructuras de datos que permiten que los programas manipulen información e manera adecuada; y 3) los documentos que describen la operación y el uso de programas.

Características de software

El software es un elemento lógico, en lugar de físico, de un sistema. Por lo tanto el software tiene características muy diferentes a las del hardware:

1. El software se desarrolla o construye, no se manufactura en el sentido clásico.
2. El software no se estropea pero se deteriora
3. A pesar de que la producción industrial tiene una tendencia hacia la construcción por componentes, la mayoría del software aún se construye sin componentes
4. Interdependencia recíproca con el hardware.
5. Posibilidad de reutilizarse para diseñar y construir otro software (Reusabilidad).
6. Posibilidad de realizarle mantenimientos no correctivos

Aplicaciones del software

El software puede aplicarse en cualquier situación en la que se haya definido un conjunto específico de pasos procedimentales (es decir, un algoritmo). Las siguientes áreas del software indican la amplitud de las aplicaciones potenciales:

- Software de sistemas
- Software de tiempo real
- Software de gestión
- Software de ingeniería y científico
- Software empotrado

- Software de computadoras personales
- Software basado en web
- Software de inteligencia artificial

Información sobre el proyecto Reservas de Hotel 1.0

El proyecto consiste en aplicación de reservas de habitaciones de hotel (software de gestión), con interfaz gráfica realizada en Visual Basic .NET 2008, programando en el entorno integrado de desarrollo (IDE) Visual Studio .NET 2008 y con acceso a una base de datos remota SQL Server 2008 R2 Express.

La aplicación accede remotamente al servidor SQL Server y permite realizar por parte de los usuarios de la aplicación las siguientes operaciones:

- Creación / modificación / eliminación de reservas
- Visualización de información de hoteles y tipos de habitación por hotel
- Alta / baja de usuario en el sistema de reserva
- Imprimir una reserva

Objetivos del proyecto

La realización de este proyecto ha tenido como objetivos:

- Practicar programación en un lenguaje orientado a objetos que utiliza la plataforma .NET Framework (Visual Basic .NET 2008)
- Utilizar el entorno integrado de desarrollo (IDE) Visual Studio .NET 2008 para reducir el tiempo de desarrollo de la aplicación
- Manejo de sentencias SQL para inserción, eliminación , actualización etc.. en la base de datos.
- Instalación, configuración y administración básica del motor de base de datos SQL Server 2008 R2 Express

Tecnologías empleadas y justificación de la elección

Tecnologías empleadas:

- Entorno de definición de datos: SQL Server 2008 R2 Express
- Desarrollo de la interfaz: Visual Basic .NET 2008
- Plataforma para crear aplicaciones y el entorno de ejecución de las mismas: .NET Framework
- Modelo de objetos de acceso a datos: ADO. NET
- Motor de base de datos: base de datos compatible con ODBC Microsoft SQL Server

He elegido Visual Basic .NET 2008 como lenguaje de programación por diversos motivos:

- a) Es un lenguaje orientado a objetos
- b) Lenguaje sencillo y fácil de manejar

- c) Incluye un nuevo modelo de acceso a datos mejorado: ADO (ActiveX Data Object)
- d) Es un lenguaje que forma parte de la plataforma .NET Framework, la cual permite el desarrollo de aplicaciones a través de uso conjunto de herramientas y servicios que proporciona.

He elegido Visual Studio .NET como entorno integrado de desarrollo por diversos motivos:

- a) Visual Studio 2008 nos provee una serie de herramientas para desarrollo, así como características de debugging, funcionalidad en base de datos y características innovadoras para la creación de aplicaciones en una variedad de plataformas.
- b) Visual Studio 2008 también nos provee a desarrolladores la habilidad de poder escoger entre múltiples versiones del Framework con el mismo entorno de desarrollo, así nosotros podemos desarrollar en la versión que queramos ya sea en .NET Framework 2.0, 3.0 o 3.5, entiendo así que soporta un gran variedad de proyectos en la versión X en el mismo entorno de desarrollo.
- c) Útil para generar aplicaciones basadas en Windows en poco tiempo: Con la herencia visual, los programadores pueden simplificar enormemente la creación de aplicaciones basadas en Windows, centralizando en formularios primarios la lógica común y la interfaz de usuario para toda la solución.
- d) Visual Studio 2008 brinda ventajas al desarrollador en 3 pilares fundamentales:
 - 1. Mejor productividad del desarrollador
 - 2. Administración del ciclo de vida de las aplicaciones.
 - 3. Desarrollo sobre últimas tecnologías.

He elegido el motor de Bases de datos SQL Server 2008 Express por varios motivos:

- 1. Es gratuito. Puede ser adecuado para una pyme. Reduciendo costes al ser gratuito.
- 2. Potente .Posee buenos registros de los benchmarks independientes tanto en transacciones totales como en coste por transacción.
- 3. Escalable. Se adapta a las necesidades de cualquier empresa.
- 4. Es el servidor de bases de datos más adecuado para trabajar con Visual Studio 2008. Se encuentra completamente integrado en Visual Studio 2008 hasta el punto que desde el propio interface gráfico de Visual Studio podemos realizar la mayoría de las tareas de administración de la base de datos.
- 5. Diseñada desde su inicio para trabajar en entornos Internet e Intranet. La aplicación de reserva de hoteles puede ser empleada tanto en entornos de red local como en entornos Internet, Intranet o Extranet.
- 6. Sencillo de instalar y con una buena herramienta de gestión (SQL Server management Studio).
- 7. Disponibilidad, fiabilidad y tolerancia a fallos.

1.2 El proceso software

Características de proceso software

En un proceso de software se establece:

- *Un marco común del proceso* definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos del software, con independencia de su tamaño o complejidad.
- Un número de *conjunto de tareas* – cada uno es una colección de tareas de trabajo de ingeniería del software, hitos de proyecto, productos de trabajo, y puntos de garantía de calidad – que permiten que la actividades del marco de trabajo se adapten a las características del proyecto del software y a los requisitos del equipo del proyecto.
- Finalmente, las actividades de protección – tales como garantía del software, gestión de configuración del software y medición – abarcan el modelo de procesos. La actividades de protección son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso.

Modelos de proceso del software

Para resolver los problemas reales de una industria, un ingeniero del software o un equipo de ingenieros debe incorporar una estrategia de desarrollo que acompañe al proceso, métodos y capas de herramientas de la ingeniería del software. Esta estrategia a menudo se llama modelo de proceso o paradigma de ingeniería del software.

Modelos:

Modelo lineal secuencial (ciclo de vida básico o modelo en cascada):

Sugiere un enfoque sistemático, secuencial, para el desarrollo del software que comienza con un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento.

Modelo de construcción de prototipos:

Fases:

- a) Recolección de requisitos, escuchar al cliente.
- b) Diseño rápido: Construcción del prototipo
- c) El prototipo lo evalúa el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar

La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que se necesita hacer.

Modelo de Desarrollo Rápido de Aplicaciones (DRA): Es un modelo de proceso de desarrollo del software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto. Es una adaptación a <<alta velocidad>> del modelo lineal secuencial en el que se logra el desarrollo rápido utilizando una construcción basada en componentes:

Fases:

- a) Modelado de gestión
- b) Modelado de datos
- c) Modelado del proceso
- d) Generación de aplicaciones
- e) Pruebas y entrega

Modelos evolutivos:

Es un modelo de proceso que se ha diseñado para acomodarse a un producto que evolucione con el tiempo. Adaptable a requisitos cambiantes. Se elaboran versiones cada vez más completas del software. Los modelos evolutivos son iterativos, interactivos y evolutivos. Se caracterizan por la forma en que permiten a los ingenieros del software desarrollar versiones cada vez más completas del software.

1. El modelo incremental combina elementos del modelo lineal secuencial con la filosofía interactiva de construcción de prototipos. Es una secuencia de aplicaciones del ciclo clásico (secuencia lineal). Cada secuencia lineal produce un <<incremento>> del software. Finaliza cuando se entrega el producto final
2. El modelo espiral conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. En el modelo espiral, el software de desarrolla en una serie de versiones incrementales. El modelo en espiral se divide en un número de actividades de marco de trabajo, también llamadas regiones de tareas. Generalmente existen seis regiones de tareas:
 - Comunicación con el cliente
 - Planificación
 - Análisis de riesgos
 - Ingeniería
 - Construcción y acción
 - Evaluación del cliente

Modelo de proceso del software utilizado en el proyecto

Para el proyecto de Reservas de hotel 1.0 se ha utilizado el modelo evolutivo incremental

1. Secuencia de aplicaciones de ciclo clásico
2. Cada iteración produce un incremento del producto (Reservas de hotel)
3. Finaliza cuando se entrega el producto final (Reservas de hotel 1.0)

En cada iteración realicé una funcionalidad completa (alta, baja, creación reserva, modificación reserva, borrado reserva, etc.).

2. PROYECTO EN VISUAL BASIC .NET: RESERVAS DE HOTEL 1.0

2.1 Introducción teórica: Definición de proyecto

Un proyecto es un esfuerzo temporal acometido para crear un único servicio o producto. Temporal quiere decir que todo proyecto tiene un comienzo claro y un final claro. Único significa que el producto o servicio es diferente de alguna forma clara de todos los productos o servicios similares.

Características de un proyecto:

- Existe un objetivo claro.
- Tiene principio y fin en el tiempo.
- Se requiere un nivel de calidad.
- Se puede identificar un conjunto de tareas.
- Es necesario la intervención de especialistas.
- Existen limitaciones en los recursos.
- Se requiere una planificación.

Fases de un proyecto

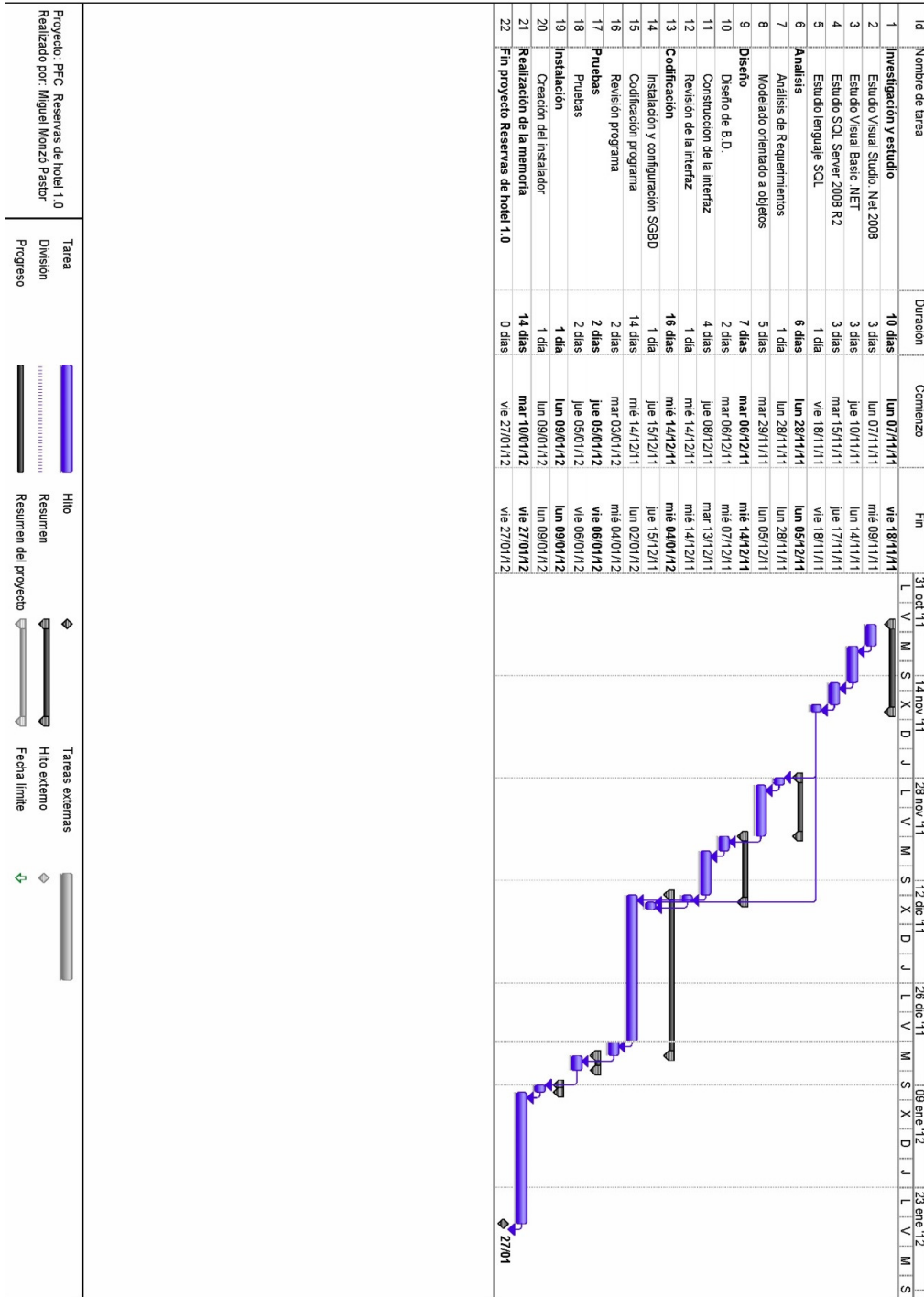
Se distinguen dos grandes fases:

- *Planeación:* Se distinguen dos sub fases:
 - Definición del problema: ¿Cuál es el problema?, ¿Dónde está la oportunidad?.

- Planificación del proyecto: ¿Cuándo se lleva a cabo el plan previsto?, ¿Cuáles son los costes?, ¿con que recursos?.
- *Ejecución:* Se distinguen tres sub fases:
 - Puesta en marcha: ¿Quién hace qué?.
 - Producción: Se llevan a cabo las tareas.
 - Conclusión del proyecto: Entrega del producto o servicio

2.2 Planificación del Proyecto con Microsoft Project 2007

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0



Informe de horas empleadas:



Horas empleadas en el PFC Reservas de hotel 1.0

| | |
|----------------------------------|-----------|
| Estudio Visual Studio. Net 2008 | 14 horas |
| Estudio Visual Basic .NET | 12 horas |
| Estudio SQL Server 2008 R2 | 6 horas |
| Estudio lenguaje SQL | 4 horas |
| Análisis de Requerimientos | 4 horas |
| Modelado orientado a objetos | 20 horas |
| Diseño de B.D. | 8 horas |
| Construcción de la interfaz | 16 horas |
| Revisión de la interfaz | 4 horas |
| Instalación y configuración SGBD | 3 horas |
| Codificación programa | 56 horas |
| Revisión programa | 4 horas |
| Pruebas | 4 horas |
| Creación del instalador | 2 horas |
| Realización de la memoria | 28 horas |
| TOTAL | 185 horas |

2.3 Análisis



2.3.1 Especificación de requerimientos

La aplicación de reserva de habitaciones de hotel tiene la siguiente especificación:

- Cada hotel se describirá por sus características básicas: nombre, dirección, categoría, teléfono, descripción y ofrecerá un conjunto de tipos de habitaciones.
- Cada tipo de habitación tendrá sus propias características básicas. Las más relevantes para la gestión de reservas son: nombre, número de habitaciones disponibles, capacidad máxima y precio por noche único (sin promociones o tarifas distintas para temporada alta o baja)
- Los clientes de la aplicación podrán realizar sus reservas en cualquiera de los hoteles que tengan disponibilidad en las fechas que correspondan
- Cada cliente tiene sus datos personales: nombre, NIF, domicilio etc..
- Las operaciones permitidas por parte de los usuarios son:
Creación/modificación/eliminación de reservas

Visualización de información de hoteles y tipos de habitación por hotel

Alta/baja de usuarios en el sistema de reservas.

- Para realizar una reserva, el cliente deberá haberse dado de alta previamente en el sistema de reservas (nombre de usuario y contraseña asociada)
- El proceso de confección de una reserva por parte del cliente se organiza en tres fases:
 - Selección de hotel (búsqueda por localidad o por nombre)
 - Comprobación de la disponibilidad de habitaciones de la capacidad deseada en las fechas indicadas.
 - Selección del tipo de habitación y confección de la reserva. Para cada reserva se toma nota de:
 - Cliente que la realiza
 - Tipo de habitación reservada (e implícitamente el hotel al que pertenece.
 - Número de ocupantes efectivos (ocupación)
 - Nombre del cliente
 - Importe por noche (no se consideran promociones o descuentos)
- El cliente puede gestionar sus reservas (modificar, eliminar, imprimir) siempre y cuando la gestión se realice antes del comienzo del periodo de reserva, como es lógico.
- Las tareas de administración serán básicas y se realizarán desde el propio servidor SQL Server, modificando los registros de las tablas.
- La configuración y mantenimiento del servidor SQL Server también será básico y abarcará lo justo y necesario para llevar a buen fin la ejecución de la aplicación.

- El administrador del servidor SQL Server podrá:
 - Modificar los datos generales de los hoteles.
 - Dar de alta nuevos hoteles y gestionar los tipos de habitación por hotel
 - Consultar, modificar y eliminar reservas.

Por simplicidad no se considera el uso de funciones HASH y cifrado para proteger las contraseñas de usuario ni las transferencias por la red.

2.3.2 Modelado Orientado a Objetos

2.3.2.1 Casos de Uso

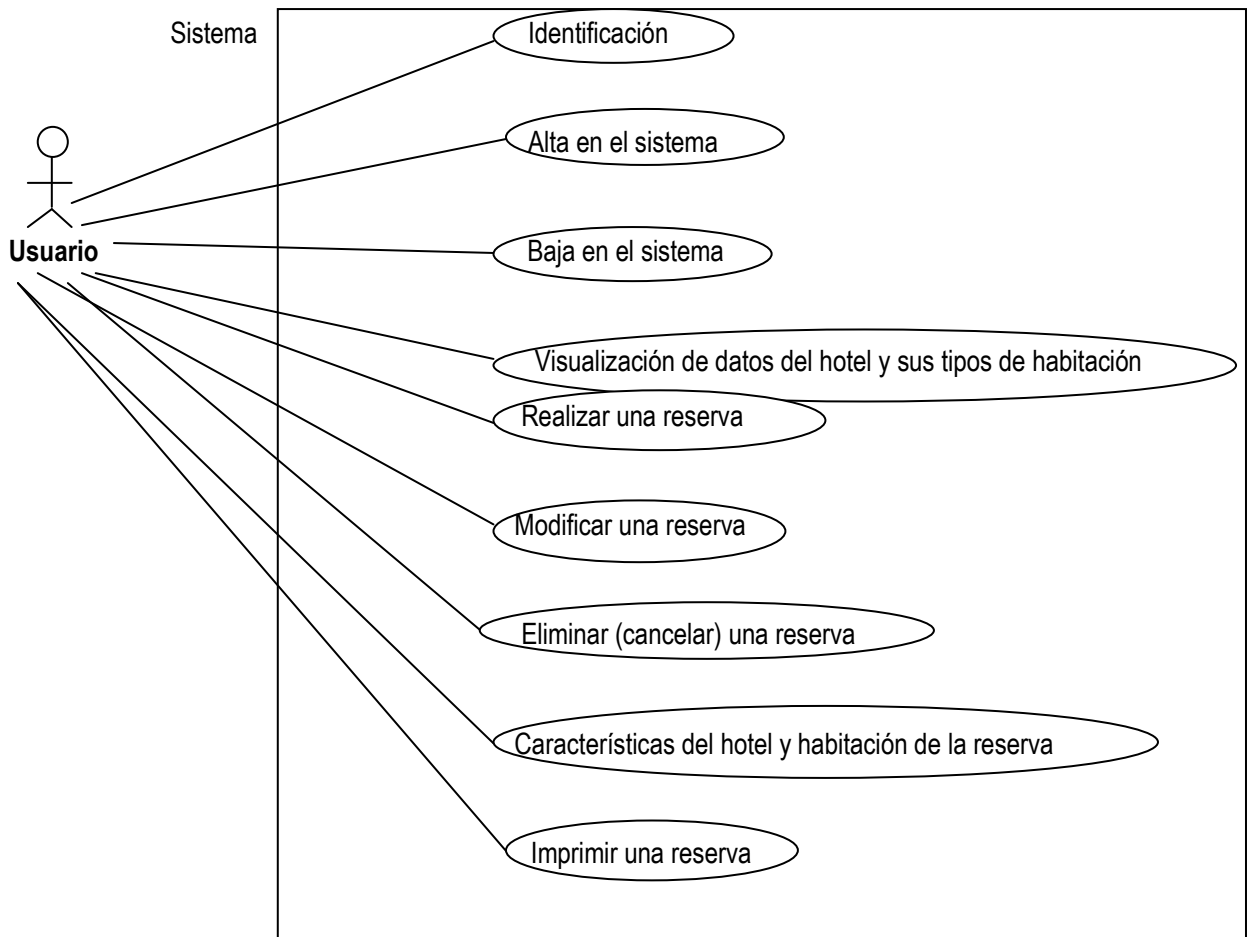
Identificación de actores

En el sistema se identifica un actor:

Usuario: Cualquier persona que accede al sistema

Descripción de casos de uso

En el siguiente diagrama se muestra los escenarios principales que podemos identificar en el sistema.



Caso de Uso: Identificación

Funcionalidad general: Verifica que el nombre de usuario y la contraseña introducidos corresponden con un usuario registrado en el sistema.

Actor principal: Usuario.

Precondición: El usuario estará previamente dado de alta (registrado) en el sistema.

Postcondición: El sistema ha validado el usuario y el password y habilita las funciones propias del perfil de usuario dado de alta (registrado).

Escenario principal:

1. El usuario ejecuta la aplicación
2. El usuario introduce un nombre de usuario y un password
3. Si el nombre de usuario y password son correctos el sistema presentará los datos del usuario validado
 - 3.1 Si el nombre de usuario no existe o el password es incorrecto, no se validará en el sistema y se mostrará un mensaje de error.

Caso de Uso: Alta en el sistema

Funcionalidad general: Permite al usuario darse de alta en el sistema (registrarse), introduciendo sus datos personales.

Actor principal: Usuario.

Precondición: El usuario no está dado de alta en el sistema o se quiere dar de alta con otros datos.

Postcondición: El usuario está dado de alta (registrado) y puede las tareas permitidas.

Escenario principal:

1. Cualquier usuario puede darse de alta (registrarse) en el sistema.
2. El usuario ejecuta la aplicación
3. El usuario entra en el sistema en modo registro
4. El usuario introduce los datos personales incluyendo nombre de usuario y contraseña
5. El usuario guarda los datos:
 - 5.1 Si falta algún dato obligatorio se mostrará un error y no se guardaran los datos ni se habilitaran las opciones de usuario.

5.2 Si el usuario ya existe se mostrará un error y no se guardaran los datos ni ser habilitaran las opciones de usuario.

5.3 Si no falta ningún dato obligatorio y el usuario no existe se dará de alta al usuario con los datos introducidos y se habilitaran las opciones disponibles para el usuario.

Caso de Uso: Baja en el sistema

Funcionalidad general: Permite al usuario darse de baja en el sistema y consecuentemente borrar todos sus datos.

Actor principal: Usuario.

Casos de uso relacionados: Identificación, Alta en el sistema.

Precondición: El usuario está dado de alta en el sistema (registrado) y validado.

Postcondición: El usuario está dado de baja en el sistema y sus datos se han borrado.

Escenario principal:

1. Si el usuario está dado de alta en el sistema entonces se realiza el Caso de uso Identificación
2. Si el usuario no está dado de alta en el sistema entonces se realiza el Caso de uso Alta en el sistema
3. El usuario selecciona la opción de darse de baja en el sistema
4. Se mostrará un mensaje de confirmación
 - 4.1 Si acepta la confirmación se dará de baja en el sistema y se borrarán todos sus datos
 - 4.2 Si no acepta la confirmación no se dará de baja en el sistema

Caso de Uso: Visualización de los datos del hotel y sus tipos de habitación

Funcionalidad general: Presenta al usuario las características del hotel y de los tipos de habitación de ese hotel seleccionado.

Actor principal: Usuario.

Casos de uso relacionados: Identificación, Alta en el sistema.

Precondición: El usuario está dado de alta en el sistema (registrado) y validado.



Postcondición: Se presenta en pantalla las características del hotel y tipo de habitación seleccionados.

Escenario principal:

1. Si el usuario está dado de alta en el sistema entonces se realiza el Caso de uso Identificación
2. Si el usuario no está dado de alta en el sistema entonces se realiza el Caso de uso Alta en el sistema
3. El usuario selecciona un hotel mediante una de estas dos opciones:
 - 3.1 Se escribe una localidad y a continuación se muestran los hoteles que existen en dicha localidad para seleccionar uno de ellos
 - 3.2 Se selecciona un hotel de la lista completa de hoteles
4. Se mostrará información de las características del hotel y se podrá seleccionar un tipo de habitación de las disponibles para el hotel seleccionado.
5. Se mostrará información del tipo de habitación seleccionado.

Caso de Uso: Realizar una reserva

Funcionalidad general: Realizar una reserva en un hotel.

Actor principal: Usuario.

Casos de uso relacionados: Identificación, Alta en el sistema.

Precondición: El usuario está dado de alta en el sistema (registrado) y validado.

Postcondición: Se realiza una reserva en el hotel y tipo de habitación seleccionados.

Escenario principal:

1. Si el usuario está dado de alta en el sistema entonces se realiza el Caso de uso Identificación
2. Si el usuario no está dado de alta en el sistema entonces se realiza el Caso de uso Alta en el sistema
3. El usuario selecciona un hotel mediante una de estas dos opciones:
 - 3.1 Se escribe una localidad y a continuación se muestran los hoteles que existen en dicha localidad para seleccionar uno de ellos
 - 3.2 Se selecciona un hotel de la lista completa de hoteles
4. Realizamos una consulta para ver disponibilidad del hotel en las fechas de entrada y salida
 - 4.1 Si la fecha de entrada es igual o anterior a la fecha actual se mostrará un mensaje de error.



4.2 Si la fecha de salida es igual o anterior a la fecha de entrada se mostrará un mensaje de error.

5. Selección de tipo de habitación, número de habitaciones reservadas y ocupación.
6. El usuario guardará la reserva realizada.

Caso de Uso: Modificación de una reserva

Funcionalidad general: Realizar una modificación de una reserva realizada previamente.

Actor principal: Usuario

Casos de uso relacionados: Identificación, Alta en el sistema, Realizar una reserva.

Precondición: El usuario está dado de alta en el sistema (registrado) y validado. He usuario ha realizado previamente al menos una reserva.

Postcondición: Modificación de una reserva realizada previamente.

Escenario principal:

1. Si el usuario está dado de alta en el sistema entonces se realiza el Caso de uso Identificación
2. Si el usuario no está dado de alta en el sistema entonces se realiza el Caso de uso Alta en el sistema.
3. Si el usuario no ha realizado la reserva entonces se realiza el caso de uso Realizar una reserva.
4. El usuario selecciona una reserva.
 - 4.1 Si la reserva tiene fecha de inicio anterior o igual a la fecha actual no se puede modificar la reserva y se mostrará un mensaje de error.
5. Seleccionamos la fecha de entrada y de salida.
 - 5.1 Si la fecha de entrada es igual o anterior a la fecha actual se mostrará un mensaje de error.
 - 5.2 Si la fecha de salida es igual o anterior a la fecha de entrada se mostrará un mensaje de error.
6. Selección de tipo de habitación, número de habitaciones reservadas y ocupación.
7. El usuario guardará la reserva modificada.

Caso de Uso: Eliminar (Cancelar) una reserva



Funcionalidad general: Realizar un borrado (cancelación) de una reserva realizada previamente.

Actor principal: Usuario

Casos de uso relacionados: Identificación, Alta en el sistema, Realizar una reserva.

Precondición: El usuario está dado de alta en el sistema (registrado) y validado. He usuario ha realizado previamente al menos una reserva.

Postcondición: Borrado (cancelación) de una reserva realizada previamente.

Escenario principal:

1. Si el usuario está dado de alta en el sistema entonces se realiza el Caso de uso Identificación
2. Si el usuario no está dado de alta en el sistema entonces se realiza el Caso de uso Alta en el sistema.
3. Si el usuario no ha realizado la reserva entonces se realiza el caso de uso Realizar una reserva.
4. El usuario selecciona una reserva
 - 4.1 Si la reserva tiene fecha de inicio anterior o igual a la fecha actual no se puede borrar (cancelar) la reserva y se mostrará un mensaje de error.
5. Se muestra información de la reserva y el usuario confirma el borrado de la reserva

Caso de Uso: Características del hotel y habitación de la reserva

Funcionalidad general: Permite ver características del hotel y habitación de una reserva seleccionada de la lista de reservas.

Actor principal: Usuario

Casos de uso relacionados: Identificación, Alta en el sistema, Realizar una reserva

Precondición: El usuario está dado de alta en el sistema (registrado) y validado. He usuario ha realizado previamente al menos una reserva.

Postcondición: Se muestra información, de las características del hotel y habitación, de la reserva seleccionada.

Escenario principal:

1. Si el usuario está dado de alta en el sistema entonces se realiza el Caso de uso Identificación
2. Si el usuario no está dado de alta en el sistema entonces se realiza el Caso de uso Alta en el sistema.
3. Si el usuario no ha realizado la reserva entonces se realiza el caso de uso Realizar una reserva.
4. El usuario selecciona una reserva de la lista de reservas realizadas
5. Se muestra información, de las características del hotel y habitación, de la reserva seleccionada.

Caso de Uso: Imprimir una reserva

Funcionalidad general: Permite imprimir información de una reserva en vigor , es decir de una reserva cuya fecha de inicio es posterior a la fecha actual.

Actor principal: Usuario

Casos de uso relacionados: Identificación, Alta en el sistema, Realizar una reserva

Precondición: El usuario está dado de alta en el sistema (registrado) y validado. He usuario ha realizado previamente al menos una reserva. La reserva tiene que estar en vigor (fecha de inicio posterior a la fecha actual)

Postcondición: Se imprime información de la reserva seleccionada.

Escenario principal:

1. Si el usuario está dado de alta en el sistema entonces se realiza el Caso de uso Identificación
2. Si el usuario no está dado de alta en el sistema entonces se realiza el Caso de uso Alta en el sistema.
3. Si el usuario no ha realizado la reserva entonces se realiza el caso de uso Realizar una reserva.
4. El usuario selecciona una reserva de la lista de reservas realizadas.
 - 4.1 Si la reserva tiene fecha de inicio anterior o igual a la fecha actual no se puede imprimir la reserva y se mostrará un mensaje de error.
5. Se muestra información de las características del hotel y habitación de la reserva seleccionada.
6. El usuario imprime la reserva.

2.3.2.2 Diagramas de secuencia

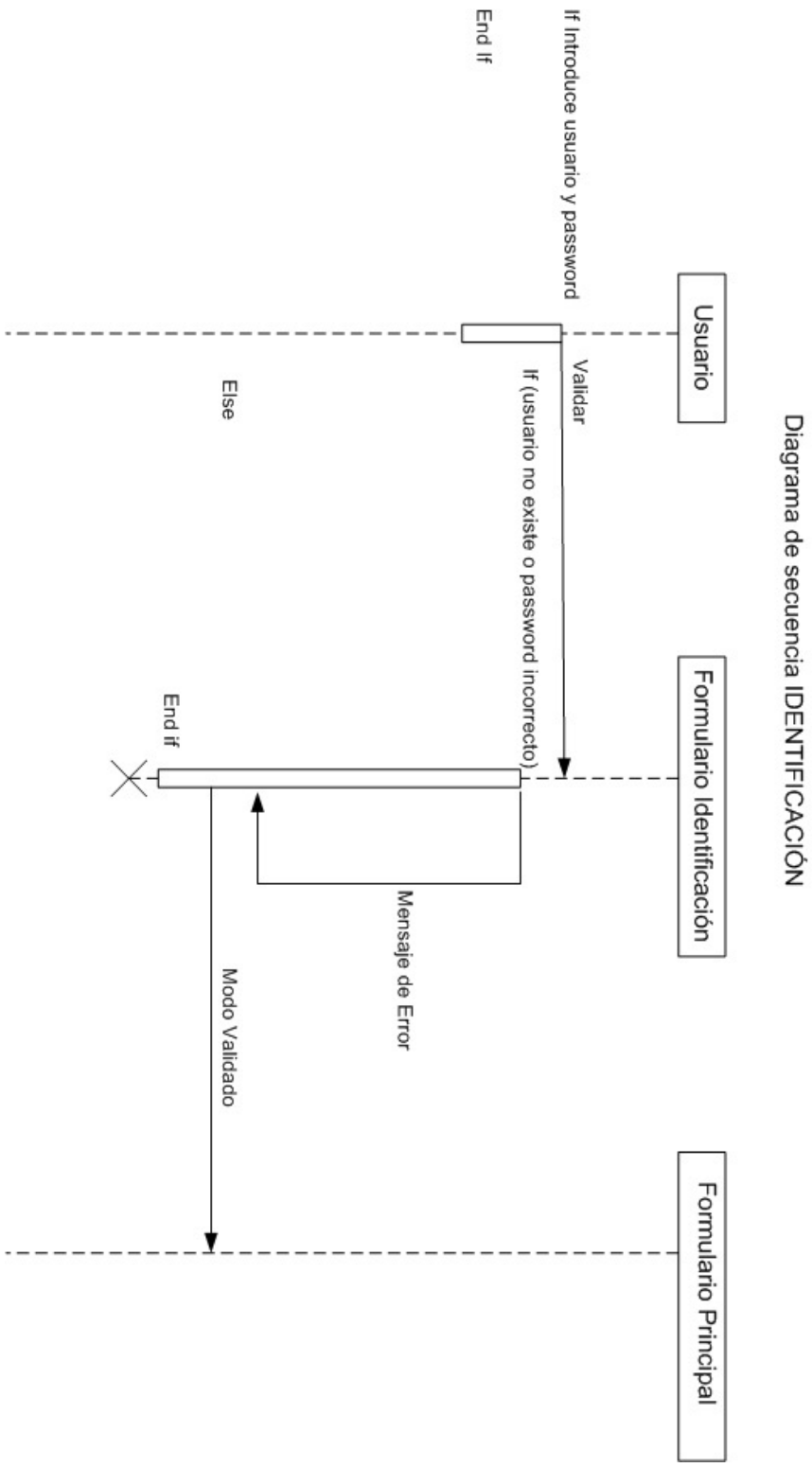
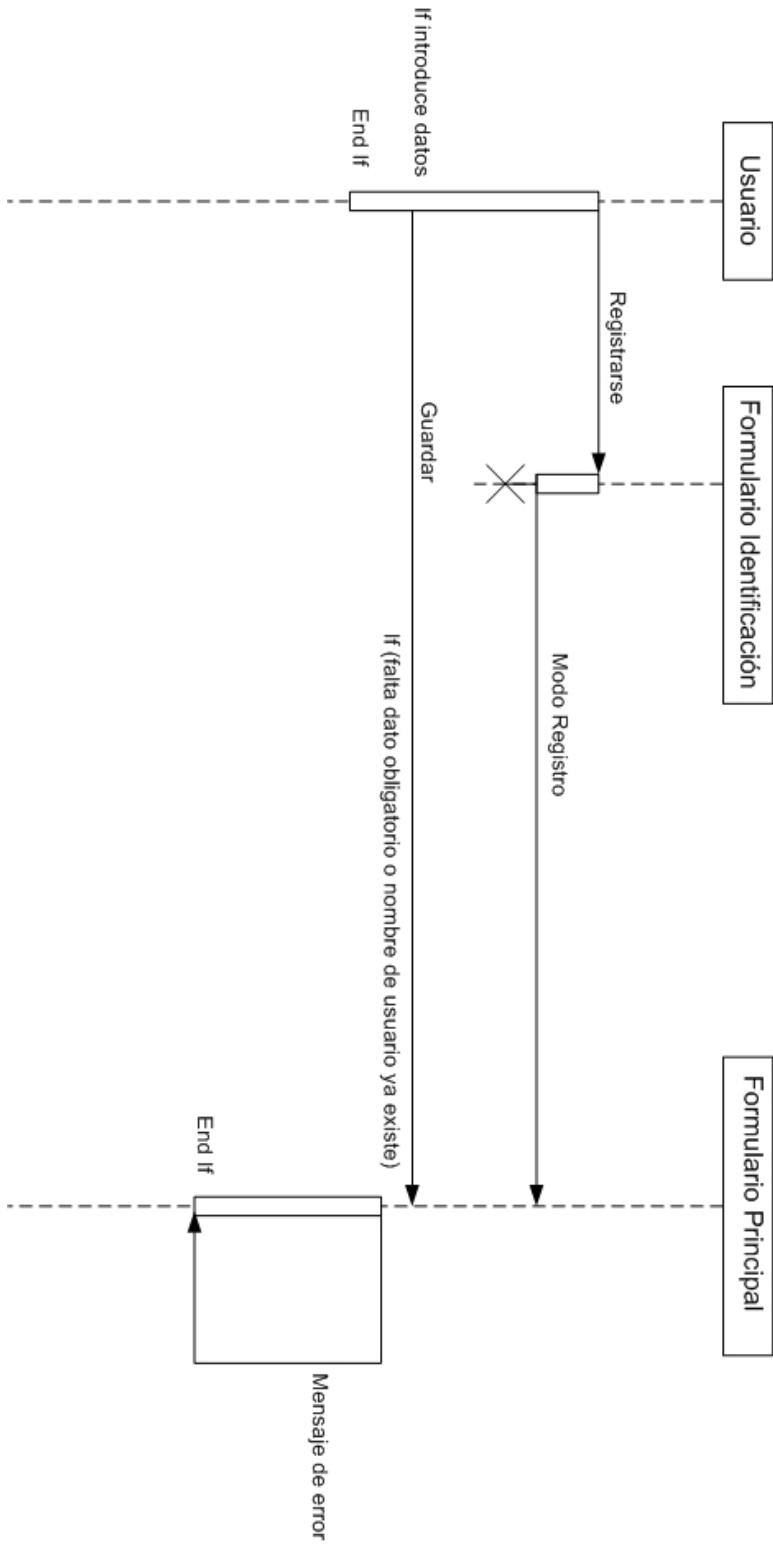


Diagrama de secuencia ALTA EN EL SISTEMA



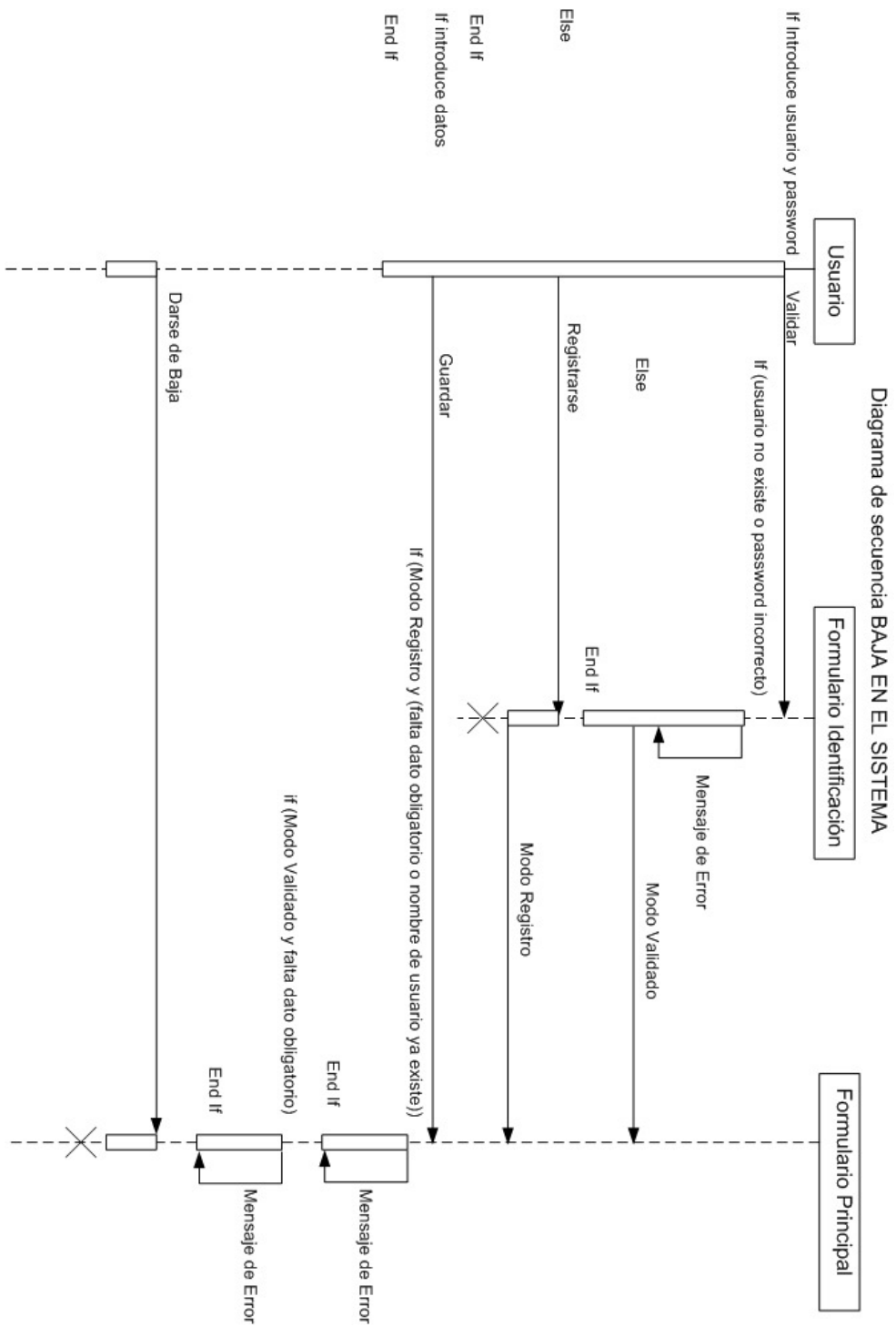
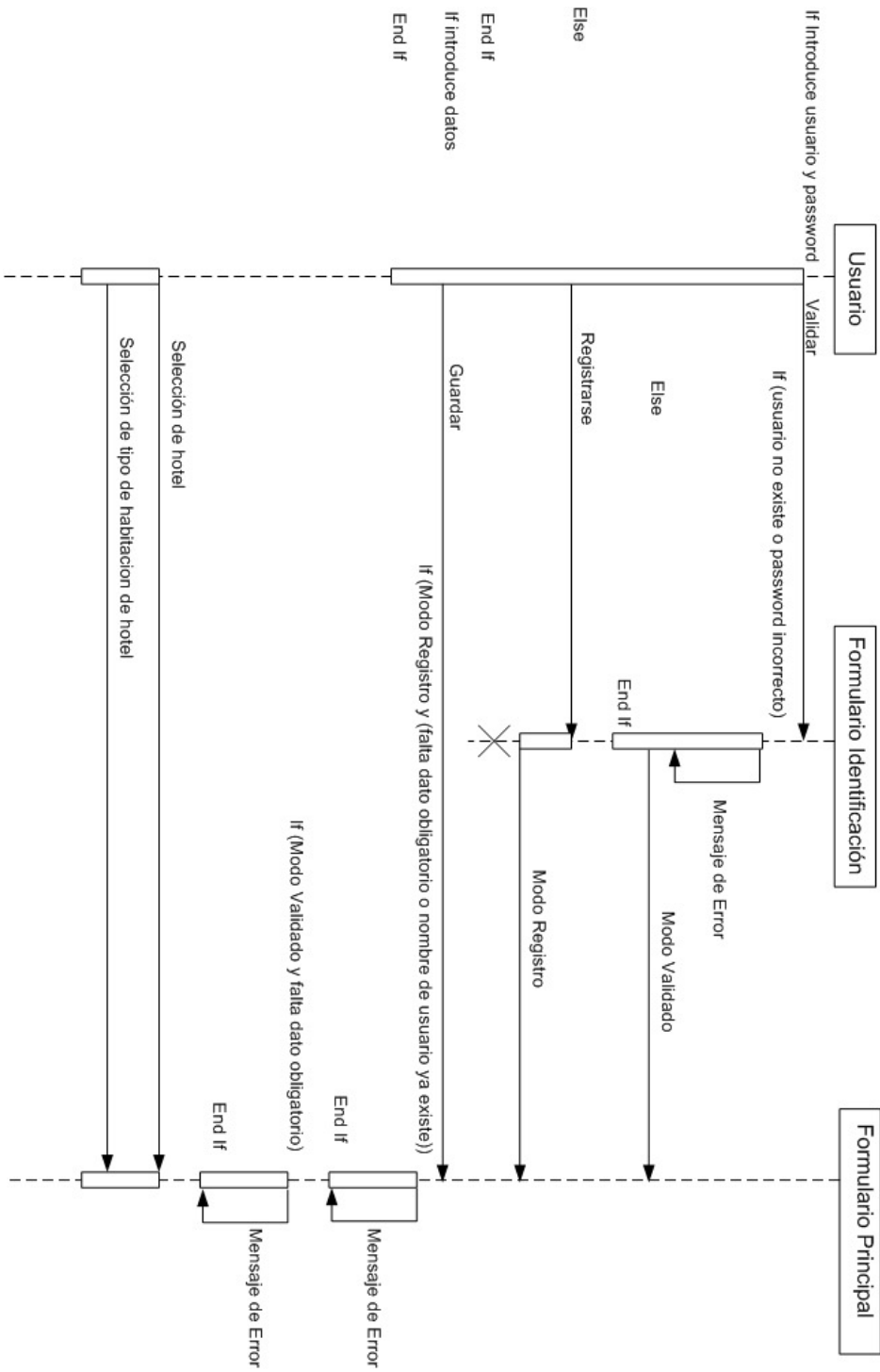
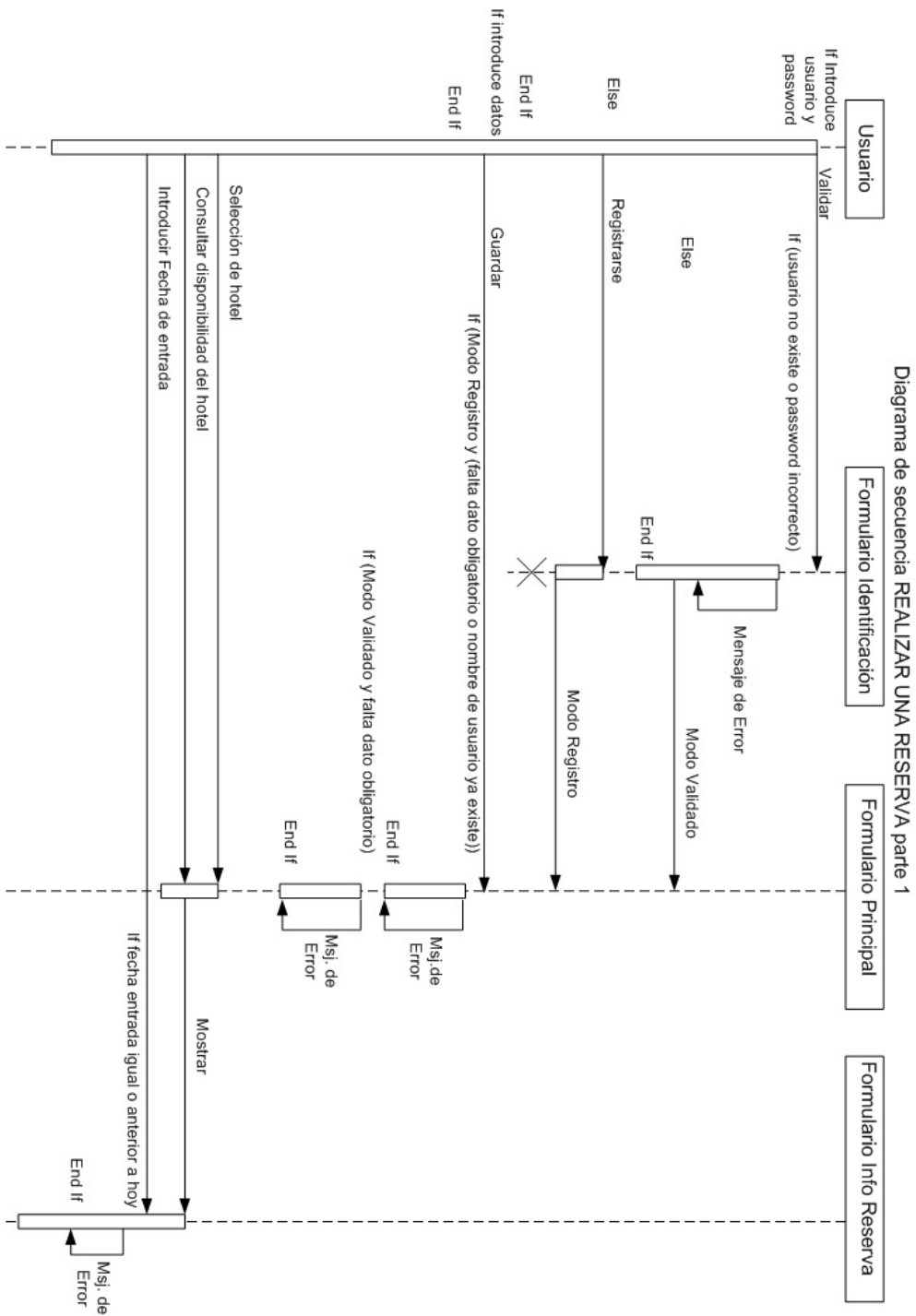
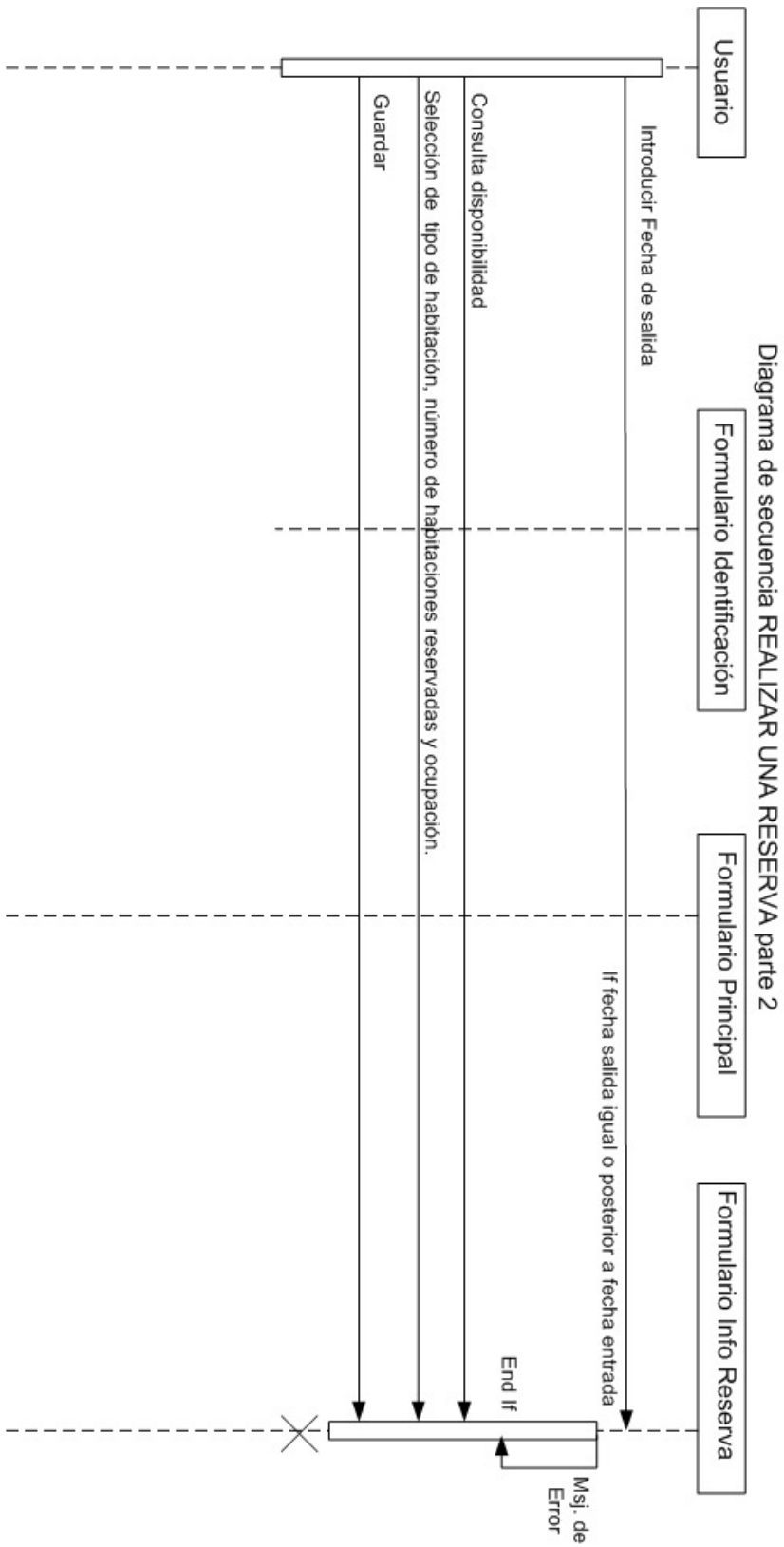
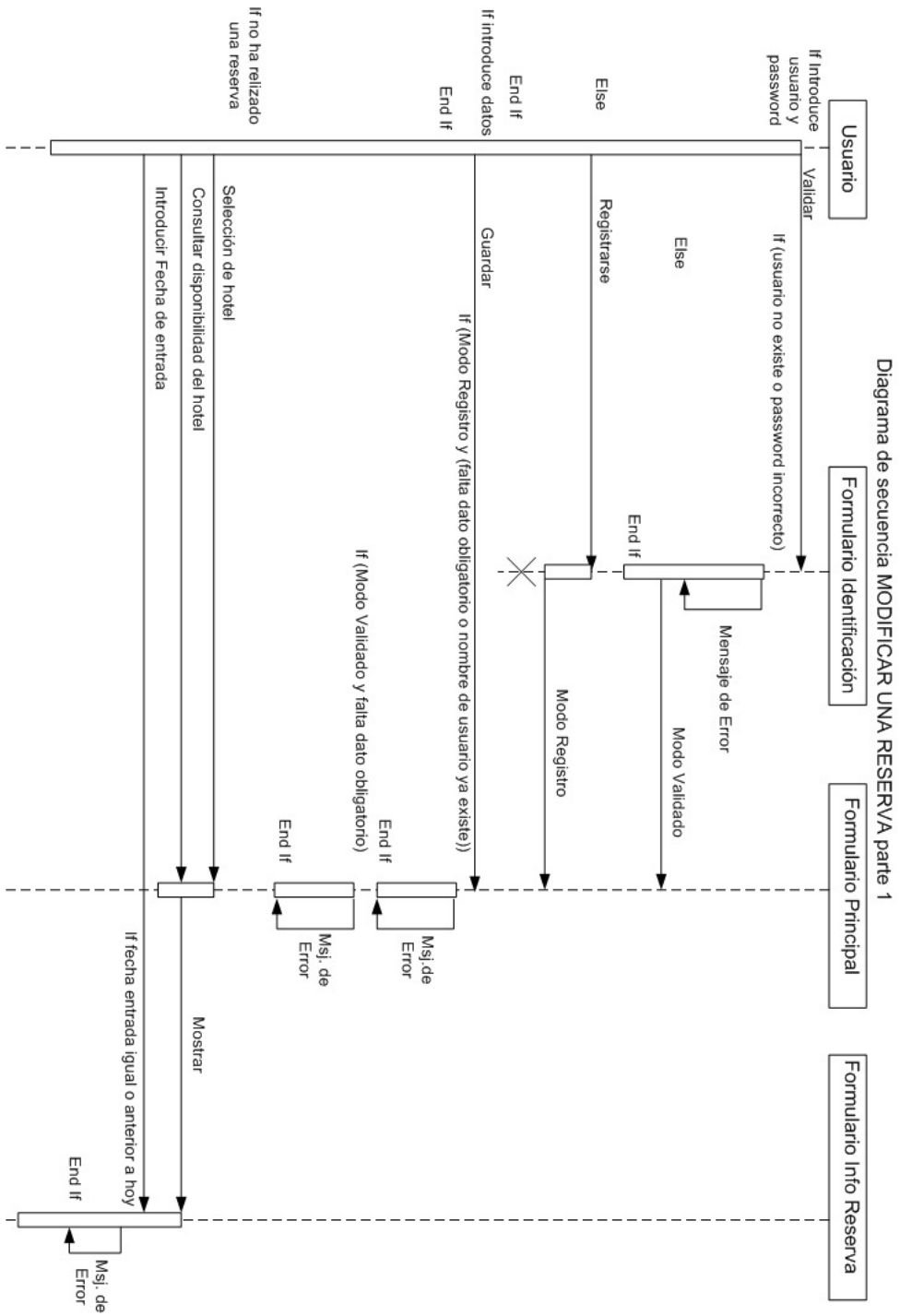


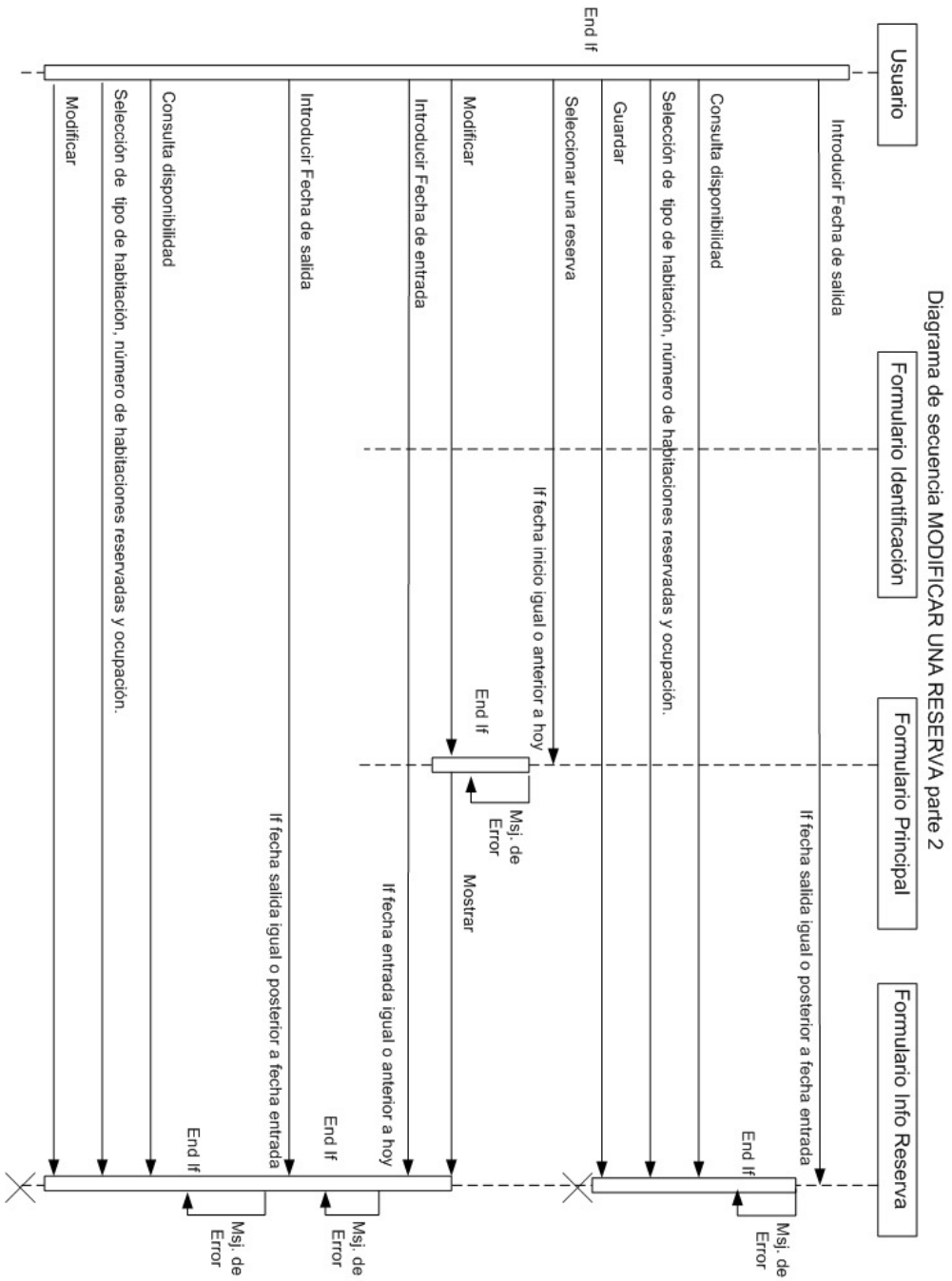
Diagrama de secuencia VISUALIZACION DE LOS DATOS DEL HOTEL Y SUS TIPOS DE HABITACION

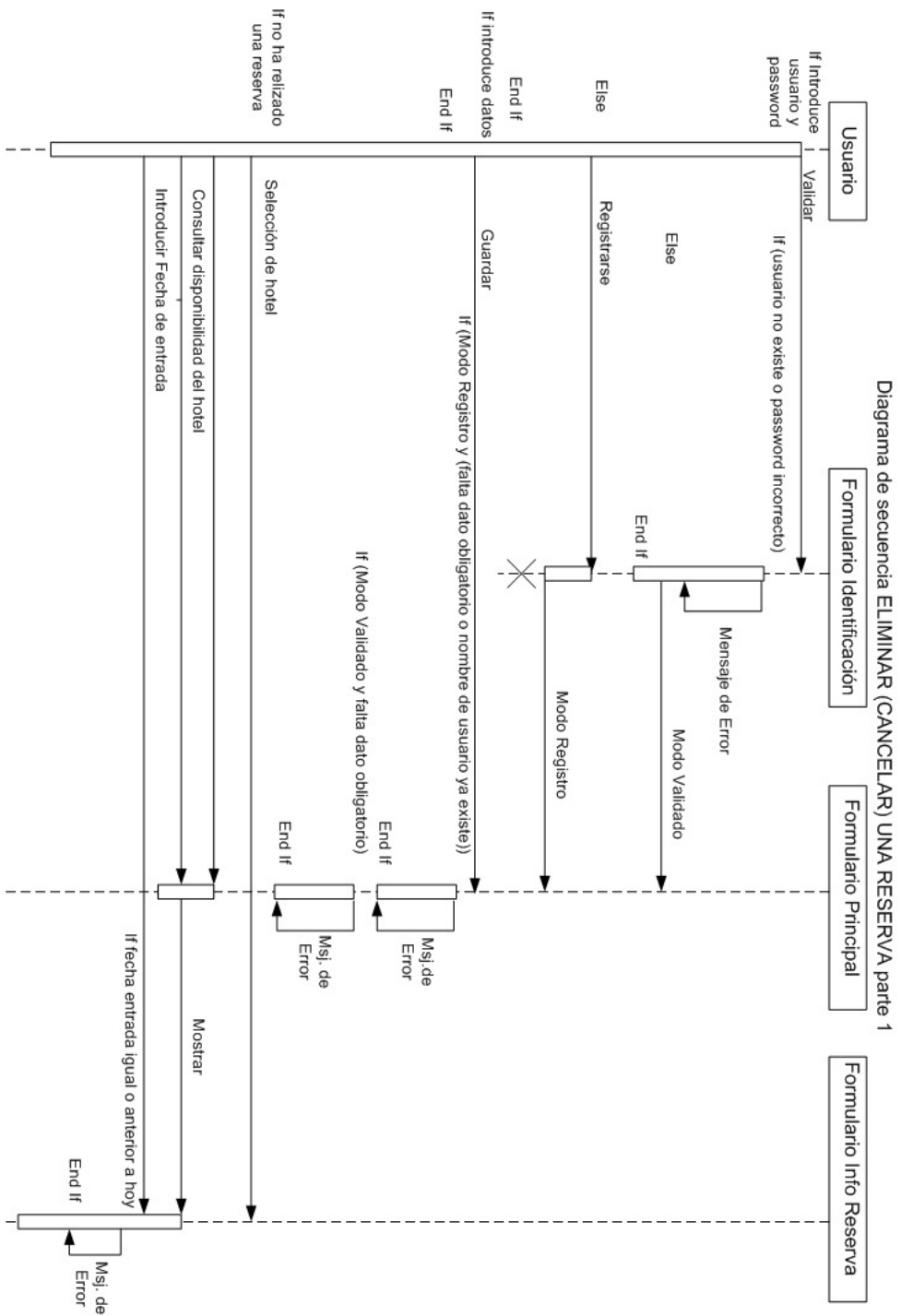


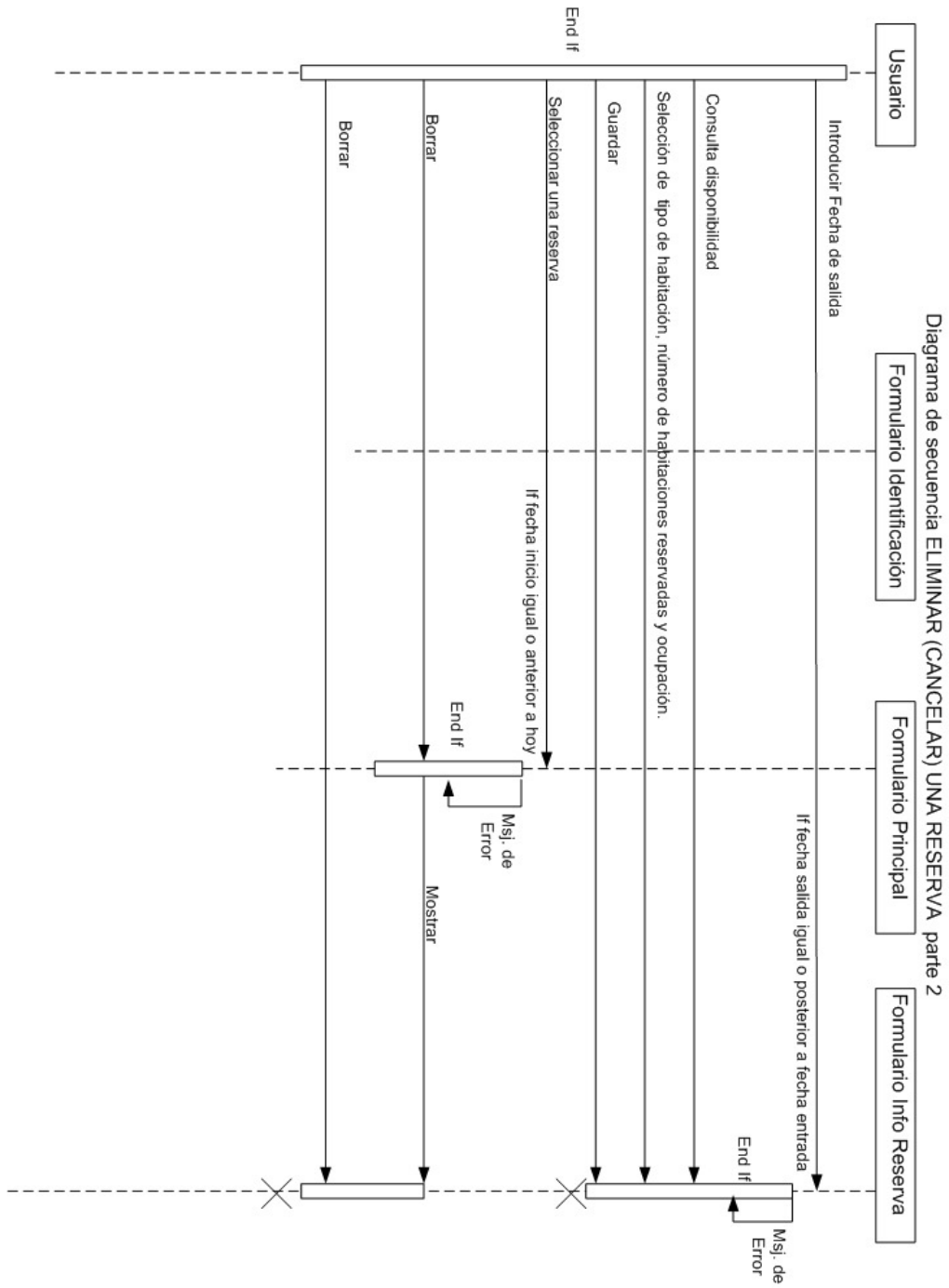


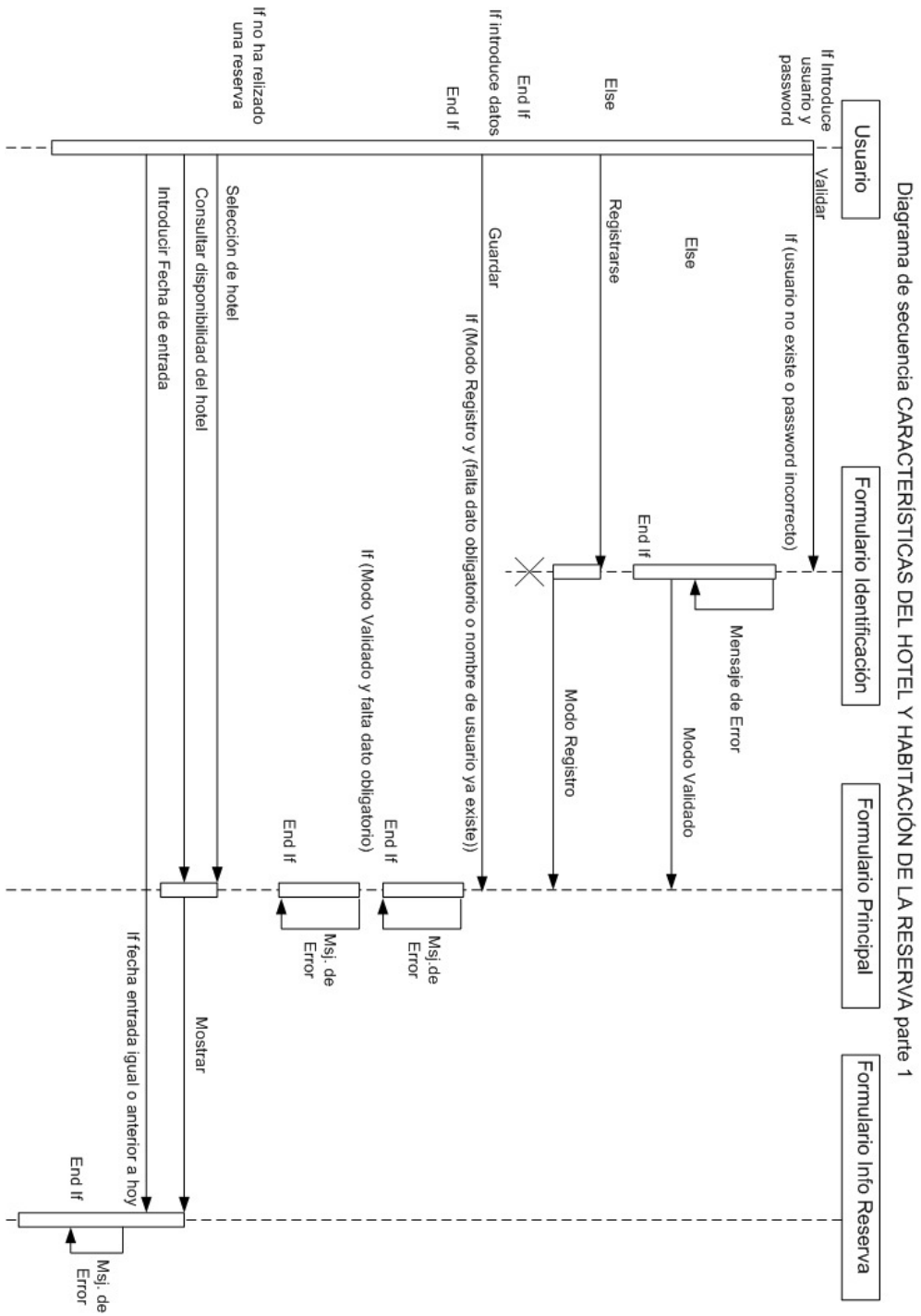


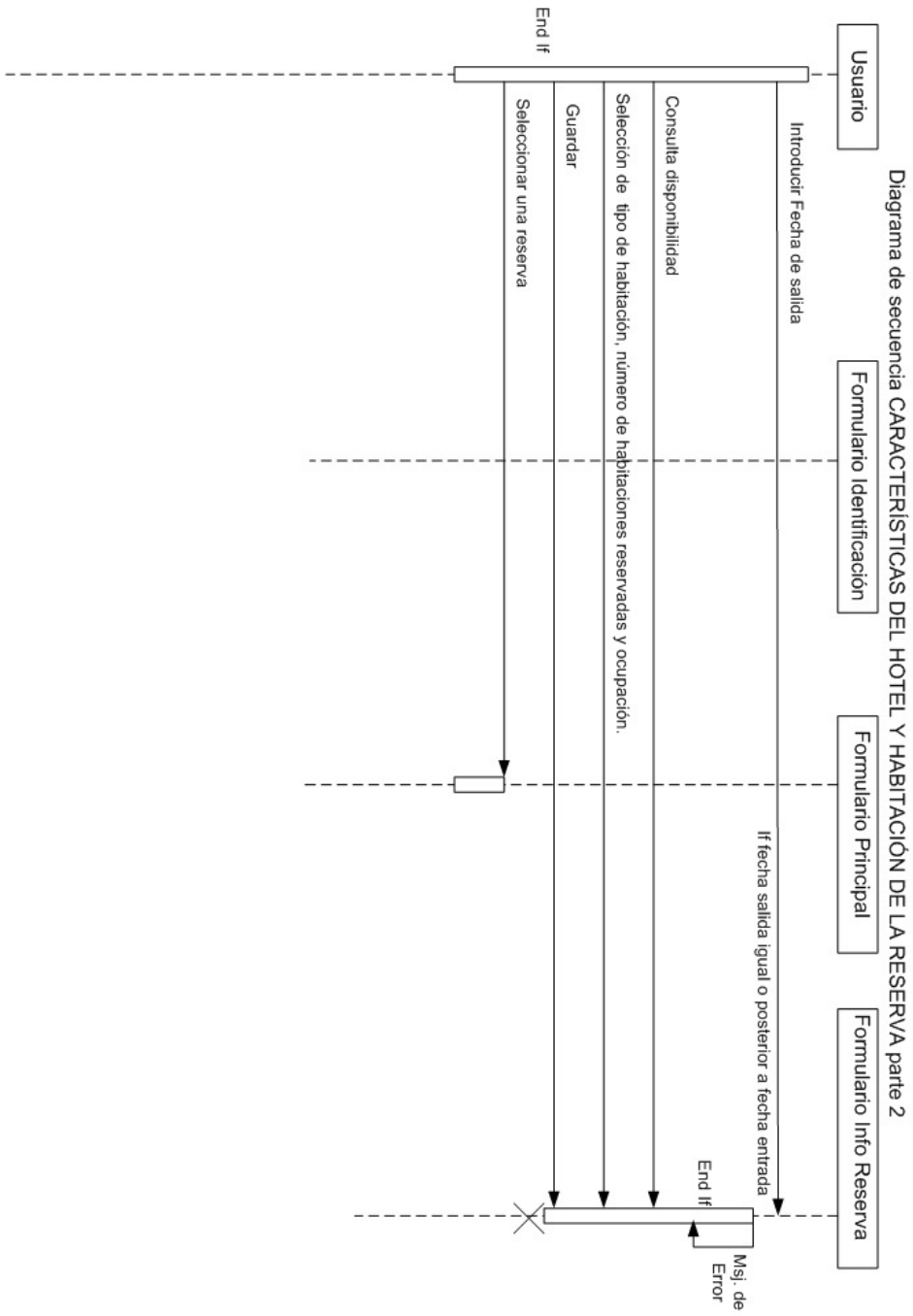


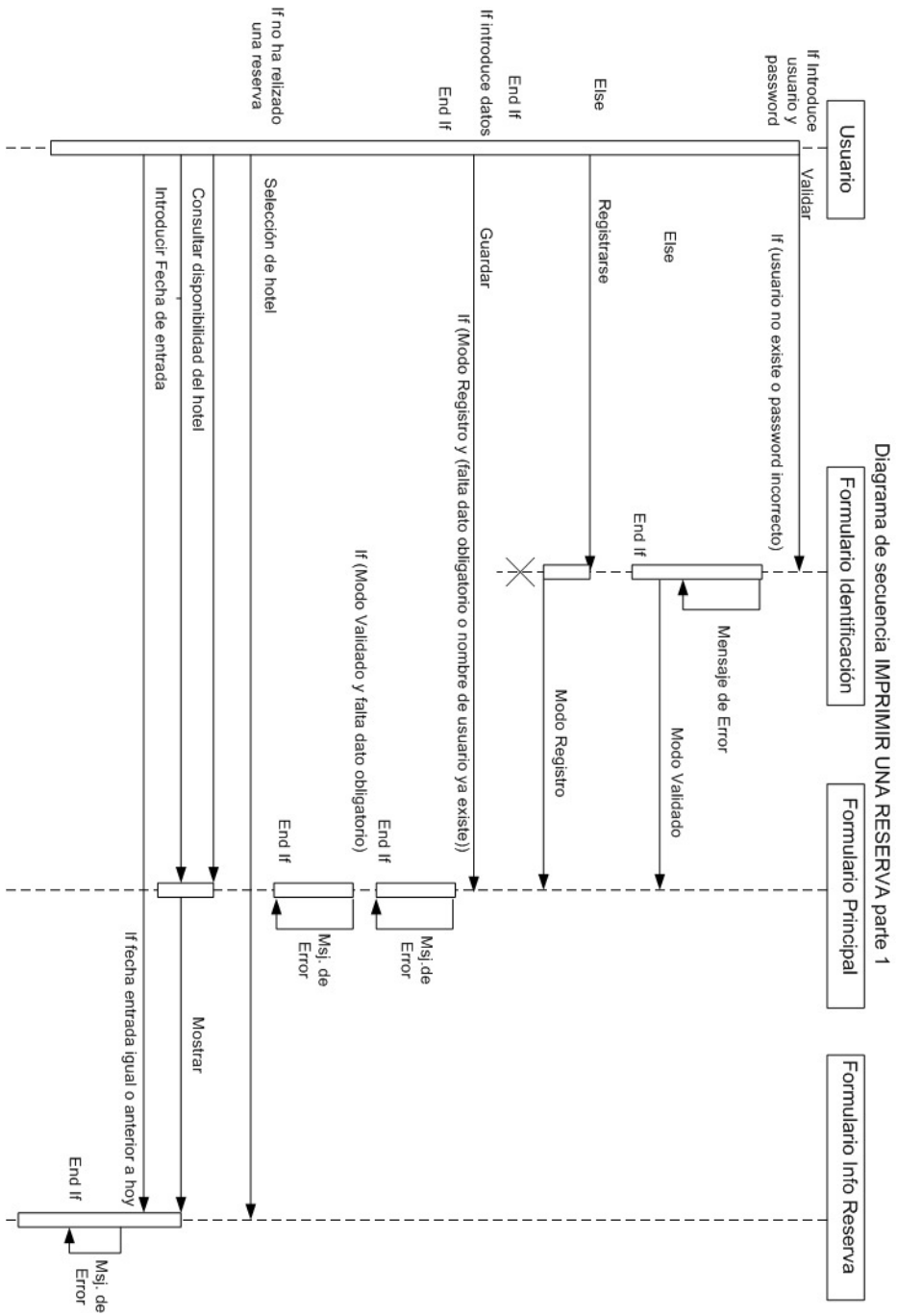


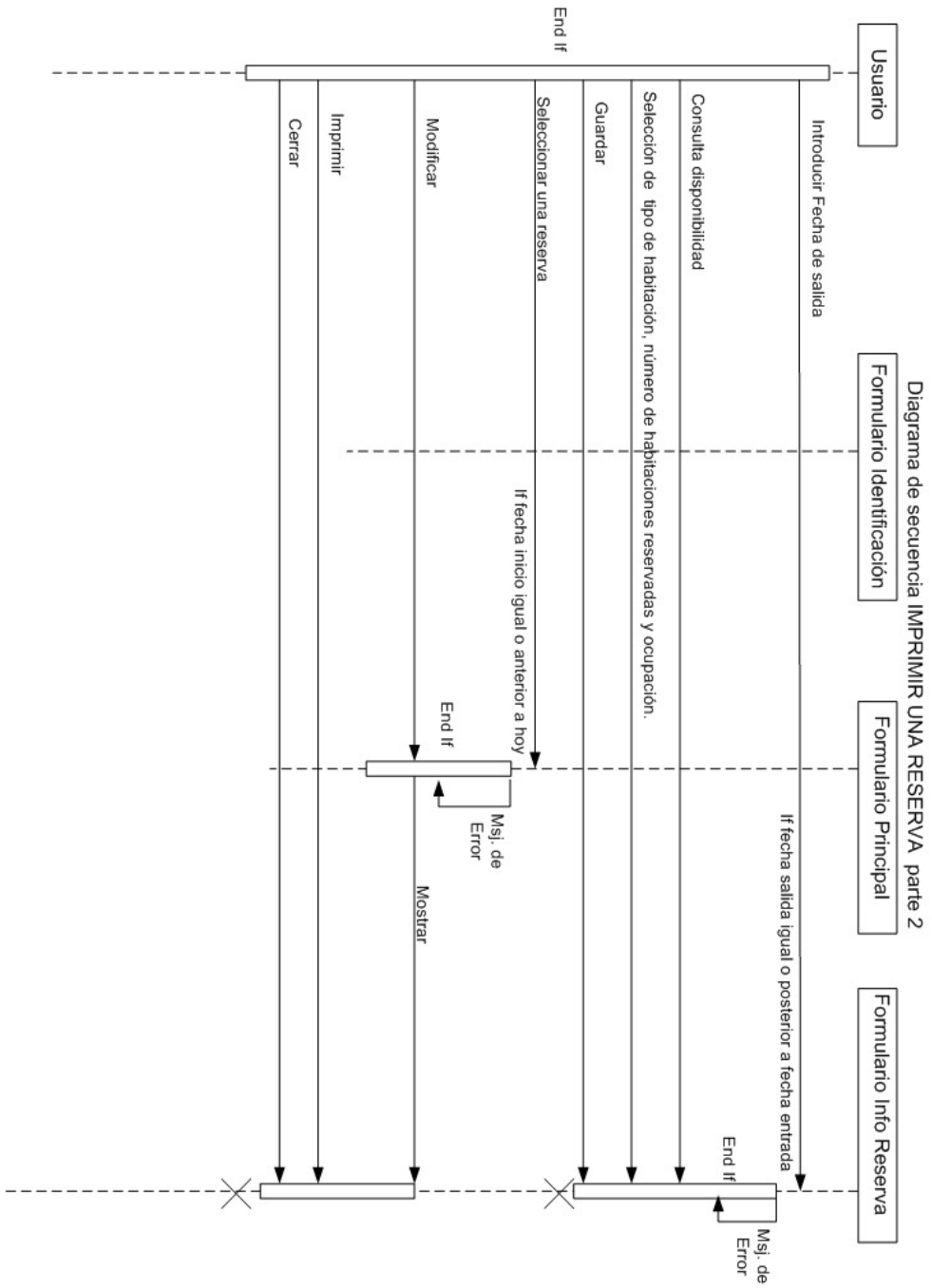




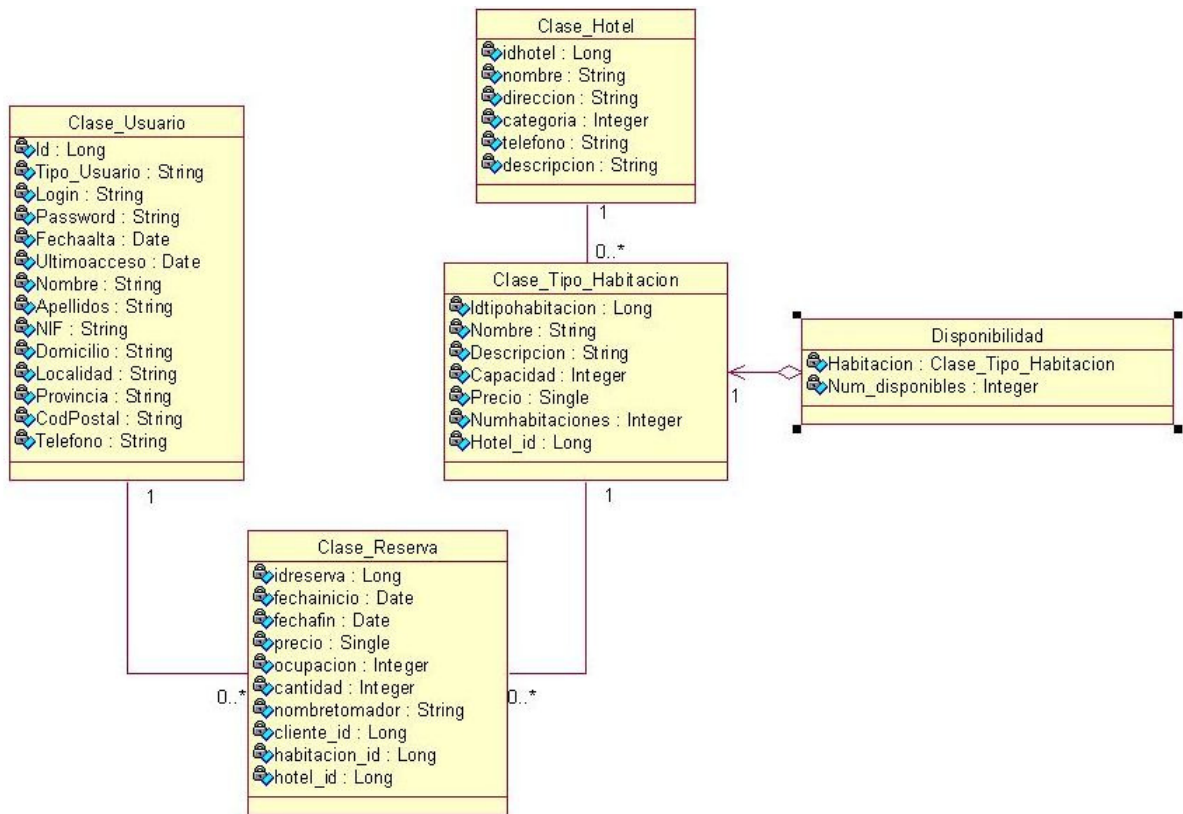








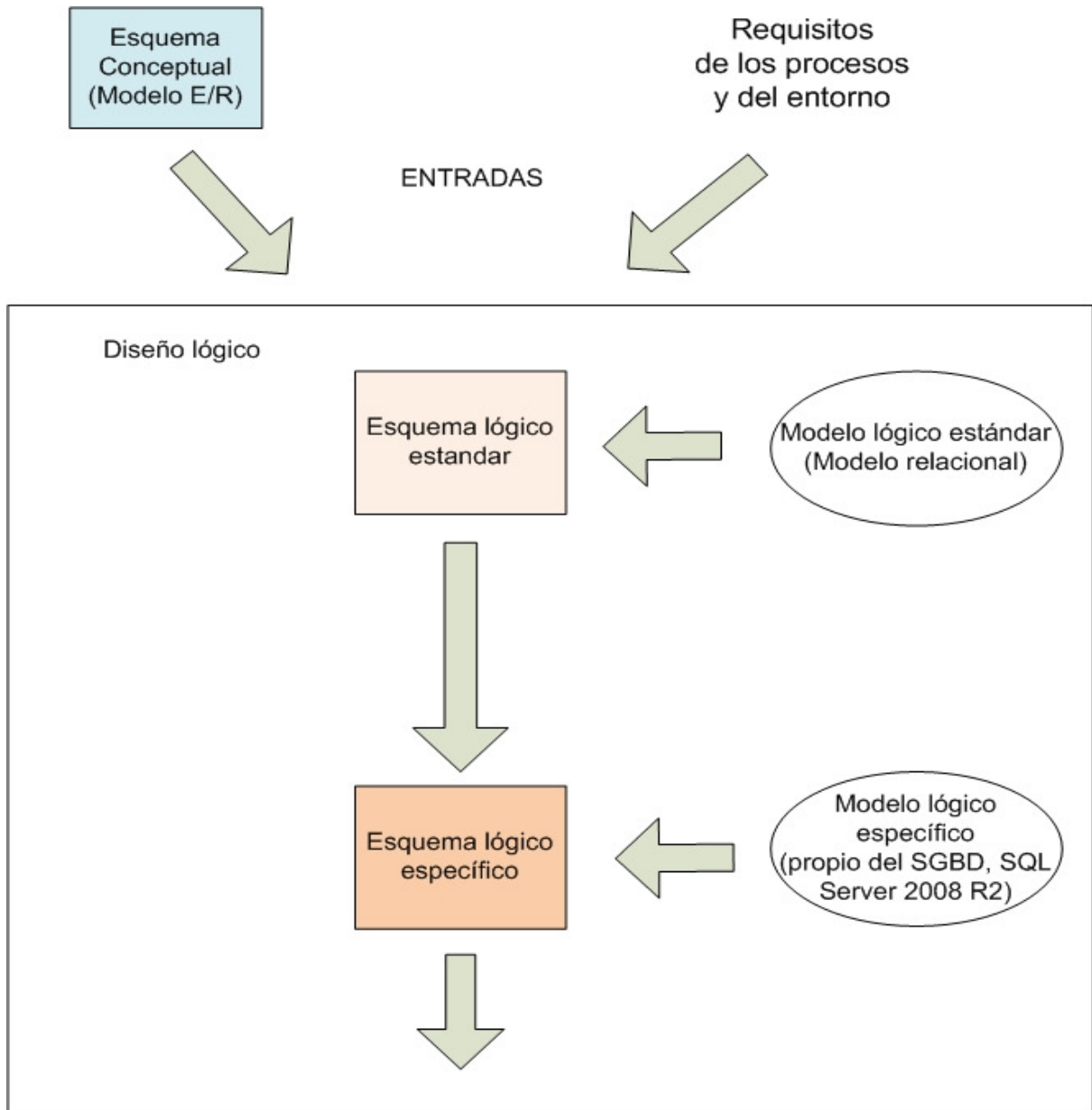
2.3.2.3 Diagrama estático de clases



2.4 Diseño

2.4.1 Diseño de la Base de Datos

2.4.1.1 Entradas y etapas del diseño lógico



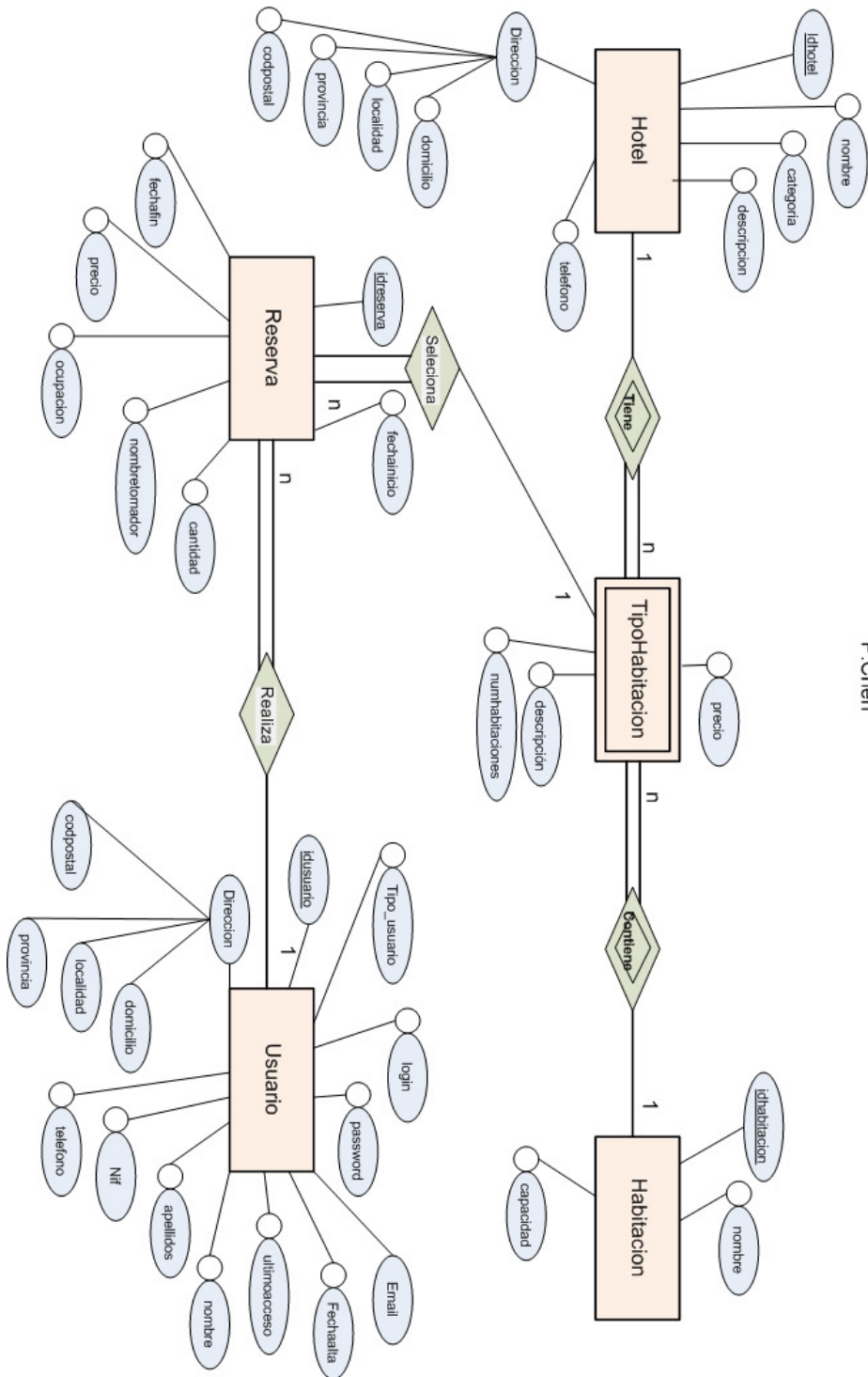
2.4.1.2 Modelo conceptual de los datos: Modelo Entidad/Relación

El modelo Entidad/Relación proporciona conceptos que permiten representar la realidad en términos de entidades (objetos), atributos y relaciones entre entidades.

Un objeto (entidad) es cualquier persona, concepto, suceso o evento, con existencia independiente, sobre la cual se almacena información.

Un objeto se define a través de sus propiedades (o atributos), es decir, a través de cualquier información que interesa del objeto y que no tiene existencia independiente de él.

Una relación es una asociación entre objetos. Cada ocurrencia de una relación asocia una ocurrencia de cada uno de los objetos relacionados



MODELO ENTIDAD / RELACION RESERVAS DE HOTEL 1.0
P.Chen



Restricciones estáticas

1. Restricciones sobre atributos

Entidad Hotel

| Atributo | Restricción de valor (o Dominio) | Restricción de valor no nulo |
|-----------------|---|-------------------------------------|
| idhotel | entero largo | valor no nulo |
| nombre | cadena de hasta 100 caracteres | valor no nulo |
| descripcion | cadena de hasta 1000 caracteres | valor no nulo |
| categoria | entero corto | valor no nulo |
| domicilio | cadena de hasta 100 caracteres | valor no nulo |
| localidad | cadena de hasta 50 caracteres | valor no nulo |
| provincia | cadena de hasta 50 caracteres | valor no nulo |
| codpostal | cadena de hasta 50 caracteres | valor no nulo |
| telefono | cadena de hasta 50 caracteres | valor no nulo |

Entidad TipoHabitacion

| Atributo | Restricción de valor (o Dominio) | Restricción de valor no nulo |
|-----------------|---|-------------------------------------|
| descripcion | cadena de hasta 1000 caracteres | valor no nulo |
| precio | decimal | valor no nulo |
| numhabitaciones | entero corto | valor no nulo |

Entidad Habitación

| Atributo | Restricción de valor (o Dominio) | Restricción de valor no nulo |
|-----------------|---|-------------------------------------|
| idhabitacion | entero largo | valor no nulo |
| nombre | cadena de hasta 100 caracteres | valor no nulo |
| capacidad | entero corto | valor no nulo |

Entidad Reserva

| Atributo | Restricción de valor (o Dominio) | Restricción de valor no nulo |
|-----------------|---|-------------------------------------|
| idreserva | entero largo | valor no nulo |
| fechainicio | fecha corta (dd-mm-aaaa) | valor no nulo |
| fechafin | fecha corta (dd-mm-aaaa) | valor no nulo |
| precio | decimal | valor no nulo |
| ocupacion | entero corto | valor no nulo |
| nombretomador | cadena de hasta 100 caracteres | valor no nulo |
| cantidad | entero | valor no nulo |

Entidad Usuario

| Atributo | Restricción de valor (o Dominio) | Restricción de valor no nulo |
|-----------------|---|-------------------------------------|
| idusuario | entero largo | valor no nulo |
| tipo_usuario | "Cliente" o "administrador" | valor no nulo |
| login | cadena de hasta 50 caracteres | valor no nulo |
| password | cadena de hasta 50 caracteres | valor no nulo |
| email | cadena de hasta 50 caracteres | permitir valor nulo |
| fechaalta | fecha corta (dd-mm-aaaa) | valor no nulo |
| ultimoacceso | fecha corta (dd-mm-aaaa) | valor no nulo |
| nombre | cadena de hasta 50 caracteres | valor no nulo |
| apellidos | cadena de hasta 50 caracteres | valor no nulo |
| nif | cadena de hasta 50 caracteres | valor no nulo |
| domicilio | cadena de hasta 100 caracteres | permitir valor nulo |
| localidad | cadena de hasta 50 caracteres | permitir valor nulo |
| codpostal | cadena de hasta 50 caracteres | permitir valor nulo |



| | | |
|-----------|-------------------------------|---------------------|
| provincia | cadena de hasta 50 caracteres | permitir valor nulo |
| telefono | cadena de hasta 50 caracteres | permitir valor nulo |

2. Restricciones sobre entidades

No puede haber dos hoteles con el mismo código de hotel ni con el mismo nombre

No puede haber dos tipos de habitaciones con el mismo código de hotel y código de habitación

No puede haber dos habitaciones con el mismo código de habitación ni con el mismo nombre

No puede haber dos reservas con el mismo código de reserva.

No puede haber dos usuarios con el mismo código de usuario ni con el mismo login

Entidad Hotel:

Atributo identificador: idhotel

Atributo identificador alternativo: nombre

Entidad débil TipoHabitacion:

Atributo identificador: idhotel , idhabitacion

Entidad Habitación:

Atributo identificador: idhabitacion

Atributo identificador alternativo: nombre

Entidad Reserva:

Atributo identificador : idreserva

Entidad Usuario:

Atributo identificador : idusuario

Atributo identificador alternativo: login

3. Restricciones sobre relaciones (restricciones de cardinalidad)

Un hotel tiene cero ó muchos tipos de habitaciones.

Un hotel tiene cero ó muchos usuarios.

Una reserva tiene una y sólo una tipo de habitación

Una reserva tiene uno y sólo un cliente

Un tipo de habitación tiene uno y sólo una habitación

Un tipo de habitación pertenece a uno y sólo un hotel

Un tipo de habitación puede tener cero o más reservas

Una habitación pertenece a cero o más tipos de habitación

Un usuario realiza cero o más reservas

2.4.1.3 Modelo lógico estándar: Modelo relacional

La estructura básica, y única, del modelo relacional es la relación, que sirve para representar tanto los objetos como las asociaciones entre ellos. Los atributos son las propiedades de las relaciones, y se definen sobre los dominios.

Un dominio es un conjunto nominado, finito y homogéneo de valores atómicos.

Un atributo es la interpretación de un determinado dominio en una relación. Un atributo está siempre asociado a una relación, mientras que un dominio tiene existencia propia con independencia de las relaciones. Un atributo representa una propiedad de la relación y toma valores de un dominio.



Una relación $r(R)$ es un conjunto de elementos denominados tuplas $\{t_j\}$. Cada tupla t_j es un conjunto de pares $(\langle A_1:V_{1j} \rangle \dots \langle A_1:V_{1j} \rangle \dots \langle A_n:V_{nj} \rangle)$ donde cada A_i es el nombre de un atributo y V_{ij} es un valor correspondiente al dominio D_i sobre el que está definido el atributo:

$$r(R) = t_j \{(\langle A_1:V_{1j} \rangle \dots \langle A_1:V_{1j} \rangle \dots \langle A_n:V_{nj} \rangle) : V_{ij} \in D_i\}$$

El número de atributos se llama grado de la relación.

El número de tuplas se llama cardinalidad de la relación. La cardinalidad varía en el transcurso del tiempo.

Restricciones de integridad en el modelo relacional

1. Restricción de unicidad

La definición de una restricción de unicidad sobre un conjunto de atributos K de la relación R expresa la siguiente propiedad: "no debe de haber en R dos tuplas que tengan el mismo valor en todos los atributos del conjunto K ".

2. Restricción de valor no nulo

La definición de una restricción de valor no nulo sobre un conjunto de atributos K de la relación R expresa la siguiente propiedad: "no debe de haber en R una tupla que tengan el valor nulo en algún atributo del K ".

3. Clave primaria

La clave primaria de una relación es un conjunto de atributos de su esquema que son elegidos para servir de identificador unívoco de sus tuplas. Debe cumplir: ser minimal (todos los atributos que lo forman sean necesarios para la función de identificación), sus atributos no pueden tener valores nulos y además este valor debe ser unívoco para cada tupla.

4. Clave ajena

Las claves ajenas son el mecanismo que proporciona el modelo relacional para expresar relaciones (como sinónimo de asociaciones) entre objetos representados en el esquema de la base de datos, la forma de hacerlo es incluir en el esquema de una relación atributos de otra (siempre que estos atributos identifiquen de forma unívoca a las tuplas de la relación). A este conjunto de atributos se le conoce como clave ajena de la relación R que hace referencia a la relación S .

5. Integridad referencial



Si una relación R tiene una clave ajena CA que hace referencia a la clave primaria CP de la relación S, todo valor de CA de coincidir con un valor de CP o ser nulo. Esta es condición de la restricción, la cual puede expresarse como un predicado:

$$R.CA = S.CP$$

Las clave ajena y clave primaria han de ser definidas sobre el mismo dominio y se permite que sobre la clave ajena se defina, si es necesario, la restricción de valor no nulo (si no se define, la clave admitirá valores nulos).

En cuanto a la acción, es de tipo específico. Si se intenta insertar una tupla en la tabla de referencia R que no cumpla la condición, la acción se rechaza. Si la condición falla debido a una operación de borrado de tuplas o modificación de la clave primaria en la tabla referenciada S, existe la posibilidad de elegir entre cuatro opciones, tanto para la operación de borrado como la de modificación:

- No action (rechaza la operación)
- Cascade (propagar la modificación o borrar las tuplas de la tabla que referencia)
- Set null (poner a nulo en la clave ajena de la tabla que referencia)
- Set default (poner valor por defecto en CA de la tabla que referencia)

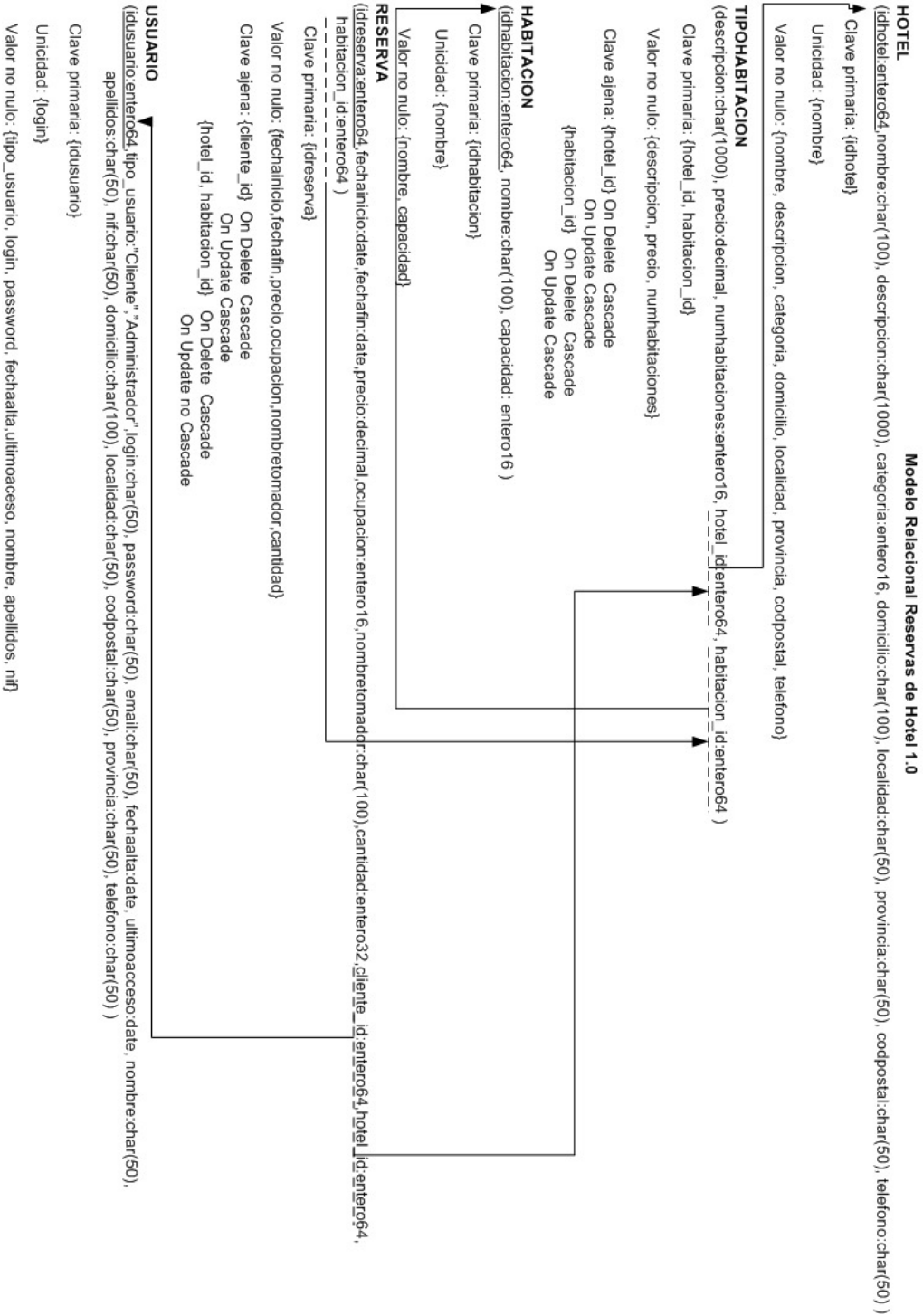
Transformación del esquema conceptual (modelo E/R) al esquema lógico estándar (Modelo relacional)

Reglas:

1. Todo tipo de entidad se convierte en una relación.
2. Todo tipo de interrelación N:M se transforma en una relación
3. Para todo tipo de interrelación 1:N se realiza lo que se denomina propagación de clave (regla general)

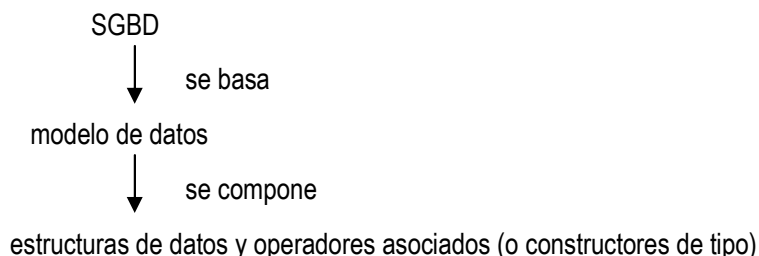
Propagación de clave: Propagar los atributos identificadores del tipo de entidad que tiene cardinalidad máxima 1 a la de N





2.4.1.4 Modelo lógico específico del SGBD: SQL Server 2008 R2 Express

Un sistema de gestión de bases de datos es una herramienta de software que permite la creación y manipulación de bases de datos definidas de acuerdo a las reglas del modelo subyacente al sistema.



Los sistemas de gestión de bases de datos proporcionan una interfaz entre los programas de aplicación que acceden a los datos y el sistema operativo, caracterizándose principalmente porque permiten una descripción unificada de los datos y la definición de vistas parciales de los mismos para distintos usuarios.

Objetivos, funciones y componentes de un SGBD

| Objetivos de las técnicas de BD | Funciones del SGBD | Componentes del SGBD |
|---|--|--|
| Descripción unificada de los datos e independiente de las aplicaciones <hr/> Independencia de las aplicaciones respecto a la representación física de los datos <hr/> Definición de vistas parciales de los datos para distintos usuarios | Definición de la base de datos a varios niveles: esquemas - esquema lógico (definición de las estructuras de la base de datos) - esquema interno (implementación de las estructuras del esquema lógico) - esquemas externos (definición de las estructuras derivadas) Establecer la correspondencia entre los esquemas | Lenguajes para la definición de esquemas y los traductores asociados |
| Gestión de la información | Manipulación: consulta y actualización Gestión y administración de la base de datos | Lenguajes de manipulación y traductores asociados Herramientas para: - reestructuración - simulación, estadísticas - impresión |



| | | |
|--|---|---|
| <p>Integridad y seguridad de los datos</p> | <p>Control de :</p> <ul style="list-style-type: none"> - la integridad semántica - los accesos concurrentes - la reconstrucción de la base de datos en caso de fallo - la seguridad | <p>Herramientas para:</p> <ul style="list-style-type: none"> - control de integridad - reconstrucción frente a fallos - control de seguridad |
|--|---|---|

Creación de las tablas en lenguaje Transact-SQL

HABITACION

```
USE [Reservas_hotel]
```

```
GO
```

```
/****** Object: Table [dbo].[Habitacion] Script Date: 12/21/2011
20:33:52 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
SET ANSI_PADDING ON
```

```
GO
```

```
CREATE TABLE [dbo].[Habitacion] (
    [idhabitacion] [bigint] IDENTITY(1,1) NOT NULL,
    [nombre] [varchar](100) NOT NULL,
    [capacidad] [tinyint] NOT NULL,
    CONSTRAINT [PK_TipoHabitacion] PRIMARY KEY CLUSTERED
(
    [idhabitacion] ASC
```


Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],

    CONSTRAINT [IX_Habitacion] UNIQUE NONCLUSTERED

(

    [nombre] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]
```

GO

SET ANSI_PADDING OFF

GO

HOTEL

USE [Reservas_hotel]

GO

```
/****** Object: Table [dbo].[Hotel]      Script Date: 12/21/2011
20:34:50 *****/
```

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

```
CREATE TABLE [dbo].[Hotel](

    [idhotel] [bigint] IDENTITY(1,1) NOT NULL,

    [nombre] [varchar](100) NOT NULL,
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
[descripcion] [varchar](1000) NOT NULL,  
[categoria] [tinyint] NOT NULL,  
[domicilio] [varchar](100) NOT NULL,  
[localidad] [varchar](50) NOT NULL,  
[provincia] [varchar](50) NOT NULL,  
[codpostal] [varchar](50) NOT NULL,  
[telefono] [varchar](50) NOT NULL,  
[administrador_id] [bigint] NOT NULL,  
CONSTRAINT [PK_Hotel] PRIMARY KEY CLUSTERED  
(  
    [idhotel] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY  
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],  
CONSTRAINT [IX_Hotel] UNIQUE NONCLUSTERED  
(  
    [nombre] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY  
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]  
  
GO  
  
SET ANSI_PADDING OFF  
  
GO  
  
RESERVA  
USE [Reservas_hotel]  
  
GO  
  
/***** Object: Table [dbo].[Reserva]      Script Date: 12/21/2011  
20:35:15 *****/  
  
SET ANSI_NULLS ON
```



GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

```
CREATE TABLE [dbo].[Reserva] (
    [idreserva] [bigint] IDENTITY(1,1) NOT NULL,
    [fechainicio] [datetime] NOT NULL,
    [fechafin] [datetime2](7) NOT NULL,
    [precio] [money] NOT NULL,
    [ocupacion] [tinyint] NOT NULL,
    [nombretomador] [varchar](100) NOT NULL,
    [cantidad] [int] NOT NULL,
    [cliente_id] [bigint] NULL,
    [hotel_id] [bigint] NULL,
    [habitacion_id] [bigint] NULL,
    CONSTRAINT [PK_Reserv] PRIMARY KEY CLUSTERED
(
    [idreserva] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

SET ANSI_PADDING OFF

GO

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
ALTER TABLE [dbo].[Reserva] WITH CHECK ADD CONSTRAINT
[FK_Reserva_TipoHabitacion] FOREIGN KEY([hotel_id], [habitacion_id])
REFERENCES [dbo].[TipoHabitacion] ([hotel_id], [habitacion_id])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
```

```
ALTER TABLE [dbo].[Reserva] CHECK CONSTRAINT
[FK_Reserva_TipoHabitacion]
GO
```

```
ALTER TABLE [dbo].[Reserva] WITH CHECK ADD CONSTRAINT
[FK_Reserva_Usuario] FOREIGN KEY([cliente_id])
REFERENCES [dbo].[Usuario] ([idusuario])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
```

```
ALTER TABLE [dbo].[Reserva] CHECK CONSTRAINT [FK_Reserva_Usuario]
GO
```

TIPOHABITACION

```
USE [Reservas_hotel]
GO
```

```
/***** Object: Table [dbo].[TipoHabitacion] Script Date:
12/21/2011 20:37:01 *****/
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
```



GO

SET ANSI_PADDING ON

GO

```
CREATE TABLE [dbo].[TipoHabitacion](
    [hotel_id] [bigint] NOT NULL,
    [habitacion_id] [bigint] NOT NULL,
    [descripcion] [varchar](1000) NOT NULL,
    [precio] [money] NOT NULL,
    [numhabitaciones] [tinyint] NOT NULL,
    CONSTRAINT [PK_TipHabitacion] PRIMARY KEY CLUSTERED
(
    [hotel_id] ASC,
    [habitacion_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

SET ANSI_PADDING OFF

GO

```
ALTER TABLE [dbo].[TipoHabitacion] WITH CHECK ADD CONSTRAINT
[FK_TipoHabitacion_Habitacion] FOREIGN KEY([habitacion_id])
REFERENCES [dbo].[Habitacion] ([idhabitacion])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
ALTER TABLE [dbo].[TipoHabitacion] CHECK CONSTRAINT  
[FK_TipoHabitacion_Habitacion]
```

```
GO
```

```
ALTER TABLE [dbo].[TipoHabitacion] WITH CHECK ADD CONSTRAINT  
[FK_TipoHabitacion_Hotel] FOREIGN KEY([hotel_id])
```

```
REFERENCES [dbo].[Hotel] ([idhotel])
```

```
ON UPDATE CASCADE
```

```
ON DELETE CASCADE
```

```
GO
```

```
ALTER TABLE [dbo].[TipoHabitacion] CHECK CONSTRAINT  
[FK_TipoHabitacion_Hotel]
```

```
GO
```

USUARIO

```
USE [Reservas_hotel]
```

```
GO
```

```
/***** Object: Table [dbo].[Usuario] Script Date: 12/21/2011  
20:37:45 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
SET ANSI_PADDING ON
```

```
GO
```

```
CREATE TABLE [dbo].[Usuario](  
    [idusuario] [bigint] IDENTITY(1,1) NOT NULL,
```

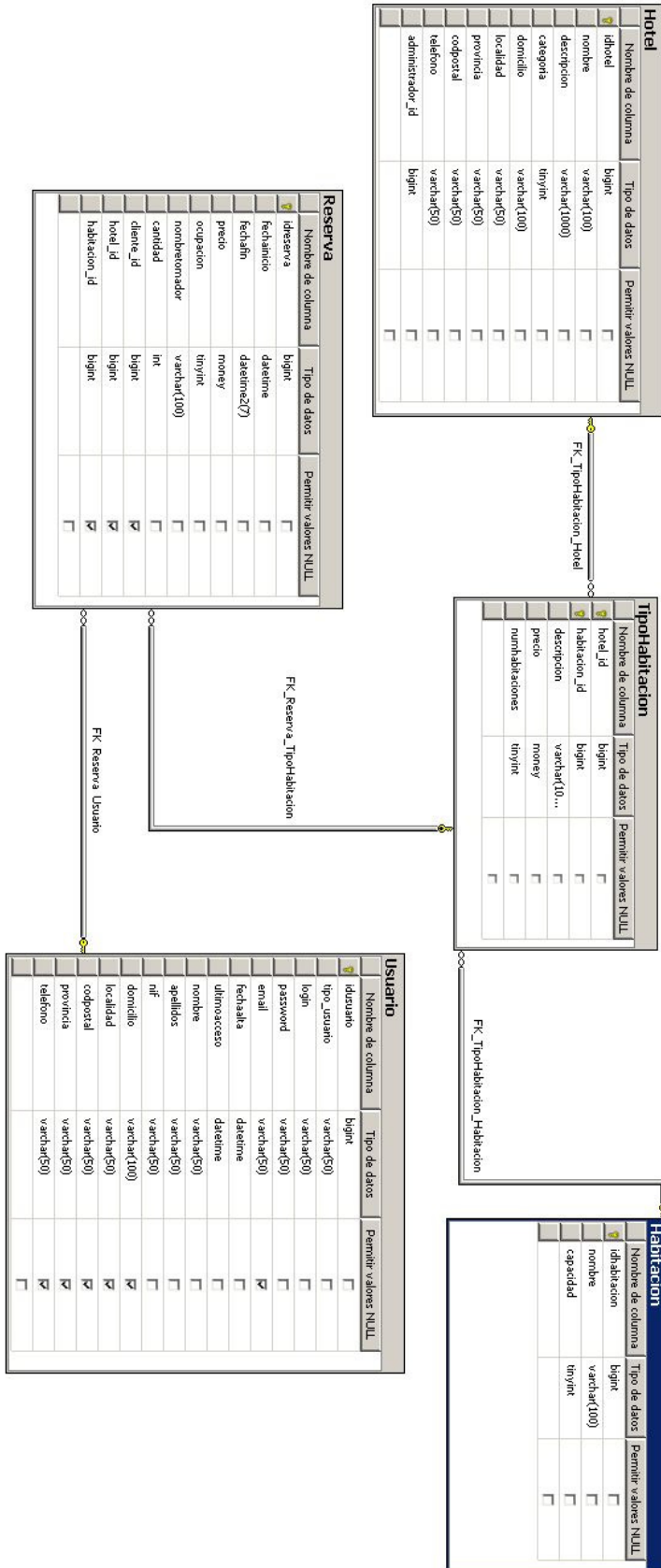


Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
[tipo_usuario] [varchar](50) NOT NULL,  
[login] [varchar](50) NOT NULL,  
[password] [varchar](50) NOT NULL,  
[email] [varchar](50) NULL,  
[fechaalta] [datetime] NOT NULL,  
[ultimoacceso] [datetime] NOT NULL,  
[nombre] [varchar](50) NOT NULL,  
[apellidos] [varchar](50) NOT NULL,  
[nif] [varchar](50) NOT NULL,  
[domicilio] [varchar](100) NULL,  
[localidad] [varchar](50) NULL,  
[codpostal] [varchar](50) NULL,  
[provincia] [varchar](50) NULL,  
[telefono] [varchar](50) NULL,  
  
CONSTRAINT [PK_Usuario] PRIMARY KEY CLUSTERED  
(  
    [idusuario] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY  
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],  
CONSTRAINT [IX_Usuario] UNIQUE NONCLUSTERED  
(  
    [login] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY  
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]  
  
GO  
  
SET ANSI_PADDING OFF  
  
GO
```



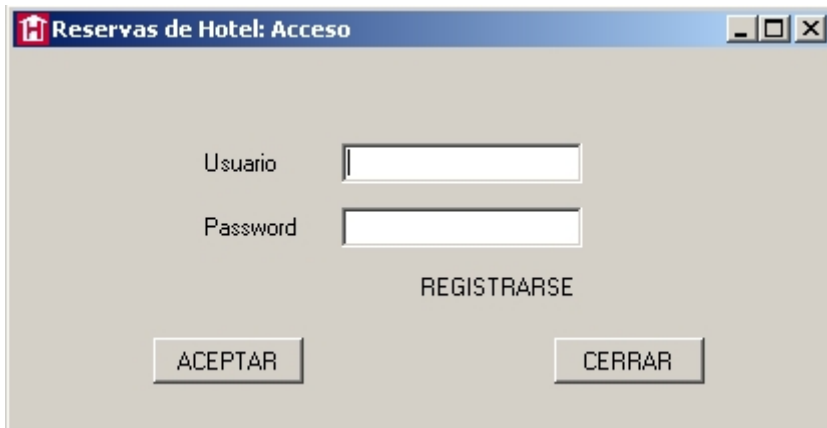
Diagrama de la Base de datos



2.4.2 Construcción y uso de la interfaz de usuario

2.4.2.1 Acceder al sistema y darse de alta en el sistema

Formulario Acceso



The screenshot shows a Windows-style window titled "Reservas de Hotel: Acceso". Inside the window, there are two text input fields. The first is labeled "Usuario" and the second is labeled "Password". Below these fields is a button labeled "REGISTRARSE". At the bottom of the window, there are two buttons: "ACEPTAR" on the left and "CERRAR" on the right.

Si el usuario esta ya registrado introduce el nombre de Usuario y el Password y pulsa ACEPTAR para acceder al formulario Usuario

Si el usuario desea salir del sistema pulsar sobre CERRAR.

Si el usuario no está registrado pulsa sobre **REGISTRARSE** para acceder al formulario Usuario darse de alta en el sistema.



This screenshot is identical to the previous one, showing the "Reservas de Hotel: Acceso" window. The only difference is that the "REGISTRARSE" button is highlighted in red, indicating it is the focus of the current instruction.

Formulario Usuario (usuario registrado)

Aparecerán las tres pestañas “USUARIO”, “RESERVA” y “LISTADO DE RESERVAS”

Pestaña USUARIO

The screenshot shows a Windows application window titled "Reservas de Hotel: Usuario". The window has a menu bar with icons for "Editar", "Guardar", "Modificar", "Borrar", "Darse de baja", "Salir", and a help icon. Below the menu bar are three tabs: "USUARIO", "RESERVA", and "LISTADO RESERVAS". The "USUARIO" tab is active and displays a form with the following fields:

(*) Campos obligatorios

Tipo de usuario:

(*)Nombre:

(*)Apellidos: (*)Login:

(*)NIF: (*)Password:

Domicilio: Localidad:

Codigo Postal: Provincia:

Email:

Teléfono:

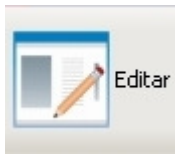
Fecha Alta: Ultimo acceso:

En la pestaña “USUARIO” aparecen todos los datos personales de un usuario del sistema (Cliente) así como la fecha de alta en el sistema y el último acceso al sistema.

En la pestaña “RESERVA” podremos acceder a la información del hotel y de sus habitaciones y también realizar una reserva.

En la pestaña “LISTADO DE RESERVAS” podremos ver las el historial de reservas realizado y si es posible modificar o borrar alguna reserva.

Opciones:



Pulsar sobre el botón "Editar" para habilitar los campos de datos de la pestaña Usuario. Los campos Nombre, Apellidos, NIF, Login Y Password son obligatorios.



Pulsar sobre el botón "Darse de baja" para eliminar todos los datos de usuario del sistema. Se pedirá confirmación.



Pulsar sobre el botón "Salir" para salir de la aplicación. Se pedirá confirmación.

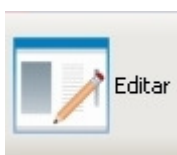


Pulsar sobre el botón "Información" para ver información sobre la aplicación

Formulario Usuario (usuario no registrado)

Inicialmente solo aparece la pestaña "USUARIO" ya que el usuario no registrado inicialmente tiene que escribir sus datos y darse de alta en el sistema para poder realizar reservas.

Opciones:



Pulsar sobre el botón "Editar" para habilitar los campos de datos de la pestaña "USUARIO". Los campos Nombre, Apellidos, NIF, Login y Password son obligatorios.

Opciones:



Una vez pulsado el botón “Editar” (para habilitar los campos) y escrito los datos de usuario pulsaremos sobre el botón “Guardar” para darse de alta en el sistema y almacenar los datos de usuario. Aparecerán las pestañas de “RESERVA” y “LISTADO DE RESERVAS”



Pulsar sobre el botón “Salir” para salir de la aplicación. Se pedirá confirmación.



Pulsar sobre el botón “Información” para ver información sobre la aplicación

2.4.2.2 Consultar información del Hotel y hacer una reserva

Pestaña RESERVA

En la pestaña “RESERVA” podremos acceder a la información del hotel y de sus habitaciones y también realizar una reserva.

The screenshot shows a Windows application window titled "Reservas de Hotel: Usuario". The window has a toolbar with icons for "Editar", "Guardar", "Modificar", "Borrar", "Darse de baja", "Salir", and a help icon. Below the toolbar are three tabs: "USUARIO", "RESERVA" (which is selected), and "LISTADO RESERVAS".

The "RESERVA" tab contains the following sections:

- Busqueda del Hotel:** Two radio buttons labeled "Localidad:" and "Hotel:". The "Hotel:" radio button is selected. Below "Localidad:" is a text input field. Below "Hotel:" is a dropdown menu with "Lista de hoteles" selected. To the right of the "Hotel:" dropdown is a button labeled "Consultar disponibilidad".
- Información del Hotel:** Fields for "Nombre:", "Dirección:", "Descripción:", "Categoría:", and "Teléfono:". The "Categoría:" field has a dropdown menu with "estrellas" selected.
- Información del tipo de habitaciones del Hotel:** Fields for "Nombre:", "Capacidad:", "Importe por noche:", and "Descripción:". The "Nombre:" field has a dropdown menu with "Lista de tipos de habitaciones" selected. The "Importe por noche:" field has a currency symbol "€" to its right.

Tenemos dos opciones para seleccionar un hotel (**Búsqueda de hotel**):

1. Busqueda por Localidad

Pulsaremos en la opción Localidad y se nos habilitará el campo para escribir la localidad donde buscar un hotel.

The screenshot shows a Windows application window titled "Reservas de Hotel: Usuario". The window has a menu bar with icons for "Editar", "Guardar", "Modificar", "Borrar", "Darse de baja", "Salir", and a help icon. Below the menu bar are three tabs: "USUARIO", "RESERVA", and "LISTADO RESERVAS". The "RESERVA" tab is active. The main content area is divided into three sections:

- Busqueda del Hotel:** This section contains two radio buttons. The "Localidad:" radio button is selected, and its corresponding text box is active. The "Hotel:" radio button is unselected. There are two dropdown menus, both currently showing "Lista de hoteles". A "Consultar disponibilidad" button is located to the right of the dropdowns.
- Información del Hotel:** This section contains four text input fields: "Nombre:", "Dirección:", "Descripción:" (with a large text area), and "Teléfono:". There is also a "Categoría:" dropdown menu followed by the text "estrellas".
- Información del tipo de habitaciones del Hotel:** This section contains three text input fields: "Nombre:" (with a dropdown menu showing "Lista de tipos de habitaciones"), "Capacidad:", and "Importe por noche:" (with a currency symbol "€"). There is also a "Descripción:" text area.

Escribimos una localidad.

The screenshot shows a Windows application window titled "Reservas de Hotel: Usuario". The window has a menu bar with three tabs: "USUARIO", "RESERVA", and "LISTADO RESERVAS". Below the menu bar is a toolbar with several icons: a pencil for "Editar", a database cylinder with a plus sign for "Guardar", a database cylinder with an up arrow for "Modificar", a database cylinder with an X for "Borrar", a person with an X for "Darse de baja", a yellow "EXIT" button for "Salir", and a blue question mark for help. The main content area is divided into three sections:

- Busqueda del Hotel:** This section has two radio buttons. The first is "Localidad:" with a text box containing "Valencia" and a dropdown menu for "Hotel:" showing "Lista de hoteles". The second is "Hotel:" with a dropdown menu for "Lista de hoteles" and a red text label "Mueva el cursor hacia la lista de hoteles". A "Consultar disponibilidad" button is on the right.
- Información del Hotel:** This section has fields for "Nombre:", "Dirección:", "Descripción:", "Categoría:" (with a small text box), and "Teléfono:" (with a small text box). The "estrellas" label is next to the "Categoría:" field.
- Información del tipo de habitaciones del Hotel:** This section has fields for "Nombre:" (with a dropdown menu showing "Lista de tipos de habitaciones"), "Capacidad:" (with a small text box), "Importe por noche:" (with a small text box and a Euro symbol "€"), and "Descripción:".

Movemos el cursor hacia la lista de desplegable de la derecha para buscar hoteles en dicha localidad.

Reservas de Hotel: Usuario

Editar Guardar Modificar Borrar Darse de baja Salir

USUARIO RESERVA LISTADO RESERVAS

Busqueda del Hotel

Localidad: valencia Hotel: Lista de hoteles

Hotel: Lista de hoteles **Consultar disponibilidad**

Información del Hotel

Nombre: Categoría: estrellas

Dirección: Teléfono:

Descripción:

Información del tipo de habitaciones del Hotel

Nombre: Lista de tipos de habitaciones Capacidad: Importe por noche: €

Descripción:

Seleccionamos un hotel de la lista desplegable (si existe).

Reservas de Hotel: Usuario

Editar Guardar Modificar Borrar Darse de baja Salir

USUARIO RESERVA LISTADO RESERVAS

Busqueda del Hotel

Localidad: Valencia Hotel: Lista de hoteles
 Hotel: Lista de hoteles

Melia Plaza
Melia Valencia

Consultar disponibilidad

Información del Hotel

Nombre: Categoría: estrellas
Dirección: Teléfono:
Descripción:

Información del tipo de habitaciones del Hotel

Nombre: Lista de tipos de habitaciones Capacidad: Importe por noche: €
Descripción:

Una vez seleccionado el hotel nos aparecerá información del hotel: Nombre, Categoría, Dirección, Teléfono y Descripción.

Reservas de Hotel: Usuario

Editar Guardar Modificar Borrar Darse de baja Salir

USUARIO RESERVA LISTADO RESERVAS

Busqueda del Hotel

Localidad: valencia Hotel: Melia Plaza

Hotel: Lista de hoteles **Consultar disponibilidad**

Información del Hotel

Nombre: Melia Plaza Categoría: 4 estrellas

Dirección: Plaza del Ayuntamiento, Valencia (Valencia), 46002 Teléfono: 963520612

Descripción:
El Meliá Plaza, por su inmejorable ubicación en la Plaza de Ayuntamiento, es el hotel en Valencia más céntrico de Valencia, a escasos 200 metros del Ayuntamiento. La fachada original de su edificio -de principios del siglo XX- combina elegantemente con la funcionalidad y modernidad de sus instalaciones y decoración.

Información del tipo de habitaciones del Hotel

Nombre: Lista de tipos de habitaciones Capacidad: Importe por noche: €

Descripción:

A continuación se nos habilitará la lista desplegable de tipos de habitaciones del hotel seleccionado.

Reservas de Hotel: Usuario

Editar Guardar Modificar Borrar Darse de baja Salir

USUARIO RESERVA LISTADO RESERVAS

Busqueda del Hotel

Localidad: valencia Hotel: Melia Plaza

Hotel: Lista de hoteles **Consultar disponibilidad**

Información del Hotel

Nombre: Melia Plaza Categoría: 4 estrellas

Dirección: Plaza del Ayuntamiento, Valencia (Valencia), 46002 Teléfono: 963520612

Descripción:
El Meliá Plaza, por su inmejorable ubicación en la Plaza de Ayuntamiento, es el hotel en Valencia más céntrico de Valencia, a escasos 200 metros del Ayuntamiento. La fachada original de su edificio -de principios del siglo XX- combina elegantemente con la funcionalidad y modernidad de sus instalaciones y decoración.

Información del tipo de habitaciones del Hotel

Nombre: Lista de tipos de habitaciones Capacidad: Importe por noche: €

Descripción:
superior cama king
estándar 1 cama queen
estándar
junior suite

Una vez seleccionado un tipo de habitación y se nos mostrará información del tipo de habitación: Nombre, Capacidad, Importe por noche y Descripción.

Reservas de Hotel: Usuario

Editar Guardar Modificar Borrar Darse de baja Salir

USUARIO RESERVA LISTADO RESERVAS

Busqueda del Hotel

Localidad: valencia Hotel: Melia Plaza

Hotel: Lista de hoteles **Consultar disponibilidad**

Información del Hotel

Nombre: Melia Plaza Categoría: 4 estrellas

Dirección: Plaza del Ayuntamiento, Valencia (Valencia), 46002 Teléfono: 963520612

Descripción:
El Meliá Plaza, por su inmejorable ubicación en la Plaza de Ayuntamiento, es el hotel en Valencia más céntrico de Valencia, a escasos 200 metros del Ayuntamiento. La fachada original de su edificio -de principios del siglo XX- combina elegantemente con la funcionalidad y modernidad de sus instalaciones y decoración.

Información del tipo de habitaciones del Hotel

Nombre: estándar Capacidad: 2 Importe por noche: 60,000 €

Descripción: Tranquilas habitaciones dotadas de todo tipo de comodidades. Estas habitaciones, con suelo de parqué, disponen de camas Queen size o dos camas individuales a su elección (según disponibilidad, bajo petición a la llegada al hotel).

2. Busqueda por nombre de hotel

Pulsaremos en la opción Hotel y se nos habilitará la lista desplegable de la derecha.

Reservas de Hotel: Usuario

Editar Guardar Modificar Borrar Darse de baja Salir

USUARIO RESERVA LISTADO RESERVAS

Busqueda del Hotel

Localidad: Hotel:

Hotel:

Información del Hotel

Nombre: Categoría: estrellas

Dirección: Teléfono:

Descripción:

Información del tipo de habitaciones del Hotel

Nombre: Capacidad: Importe por noche: €

Descripción:

Seleccionamos un hotel de la lista desplegable.

Reservas de Hotel: Usuario

Editar Guardar Modificar Borrar Darse de baja Salir

USUARIO RESERVA LISTADO RESERVAS

Busqueda del Hotel

Localidad: Hotel: Lista de hoteles

Hotel: Lista de hoteles

Consultar disponibilidad

Información del Hotel

Nombre: Gran Melia Fenix
Melia Avenida America
Melia Plaza
Melia Valencia
Tryp Almussafes

Categoría: estrellas

Dirección: Teléfono:

Descripción:

Información del tipo de habitaciones del Hotel

Nombre: Lista de tipos de habitaciones Capacidad: Importe por noche: €

Descripción:

Una vez seleccionado el hotel nos aparecerá información del hotel: Nombre, Categoría, Dirección, Teléfono y Descripción.

Reservas de Hotel: Usuario

Editar Guardar Modificar Borrar Darse de baja Salir

USUARIO RESERVA LISTADO RESERVAS

Busqueda del Hotel

Localidad: Hotel:

Hotel:

Información del Hotel

Nombre: Categoría: estrellas

Dirección: Teléfono:

Descripción:

El Meliá Plaza, por su inmejorable ubicación en la Plaza de Ayuntamiento, es el hotel en Valencia más céntrico de Valencia, a escasos 200 metros del Ayuntamiento. La fachada original de su edificio -de principios del siglo XX- combina elegantemente con la funcionalidad y modernidad de sus instalaciones y decoración.

Información del tipo de habitaciones del Hotel

Nombre: Capacidad: Importe por noche: €

Descripción:

A continuación se nos habilitará la lista desplegable de tipos de habitaciones del hotel seleccionado.

Reservas de Hotel: Usuario

Editar Guardar Modificar Borrar Darse de baja Salir

USUARIO RESERVA LISTADO RESERVAS

Busqueda del Hotel

Localidad: Hotel: Lista de hoteles

Hotel: Melia Plaza **Consultar disponibilidad**

Información del Hotel

Nombre: Melia Plaza Categoría: 4 estrellas

Dirección: Plaza del Ayuntamiento, Valencia (Valencia), 46002 Teléfono: 963520612

Descripción:
El Meliá Plaza, por su inmejorable ubicación en la Plaza de Ayuntamiento, es el hotel en Valencia más céntrico de Valencia, a escasos 200 metros del Ayuntamiento. La fachada original de su edificio -de principios del siglo XX- combina elegantemente con la funcionalidad y modernidad de sus instalaciones y decoración.

Información del tipo de habitaciones del Hotel

Nombre: Lista de tipos de habitaciones Capacidad: Importe por noche: €

Descripción:
superior cama king
estándar 1 cama queen
estándar
junior suite

Una vez seleccionado un tipo de habitación y se nos mostrará información del tipo de habitación: Nombre, Capacidad, Importe por noche y Descripción.

Realizar una Reserva

Una vez seleccionado un hotel se nos habilitará el botón



Si pulsamos dicho botón se no mostrará el formulario “Información de la Reserva”.

Formulario Información de la reserva

En el formulario “Información de la reserva” tenemos información del hotel y de la reserva.

Reservas de Hotel: Información de la reserva

Imprimir Guardar reserva Modificar reserva Borrar reserva Cerrar

RESERVA

Cliente:
Nombre tomador:
Hotel:
Dirección:
Habitación: **Hab libres:**
Habitaciones reservadas: **Capacidad:** **Ocupación:**
Fecha de entrada:
Fecha de salida:
Importe hab/noche: € **Precio por noche TOTAL:** €

Datos obligatorios

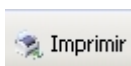
Habitación:
Habitaciones reservadas: **Ocupación:**

Disponibilidad del hotel

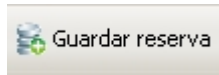
Fecha de entrada:
Fecha de salida:

CONSULTA

Opciones:



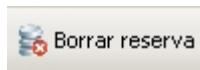
Si pulsamos el botón “Imprimir” nos saldrá el formulario de selección de impresora y características de dicha impresora.



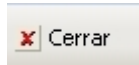
Si pulsamos el botón “Guardar reserva” se almacenarán los datos de la nueva reserva en el sistema



Si pulsamos el botón “Modificar reserva” se almacenarán los datos modificados de la reserva en el sistema. la



Si pulsamos el botón “Borrar reserva” se borrarán los datos de la reserva en el sistema.



Si pulsamos el botón “Cerrar” se cerrará el formulario de Información de la reserva

Pasos para realizar una reserva:

1. Primero tenemos que seleccionar un hotel (por localidad o por nombre). Se habilitará el botón



Reservas de Hotel: Usuario

Editar Guardar Modificar Borrar Darse de baja Salir

USUARIO RESERVA LISTADO RESERVAS

Busqueda del Hotel

Localidad: Hotel:

Hotel: **Consultar disponibilidad**

Información del Hotel

Nombre: Categoría: estrellas

Dirección: Teléfono:

Descripción:
El Meliá Plaza, por su inmejorable ubicación en la Plaza de Ayuntamiento, es el hotel en Valencia más céntrico de Valencia, a escasos 200 metros del Ayuntamiento. La fachada original de su edificio -de principios del siglo XX- combina elegantemente con la funcionalidad y modernidad de sus instalaciones y decoración.

Información del tipo de habitaciones del Hotel

Nombre: Capacidad: Importe por noche: €

Descripción:

2. Pulsaremos el botón

**Consultar
disponibilidad**

para acceder al formulario Información de la reserva

Reservas de Hotel: Información de la reserva

Imprimir Guardar reserva Modificar reserva Borrar reserva Cerrar

RESERVA

Cliente: Miguel Monzo Pastor

Nombre tomador: Miguel Monzo Pastor

Hotel: Melia Plaza

Dirección: Plaza del Ayuntamiento, Valencia (Valencia), 46002

Habitación: Hab libres:

Habitaciones reservadas: 1 Capacidad: Ocupación: 1

Fecha de entrada: lunes, 05 de diciembre de 2011

Fecha de salida: lunes, 05 de diciembre de 2011

Importe hab/noche: 0 € Precio por noche TOTAL: 0 €

Datos obligatorios

Habitación: Tipos de habitaciones disponibles

Habitaciones reservadas: 1 Ocupación: 1

Disponibilidad del hotel

Fecha de entrada: lunes, 05 de diciembre de 2011

Fecha de salida: lunes, 05 de diciembre de 2011

CONSULTA

3. Seleccionaremos Fecha de entrada y Fecha de salida correctas

4. Pulsamos el botón **CONSULTA**.

Si las fechas de entrada y salida son incorrectas se nos mostrarán mensajes de error.

Si las fechas de entrada y salida son correctas se nos habilitará la lista desplegable de los tipos de habitación disponibles del hotel seleccionado para las fechas introducidas.

Reservas de Hotel: Información de la reserva

Imprimir Guardar reserva Modificar reserva Borrar reserva Cerrar

RESERVA

Cliente:
Nombre tomador:
Hotel:
Dirección:
Habitación: **Hab libres:**
Habitaciones reservadas: **Capacidad:** **Ocupación:**
Fecha de entrada:
Fecha de salida:
Importe hab/noche: € **Precio por noche TOTAL:** €

Datos obligatorios

Habitación:
Habitaciones reservadas: **Ocupación:**

Disponibilidad del hotel

Fecha de entrada:
Fecha de salida:

CONSULTA

5. Seleccionamos una habitación de la lista desplegable.

Reservas de Hotel: Información de la reserva

Imprimir Guardar reserva Modificar reserva Borrar reserva Cerrar

RESERVA

Cliente: Miguel Monzo Pastor
Nombre tomador: Miguel Monzo Pastor
Hotel: Melia Plaza
Dirección: Plaza del Ayuntamiento, Valencia (Valencia), 46002

Habitación: **Hab libres:**
Habitaciones reservadas: **Capacidad:** **Ocupación:**

Fecha de entrada: jueves, 22 de diciembre de 2011
Fecha de salida: sábado, 31 de diciembre de 2011

Importe hab/noche: € **Precio por noche TOTAL:** €

Datos obligatorios

Habitación: Tipos de habitaciones disponibles
Habitaciones: superior cama king
 estándar 1 cama queen
 estándar
 junior suite

Disponibilidad del hotel

Fecha de entrada: jueves , 22 de diciembre de 2011
Fecha de salida: sábado , 31 de diciembre de 2011

CONSULTA

Se nos habilitará las opciones "Habitaciones reservadas" y "Ocupación" de la agrupación Datos obligatorios.

6. Se nos mostrará información del tipo de habitación seleccionado útil para realizar la reserva

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

(Habitaciones libres, Importe de la habitación por noche, Capacidad etc..) . Escogeremos el número de habitaciones que queremos reservar (Habitaciones reservadas) y la ocupación de cada habitación (Ocupación). El campo "Precio por noche TOTAL" se actualizará automáticamente.

Reservas de Hotel: Información de la reserva

Imprimir Guardar reserva Modificar reserva Borrar reserva Cerrar

RESERVA

Cliente: Miguel Monzo Pastor

Nombre tomador: Miguel Monzo Pastor

Hotel: Melía Plaza

Dirección: Plaza del Ayuntamiento, Valencia [Valencia], 46002

Habitación: estándar 1 cama queen Hab libres: 5

Habitaciones reservadas: 3 Capacidad: 2 Ocupación: 2

Fecha de entrada: jueves, 22 de diciembre de 2011

Fecha de salida: sábado, 31 de diciembre de 2011

Importe hab/noche: 90 € Precio por noche TOTAL: 2700 €

Datos obligatorios

Habitación: estándar 1 cama queen


Habitaciones reservadas: 3 Ocupación: 2

Disponibilidad del hotel


Fecha de entrada: jueves, 22 de diciembre de 2011

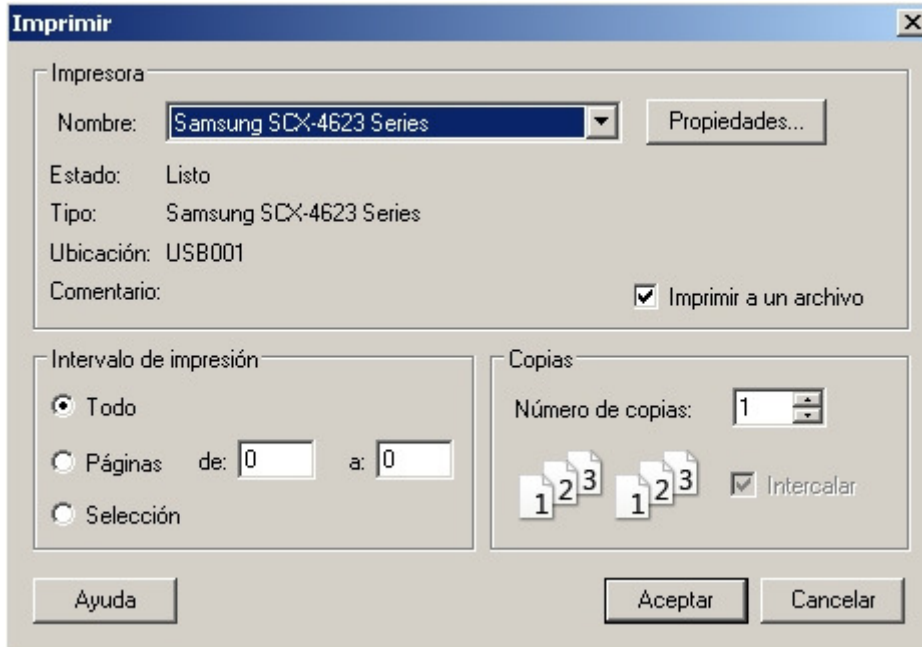
Fecha de salida: sábado, 31 de diciembre de 2011

CONSULTA

7. Pulsaremos el botón  para almacenar la reserva en el sistema.

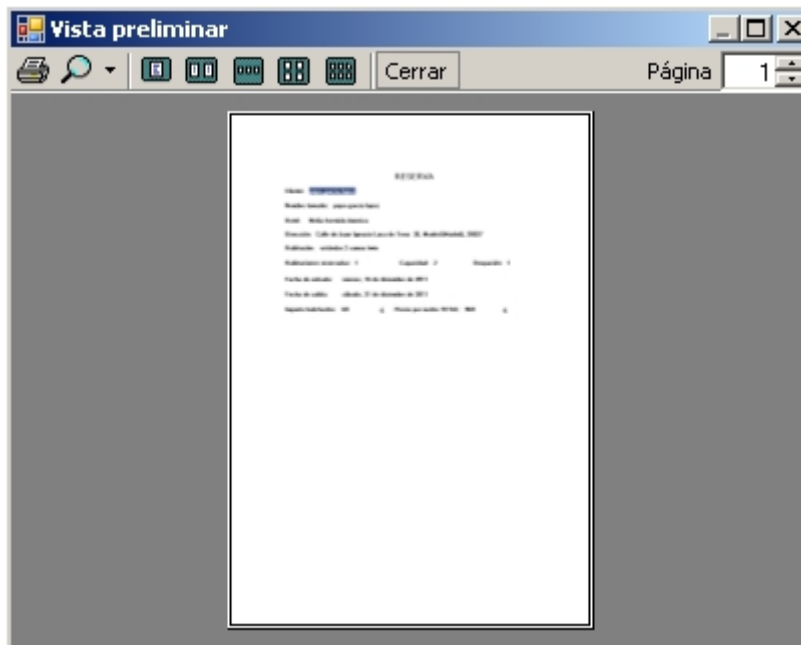
2.4.2.3 Imprimir una reserva

8. Pulsaremos el botón  para imprimir los datos de la reserva. Nos saldrá el formulario de selección de impresora y características de dicha impresora.



Seleccionaremos la impresora y otras características y pulsaremos el botón Aceptar.

A continuación se nos mostrará la Vista Preliminar.

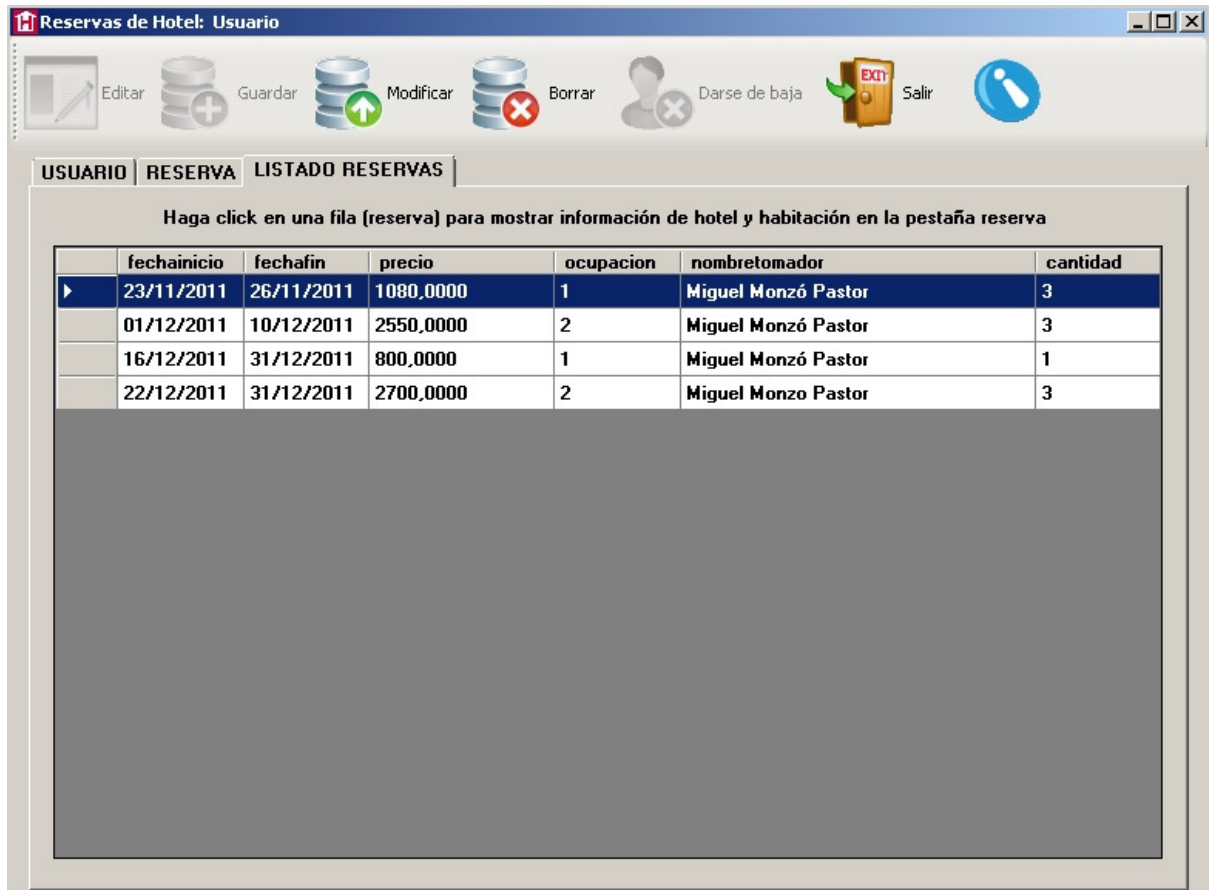


Si estamos de acuerdo y todo es correcto pulsaremos en el botón Imprimir para imprimir la reserva realizada.

2.4.2.4 Consultar información del Hotel y de la habitación de una reserva

Pestaña LISTADO DE RESERVAS

En la pestaña de “LISTADO DE RESERVAS” podremos acceder a la información básica de las reservas que hemos realizado a lo largo del tiempo (historial) y que actualmente tenemos almacenadas en el sistema.



Podremos seleccionar una reserva simplemente pulsando sobre la fila correspondiente (se nos mostrará en azul)

Opciones:



Pulsar sobre el botón "Modificar" para mostrar el formulario Información de Reserva y poder modificar una reserva en vigor. Podremos modificar la reserva si la fecha de inicio de la reserva es posterior a la fecha actual.



Pulsar sobre el botón "Borrar" para mostrar el formulario Información de Reserva y poder borrar una reserva en vigor. Podremos borrar (anular) la reserva si la fecha de inicio de la reserva es posterior a la fecha actual



Pulsar sobre el botón "Salir" para salir de la aplicación. Se pedirá confirmación.



Pulsar sobre el botón "Información" para ver información sobre la aplicación

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

Para consultar información del Hotel y de la habitación de una reserva pulsaremos sobre la fila deseada correspondiente a una reserva y se nos mostrará información en la pestaña "RESERVA"

Reservas de Hotel: Usuario

Editar Guardar Modificar Borrar Darse de baja Salir

USUARIO RESERVA LISTADO RESERVAS

Haga click en una fila (reserva) para mostrar información de hotel y habitación en la pestaña reserva

| fechainicio | fechafin | precio | ocupacion | nombretomador | cantidad |
|-------------|------------|-----------|-----------|---------------------|----------|
| 23/11/2011 | 26/11/2011 | 1080,0000 | 1 | Miguel Monzó Pastor | 3 |
| 01/12/2011 | 10/12/2011 | 2550,0000 | 2 | Miguel Monzó Pastor | 3 |
| 16/12/2011 | 31/12/2011 | 800,0000 | 1 | Miguel Monzó Pastor | 1 |
| 22/12/2011 | 31/12/2011 | 2700,0000 | 2 | Miguel Monzo Pastor | 3 |

Reservas de Hotel: Usuario

Editar Guardar Modificar Borrar Darse de baja Salir

USUARIO RESERVA LISTADO RESERVAS

Busqueda del Hotel

Localidad: Hotel:

Hotel:

Información del Hotel

Nombre: Categoría: estrellas

Dirección: Teléfono:

Descripción:
El Hotel Meliá Avenida América se encuentra ubicado entre el Aeropuerto Internacional Madrid Barajas y el Paseo de la Castellana, en una zona de gran actividad empresarial de Madrid.

Información del tipo de habitaciones del Hotel

Nombre: Capacidad: Importe por noche: €

Descripción:
Acogedora y confortable, esta habitación con cama queen size combina la elegancia con el más moderno diseño ofreciendo a nuestros clientes todos los detalles necesarios para una agradable y relajada estancia.

2.4.2.5 Modificar una reserva

Para modificar una reserva se tiene que cumplir que la fecha de inicio de la reserva sea posterior a la fecha actual. Si esta condición no se cumple se nos mostrará un mensaje de error indicando que no se puede modificar la reserva.

Pasos para modificar una reserva:

1. Seleccionamos en la pestaña "LISTADO DE RESERVAS" una reserva que cumpla la condición para poder modificarla para ello hacemos clic en la fila correspondiente.

Reservas de Hotel: Usuario

Editar Guardar Modificar Borrar Darse de baja Salir

USUARIO RESERVA LISTADO RESERVAS

Haga click en una fila (reserva) para mostrar información de hotel y habitación en la pestaña reserva

| fechainicio | fechafin | precio | ocupacion | nombretomador | cantidad |
|-------------|------------|-----------|-----------|---------------------|----------|
| 23/11/2011 | 26/11/2011 | 1080,0000 | 1 | Miguel Monzó Pastor | 3 |
| 01/12/2011 | 10/12/2011 | 2550,0000 | 2 | Miguel Monzó Pastor | 3 |
| 16/12/2011 | 31/12/2011 | 800,0000 | 1 | Miguel Monzó Pastor | 1 |
| 22/12/2011 | 31/12/2011 | 2700,0000 | 2 | Miguel Monzo Pastor | 3 |

2. Pulsamos el botón



para mostrar el formulario de "Información de la reserva"

Reservas de Hotel: Información de la reserva

Imprimir Guardar reserva Modificar reserva Borrar reserva Cerrar

RESERVA

Cliente:

Nombre tomador:

Hotel:

Dirección:

Habitación: Hab libres:

Habitaciones reservadas: Capacidad: Ocupación:

Fecha de entrada:

Fecha de salida:

Importe hab/noche: € Precio por noche TOTAL: €

Datos obligatorios

Habitación:

Habitaciones reservadas: Ocupación:

Disponibilidad del hotel

Fecha de entrada:

Fecha de salida:

CONSULTA

3. En el formulario "Información de la reserva" nos aparecerá información relativa a la reserva que queremos modificar. Procederemos a realizar una nueva consulta ver la disponibilidad del hotel en las fechas indicadas entre la fecha de entrada y la fecha de salida.

Para ello seleccionamos la fecha de entrada y la fecha de salida (pueden ser las mismas fechas que la reserva original o no) y pulsaremos el botón **CONSULTA**. Se habilitará la lista desplegable del tipo de habitaciones del hotel.

Reservas de Hotel: Información de la reserva

Imprimir Guardar reserva Modificar reserva Borrar reserva Cerrar

RESERVA

Cliente: Miguel Monzo Pastor
Nombre tomador: Miguel Monzo Pastor
Hotel: Gran Melia Fenix
Dirección: Calle de Hermosilla 2, Madrid (Madrid), 28001
Habitación: colon **Hab libres:**
Habitaciones reservadas: 3 **Capacidad:** 1 **Ocupación:** 1
Fecha de entrada: viernes, 16 de diciembre de 2011
Fecha de salida: sábado, 31 de diciembre de 2011
Importe hab/noche: 50 € **Precio por noche TOTAL:** 2400 €

Datos obligatorios

Habitación: Tipos de habitaciones disponibles
Habitaciones reservadas: 0 **Ocupación:** 0

Disponibilidad del hotel

Fecha de entrada: viernes, 16 de diciembre de 2011
Fecha de salida: sábado, 31 de diciembre de 2011

CONSULTA

4. Seleccionaremos un tipo de habitación de la lista desplegable (puede ser la misma que la reserva original o no) y se nos habilitará las opciones de "Habitaciones reservadas" y "Ocupación". A continuación podremos modificar dichas opciones dependiendo de la disponibilidad del hotel (Habitaciones libres y Capacidad).

Reservas de Hotel: Información de la reserva

Imprimir Guardar reserva Modificar reserva Borrar reserva Cerrar

RESERVA

Cliente: Miguel Monzo Pastor

Nombre tomador: Miguel Monzo Pastor

Hotel: Gran Melia Fenix

Dirección: Calle de Hermosilla 2, Madrid (Madrid), 28001

Habitación: colon Hab libres: 4

Habitaciones reservadas: 3 Capacidad: 1 Ocupación: 1

Fecha de entrada: viernes, 16 de diciembre de 2011

Fecha de salida: sábado, 31 de diciembre de 2011

Importe hab/noche: 50 € Precio por noche TOTAL: 2400 €

Datos obligatorios

Habitación: colon


Habitaciones reservadas: 3 Ocupación: 1


Disponibilidad del hotel

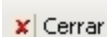
Fecha de entrada: viernes, 16 de diciembre de 2011

Fecha de salida: sábado, 31 de diciembre de 2011

CONSULTA

5. Por último pulsaremos el botón  para almacenar los cambios realizados.

6. También tendremos la posibilidad de imprimir la reserva pulsando en el botón  o salir sin

 Cerrar

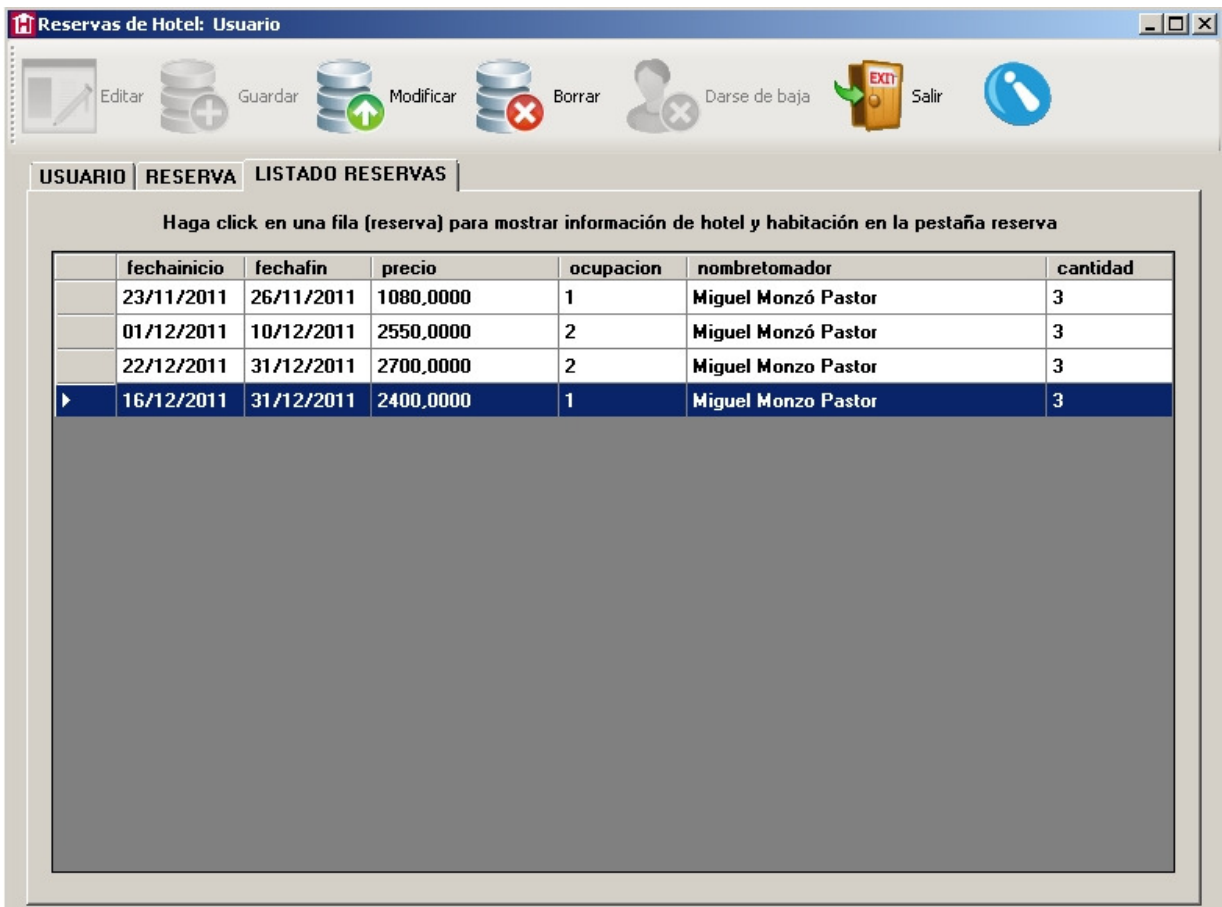
guardar los cambios pulsando el botón


2.4.2.6 Borrar (cancelar) una reserva

Para borrar (cancelar) una reserva se tiene que cumplir que la fecha de inicio de la reserva sea posterior a la fecha actual. Si esta condición no se cumple se nos mostrará un mensaje de error indicando que no se puede borrar (cancelar) la reserva.

Pasos para borrar (cancelar) una reserva:


1. Seleccionamos en la pestaña “LISTADO DE RESERVAS” una reserva que cumpla la condición para poder borrar (cancelar) para ello hacemos clic en la fila correspondiente.

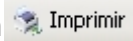


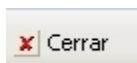
2. Pulsamos el botón  para mostrar el formulario de “Información de la reserva” y



poder realizar el proceso de borrar (cancelar) la reserva.

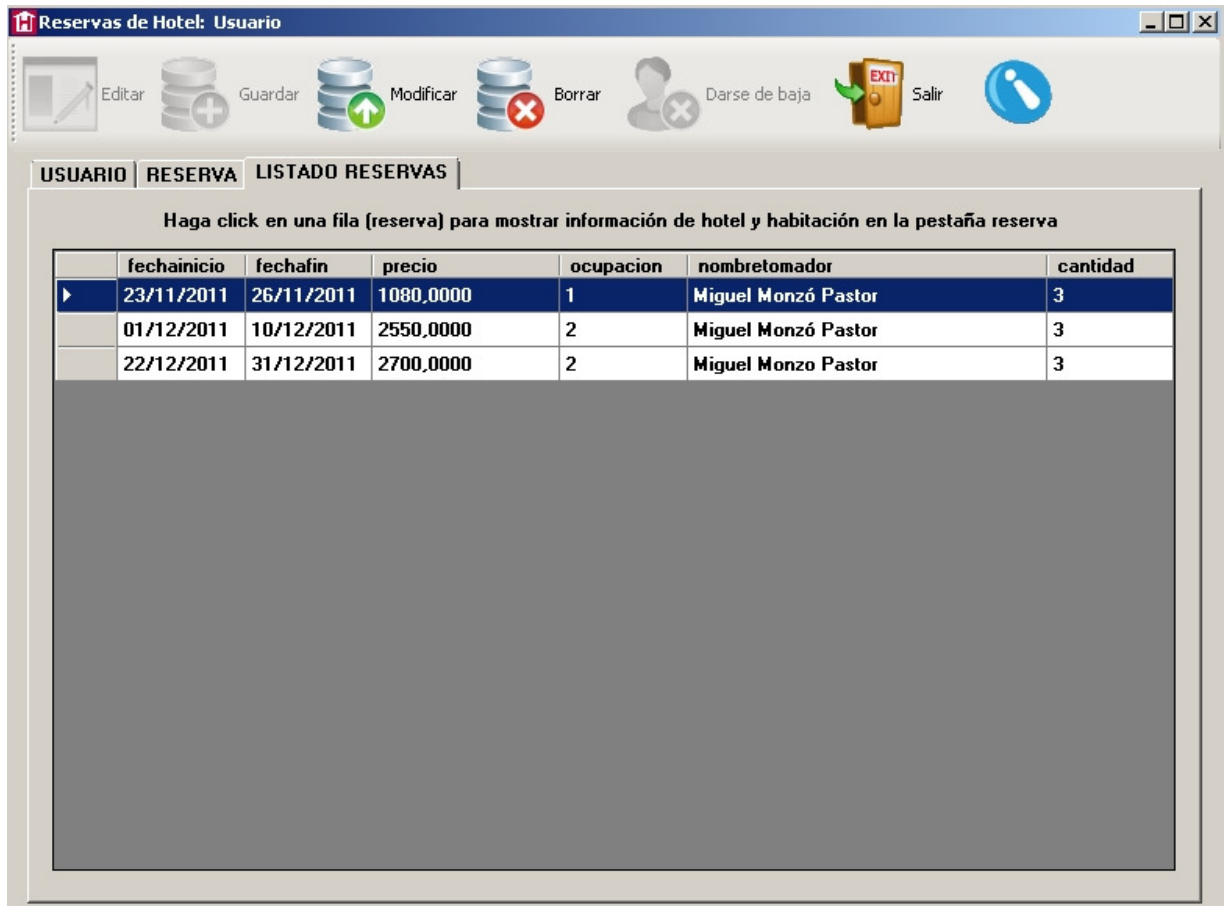
4. Como podremos observar las opciones de “Datos obligatorios” y “Disponibilidad del hotel” están deshabilitadas. Procederemos entonces a confirmar el borrado (cancelación) de la reserva pulsando el botón 

5. También tendremos la posibilidad de imprimir la reserva pulsando en el botón  o salir sin



Borrar (cancelar) pulsando el botón

6. En la pestaña de "LISTADO DE RESERVAS" ya no aparecerá la línea correspondiente a la reserva borrada (cancelada)



2.5 Codificación

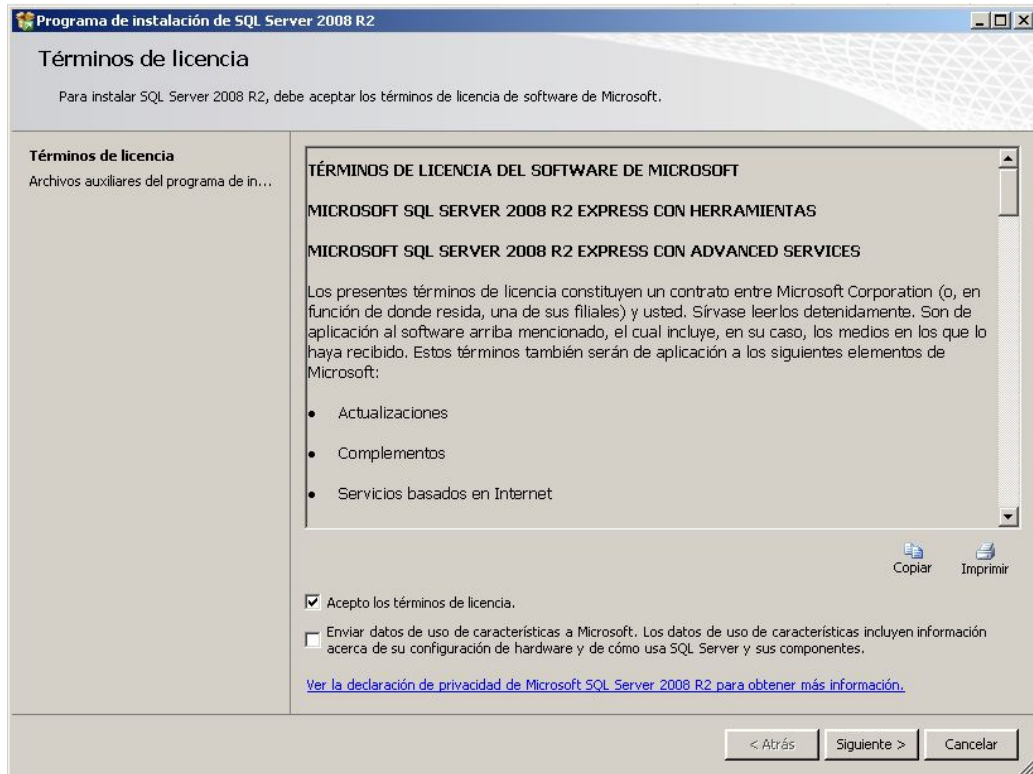
2.5.1 Instalación y configuración del SGBD SQL Server 2008 R2

1 - INSTALACION DE SQL SERVER 2008 R2 EXPRESS

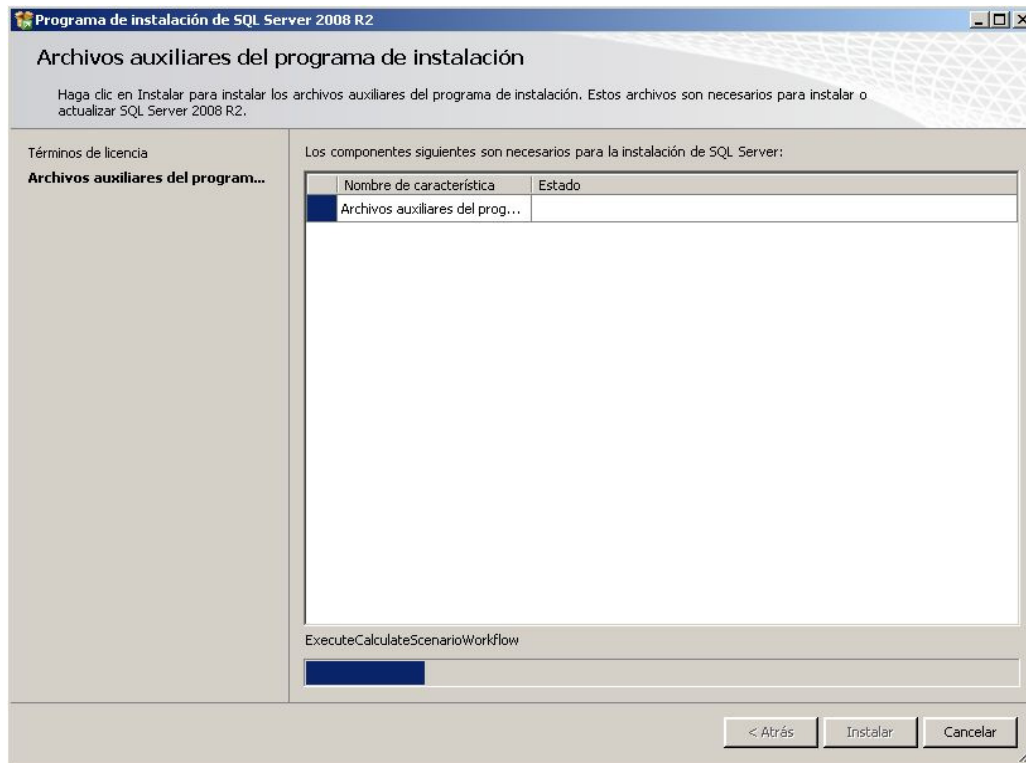
Paso 1: Iniciamos el proceso de instalación seleccionando la opción *“Nueva instalación o agregar características a una instalación existente”*.



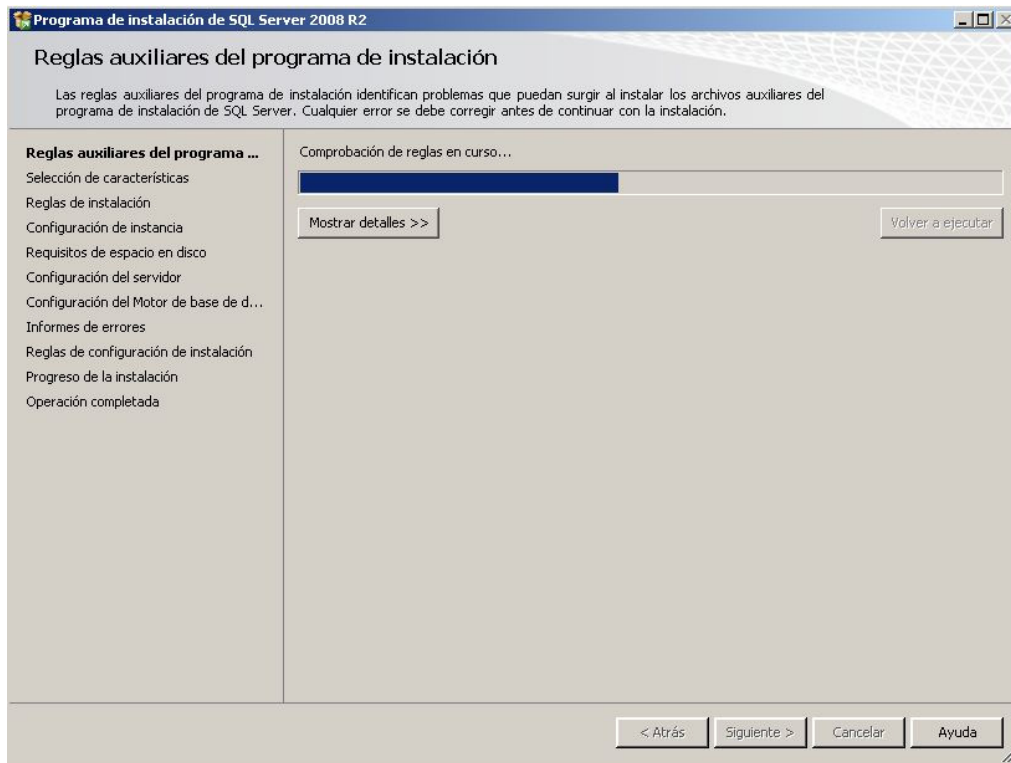
Paso 2: Aceptamos los términos de licencia y pulsamos Siguiente.



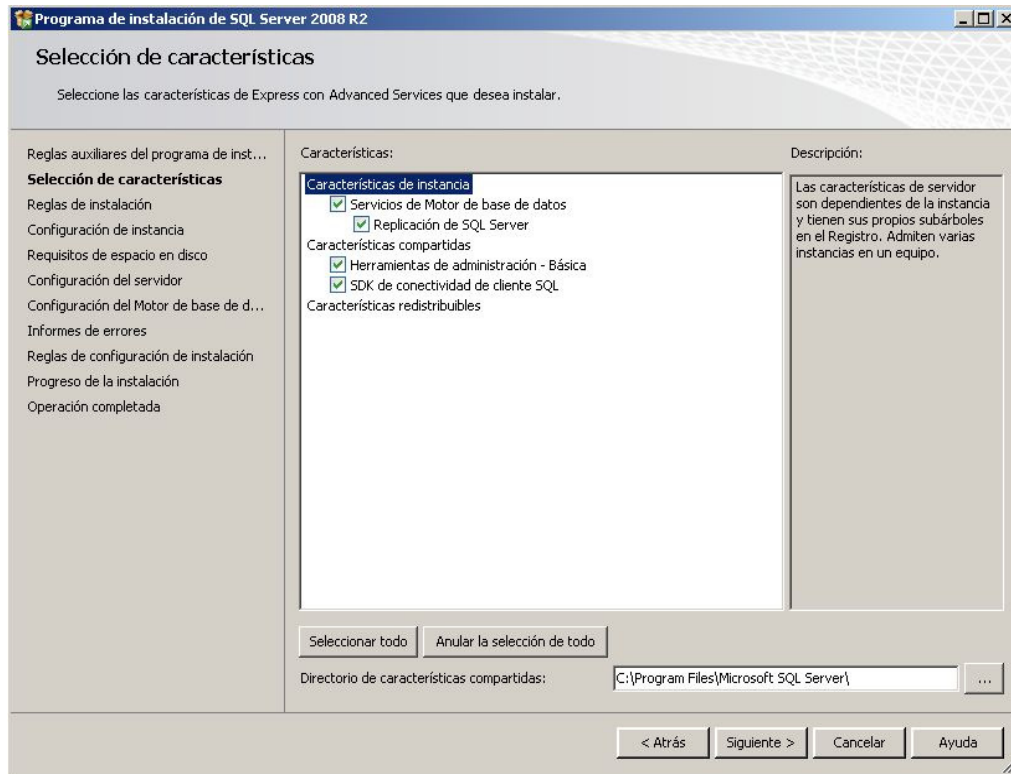
Paso 3: Se instalan automáticamente los archivos auxiliares del programa de instalación.



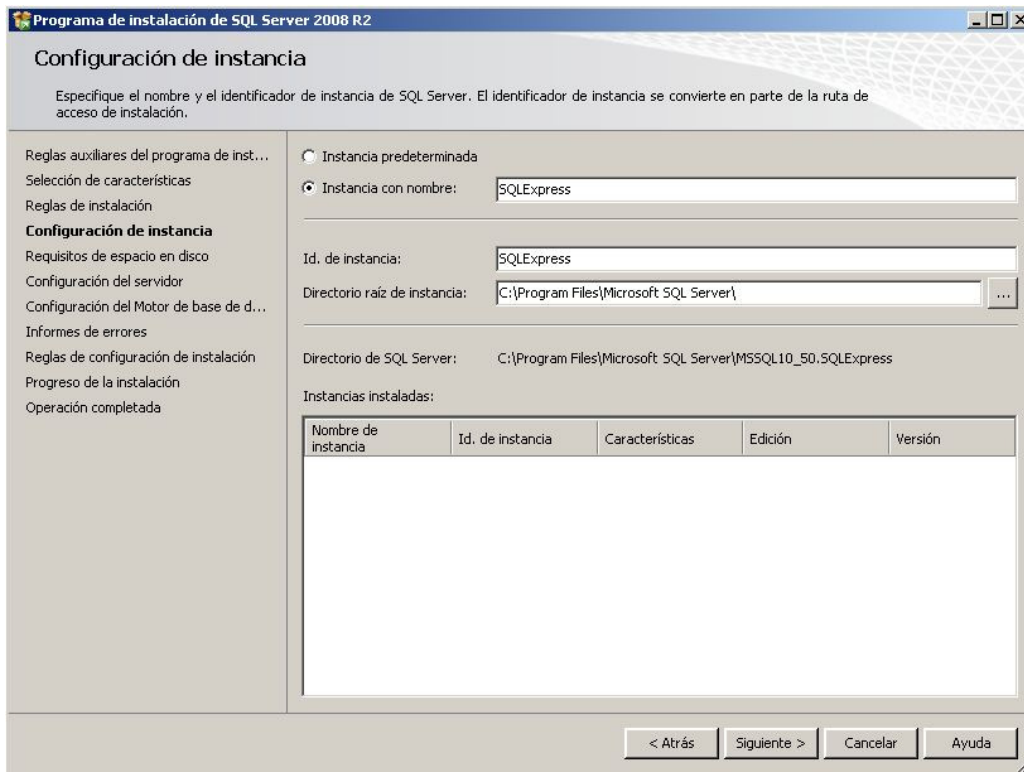
Paso 4: Se comprueban, mediante reglas, que no hay ningún problema al instalar los archivos auxiliares de instalación.



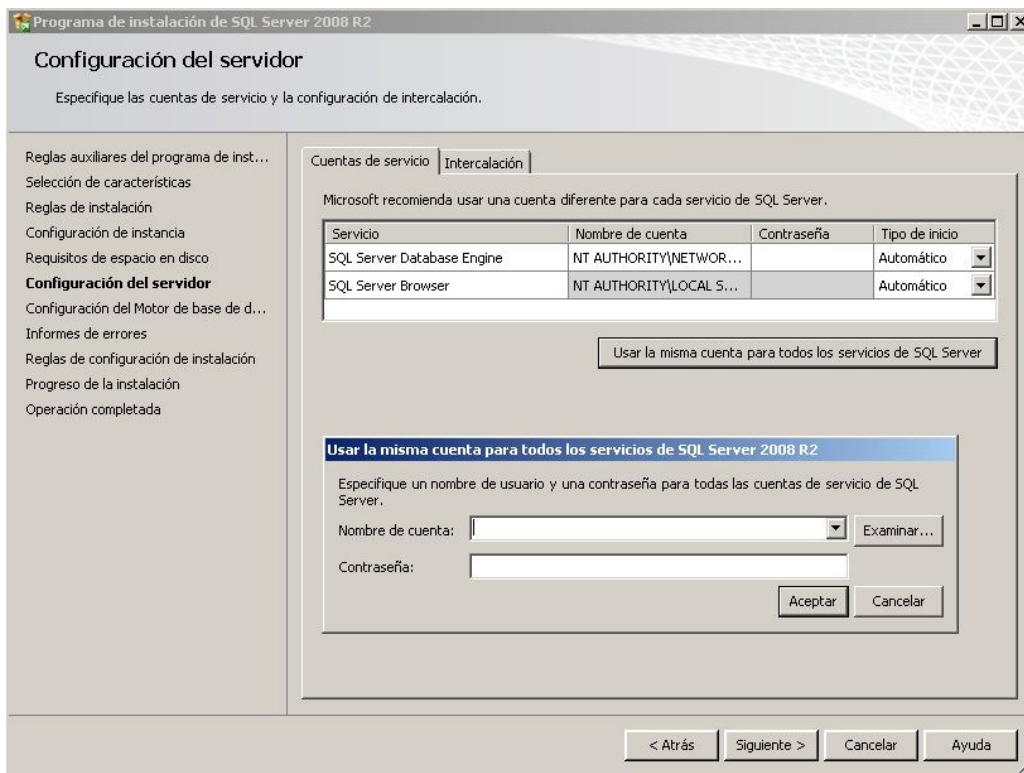
Paso 5: Selección de características. Seleccionamos las características que queremos que se instalen (seleccionamos todas las características) y pulsamos Siguiete.



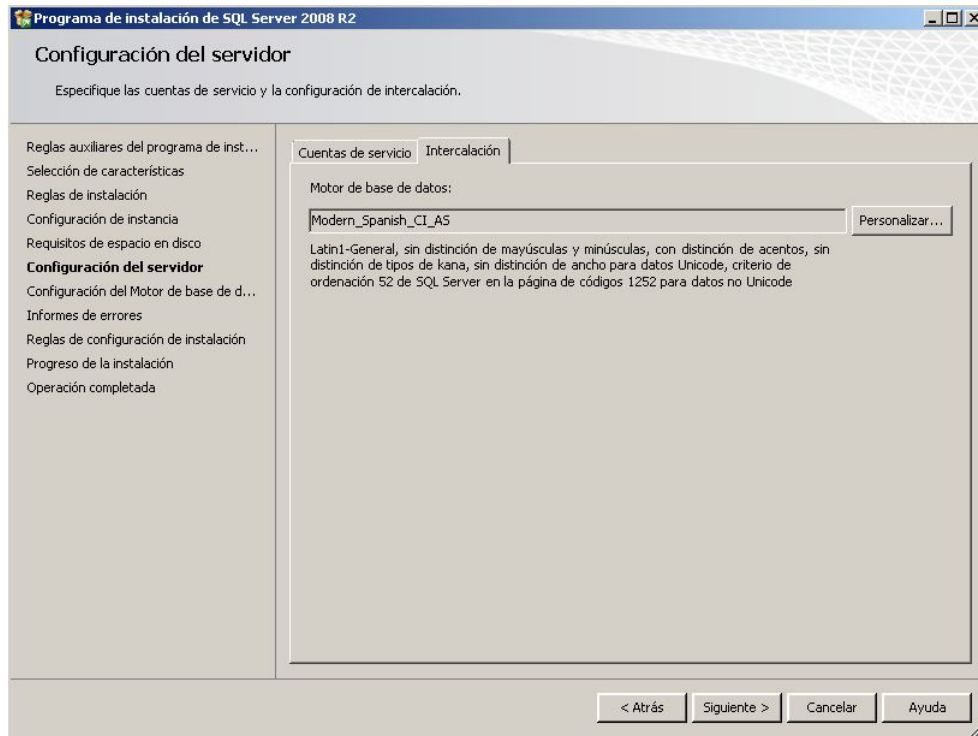
Paso 6: Configuración de Instancia. Seleccionamos la opción Instancia con nombre (SQLEXPRESS), el resto de opciones aparecen por defecto y pulsamos Siguiente.



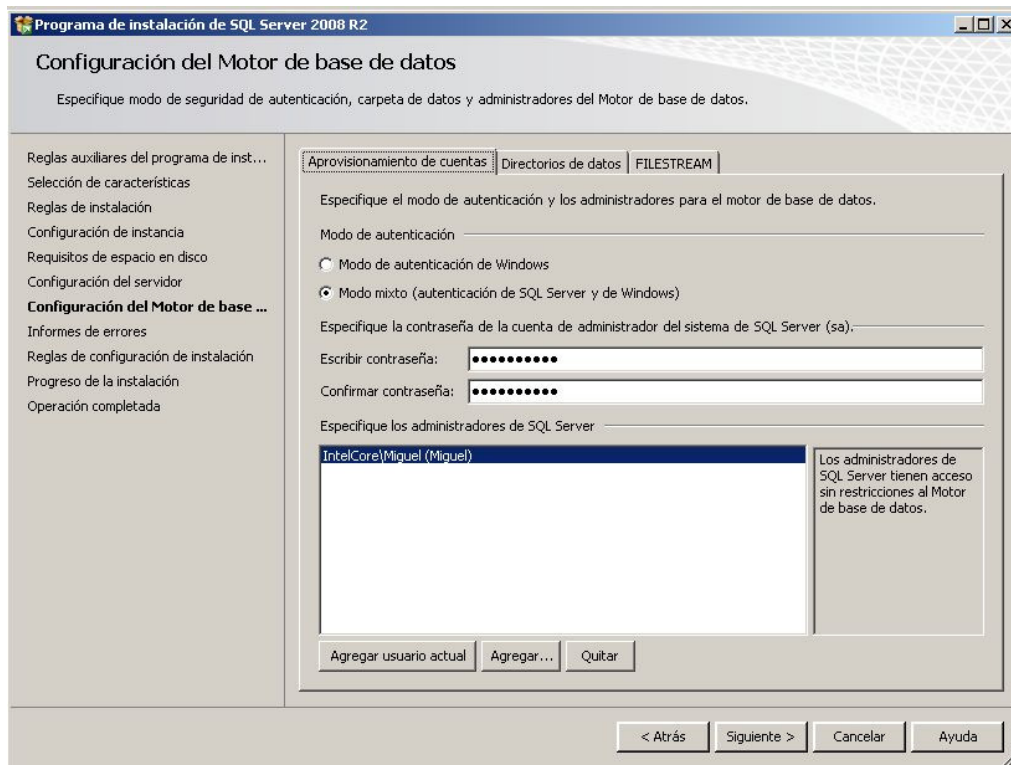
Paso 7: Configuración del servidor. En la pestaña cuentas de usuario pulsamos la opción “Usar la misma cuenta para todos los servicios de SQL Server”. A continuación seleccionamos una cuenta de administrador dada de alta en el sistema (en el ejemplo Miguel), pulsamos el botón examinar y seleccionamos la cuenta. Seguidamente introducimos una contraseña y pulsamos Aceptar.



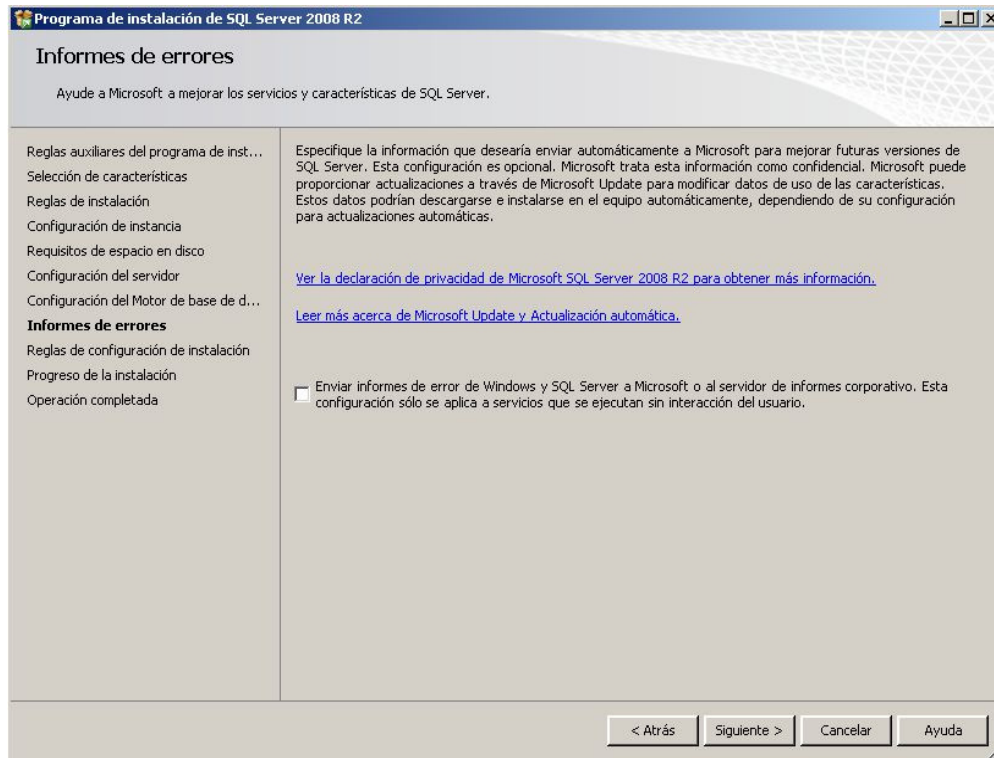
Paso 8: En la pestaña *Intercalación*, dejamos la opción por defecto *Modern_Spanish_CI_AS* y pulsamos *Siguiente*.



Paso 9: Configuración del motor de base de datos. Seleccionamos la pestaña *Aprovisionamiento de cuentas* y en la opción Modo de autenticación seleccionamos *Modo Mixto (autenticación de SQL Server y de Windows)*. A continuación escribimos la contraseña de la cuenta de administrador del sistema SQL Server. El resto de opciones no las modificamos. Pulsamos *Siguiete*.

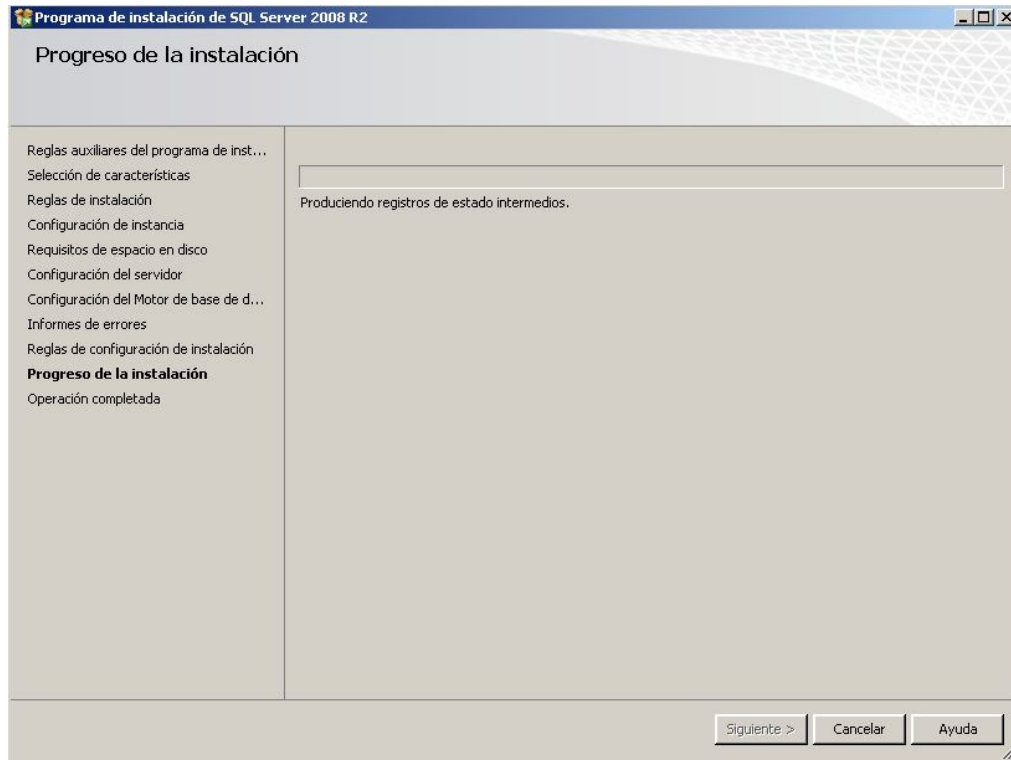


Paso 10: Informe de errores. Esta opción es optativa y se deja a elección del usuario. Pulsamos Siguiete.

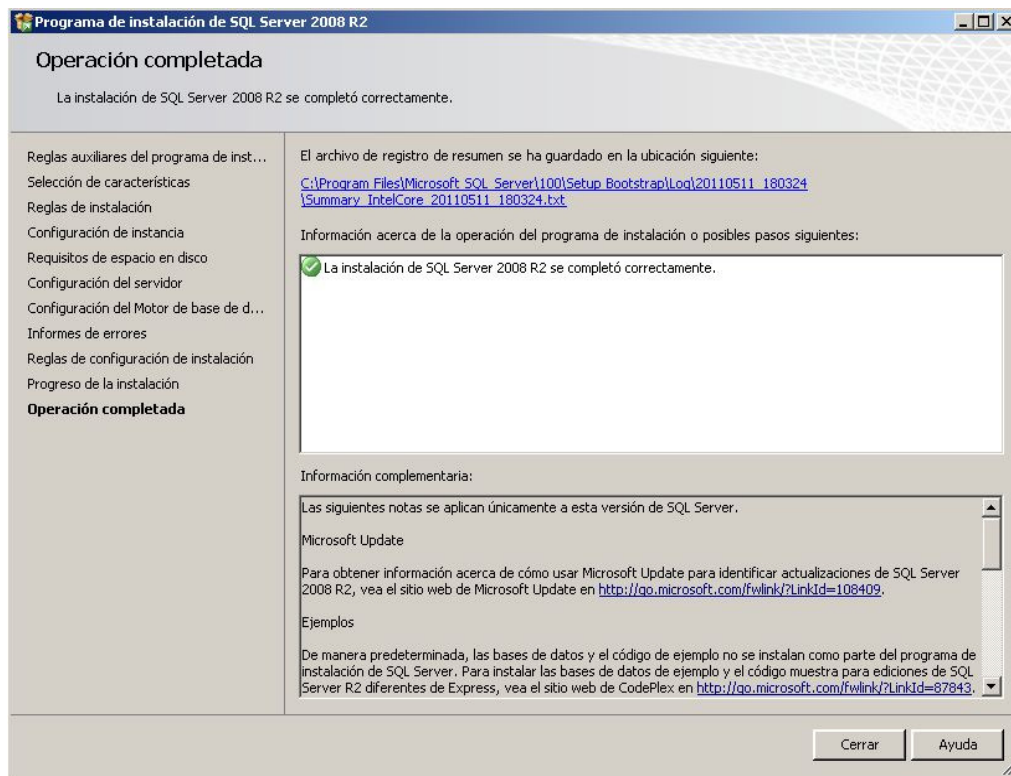


Paso 11: Progreso de instalación. Comienza el proceso de instalación y configuración de todos los archivos.





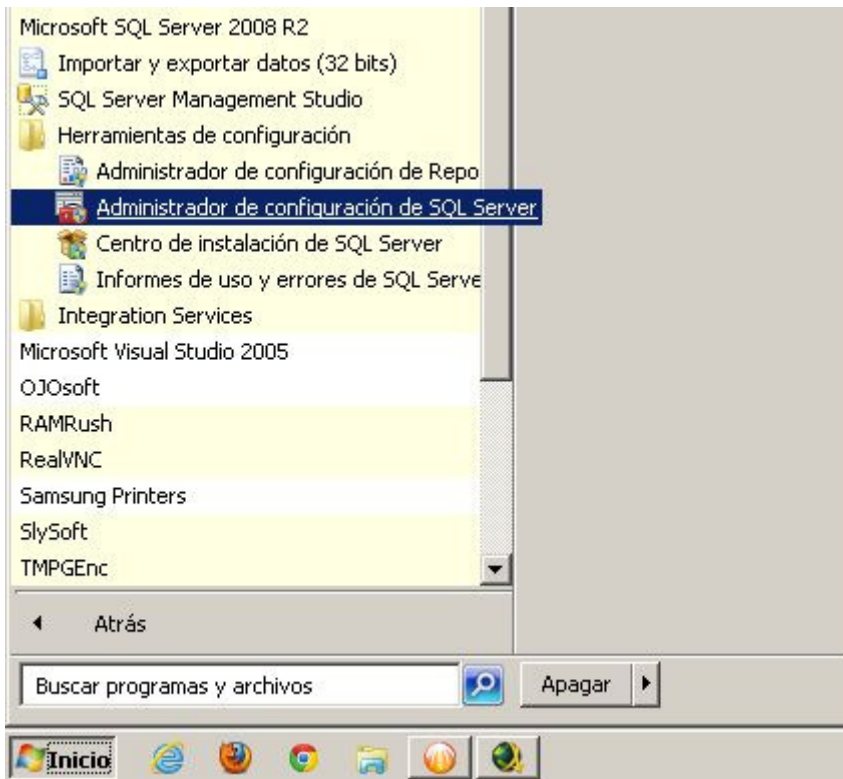
Paso 12: Operación completada. Si no ha ocurrido ningún error durante el proceso de instalación y configuración de los archivos se nos muestra un mensaje de finalización de la instalación completada correctamente. Pulsamos Cerrar.



2 - CONFIGURACIÓN PARA ACCESO REMOTO MEDIANTE TCP/IP

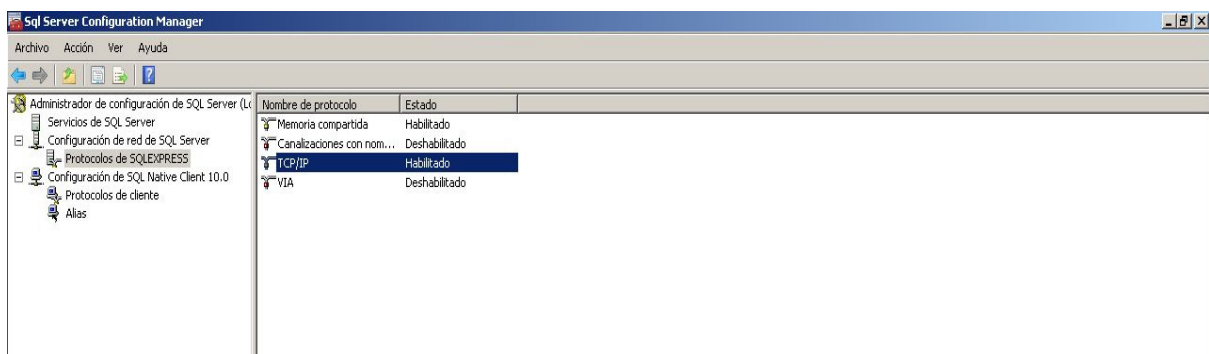


Paso 1: Seleccionamos en el menú de inicio, *Microsoft SQL Server 2008 R2 -> Herramientas de configuración -> Administrador de configuración de SQL Server*



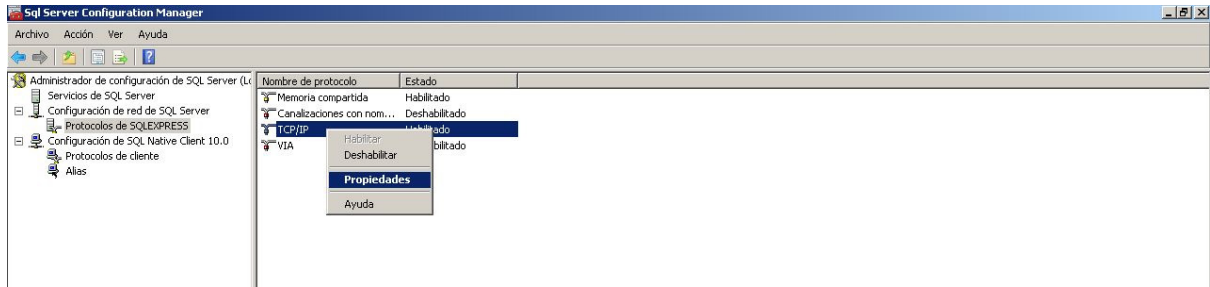
Paso 2: Vamos a la opción *Configuración de red de SQL Server -> Protocolos de SQLExpress ->TCP/IP*, pulsar botón derecho y seleccionar la opción *Habilitar*

Vamos a la opción *Configuración de red de SQL Server -> Protocolos de SQLExpress ->Memoria compartida*, pulsar botón derecho y seleccionar la opción *Habilitar*

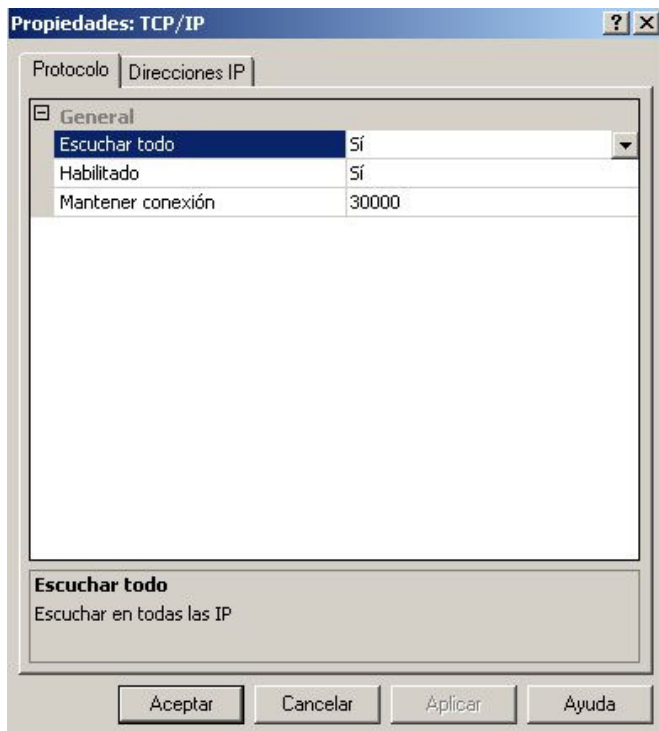


Paso 3: Vamos a la opción *Configuración de red de SQL Server -> Protocolos de SQLExpress ->TCP/IP*, pulsamos botón derecho sobre *TCP/IP* y seleccionamos la opción *Propiedades*

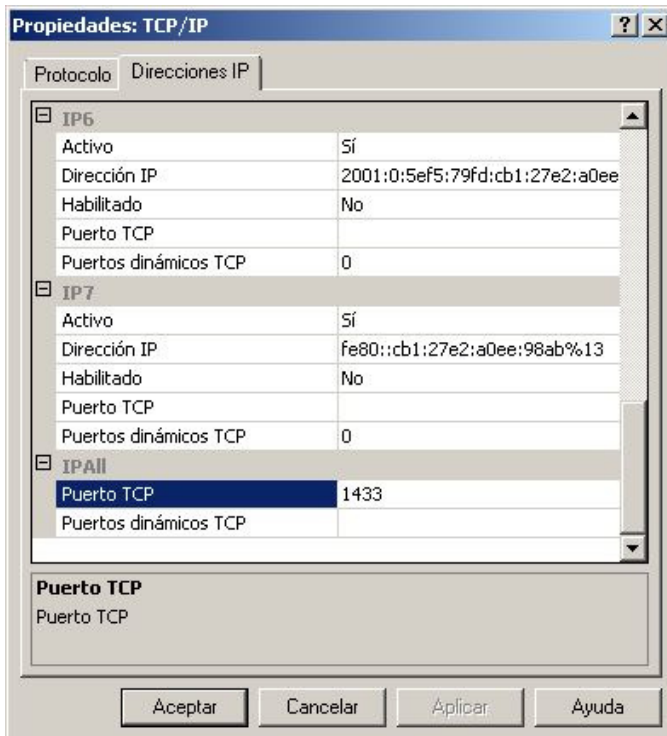
Aplicación en Visual Basic .NET: Reservas de Hotel 1.0



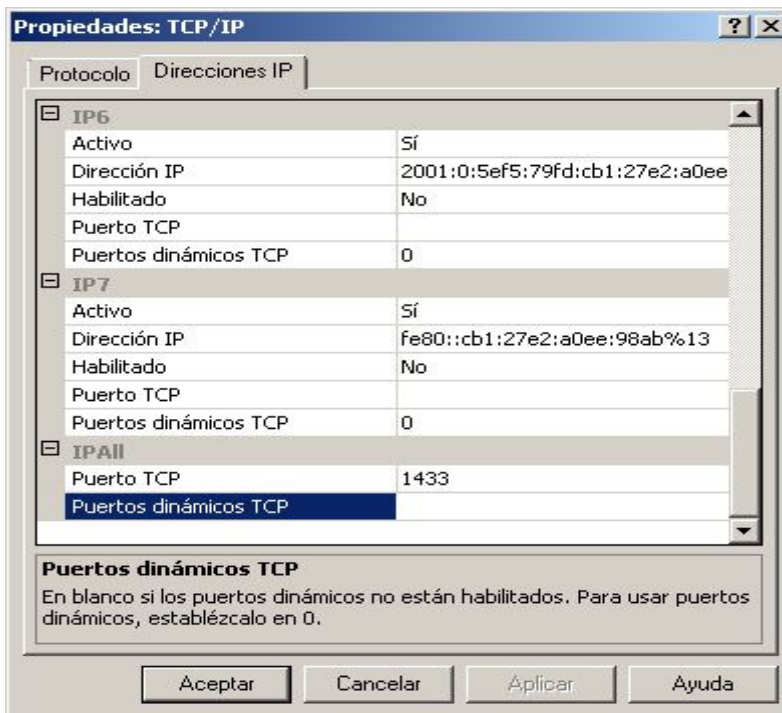
Paso 4: En la pestaña *Protocolo*, seleccionamos la opción *Escuchar todo* -> *Si*



Paso 5: En la pestaña *Direcciones IP*. En la opción *IPAll* -> *Puerto TCP* colocar 1433



Paso 6: En la pestaña *Direcciones IP*. En la opción *IPAll* -> *Puertos dinámicos TCP* dejarlo en blanco para no habilitar los puertos dinámicos. Pulsar *Aceptar*.

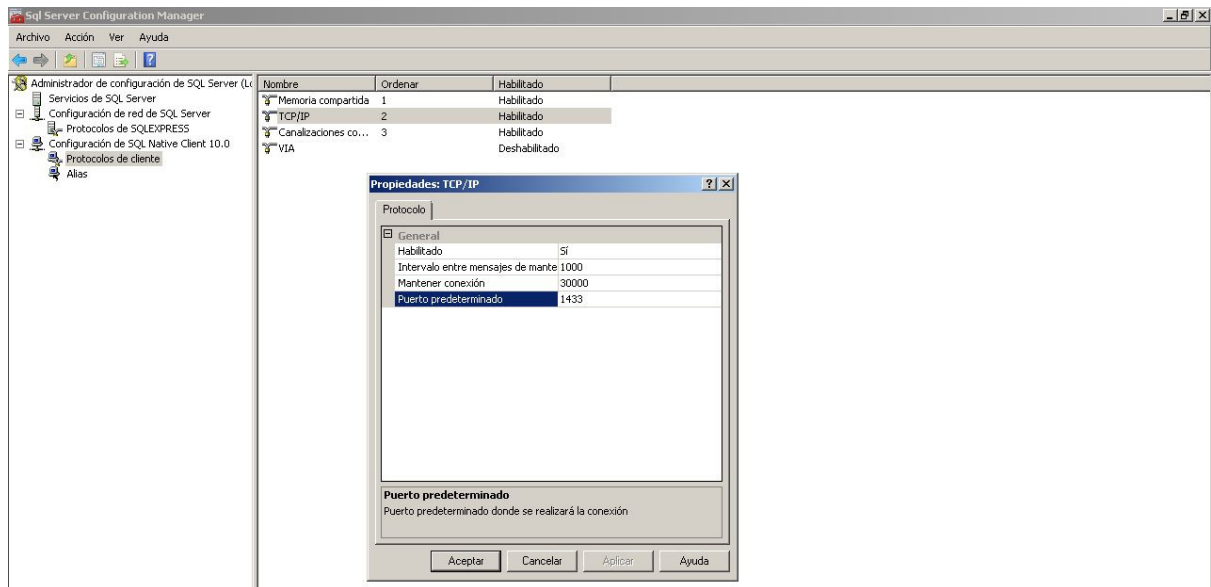


Paso 7: En *Configuración de SQL Native Client 10.0* → *Protocolos de cliente* -> *TCP/IP*, botón derecho y seleccionar *Habilitar*

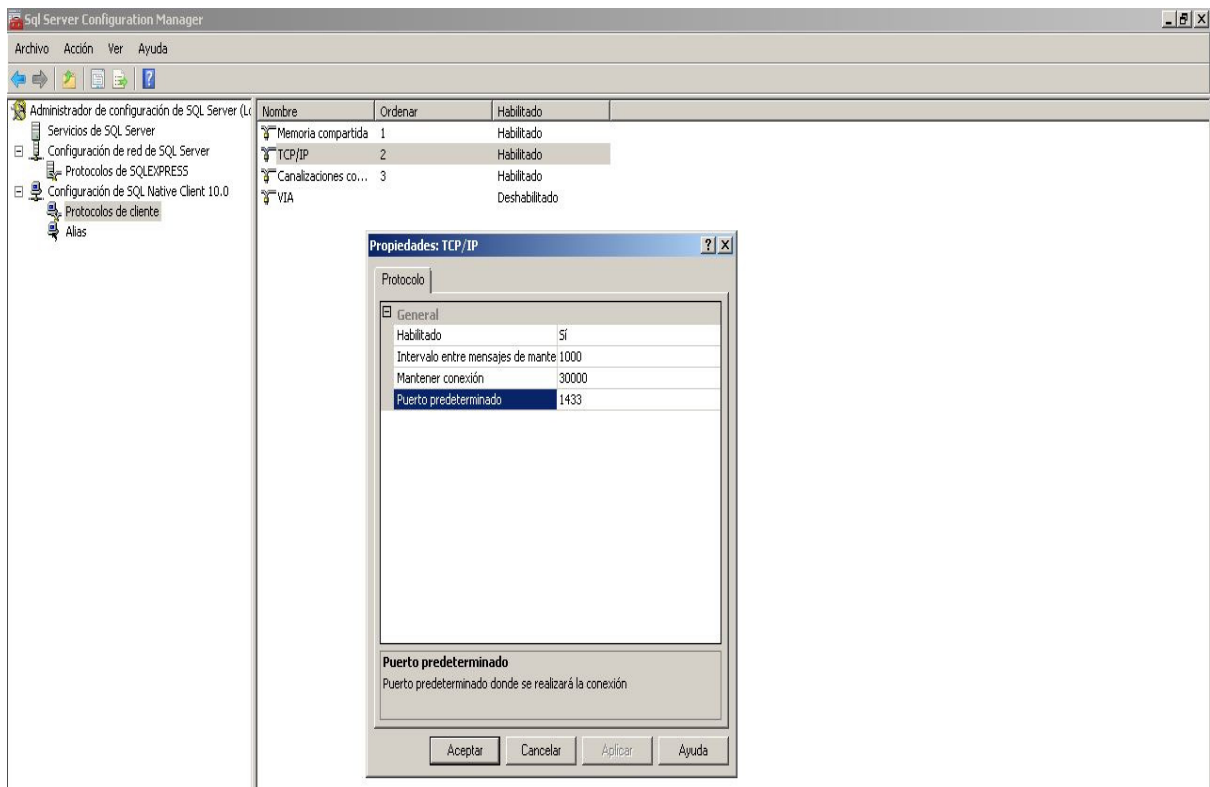
Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

En Configuración de SQL Native Client 10.0 → Protocolos de cliente →Canalizaciones con nombre, botón derecho y seleccionar *Habilitar*

En Configuración de SQL Native Client 10.0 → Protocolos de cliente →Memoria compartida, botón derecho y seleccionar *Habilitar*



Paso 8: Vemos que en Configuración de SQL Native Client 10.0 → Protocolos de cliente →TCP/IP, botón derecho , en protocolo tenemos como puerto predeterminado el que hemos puesto anteriormente 1433



Paso 9: Una vez hechos los cambios reiniciamos el servicio SQL Server (SQLEXPRESS). Para ello vamos a la opción *Servicios de SQL Server* -> *SQL Server (SQLEXPRESS)*, pulsamos botón derecho y seleccionamos la opción *Reiniciar* para que se hagan efectivos los cambios



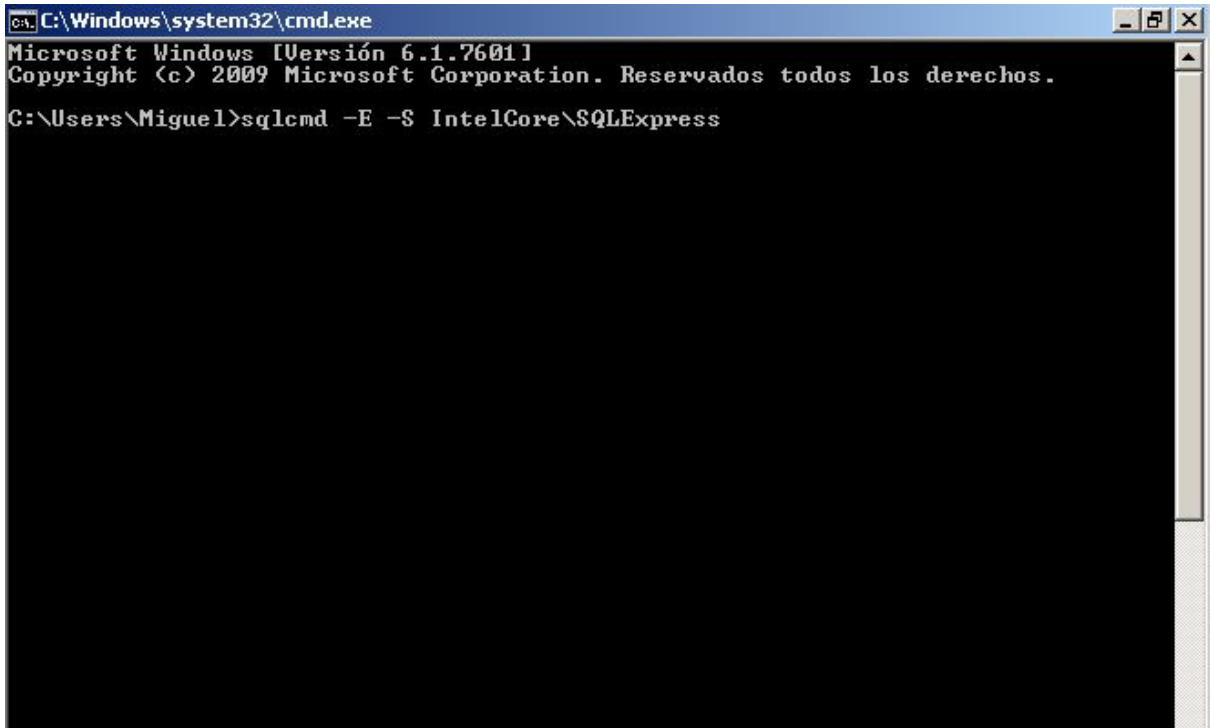
Paso 10: Ahora probaremos la conectividad con SQL Server Express. Para ello abriremos una ventana de línea de comandos (comando cmd) y ejecutamos el comando:

sqlcmd -E -S <servidorSQLExpress> (<nombre de servidor\Instancia> -> por ejemplo IntelCore\SQLEXPRESS)

En el caso en que tengas seleccionado el Modo mixto como método de autenticación a SQL Server Express entonces también se puede utilizar el comando:

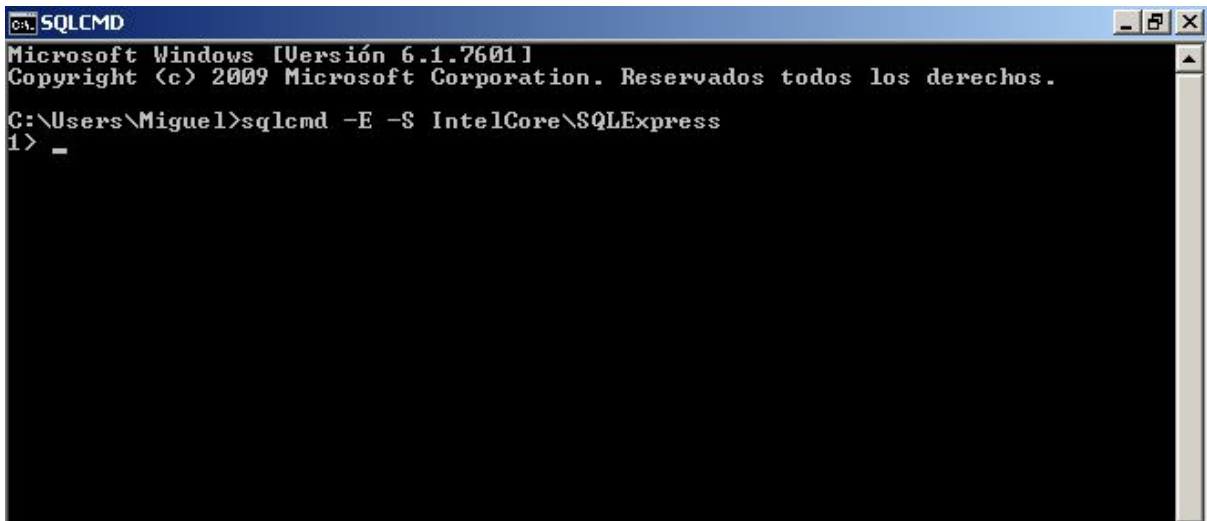
Sqlcmd -Usa -P<clave de acceso sa> -S <servidorSQLEXPRESS>

La clave de acceso sa es la de windows.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.
C:\Users\Miguel>sqlcmd -E -S IntelCore\SQLEXPRESS
```

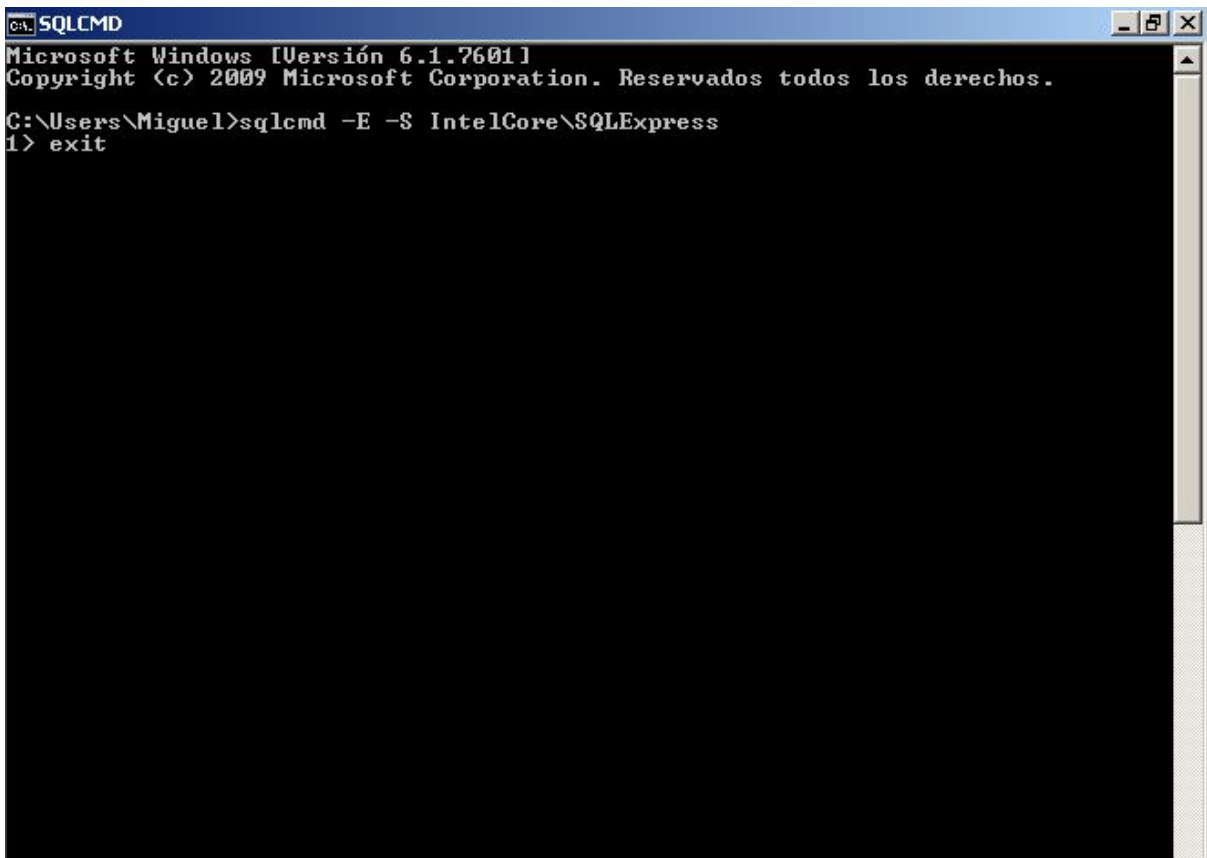
Paso 11: Si el prompt de comando queda como 1>, es señal de que una conexión fue establecida con éxito



```
SQLCMD
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

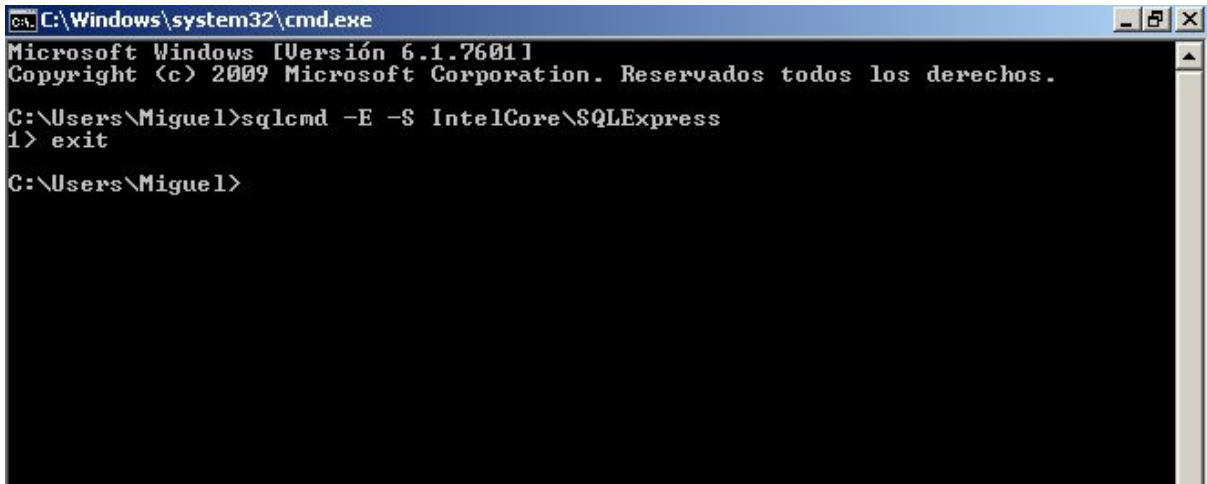
C:\Users\Miguel>sqlcmd -E -S IntelCore\SQLEXPRESS
1> _
```

Paso 12: Pulsamos exit para salir.



```
SQLCMD
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Miguel>sqlcmd -E -S IntelCore\SQLEXPRESS
1> exit
```

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Miguel>sqlcmd -E -S IntelCore\SQLExpress
1> exit

C:\Users\Miguel>
```

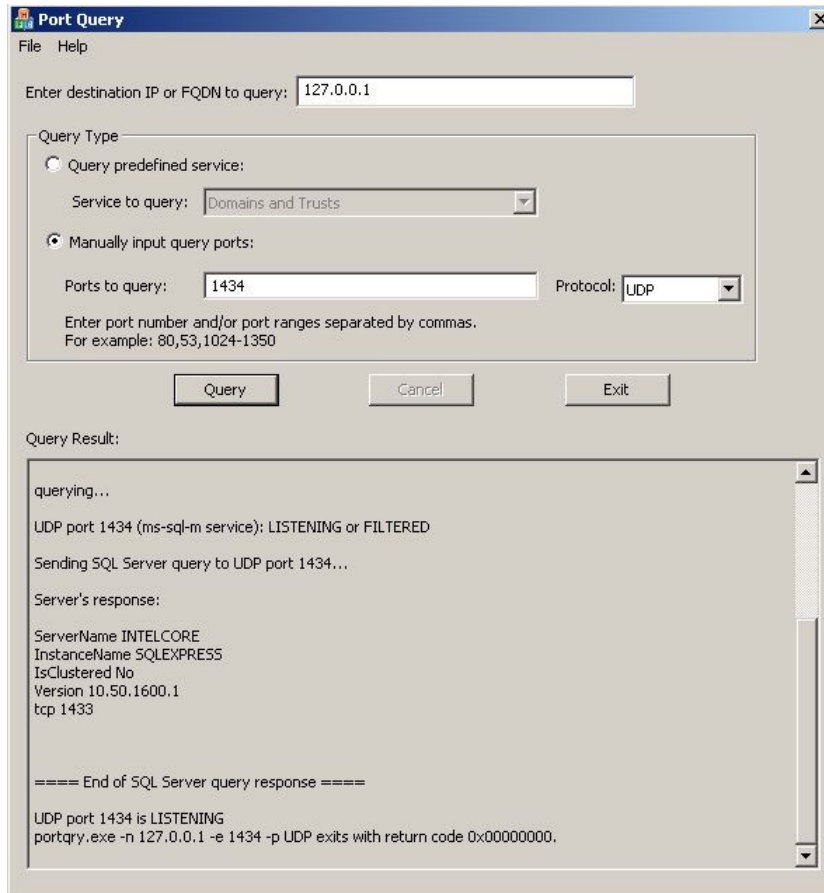
Paso 13: Además de la utilidad sqlcmd.exe, también podemos probar la conectividad remota con SQL Server Express utilizando la utilidad Port Qry GUI.

Para ello ejecutamos la aplicación portqueryui.exe para solucionar problemas de conectividad con el protocolo TCP/IP. Al ser ejecutado mostrará el estado del puerto TCP ó UDP, sea para una máquina local o remota.

Caso 1: Máquina local puerto UDP 1433

Realizamos una petición manual al puerto UDP 1434 en la maquina local (IP 127.0.0.1)

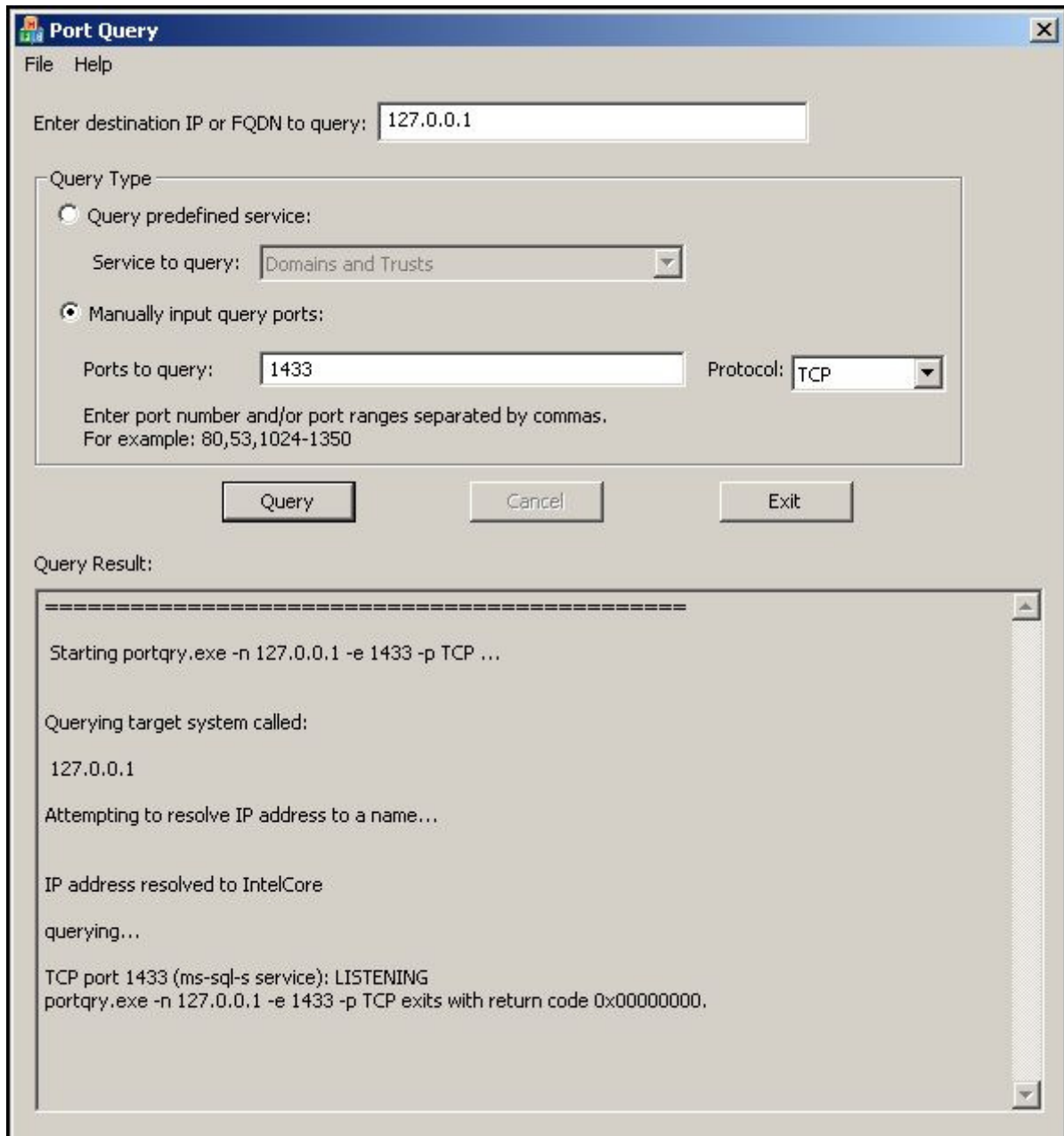
Si se ha realizado correctamente se nos muestra información de respuesta del servidor (LISTENING)



Caso 2: Máquina local puerto TCP 1433

Realizamos una petición manual al puerto TCP 1433 en la maquina local (IP 127.0.0.1)

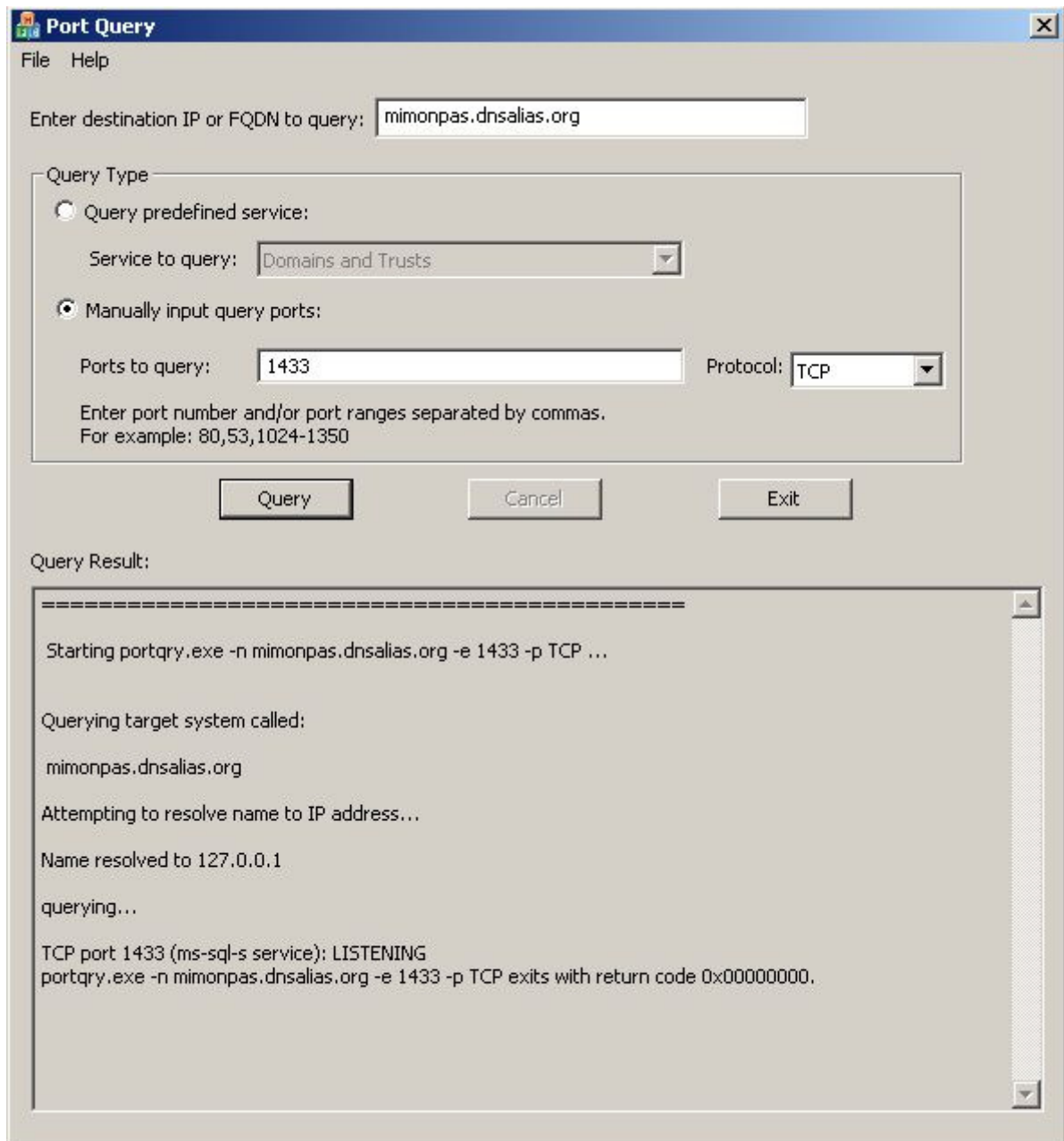
Si se ha realizado correctamente se nos muestra información de respuesta del servidor (LISTENING)



Caso 3: Máquina remota puerto TCP 1433 (en el ejemplo máquina remota mimonpas.alias.org)

Realizamos una petición manual al puerto TCP 1433 en la maquina remota (mimonpas.dnsalias.org) -> en la intranet se resuelve el nombre como 127.0.0.1 y en Internet se resuelve el nombre con la ip dinámica que se tenga en ese momento proporcionada por el operador que proporciona acceso a internet

Si se ha realizado correctamente se nos muestra información de respuesta del servidor (LISTENING)



3 – APERTURA DE PUERTOS EN EL ROUTER (NAT) PARA ACCESO REMOTO MEDIANTE TCP/IP A SQL SERVER 2008 R2 EXPRESS

Mediante NAT se dirige el tráfico de entrada desde una WAN (identificado por el protocolo y el puerto externo) a un servidor interno con una IP privada dentro de una LAN.

Paso 1: Abrimos el menú web del router introduciendo la dirección IP del router en la red local (ejemplo 192.168.1.1) . A continuación vamos a la opción NAT (Traducción de direcciones de red)

COMTREND ADSL Router

NAT -- Virtual Servers Setup

Virtual Server allows you to direct incoming traffic from WAN side (identified by Protocol and External port) to the Internal server with private IP address on the LAN side. The Internal port is required only if the external port needs to be converted to a different port number used by the server on the LAN side. A maximum 32 entries can be configured.

Add Remove

| Server Name | External Port Start | External Port End | Protocol | Internal Port Start | Internal Port End | Server IP Address | Remove |
|-------------|---------------------|-------------------|----------|---------------------|-------------------|-------------------|--------------------------|
| BRComet | 12775 | 12775 | TCP | 12775 | 12775 | 192.168.1.2 | <input type="checkbox"/> |
| Emule1 | 15125 | 15125 | TCP | 15125 | 15125 | 192.168.1.2 | <input type="checkbox"/> |
| Emule1 | 36207 | 36207 | UDP | 36207 | 36207 | 192.168.1.2 | <input type="checkbox"/> |
| BRComet2 | 12775 | 12775 | TCP | 12775 | 12775 | 192.168.1.15 | <input type="checkbox"/> |
| Emule_2 | 8584 | 8584 | TCP | 8584 | 8584 | 192.168.1.4 | <input type="checkbox"/> |
| Emule_2 | 26048 | 26048 | UDP | 26048 | 26048 | 192.168.1.4 | <input type="checkbox"/> |
| BRComet_2 | 12295 | 12295 | TCP | 12295 | 12295 | 192.168.1.4 | <input type="checkbox"/> |
| VNC | 5900 | 5900 | TCP | 5900 | 5900 | 192.168.1.4 | <input type="checkbox"/> |

Add Remove

Paso 2: Vamos a abrir el puerto 1434 UDP en el router y direccionarlo a la dirección IP del ordenador donde se está escuchando dicho puerto .Para ello pulsamos sobre el botón Add (añadir nueva NAT). En la opción Nombre de servidor escribimos SQL_SERVER_UDP.

En IP Server Address (dirección IP del servidor) ponemos la dirección IP local donde está instalado el servidor SQL Server (en el ejemplo 192.168.1.4)

En External Port Start ponemos el puerto de escucha externo: 1434

En External Port End ponemos el mismo puerto: 1434

En protocolo ponemos UDP

La demás opciones se ponen automáticamente (Internal Port Start e Internal Port End).Pulsamos Salvar o Aplicar y reiniciamos el Router

COMTREND ADSL Router

NAT -- Virtual Servers

Select the service name, and enter the server IP address and click "Save/Apply" to forward IP packets for this service to the specified se
Port End" normally and will be the same as the "Internal Port Start" or "External Port End" if either one is modified.
 Remaining number of entries that can be configured:24

Server Name:

Select a Service:

Custom Server:

Server IP Address:

| External Port Start | External Port End | Protocol | Internal Port Start | Internal Port End |
|---------------------|-------------------|----------|---------------------|-------------------|
| 1434 | 1434 | UDP | 1434 | 1434 |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

Paso 3: Vamos a abrir el puerto 1433 TCP en el router y direccionarlo a la dirección IP del ordenador donde se está escuchando dicho puerto .Para ello pulsamos sobre el botón Add (añadir nueva NAT). En la opción Nombre de servidor escribimos SQL_SERVER_TCP.

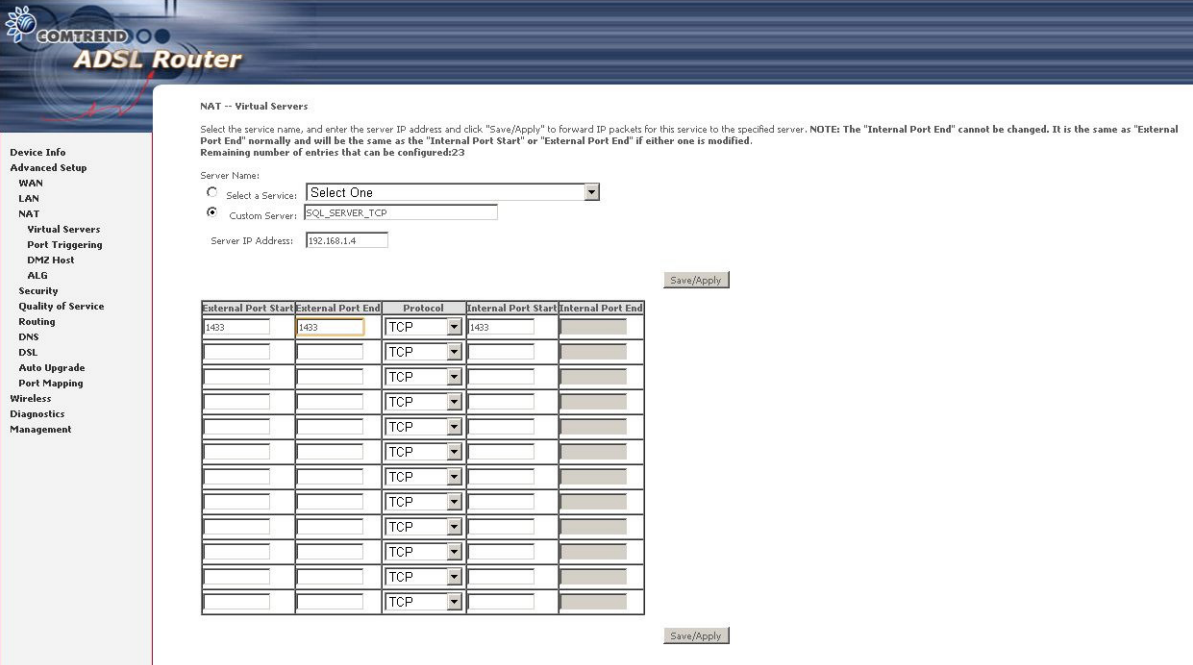
En IP Server Address (dirección IP del servidor) ponemos la dirección IP local donde está instalado el servidor SQL Server (en el ejemplo 192.168.1.4)

En External Port Start ponemos el puerto de escucha externo: 1433

En External Port End ponemos el mismo puerto: 1433

En protocolo ponemos TCP

La demás opciones se ponen automáticamente (Internal Port Start e Internal Port End). Pulsamos Salvar o Aplicar y reiniciamos el Router



The screenshot shows the configuration page for NAT Virtual Servers on a COMTREND ADSL Router. The interface includes a sidebar with navigation options and a main configuration area. The main area has a title "NAT -- Virtual Servers" and a note: "Select the service name, and enter the server IP address and click 'Save/Apply' to forward IP packets for this service to the specified server. NOTE: The 'Internal Port End' cannot be changed. It is the same as 'External Port End' normally and will be the same as the 'Internal Port Start' or 'External Port End' if either one is modified. Remaining number of entries that can be configured:23".

Below the note, there are fields for "Server Name:" with a dropdown menu set to "Select One", "Select a Service:" with a radio button, "Custom Server:" with a text box containing "SQL_SERVER_TCP", and "Server IP Address:" with a text box containing "192.168.1.4".

A "Save/Apply" button is located to the right of the form. Below the form is a table with the following columns: "External Port Start", "External Port End", "Protocol", "Internal Port Start", and "Internal Port End".

| External Port Start | External Port End | Protocol | Internal Port Start | Internal Port End |
|---------------------|-------------------|----------|---------------------|-------------------|
| 1433 | 1433 | TCP | 1433 | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |
| | | TCP | | |

Another "Save/Apply" button is located at the bottom right of the table area.

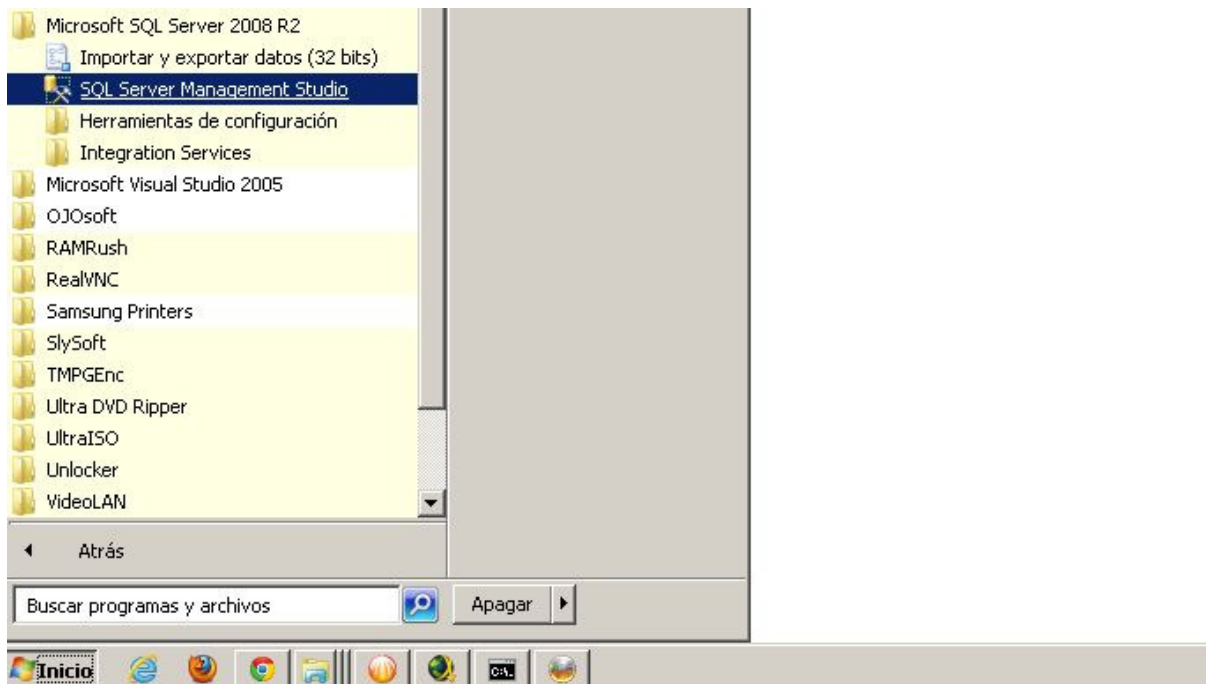
4 – CONFIGURACIÓN y ADMINISTRACIÓN DEL SERVIDOR SQL SERVER 2008 R2 EXPRESS

Vamos a utilizar una herramienta que nos permite administrar nuestra base de datos, realizar comprobaciones, y otro tipo de tareas.

Esta herramienta es SQL Server Management Studio, la cual hay que instalarla a parte si no se ha instalado conjuntamente con SQL Server 2008 R2 Express.

También está la posibilidad de utilizar un panel de exploración de base de datos en el entorno de Visual Studio.

Paso 1: Vamos a ejecutar la herramienta. Para ello vamos al menú de inicio y seleccionamos Microsoft SQL Server 2008 R2 y a continuación SQL Server Management Studio



Paso 2: Se nos muestra una ventana de conexión con el servidor.

Seleccionamos lo siguiente:

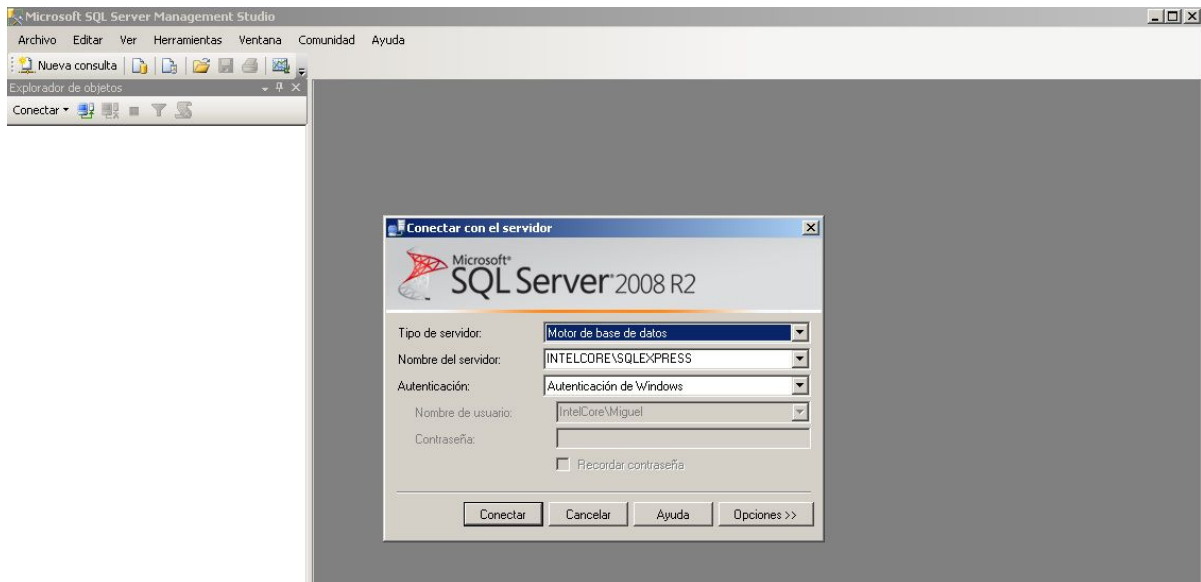
- Tipo de servidor: Motor de base de datos
- Nombre del servidor: Elegiremos el servidor con el que conectaremos. Por ejemplo: INTELCORE\SQLEXPRESS. Debes recordar este nombre, porque será el que tengas que utilizar en las cadenas de conexión desde .NET
- Autenticación: Hay dos opciones "Autenticación de Windows" o "Autenticación de SQL Server". Si elegimos la opción "Autenticación de Windows" estaremos conectando con la misma cuenta de usuario con la que iniciamos nuestro sistema. Si seleccionamos "Autenticación de SQL

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

Server” debemos indicar con que cuenta vamos a conectar e introducir la contraseña específica para conectar con SQL Server.

En nuestro caso trabajaremos con la “Autenticación de Windows” ya que de otro modo podemos encontrar problemas por la limitación de la versión gratuita.

Pulsamos en Conectar.



Paso 3: Si todo ha resultado correcto, arranca nuestra herramienta de administración. Vamos a crear una cuenta de usuario para la conexión a la base de datos.

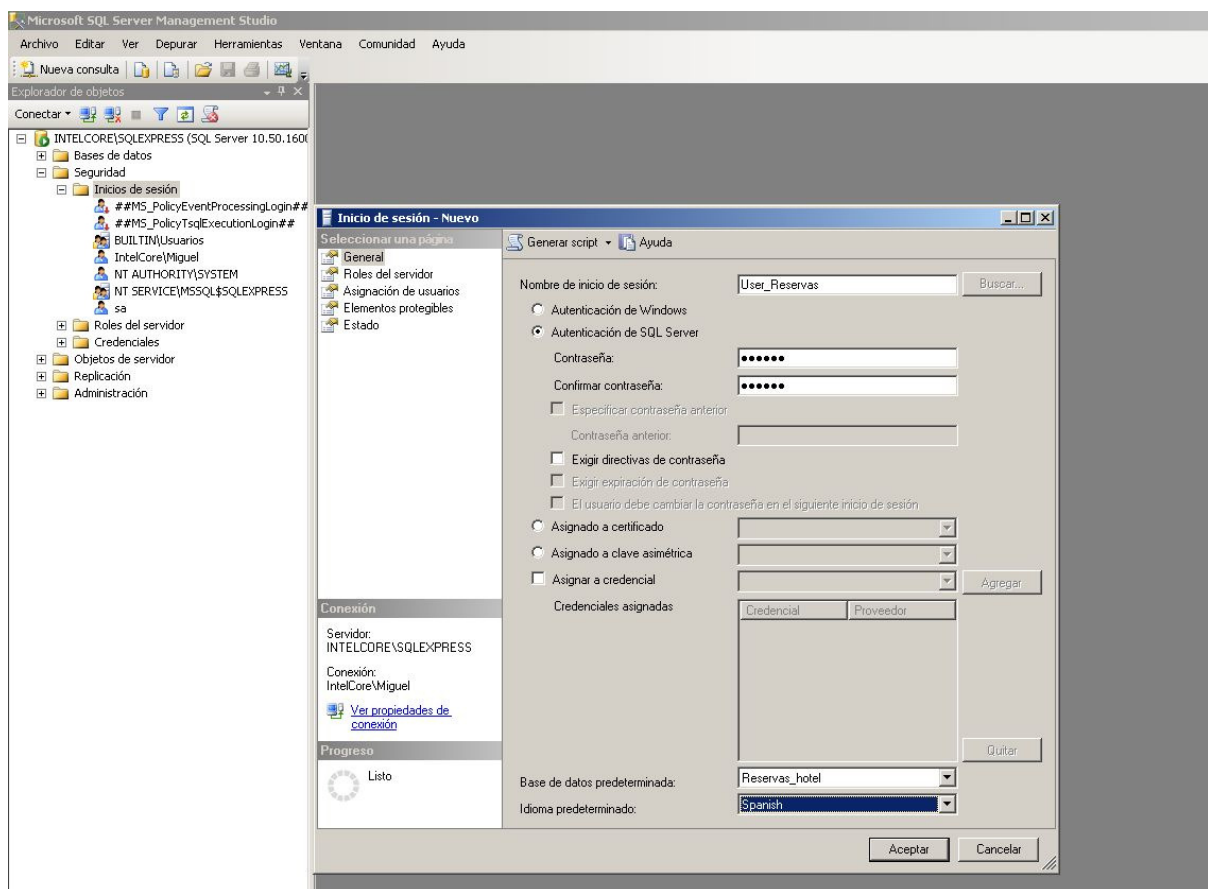
Vamos a la opción “Seguridad” -> “Inicios de sesión”, pulsamos botón derecho del ratón y seleccionamos en el desplegable la opción “Nuevo Inicio de sesión”.



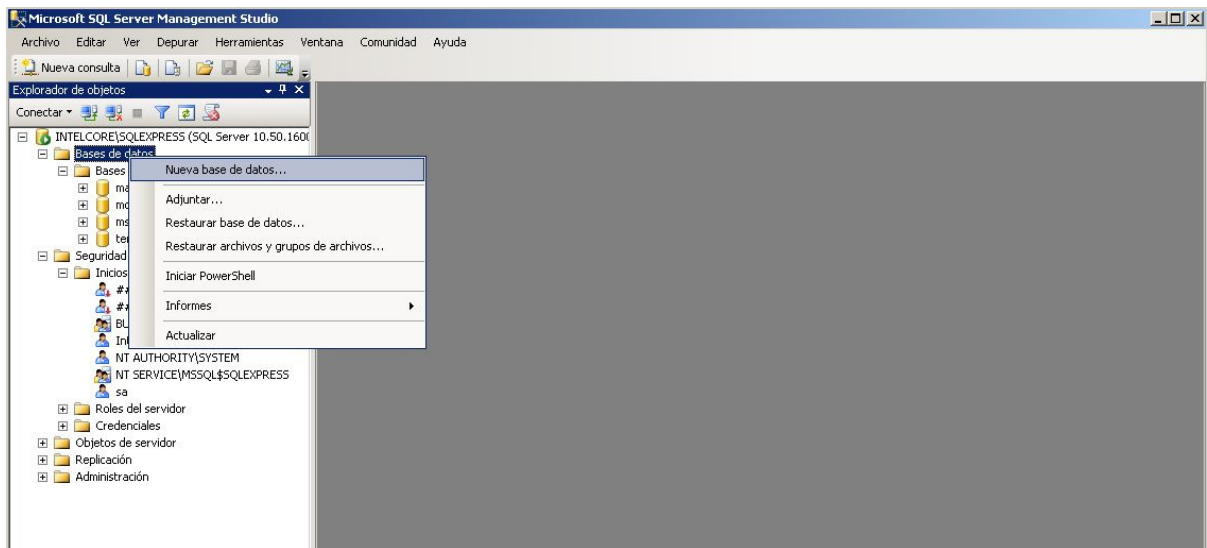
Paso 4: En la siguiente ventana comienza el proceso de creación de la cuenta

En la opción “**General**” indicamos:

- *Nombre de inicio de sesión:* En el ejemplo User_Reservas.
- *Tipo de autenticación para esta cuenta.* Podemos elegir entre autenticación de Windows o autenticación de SQL Server. En el ejemplo crearemos una cuenta de SQL Server, por lo tanto la seleccionamos y le indicamos la contraseña.
- Vemos que tenemos la posibilidad de forzar las *políticas de contraseñas*, las cuales se aseguran que las contraseñas cumplan unas determinadas condiciones de formato, de caducidad, y la posibilidad de que el usuario cambie su contraseña en el momento de iniciar por primera vez su sesión. En el ejemplo no exigimos directivas de contraseña
- *Base de datos por defecto:* Inicialmente es master pero más adelante cuando creemos nuestra base de datos escribiremos Reservas_hotel
- *Idioma predeterminado:* Pondremos Spanish.

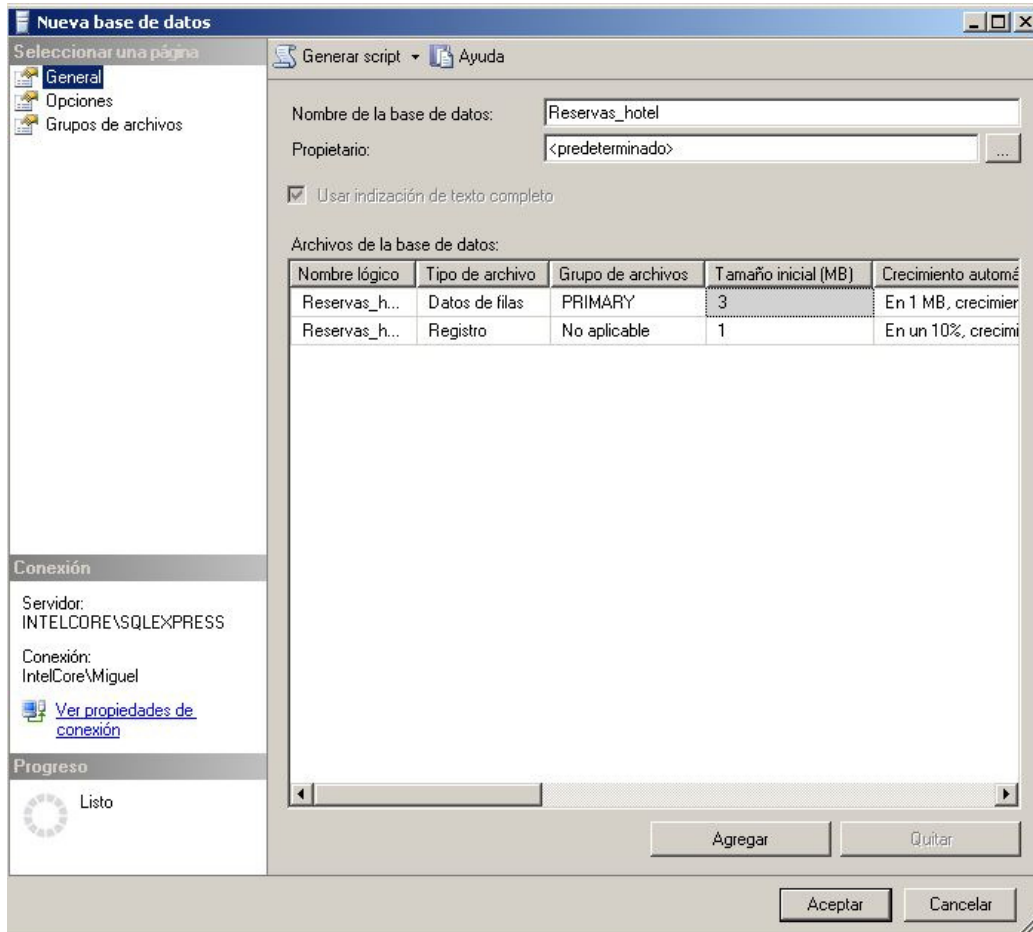


Paso 5: Vamos a crear la base de datos. Seleccionamos la opción “**Bases de datos**”, pulsamos botón derecho y en el menú desplegable seleccionamos la opción “**Nueva base de datos**”.



Paso 6: Durante la tarea de creación de nuevas bases de datos, se fija el nombre (En el ejemplo Reservas_hotel), sus propiedades, y la ubicación física de los archivos que lo componen.

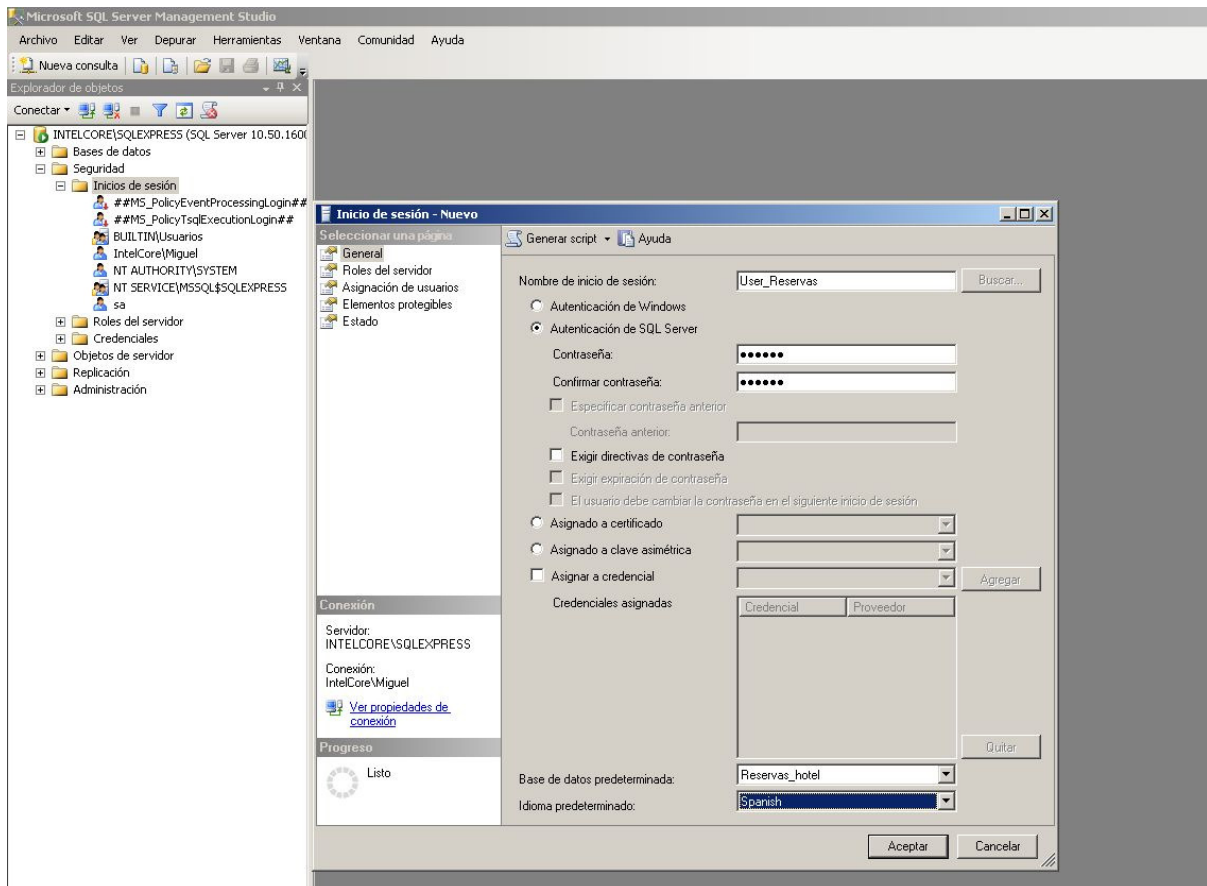
- En la opción “**General**”: Rellenamos los datos de nombre de la base de datos (En el proyecto Reservas_hotel) el resto de opciones las dejamos como están.



Paso 7: Ahora si podemos seleccionar en “Inicio de sesión” -> “User_Rservas” , botón derecho “Propiedades” , en la opción “General” en las opciones:

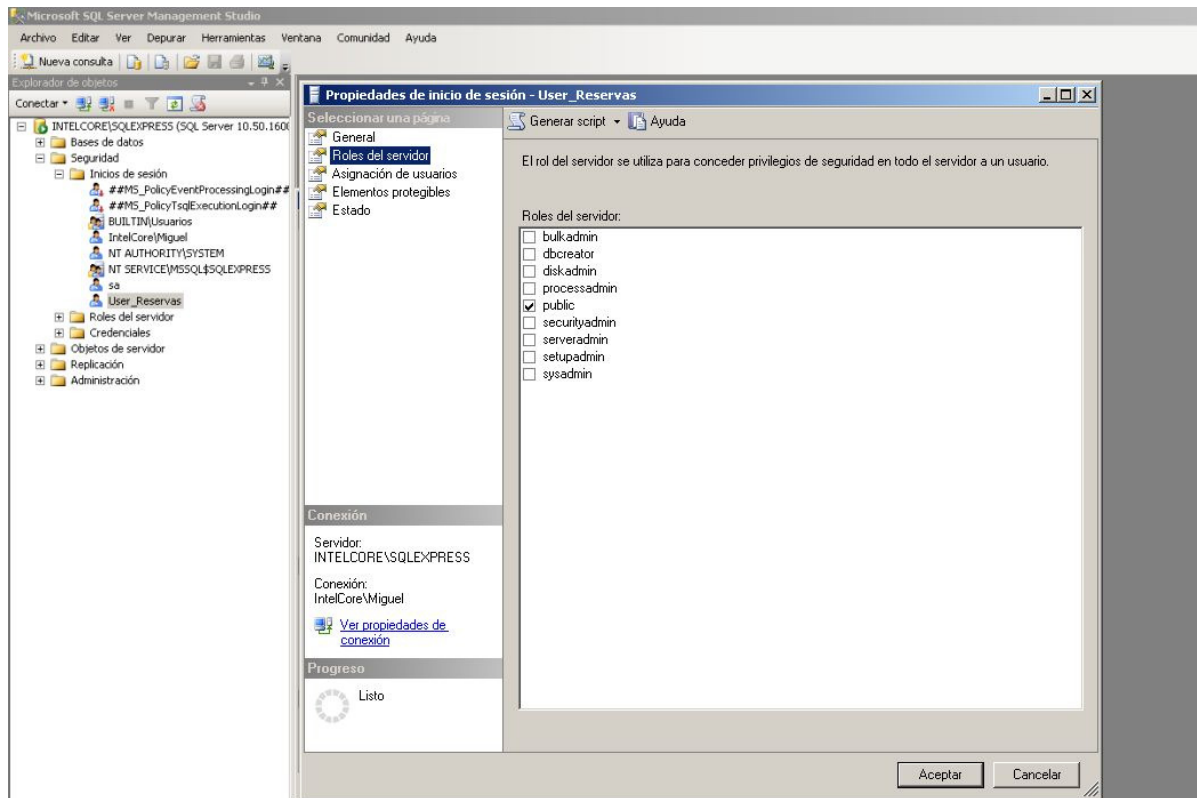
Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

- Base de datos predeterminada: Reservas_hotel (la hemos creado anteriormente)
- Idioma predeterminado: Se deja como está, en Spanish



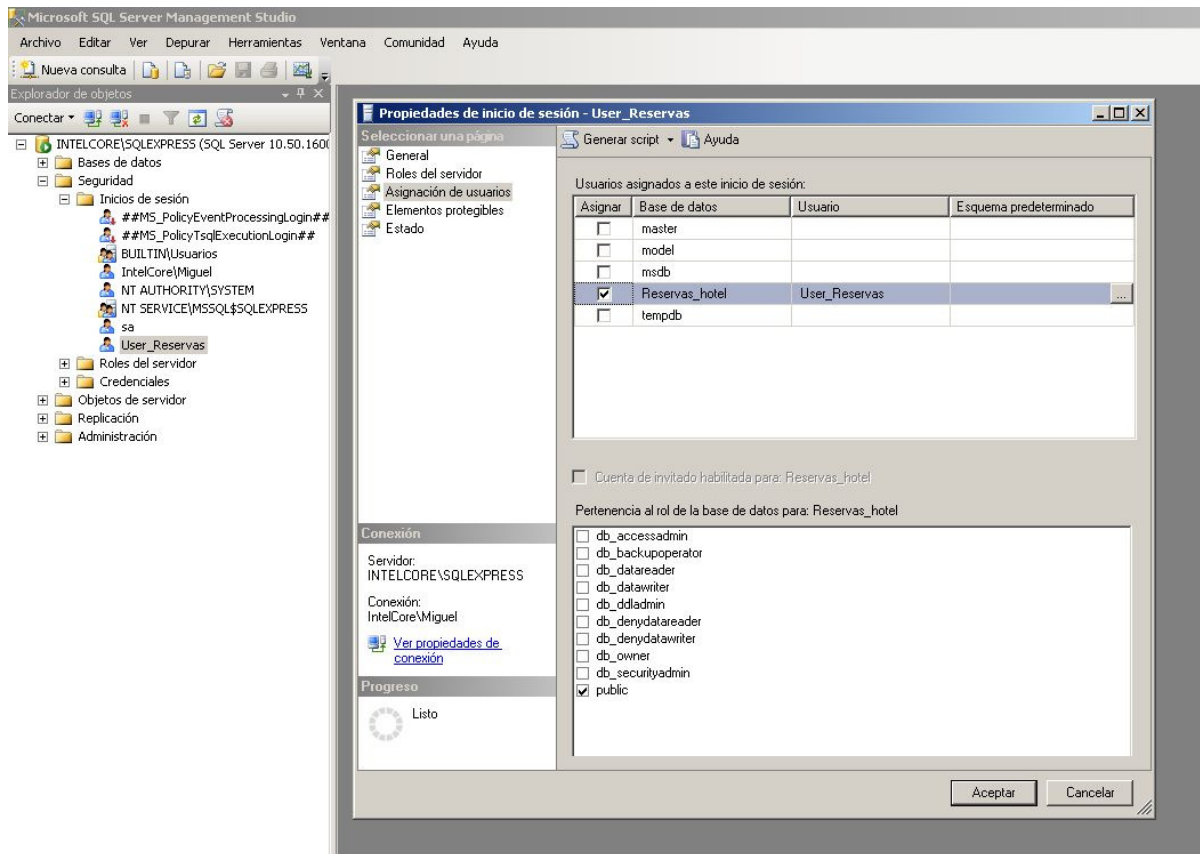
En la opción “Roles del servidor” seleccionamos la pestaña **public**





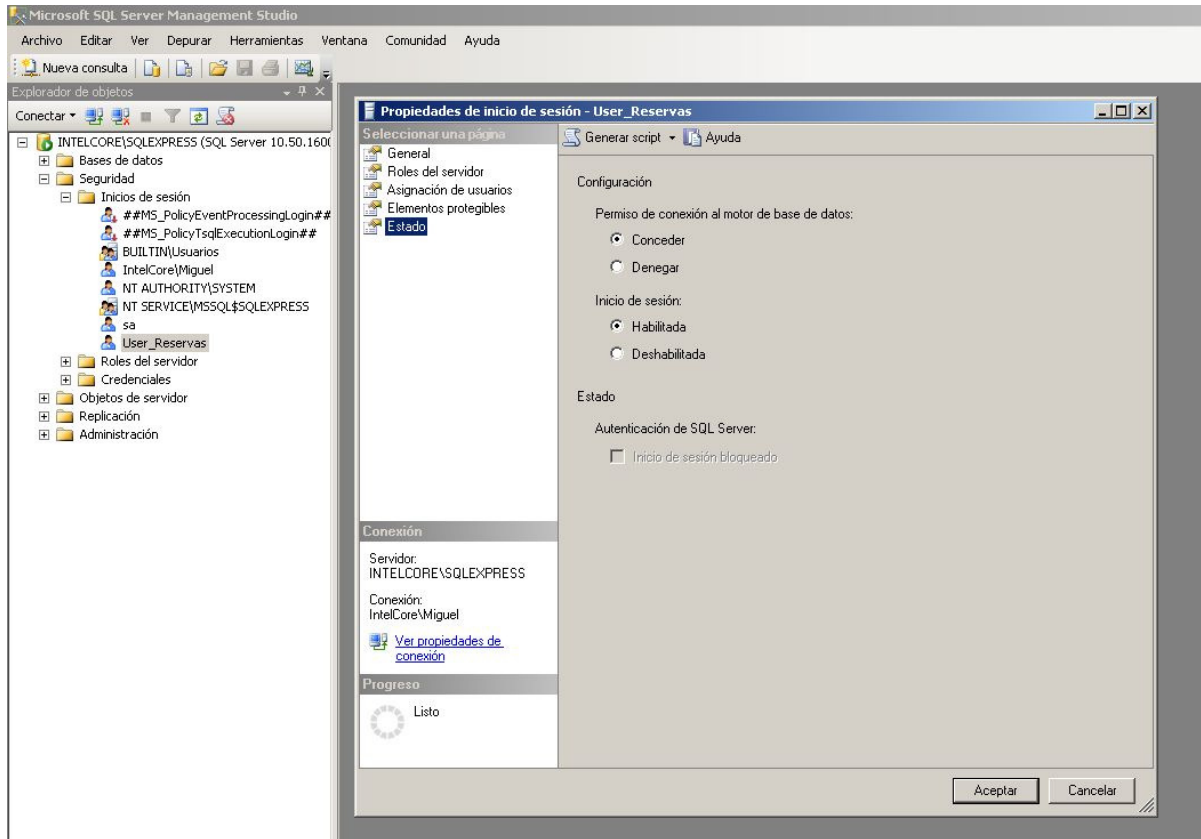
En la opción “Asignación de usuarios” . Marcamos la casilla “asignar” a Reservas_hotel y seleccionamos el usuario creado anteriormente “User_Reservas”

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

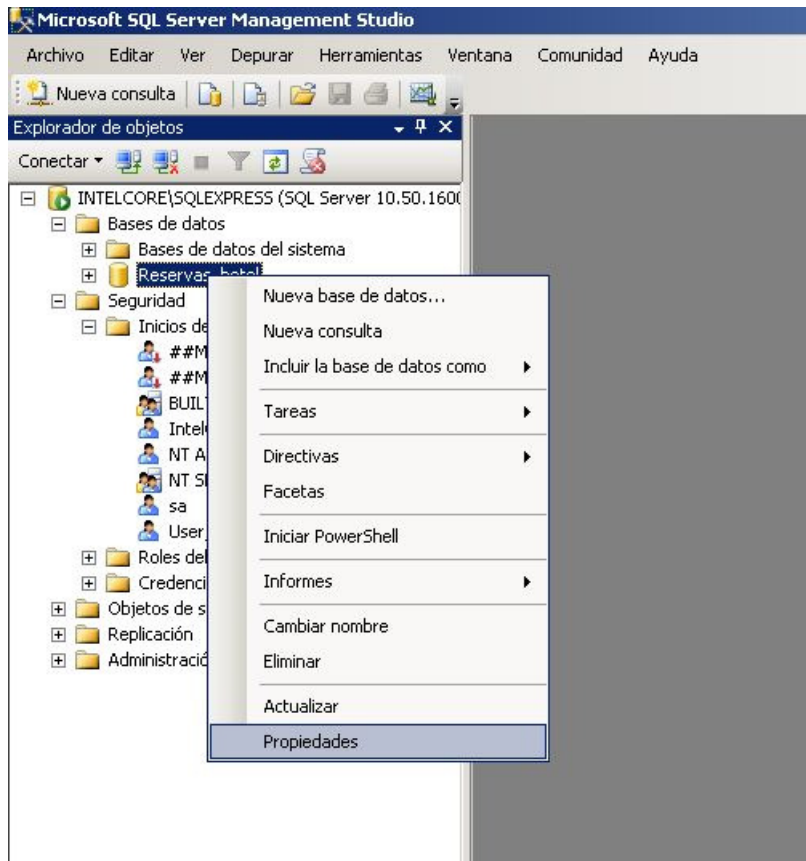


En la opción “Estado”. Permiso de conexión de base de datos -> Conceder. Inicio de sesión -> habilitada

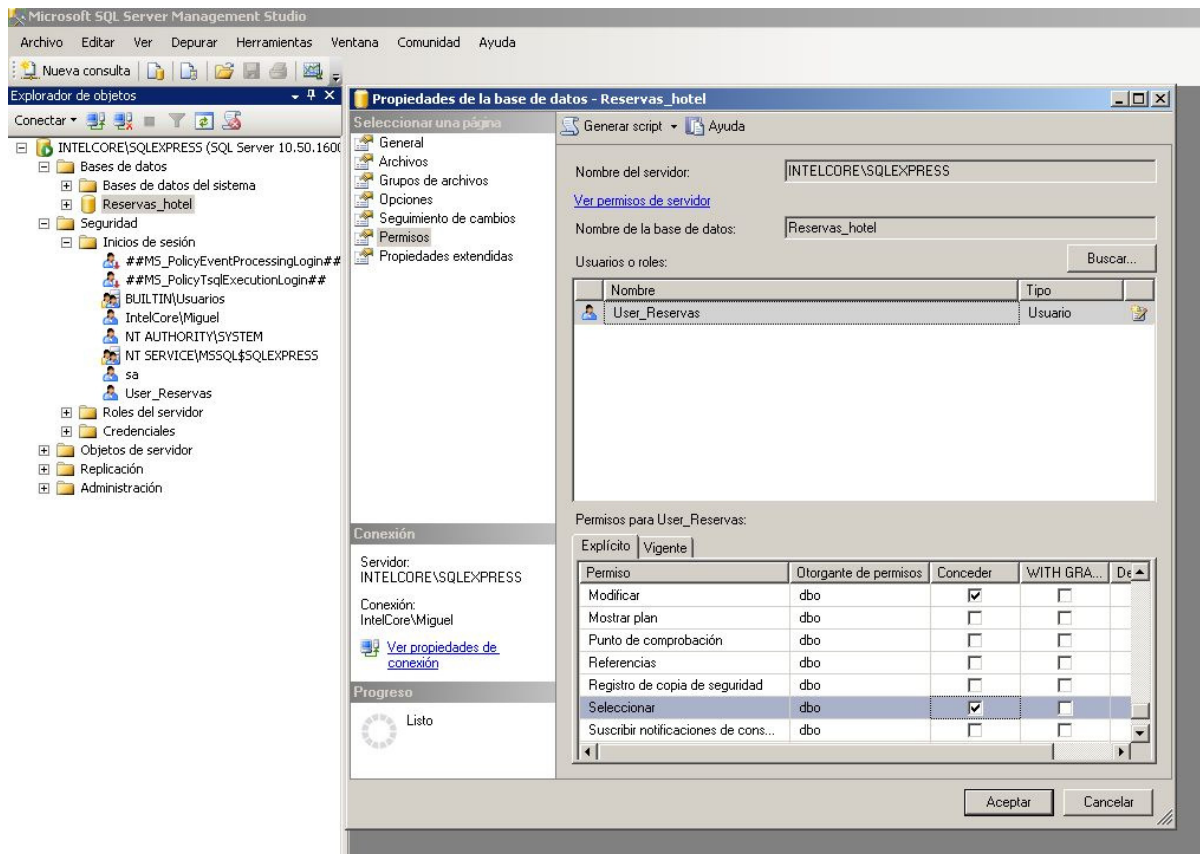




Paso 8 : Seleccionamos la base de datos Reservas_hotel, botón derecho “Propiedades”



En la opción **Permisos**. En permisos para **User_Reservas** -> **Explícitos**, concedemos los permisos de Actualizar, Conectar, Ejecutar, Eliminar, Insertar, Modificar y Seleccionar.



2.5.2 Codificación del programa (listado completo y comentado)

FORMULARIO ACCESO

```
Imports System.Data.SqlClient
```

```
Public Class Form_acceso
```

```
    Private Sub TextBox_Usuario_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles TextBox_Usuario.Click
```

```
        TextBox_Usuario.Text = ""
```

```
        Mensaje.Text = ""
```

```
    End Sub
```

```
    Private Sub TextBox_Password_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles TextBox_Password.Click
```

```
        TextBox_Password.Text = ""
```

```
        Mensaje.Text = ""
```

```
    End Sub
```

```
    Private Sub Button_Cerrar_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button_Cerrar.Click
```

```
        Me.Close()
```

```
    End Sub
```

```
    Private Sub Button_Aceptar_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button_Aceptar.Click
```

```
        Dim Resultado As Integer
```

```
        'Comprobamos que los campos usuario y password no esten vacios
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
If (TextBox_Usuario.Text.Trim Like "") And
(TextBox_Password.Text.Trim Like "") Then

    Mensaje.Text = "Porfavor introduzca usuario y contraseña"

ElseIf TextBox_Usuario.Text.Trim Like "" Then

    Mensaje.Text = "Porfavor introduzca usuario"

ElseIf TextBox_Password.Text.Trim Like "" Then

    Mensaje.Text = "Porfavor introduzca password"

Else

    'Quitamos todos los caracteres de espacio en blanco a la
derecha y a la izq del texto introducido

    'en los TextBox Usuario y Password

    Usuario = TextBox_Usuario.Text.Trim

    Password = TextBox_Password.Text.Trim

Try

    'Crear comando escalar

    'La consulta "Consulta_SQL" devuelve el numero de
filas de la tabla Usuario

    'donde el campo login es igual al login introducido
por teclado

    Consulta_SQL = "SELECT count(*) FROM Usuario WHERE
login = @Usuario"

    'Crear comando

    Comando = New SqlCommand(Consulta_SQL, Conexion)

    ' Añadir parametros al comando

    Comando.Parameters.Add(New SqlParameter("@Usuario",
SqlDbType.VarChar))

    Comando.Parameters("@Usuario").Value = Usuario
```

```

        'En la variable Resultado tenemos el resultado de la
consulta
        'Con metodo ExecuteScalar recuperamo un único valor de
un origen de datos:
        'La primera columna de la primera fila del conjunto de
resultados devuelto por la consulta
        'Esto requiere menos código que el uso del método
ExecuteReader y la posterior
        'realización de las operaciones necesarias para
generar un único valor a partir de los datos devueltos
        'por un objeto OleDbDataReader.
        'La utilizacion de este método tiene sentido cuando
estamos ejecutando una sentencia SQL del tipo
        'Count(*)
Resultado = Comando.ExecuteScalar()

Comando = Nothing

Catch ex As SqlException
objeto
        ' si se produce un error, lo capturamos mediante el
        ' de excepciones particular para el proveedor de SQL
Server
        MessageBox.Show("ERROR 1:" & ControlChars.CrLf &
ex.Message & _
        ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor")
Exit Sub

End Try

Try
        If (Resultado = 0) Then ' Si no hay ningun usuario
registrado con el login introducido
        Mensaje.Text = "El usuario no existe,si lo desea
puede registrarse"

```



```

Else
    'Si hay un usuario con el mismo login que el login
    introducido

    'La consulta "Consulta_SQL" devuelve el numero de
    usuarios de la tabla Usuario

    'donde el campo login y el campo password es igual
    al login y password respectivamente introducido por teclado

    Consulta_SQL = "SELECT count(*) FROM Usuario WHERE
    login = @Usuario and password = @Password"

    Comando = New SqlCommand(Consulta_SQL, Conexion)

    Comando.Parameters.Add(New
    SqlParameter("@Usuario", SqlDbType.VarChar))

    Comando.Parameters.Add(New
    SqlParameter("@Password", SqlDbType.VarChar))

    Comando.Parameters("@Usuario").Value = Usuario
    Comando.Parameters("@Password").Value = Password

    Resultado = Comando.ExecuteNonQuery()

    Comando = Nothing

    If (Resultado = 0) Then 'Si hay un usuario con el
    mismo login pero no con el mismo password

        Mensaje.Text = "Usuario válido, password
    incorrecto"

    Else

        'Hay un usuario con el mismo login y password
        que el introducido por teclado

        Me.Hide()

        'Me.Close()

        Registrarse = False 'No estamos en el modo
    registrarse

```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```

                                'Mostramos el formulario Form_usuario y lo
hacemos activo

                                Form_usuario.Visible = True

                                Form_usuario.Show()

                                Form_usuario.Activate()

                                End If

                                End If

                                Catch ex As SqlException

                                ' si se produce un error, lo capturamos mediante el
objeto

                                ' de excepciones particular para el proveedor de SQL
Server

                                MessageBox.Show("ERROR 2" & ControlChars.CrLf &
ex.Message & _

                                ControlChars.CrLf & "SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor")

                                Exit Sub

                                End Try

                                End If

                                End Sub

                                'Evento se se produce cuando el raton se situa sobre la etiqueta
L_Registrase
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Private Sub Label4_MouseEnter(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles L_Registrase.MouseEnter
```

```
    L_Registrase.ForeColor = Color.Red
```

```
End Sub
```

```
'Evento se se produce cuando el raton deja de estar sobre la etiqueta L_Registrase
```

```
Private Sub Label4_MouseLeave(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles L_Registrase.MouseLeave
```

```
    L_Registrase.ForeColor = Color.Black
```

```
End Sub
```

```
'Evento que se produce cuando se pulsa sobre la etiqueta L_Registrase
```

```
Private Sub L_Registrase_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles L_Registrase.Click
```

```
    Me.Hide()
```

```
    Registrarse = True 'Estamos en el modo registrarse
```

```
    'Mostramos el formulario Form_usuario y lo hacemos activo
```

```
    Form_usuario.Visible = True
```

```
    Form_usuario.Show()
```

```
    Form_usuario.Activate()
```

```
End Sub
```

```
'Evento que se produce cuando se carga el formulario Form_acceso
```

```
Private Sub Form_acceso_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
    'Borramos todo de los TextBox y Label
```

```
    TextBox_Usuario.Text = ""
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
TextBox_Password.Text = ""

Mensaje.Text = ""

End Sub

Private Sub Form_acceso_Shown(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Shown

    Try

        ' Paso la cadena de conexion

        Conexion.ConnectionString = "Data Source =
mimonpas.dnsalias.org\SQLExpress,1433;Network=dbmssocn;" & _
        "Initial Catalog= Reservas_Hotel;" & "User id =
User_Reservas; password = dorado;Connection Timeout=60;"

        Conexion.Open()

    Catch ex As Exception

        ' si se produce un error, lo capturamos mediante el objeto
        ' de excepciones particular para el proveedor de SQL
Server

        MessageBox.Show("ERROR 0" & ControlChars.CrLf & ex.Message
& _
        ControlChars.CrLf & "SERVIDOR:SQL SERVER 2008 R2 Express.
Miguel Monzo Pastor")

        Application.Exit()

    End Try

End Sub
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Private Sub Form_acceso_FormClosing(ByVal sender As System.Object,  
ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles  
MyBase.FormClosing  
  
    Try  
  
        Conexion.Close()  
  
        Conexion = Nothing  
  
    Catch ex As Exception  
  
        MessageBox.Show("ERROR 2.1" & ControlChars.CrLf &  
ex.Message & _  
ControlChars.CrLf & "SERVIDOR:SQL SERVER 2008 R2 Express.  
Miguel Monzo Pastor")  
  
    End Try  
  
End Sub  
  
End Class
```

FORMULARIO USUARIO

```
Imports System.Data.SqlClient
```



```

Public Class Form_usuario

    Private Id As Int64

    Private localidad As String

    'Evento que se ejecuta antes de que se muestre un formulario por
    primera vez.

    Private Sub Form_usuario_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load

        Dim Fecha1 As DateTime

        Dim Fecha2 As DateTime

        Dim fecha As DateTime

        'Variables para la conexión y consulta a la base de datos

        Dim oDataAdapter As SqlDataAdapter

        Dim oDataSet As DataSet

        Dim oCB As SqlCommandBuilder

        Dim Datos As SqlDataReader

        If Registrarse = True Then 'Si estamos en el modo registrarse

            'Creamos objeto Cliente

            Objeto_usuario = New Clase_usuario("", "")

            ' Inicializamos todos los campos del formulario

            fecha = DateTime.Now

            Resp_Datos_usuario.Text = ""

            TB_Tipo_usuario.Text = "Cliente"

            TB_Nombre.Text = ""
        
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
TB_NIF.Text = ""
TB_Localidad.Text = ""
TB_Codigo_Postal.Text = ""
TB_Email.Text = ""
TB_Login.Text = ""
TB_Password.Text = ""
Tb_Fecha_alta.Text = fecha.ToLongDateString
TB_Apellidos.Text = ""
TB_Domicilio.Text = ""
TB_Provincia.Text = ""
TB_Telefono.Text = ""

TB_Ultimo_acceso.Text = fecha.ToString("G") 'Fecha corta,
hora larga

Panel_Usuario.Enabled = False 'No habilitamos el Panel de
usuario, no permitimos modificacion

Resp_Datos_Usuario2.Visible = False

TabControl_Usuario.TabPages.Remove(TP_Reserva) 'No
mostramos la pestaña Reserva

TabControl_Usuario.TabPages.Remove(TP_List_Res) 'No
mostramos la pestaña Listado de Reservas

Else 'si no estamos en modo registrarse

' creo un objeto cliente

Objeto_Usuario = New Clase_Usuario(Usuario, Password)

Panel_Usuario.Enabled = False 'No habilitamos el Panel de
usuario, no permitimos modificacion

TB_Login.Text = Usuario

TB_Password.Text = Password
```

Try

```
'Hacemos una consulta a la base de datos para
averiguar los datos del usuario

'Crear comando escalar

'La consulta "Consulta_SQL" devuelve todos campos de
la tabla Usuario

'donde el campo login y el campo password es igual al
login y password respectivamente introducido por teclado

Consulta_SQL = "SELECT * FROM Usuario WHERE login =
@Usuario and password = @Password"

'Crear comando

Comando = New SqlCommand(Consulta_SQL, Conexion)

Comando.Parameters.Add(New SqlParameter("@Usuario",
SqlDbType.VarChar))

Comando.Parameters.Add(New SqlParameter("@Password",
SqlDbType.VarChar))

Comando.Parameters("@Usuario").Value = Usuario

Comando.Parameters("@Password").Value = Password

Datos = Comando.ExecuteReader() ' obtener DataReader .
la sentencia a ejecutar devuelve un conjunto de registros

If Datos.Read() Then

'Leemos todos lo campos de un registro de la
tabla usuario

Id = Datos("idusuario")

TB_Tipo_Usuario.Text = Datos("tipo_usuario")

TB_Nombre.Text = Datos("nombre")

TB_NIF.Text = Datos("nif")
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
TB_Localidad.Text = Datos("localidad")
TB_Codigo_Postal.Text = Datos("codpostal")
TB_Email.Text = Datos("email")
TB_Login.Text = Datos("login")
TB_Password.Text = Datos("password")
TB_Apellidos.Text = Datos("apellidos")
TB_Domicilio.Text = Datos("domicilio")
TB_Provincia.Text = Datos("provincia")
TB_Telefono.Text = Datos("telefono")
Fecha1 = Datos("fechaalta")
Fecha2 = Datos("ultimoacceso")
TB_Ultimo_acceso.Text = Fecha2.ToString("G")
Tb_Fecha_alta.Text = Fecha1.ToLongDateString

End If

Datos.Close()

'liberamos recursos
Datos = Nothing
Comando = Nothing

Catch ex As SqlException

    ' Si hay un error en el acceso a los datos volvemos a
    mostrar el formulario de acceso

    MessageBox.Show("ERROR 3:" & ControlChars.CrLf &
ex.Message & _
ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

    Form_acceso.Show()

    Me.Close()

End Try
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
'Actualizamos el objeto usuario con los datos leídos de la
base de datos
```

```
Objeto_Usuario.Id_valor = Id
```

```
Objeto_Usuario.Tipo_Usuario_valor =
TB_Tipo_Usuario.Text.Trim
```

```
Objeto_Usuario.Nombre_valor = TB_Nombre.Text.Trim
```

```
Objeto_Usuario.NIF_valor = TB_NIF.Text.Trim
```

```
Objeto_Usuario.Localidad_valor = TB_Localidad.Text.Trim
```

```
Objeto_Usuario.Codpostal_valor =
TB_Codigo_Postal.Text.Trim
```

```
Objeto_Usuario.Email_valor = TB_Email.Text.Trim
```

```
Objeto_Usuario.Fechaalta_valor = Fecha1
```

```
Objeto_Usuario.Apellidos_valor = TB_Apellidos.Text.Trim
```

```
Objeto_Usuario.Domicilio_valor = TB_Domicilio.Text.Trim
```

```
Objeto_Usuario.Provincia_valor = TB_Provincia.Text.Trim
```

```
Objeto_Usuario.Telefono_valor = TB_Telefono.Text.Trim
```

```
Objeto_Usuario.Ultimoacceso_valor = Fecha2
```

```
Try
```

```
'Hacemos una consulta a la base de datos para
averiguar que reservas tiene hechas el cliente
```

```
'La consulta "Consulta_SQL" devuelve todos campos de
la tabla Reserva
```

```
'donde el campo cliente_id es igual al identificado de
usuario
```

```
Consulta_SQL = "SELECT * FROM Reserva WHERE cliente_id
= @cliente_id"
```

```
Comando = New SqlCommand(Consulta_SQL, Conexion)
```

```
Comando.Parameters.Add(New SqlParameter("@cliente_id",
SqlDbType.BigInt))
```

```
Comando.Parameters("@cliente_id").Value =
Objeto_Usuario.Id_valor
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
'crear adaptador
oDataAdapter = New SqlDataAdapter(Comando)

'crear commandbuilder
oCB = New SqlCommandBuilder(oDataAdapter)

'crear dataset
'El dataset es un almacen de datos.Representa una base
de datos en memoria y desconectada del proveedor de datos,
'que contiene tablas y sus relaciones
oDataSet = New DataSet()
oDataAdapter.Fill(oDataSet, "Reservas")

'asignar Dataset al DataGridView
Me.DG_Reservas.DataSource = oDataSet
Me.DG_Reservas.DataMember = "Reservas"

'Modificamos propiedades por codigo del DataGridView
DG_Reservas.Un DataGridView es un control nos permite realizar
'enlace complejo de datos con ADO.NET
Me.DG_Reservas.Columns(0).Visible = False 'Oculta la
columna 0 del DataGridView DG_Reservas
Me.DG_Reservas.Columns(7).Visible = False
Me.DG_Reservas.Columns(8).Visible = False
Me.DG_Reservas.Columns(9).Visible = False
Me.DG_Reservas.Columns(3).MinimumWidth = 120 'Ancho
minimo de la columna: 120 pixeles
Me.DG_Reservas.Columns(5).MinimumWidth = 230
Me.DG_Reservas.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill 'Ajustar ancho de la columna

Catch ex As SqlException
```


Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
' Si hay un error en el acceso a los datos volvemos a
mostrar el formulario de acceso

        MessageBox.Show("ERROR 4:" & ControlChars.CrLf &
ex.Message & _
        ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor")

        Form_acceso.Show()

        Me.Close()

    End Try

End If

Resp_Datos_Usuario2.Text = ""

TB_Localidad_Hotel.Enabled = False

CB_Hotel2.Enabled = False

CB_Hotel1.Enabled = False

CB_Tipo_Habitacion.Enabled = False

B_Reserva.Enabled = False

Try

    'Hacemos una consulta a la base de datos para averiguar
los hoteles dados de alta en el sistema

    'La consulta "Consulta_SQL" devuelve el campo nombre de la
tabla Hotel

    Consulta_SQL = "SELECT nombre FROM Hotel"

    Comando = New SqlCommand(Consulta_SQL, Conexion)

    Datos = Comando.ExecuteReader()

    While Datos.Read()

        CB_Hotel1.Items.Add(Datos("nombre"))

    End While

    Datos.Close()
```



```
Datos = Nothing

Comando = Nothing

Catch ex As SqlException

    ' Si hay un error en el acceso a los datos volvemos a
    mostrar el formulario de acceso

    MessageBox.Show("ERROR 5:" & ControlChars.CrLf &
ex.Message & _

    ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

    Form_acceso.Show()

    Me.Close()

End Try

End Sub

' Evento que se produce antes de cerrar el formulario
"Form_usuario"

Private Sub Form_usuario_FormClosing(ByVal sender As
System.Object, ByVal e As System.Windows.Forms.FormClosingEventArgs)
Handles MyBase.FormClosing

    Application.Exit() ' Cierra todas las ventanas de la
aplicacion una vez procesados todos los mensajes

End Sub
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Private Sub TB_Nombre_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TB_Nombre.Click
```

```
    TB_Nombre.Text = ""
```

```
End Sub
```

```
Private Sub TB_Apellidos_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TB_Apellidos.Click
```

```
    TB_Apellidos.Text = ""
```

```
End Sub
```

```
Private Sub TB_NIF_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TB_NIF.Click
```

```
    TB_NIF.Text = ""
```

```
End Sub
```

```
Private Sub TB_Domicilio_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TB_Domicilio.Click
```

```
    TB_Domicilio.Text = ""
```

```
End Sub
```

```
Private Sub TB_Codigo_Postal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TB_Codigo_Postal.Click
```

```
    TB_Codigo_Postal.Text = ""
```

```
End Sub
```

```
Private Sub TB_Email_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TB_Email.Click
```

```
    TB_Email.Text = ""
```

```
End Sub
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Private Sub Tb_Fecha_alta_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Tb_Fecha_alta.Click  
  
    Tb_Fecha_alta.Text = ""  
  
End Sub  
  
Private Sub TB_Login_Click(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles TB_Login.Click  
  
    TB_Login.Text = ""  
  
End Sub  
  
Private Sub TB_Password_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles TB_Password.Click  
  
    TB_Password.Text = ""  
  
End Sub  
  
Private Sub TB_Localidad_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles TB_Localidad.Click  
  
    TB_Localidad.Text = ""  
  
End Sub  
  
Private Sub TB_Provincia_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles TB_Provincia.Click  
  
    TB_Provincia.Text = ""  
  
End Sub  
  
Private Sub TB_Telefono_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles TB_Telefono.Click  
  
    TB_Telefono.Text = ""
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
End Sub
```

```
Private Sub TB_Ultimo_acceso_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles TB_Ultimo_acceso.Click
```

```
    TB_Ultimo_acceso.Text = ""
```

```
End Sub
```

```
Private Sub TB_Nombre_TextChanged(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles TB_Nombre.TextChanged
```

```
    Objeto_Usuario.Nombre_valor = TB_Nombre.Text.Trim
```

```
    If Registrarse = True Then
```

```
        Resp_Datos_Usuario2.Text = "Rellene los campos y pulse  
GUARDAR almacenar los datos"
```

```
    Else
```

```
        Resp_Datos_Usuario2.Text = "Campo modificado pulse  
MODIFICAR almacenar los cambios"
```

```
    End If
```

```
End Sub
```

```
Private Sub TB_Apellidos_TextChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
TB_Apellidos.TextChanged
```

```
    Objeto_Usuario.Apellidos_valor = TB_Apellidos.Text.Trim
```

```
    If Registrarse = True Then
```

```
        Resp_Datos_Usuario2.Text = "Rellene los campos y pulse  
GUARDAR almacenar los datos"
```

```
    Else
```

```
        Resp_Datos_Usuario2.Text = "Campo modificado pulse  
MODIFICAR almacenar los cambios"
```

```
    End If
```

```
End Sub
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Private Sub TB_NIF_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TB_NIF.TextChanged

    Objeto_Usuario.NIF_valor = TB_NIF.Text.Trim

    If Registrarse = True Then

        Resp_Datos_Usuario2.Text = "Rellene los campos y pulse
GUARDAR almacenar los datos"

    Else

        Resp_Datos_Usuario2.Text = "Campo modificado pulse
MODIFICAR almacenar los cambios"

    End If

End Sub

Private Sub TB_Login_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TB_Login.TextChanged

    Objeto_Usuario.Login_valor = TB_Login.Text.Trim

    If Registrarse = True Then

        Resp_Datos_Usuario2.Text = "Rellene los campos y pulse
GUARDAR almacenar los datos"

    Else

        Resp_Datos_Usuario2.Text = "Campo modificado pulse
MODIFICAR almacenar los cambios"

    End If

End Sub

Private Sub TB_Password_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TB_Password.TextChanged

    Objeto_Usuario.Password_valor = TB_Password.Text.Trim

    If Registrarse = True Then

        Resp_Datos_Usuario2.Text = "Rellene los campos y pulse
GUARDAR almacenar los datos"

    Else

        Resp_Datos_Usuario2.Text = "Campo modificado pulse
MODIFICAR almacenar los cambios"

    End If

End Sub
```



```
End Sub
```

```
Private Sub TB_Domicilio_TextChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
TB_Domicilio.TextChanged  
  
Objeto_Usuario.Domicilio_valor = TB_Domicilio.Text.Trim  
  
If Registrarse = True Then  
  
Resp_Datos_Usuario2.Text = "Rellene los campos y pulse  
GUARDAR almacenar los datos"  
  
Else  
  
Resp_Datos_Usuario2.Text = "Campo modificado pulse  
MODIFICAR almacenar los cambios"  
  
End If  
  
End Sub
```

```
Private Sub TB_Localidad_TextChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
TB_Localidad.TextChanged  
  
Objeto_Usuario.Localidad_valor = TB_Localidad.Text.Trim  
  
If Registrarse = True Then  
  
Resp_Datos_Usuario2.Text = "Rellene los campos y pulse  
GUARDAR almacenar los datos"  
  
Else  
  
Resp_Datos_Usuario2.Text = "Campo modificado pulse  
MODIFICAR almacenar los cambios"  
  
End If  
  
End Sub
```

```
Private Sub TB_Codigo_Postal_TextChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
TB_Codigo_Postal.TextChanged  
  
Objeto_Usuario.Codpostal_valor = TB_Codigo_Postal.Text.Trim  
  
If Registrarse = True Then  
  
Resp_Datos_Usuario2.Text = "Rellene los campos y pulse  
GUARDAR almacenar los datos"
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Else

    Resp_Datos_Usuario2.Text = "Campo modificado pulse
MODIFICAR almacenar los cambios"

End If

End Sub

Private Sub TB_Provincia_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TB_Provincia.TextChanged

    Objeto_Usuario.Provincia_valor = TB_Provincia.Text.Trim

    If Registrarse = True Then

        Resp_Datos_Usuario2.Text = "Rellene los campos y pulse
GUARDAR almacenar los datos"

    Else

        Resp_Datos_Usuario2.Text = "Campo modificado pulse
MODIFICAR almacenar los cambios"

    End If

End Sub

Private Sub TB_Email_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TB_Email.TextChanged

    Objeto_Usuario.Email_valor = TB_Email.Text.Trim

    If Registrarse = True Then

        Resp_Datos_Usuario2.Text = "Rellene los campos y pulse
GUARDAR almacenar los datos"

    Else

        Resp_Datos_Usuario2.Text = "Campo modificado pulse
MODIFICAR almacenar los cambios"

    End If

End Sub

Private Sub TB_Telefono_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TB_Telefono.TextChanged

    Objeto_Usuario.Telefono_valor = TB_Telefono.Text.Trim
```


Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
If Registrarse = True Then

    Resp_Datos_Usuario2.Text = "Rellene los campos y pulse
GUARDAR almacenar los datos"

Else

    Resp_Datos_Usuario2.Text = "Campo modificado pulse
MODIFICAR almacenar los cambios"

End If

End Sub

'Evento que se produce al pulsar sobre el boton Salir

Private Sub TSB_Salir_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TSB_Salir.Click

    Dim Result As DialogResult

    Result = MessageBox.Show("¿Seguro que desea salir de la
aplicación?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

    If Result = DialogResult.Yes Then

        Me.Close()

    End If

End Sub

' Evento que se produce al pulsar sobre el boton "Editar" de la
barra de herramientas

Private Sub TSB_Editar_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TSB_Editar.Click

    Panel_Usuario.Enabled = True 'Habilito el panel de usuario
para modificacion

    TB_Ultimo_acceso.Enabled = False ' Deshabilito TextBox
porque esde solo lectura

    Tb_Fecha_alta.Enabled = False ' Deshabilito TextBox porque
esde solo lectura

    TB_Tipo_Usuario.Enabled = False ' Deshabilito TextBox porque
esde solo lectura
```



```

If Registrarse = True Then

    Resp_Datos_Usuario2.Visible = True

    Resp_Datos_Usuario2.Text = "Rellene los campos y pulse
GUARDAR almacenar los datos"

    'Si esta en modo registrarse habilito el boton guardar
    TSB_Guardar.Enabled = True

    TSB_Modificar.Enabled = False

Else

    ' Si no estamos en modo registro el login no se puede
    modificar

    TB_Login.Enabled = False

    Resp_Datos_Usuario2.Visible = True

    Resp_Datos_Usuario2.Text = "Rellene los campos y pulse
GUARDAR almacenar los datos"

    'Si no esta en modo registrarse deshabilito el boton
    guardar y habilito el boton modificar

    TSB_Guardar.Enabled = False

    TSB_Modificar.Enabled = True

End If

    editable = True ' Variable global "editable", indica si
    podemos editar o no la infoemacion de usuario

    TSB_Editar.Enabled = False 'Deshabilitamos el boton editar

End Sub

    ' Evento que se produce al pulsar sobre el boton "Guardar" de la
    barra de herramientas

    Private Sub TSB_Guardar_Click(ByVal sender As System.Object, ByVal
    e As System.EventArgs) Handles TSB_Guardar.Click

        'Variables para la conexión y consulta a la base de datos

        Dim Resultado As Integer = 0

        Dim Datos As SqlDataReader

```

```
'Comprobamos que los datos obligatorios estan introducidos
If TB_Nombre.Text = "" Then
    Resp_Datos_Usuario.Text = "Porfavor introduzca Nombre"
ElseIf TB_Apellidos.Text = "" Then
    Resp_Datos_Usuario.Text = "Porfavor introduzca Apellidos"
ElseIf TB_NIF.Text = "" Then
    Resp_Datos_Usuario.Text = "Porfavor introduzca NIF"
ElseIf TB_Login.Text = "" Then
    Resp_Datos_Usuario.Text = "Porfavor introduzca Login"
ElseIf TB_Password.Text = "" Then
    Resp_Datos_Usuario.Text = "Porfavor introduzca Password"
Else
    Try
        'Hacemos una consulta a la base de datos para
        averiguar si ya hay un usuario con el mismo login

        'Crear comando escalar

        'La consulta "Consulta_SQL" devuelve el numero de
        filas de la tabla Usuario

        'donde el campo login es igual al login introducido
        por teclado

        Consulta_SQL = "SELECT count(*) FROM Usuario WHERE
        login = @Usuario"

        'Crear comando

        Comando = New SqlCommand(Consulta_SQL, Conexion)

        ' Añadir parametros al comando

        Comando.Parameters.Add(New SqlParameter("@Usuario",
        SqlDbType.VarChar))
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Comando.Parameters("@Usuario").Value =
TB_Login.Text.Trim

Resultado = Comando.ExecuteNonQuery() 'Ejecuta comando.
Devuelve la primera columna de la primera fila del conjunto de
registros

'el resto de datos nos se tendran en cuenta

Comando = Nothing

If (Resultado > 0) Then ' Hay un usuario con el mismo
login

    MessageBox.Show("EL LOGIN YA EXISTE --> INTRODUCZA
NUEVO LOGIN")

    Exit Sub ' Salimos del procedimiento

End If

Catch ex As SqlException

    ' Si hay un error en el acceso a los datos mostramos
mensaje de error y salimos del procedimiento

    MessageBox.Show("ERROR 6:" & ControlChars.CrLf &
ex.Message & _
ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

    MessageBox.Show("DATOS DE USUARIO NO GUARDADOS")

Exit Sub ' Salimos del procedimiento TSB_Guardar_Click
```

```
End Try
```

```
Try
```

```
    'Hacemos una insercion en la base de datos de los
datos de informacion de usuario introducidos por teclado

    'Crear comando escalar

    'La consulta "Consulta_SQL" Insertamos un registro en
la tabla usuario con los datos introducidos por el usuario en el
formulario

    Consulta_SQL = "INSERT INTO " & _

"Usuario(tipo_usuario,login,password,email,fechaalta,ultimoacceso,nomb
re,apellidos,nif,domicilio,localidad,codpostal,provincia,telefono) " &
-
    "VALUES
(@tipo_usuario,@login,@password,@email,@fechaalta,@ultimoacceso,@nombr
e,@apellidos,@nif,@domicilio,@localidad,@codpostal,@provincia,@telefon
o) "

    'Crear comando

    Comando = New SqlCommand(Consulta_SQL, Conexion)

    ' Añadir parametros al comando

    Comando.Parameters.Add(New
SqlParameter("@tipo_usuario", SqlDbType.VarChar))

    Comando.Parameters.Add(New SqlParameter("@login",
SqlDbType.VarChar))

    Comando.Parameters.Add(New SqlParameter("@password",
SqlDbType.VarChar))

    Comando.Parameters.Add(New SqlParameter("@email",
SqlDbType.VarChar))
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Comando.Parameters.Add(New SqlParameter("@fechaalta",
SqlDbType.DateTime))

Comando.Parameters.Add(New
SqlParameter("@ultimoacceso", SqlDbType.DateTime))

Comando.Parameters.Add(New SqlParameter("@nombre",
SqlDbType.VarChar))

Comando.Parameters.Add(New SqlParameter("@apellidos",
SqlDbType.VarChar))

Comando.Parameters.Add(New SqlParameter("@nif",
SqlDbType.VarChar))

Comando.Parameters.Add(New SqlParameter("@domicilio",
SqlDbType.VarChar))

Comando.Parameters.Add(New SqlParameter("@localidad",
SqlDbType.VarChar))

Comando.Parameters.Add(New SqlParameter("@provincia",
SqlDbType.VarChar))

Comando.Parameters.Add(New SqlParameter("@codpostal",
SqlDbType.VarChar))

Comando.Parameters.Add(New SqlParameter("@telefono",
SqlDbType.VarChar))

'Actulizamos el objeto usuario con los datos obtenidos
en el formulario

Objeto_Usuario.Fechaalta_valor =
Convert.ToDateTime(Tb_Fecha_alta.Text)

Objeto_Usuario.Ultimoacceso_valor = DateTime.Now

Objeto_Usuario.Tipo_Usuario_valor =
TB_Tipo_Usuario.Text.Trim

Objeto_Usuario.Nombre_valor = TB_Nombre.Text.Trim

Objeto_Usuario.NIF_valor = TB_NIF.Text.Trim

Objeto_Usuario.Localidad_valor =
TB_Localidad.Text.Trim

Objeto_Usuario.Codpostal_valor =
TB_Codigo_Postal.Text.Trim

Objeto_Usuario.Email_valor = TB_Email.Text.Trim

Objeto_Usuario.Apellidos_valor =
TB_Apellidos.Text.Trim
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Objeto_Usuario.Domicilio_valor =
TB_Domicilio.Text.Trim

Objeto_Usuario.Provincia_valor =
TB_Provincia.Text.Trim

Objeto_Usuario.Telefono_valor = TB_Telefono.Text.Trim

Objeto_Usuario.Login_valor = TB_Login.Text.Trim

Objeto_Usuario.Password_valor = TB_Password.Text.Trim

' Doy valores a los parametros

Comando.Parameters("@tipo_usuario").Value =
Objeto_Usuario.Tipo_Usuario_valor

Comando.Parameters("@login").Value =
Objeto_Usuario.Login_valor

Comando.Parameters("@password").Value =
Objeto_Usuario.Password_valor

Comando.Parameters("@email").Value =
Objeto_Usuario.Email_valor

Comando.Parameters("@fechaalta").Value =
Objeto_Usuario.Fechaalta_valor

Comando.Parameters("@ultimoacceso").Value =
Objeto_Usuario.Ultimoacceso_valor

Comando.Parameters("@nombre").Value =
Objeto_Usuario.Nombre_valor

Comando.Parameters("@apellidos").Value =
Objeto_Usuario.Apellidos_valor

Comando.Parameters("@nif").Value =
Objeto_Usuario.NIF_valor

Comando.Parameters("@domicilio").Value =
Objeto_Usuario.Domicilio_valor

Comando.Parameters("@localidad").Value =
Objeto_Usuario.Localidad_valor

Comando.Parameters("@provincia").Value =
Objeto_Usuario.Provincia_valor

Comando.Parameters("@codpostal").Value =
Objeto_Usuario.Codpostal_valor

Comando.Parameters("@telefono").Value =
Objeto_Usuario.Telefono_valor
```



```

Resultado = Comando.ExecuteNonQuery()

Comando = Nothing

' Hacemos una nueva consulta a la base de datos para
averiguar el identificador de usuario que se ha creado

' al insertar un registro en la tabla usuario

'La consulta "Consulta_SQL" devuelve el identificador
de usuario de un registro de la tabla usuario

'donde el campo login y el campo password es igual al
login y password respectivamente introducidos en el formulario

Consulta_SQL = "SELECT idusuario FROM Usuario WHERE
login = @login and password = @Password"

Comando = New SqlCommand(Consulta_SQL, Conexion)

Comando.Parameters.Add(New SqlParameter("@login",
SqlDbType.VarChar))

Comando.Parameters.Add(New SqlParameter("@password",
SqlDbType.VarChar))

Comando.Parameters("@login").Value =
Objeto_Usuario.Login_valor

Comando.Parameters("@password").Value =
Objeto_Usuario.Password_valor

Datos = Comando.ExecuteReader()

If Datos.Read() Then

    Id = Datos("idusuario") 'Asignamos a Id el calor
del campo idusuario de la tabla usuario

End If

Datos.Close()

```



```
Objeto_Usuario.Id_valor = Id

Comando = Nothing

Datos = Nothing

Catch ex As SqlException

    ' Si hay un error en el acceso a los datos salimos del
procedimiento

    MessageBox.Show("ERROR 7:" & ControlChars.CrLf &
ex.Message & _
ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

    Resp_Datos_Usuario.Text = ""

    MessageBox.Show("DATOS DE USUARIO NO GUARDADOS")

Exit Sub

End Try

' Si no hay un error en el acceso a los datos

MessageBox.Show("DATOS DE USUARIO GUARDADOS
CORRECTAMENTE")

TSB_Editar.Enabled = True 'Habilito el boton Editar

TSB_Guardar.Enabled = False 'Deshabilito el boton Guardar

TSB_Modificar.Enabled = False 'Deshabilito el boton
Modificar

TSB_Baja.Enabled = True 'Habilito el boton baja

Resp_Datos_Usuario2.Text = ""
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Resp_Datos_Usuario2.Visible = True

Panel_Usuario.Enabled = False 'Deshabilito el panel de
usuario

    TabControl_Usuario.TabPages.Add(TP_Reserva) ' Habilitamos
la pestaña Reserva

    TabControl_Usuario.TabPages.Add(TP_List_Res) ' Habilitamos
la pestaña Listado de reservas

    Registrarse = False

    Resp_Datos_Usuario2.Text = ""

End If

End Sub

' Evento que se produce al pulsar sobre el boton "Modificar" de la
barra de herramientas

Private Sub TSB_Modificar_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TSB_Modificar.Click

    'Variables para la conexión y consulta a la base de datos

    Dim Resultado As Integer

    Dim Datos As SqlDataReader

    If Descarga_reserva = False Then ' Si el formulario
Form_reserva esta cargado

        Form_Reserva.Close() ' Cerramos el formulario

    End If

    If list_reservas = True Then ' Si estoy en la pestaña listado
de reservas
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
If Me.DG_Reservas.CurrentRow.Selected = False Then 'Si no
he seleccionado ninguna fila del DataGridView DG_Reservas

    MessageBox.Show("Porfavor seleccione una reserva para
modificar o borrar")

    Exit Sub

    ElseIf DateDiff(DateInterval.Day, Now.Date,
CDate(DG_Reservas.CurrentRow.Cells(1).Value)) <= 0 Then

        'Si la fecha de hoy es posterior a fecha de inicio de
la reserva -> no se puede modificar la reserva

        MessageBox.Show("Seleccione una reserva para modificar
que tenga una fecha de inicio posterior a la fecha de hoy")

        Exit Sub

    Else

        ' Actulizamos el objeto reserva con los valores del
DataGridView DG_Reservas

        Objeto_Reserva.idreserva_valor =
Me.DG_Reservas.CurrentRow.Cells(0).Value

        Objeto_Reserva.fechainicio_valor =
Me.DG_Reservas.CurrentRow.Cells(1).Value

        Objeto_Reserva.fechafin_valor =
Me.DG_Reservas.CurrentRow.Cells(2).Value

        Objeto_Reserva.precio_valor =
CSng(Me.DG_Reservas.CurrentRow.Cells(3).Value)

        Objeto_Reserva.ocupacion_valor =
Me.DG_Reservas.CurrentRow.Cells(4).Value

        Objeto_Reserva.nombretomador_valor =
Me.DG_Reservas.CurrentRow.Cells(5).Value

        Objeto_Reserva.cantidad_valor =
Me.DG_Reservas.CurrentRow.Cells(6).Value

        Objeto_Reserva.cliente_id_valor =
Me.DG_Reservas.CurrentRow.Cells(7).Value

        Objeto_Reserva.hotel_id_valor =
Me.DG_Reservas.CurrentRow.Cells(8).Value

        Objeto_Reserva.habitacion_id_valor =
Me.DG_Reservas.CurrentRow.Cells(9).Value
```



Try

```

        'Hacemos una consulta a la base de datos para
        averiguar datos sobre el tipo de habitacion de la reserva

        ' La consulta "Consulta_SQL" realiza la operacion
        INNER JOIN, la cual realiza una combinacion con todos aquellos

        ' registros de las dos tablas (Habitacion y
        TipoHabitacion) en el que el campo comun de ambas (identificador de
        habitacion)

        ' tenga el mismo valor (el valor del identificador
        de la habitacion) y donde el campo habitacion_id de la tabla

        ' TipoHabitacion tenga el valor del indentificador
        de habitacion de la reserva y el hotel_id tenga el valor del hotel de
        la reserva

        ' Seleccionamos los campos de la tabla Habitacion
        y TipoHabitacion.

        Consulta_SQL = "SELECT
        TipoHabitacion.hotel_id,TipoHabitacion.habitacion_id,TipoHabitacion.de
        scripcion,TipoHabitacion.precio," & _

        "TipoHabitacion.numhabitaciones,Habitacion.nombre,Habitacion.capacidad
        FROM Habitacion INNER JOIN TipoHabitacion ON " & _

        "TipoHabitacion.habitacion_id =
        Habitacion.idhabitacion WHERE TipoHabitacion.habitacion_id =
        @idhabitacion and TipoHabitacion.hotel_id = @idhotel"

        Comando = New SqlCommand(Consulta_SQL, Conexion)

        Comando.Parameters.Add(New
        SqlParameter("@idhabitacion", SqlDbType.BigInt))

        Comando.Parameters.Add(New
        SqlParameter("@idhotel", SqlDbType.BigInt))

        Comando.Parameters("@idhabitacion").Value =
        Objeto_Reserva.habitacion_id_valor

        Comando.Parameters("@idhotel").Value =
        Objeto_Reserva.hotel_id_valor
    
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Datos = Comando.ExecuteReader()

If Datos.Read() Then

    ' Actualizamos el objeto habitacion

    Objeto_Habitacion.Hotel_id_valor =

Datos("hotel_id")

    Objeto_Habitacion.Idtipohabitacion_valor =

Datos("habitacion_id")

    Objeto_Habitacion.Descripcion_valor =

Datos("descripcion")

    Objeto_Habitacion.Precio_valor =

Datos("precio")

    Objeto_Habitacion.Numhabitaciones_valor =

Datos("numhabitaciones")

    Objeto_Habitacion.Nombre_valor =

Datos("nombre")

    Objeto_Habitacion.Capacidad_valor =

Datos("capacidad")

End If

Datos.Close()

Comando = Nothing

Datos = Nothing

Catch ex As SqlException

    ' Si hay un error en el acceso a los datos
volvemos a mostrar el formulario de acceso

    MessageBox.Show("ERROR 8:" & ControlChars.CrLf &
ex.Message & _

        ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008
R2 Express. Miguel Monzo Pastor ")

    MessageBox.Show("ERROR EN EL ACCESO A LA BASE DE
DATOS, INTENTELO DE NUEVO")

Exit Sub 'Salimos del procedimiento
```



```

        'Form_acceso.Show()

        'Me.Close()

    End Try

    Try

        'Hacemos una consulta a la base de datos para
        averiguar datos sobre el hotel donde se ha realizado la reserva

        ' La consulta "Consulta_SQL" realiza la operacion
        INNER JOIN, la cual realiza una combinacion con todos aquellos

        ' registros de las dos tablas (Hotel y
        TipoHabitacion) en el que el campo comun de ambas (identificador de
        hotel)

        ' tenga el mismo valor (el valor del identificador
        de hotel de la reserva) y donde el campo hotel_id de la tabla

        ' TipoHabitacion sea igual al identificador del
        hotel

        ' Seleccionamos los campos de la tabla hotel.

        Consulta_SQL = "SELECT Hotel.idhotel,Hotel.nombre
        ,Hotel.descripcion,Hotel.categoria," & _

        "Hotel.domicilio,Hotel.localidad,Hotel.provincia,Hotel.codpostal,Hotel
        .telefono FROM Hotel INNER JOIN TipoHabitacion ON " & _

        "Hotel.idhotel = TipoHabitacion.hotel_id WHERE
        TipoHabitacion.hotel_id = @hotel_id"

        Comando = New SqlCommand(Consulta_SQL, Conexion)

        Comando.Parameters.Add(New
        SqlParameter("@hotel_id", SqlDbType.BigInt))

        Comando.Parameters("@hotel_id").Value =
        Objeto_Habitacion.Hotel_id_valor

        Datos = Comando.ExecuteReader()

        If Datos.Read() Then

            ' Actualizamos el objeto Hotel

            Hotel.idhotel_valor = Datos("idhotel")

            Hotel.nombre_valor = Datos("nombre")

            Hotel.descripcion_valor = Datos("descripcion")
    
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
        Hotel.categoria_valor = Datos("categoria")

        Hotel.direccion_valor = Datos("domicilio") &
", " & Datos("localidad") _
        & " " & "(" & Datos("provincia") & ")" & ", "
& Datos("codpostal")

        Hotel.telefono_valor = Datos("telefono")

End If

Datos.Close()

Datos = Nothing

Comando = Nothing

hotel2 = Hotel.nombre_valor

idhotel1 = Hotel.idhotel_valor

hotel1 = Hotel.nombre_valor

Catch ex As SqlException

    ' Si hay un error en el acceso a los datos
volvemos a mostrar el formulario de acceso

    MessageBox.Show("ERROR 9:" & ControlChars.CrLf &
ex.Message & _
        ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008
R2 Express. Miguel Monzo Pastor ")

    MessageBox.Show("ERROR EN EL ACCESO A LA BASE DE
DATOS, INTENTELO DE NUEVO")

Exit Sub 'Salimos del procedimiento

'Form_acceso.Show()

'Me.Close()

End Try

MessageBox.Show("REALICE UNA CONSULTA PARA VER
DISPONIBILIDAD DEL HOTEL")

Ver_reserva = 1 'Indica se muestra el formulario
Form_Reserva desde la pestaña listado de reservas
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
' Deshabilitamos los botones Guardar, Modificar y
Borrar del formulario Form_Reserva

Form_Reserva.TSB_Guardar_Reserva.Enabled = False

Form_Reserva.TSB_Modificar_Reserva.Enabled = False

Form_Reserva.TSB_Borrar_Reserva.Enabled = False

' Mostramos el formulario de Reservas

Form_Reserva.Visible = True

Form_Reserva.Show()

End If

Else

' Estoy en la pestaña usuario

If TB_Nombre.Text = "" Then

    Resp_Datos_Usuario.Text = "Porfavor introduzca Nombre"

ElseIf TB_Apellidos.Text = "" Then

    Resp_Datos_Usuario.Text = "Porfavor introduzca
Apellidos"

ElseIf TB_NIF.Text = "" Then

    Resp_Datos_Usuario.Text = "Porfavor introduzca NIF"

ElseIf TB_Login.Text = "" Then

    Resp_Datos_Usuario.Text = "Porfavor introduzca Login"

ElseIf TB_Password.Text = "" Then

    Resp_Datos_Usuario.Text = "Porfavor introduzca
Password"

Else

' Todos los campos obligatorios introducidos

Try
```


Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
'Hacemos una actualizacion de la base de datos
concretamente en la tabla usuario

' La consulta "Consulta_SQL" realiza una
actualizacion de un registro de la tabla usuario donde

' el identificador de usuario es el del usuario
actual

Consulta_SQL = "UPDATE Usuario " & _
               "SET tipo_usuario = @tipo_usuario,login =
@login,password = @password,email = @email,fechaalta = @fechaalta," &
_
               "ultimoacceso = @ultimoacceso,nombre =
@nombre,apellidos = @apellidos,nif = @nif,domicilio = @domicilio," & _
               "localidad = @localidad,codpostal =
@codpostal,provincia = @provincia,telefono = @telefono " & _
               "WHERE idusuario = @idusuario;"

'Crear comando
Comando = New SqlCommand(Consulta_SQL, Conexion)

' Añadir parametros al comando

Comando.Parameters.Add(New
SqlParameter("@tipo_usuario", SqlDbType.VarChar))

Comando.Parameters.Add(New SqlParameter("@login",
SqlDbType.VarChar))

Comando.Parameters.Add(New
SqlParameter("@password", SqlDbType.VarChar))

Comando.Parameters.Add(New SqlParameter("@email",
SqlDbType.VarChar))

Comando.Parameters.Add(New
SqlParameter("@fechaalta", SqlDbType.DateTime))

Comando.Parameters.Add(New
SqlParameter("@ultimoacceso", SqlDbType.DateTime))
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Comando.Parameters.Add(New SqlParameter("@nombre",
SqlDbType.VarChar))

Comando.Parameters.Add(New
SqlParameter("@apellidos", SqlDbType.VarChar))

Comando.Parameters.Add(New SqlParameter("@nif",
SqlDbType.VarChar))

Comando.Parameters.Add(New
SqlParameter("@domicilio", SqlDbType.VarChar))

Comando.Parameters.Add(New
SqlParameter("@localidad", SqlDbType.VarChar))

Comando.Parameters.Add(New
SqlParameter("@provincia", SqlDbType.VarChar))

Comando.Parameters.Add(New
SqlParameter("@codpostal", SqlDbType.VarChar))

Comando.Parameters.Add(New
SqlParameter("@telefono", SqlDbType.VarChar))

Comando.Parameters.Add(New
SqlParameter("@idusuario", SqlDbType.BigInt))

' Doy valores a los parametros

Objeto_Usuario.Fechaalta_valor =
Convert.ToDateTime(Tb_Fecha_alta.Text)

Objeto_Usuario.Ultimoacceso_valor = DateTime.Now

Comando.Parameters("@tipo_usuario").Value =
Objeto_Usuario.Tipo_Usuario_valor

Comando.Parameters("@login").Value =
Objeto_Usuario.Login_valor

Comando.Parameters("@password").Value =
Objeto_Usuario.Password_valor

Comando.Parameters("@email").Value =
Objeto_Usuario.Email_valor

Comando.Parameters("@fechaalta").Value =
Objeto_Usuario.Fechaalta_valor

Comando.Parameters("@ultimoacceso").Value =
Objeto_Usuario.Ultimoacceso_valor

Comando.Parameters("@nombre").Value =
Objeto_Usuario.Nombre_valor
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Comando.Parameters("@apellidos").Value =
Objeto_Usuario.Apellidos_valor

Comando.Parameters("@nif").Value =
Objeto_Usuario.NIF_valor

Comando.Parameters("@domicilio").Value =
Objeto_Usuario.Domicilio_valor

Comando.Parameters("@localidad").Value =
Objeto_Usuario.Localidad_valor

Comando.Parameters("@provincia").Value =
Objeto_Usuario.Provincia_valor

Comando.Parameters("@codpostal").Value =
Objeto_Usuario.Codpostal_valor

Comando.Parameters("@telefono").Value =
Objeto_Usuario.Telefono_valor

Comando.Parameters("@idusuario").Value =
Objeto_Usuario.Id_valor

Resultado = Comando.ExecuteNonQuery()

Comando = Nothing

Catch ex As SqlException

    ' Si hay un error en el acceso a los datos salimos
del procedimiento

    MessageBox.Show("ERROR 10:" & ControlChars.CrLf &
ex.Message & _
ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

    MessageBox.Show("DATOS DE USUARIO NO MODIFICADOS
CORRECTAMENTE")

Exit Sub ' salir del procedimiento

End Try
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
' Si no hay error en el acceso a la base de datos
MessageBox.Show("DATOS DE USUARIO MODIFICADOS
CORRECTAMENTE")

Resp_Datos_Usuario.Text = ""
Resp_Datos_Usuario2.Text = ""
Panel_Usuario.Enabled = False
TSB_Baja.Enabled = True
TSB_Modificar.Enabled = False
TSB_Guardar.Enabled = False
TSB_Editar.Enabled = True

End If

End If

End Sub

' Evento que se produce al pulsar sobre el boton "Baja" de la
barra de herramientas

Private Sub TSB_Baja_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TSB_Baja.Click

    Dim Resultado As Integer

    Dim Result1 As DialogResult

    Result1 = MessageBox.Show("¿Seguro que desea darse de baja?",
"", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

    If Result1 = DialogResult.Yes Then

        Try

            'Borramos de la base de datos al usuario

            ' La consulta "Consulta_SQL" realiza una borrado de un
registro de la tabla usuario donde
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
' el identificador de usuario es el del usuario actual
Consulta_SQL = "DELETE FROM Usuario WHERE idusuario =
@idusuario"

Comando = New SqlCommand(Consulta_SQL, Conexion)

Comando.Parameters.Add(New SqlParameter("@idusuario",
SqlDbType.BigInt))

Comando.Parameters("@idusuario").Value =
Objeto_Usuario.Id_valor

Resultado = Comando.ExecuteNonQuery()

Catch ex As SqlException

' Si no hay error en el acceso a la base de datos
MessageBox.Show("ERROR 11:" & ControlChars.CrLf &
ex.Message & _
ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor")

MessageBox.Show("NO SE HA PODIDO REALIZAR LA BAJA")

Exit Sub

End Try

Dim Result2 As DialogResult

Result2 = MessageBox.Show("Baja en el sistema realizada
correctamente,pulse ACEPTAR para Salir de la aplicación", "Baja del
sistema", MessageBoxButtons.OK)

If Result2 = DialogResult.OK Then

Application.Exit()

End If

End If

End Sub
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
'Evento que se produce cuando el control "TP_Reserva" es el
elemento activo del formulario

' (cuando hemos selecciona la pestaña Reserva)

Private Sub TP_Reserva_Enter(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TP_Reserva.Enter

    Label7.Text = "" 'Borramos el contenido de la etiqueta de
información

    ' Habilitamos y deshabilitamos los botones correspondientes

    TSB_Guardar.Enabled = False

    TSB_Editar.Enabled = False

    TSB_Baja.Enabled = False

    TSB_Borrar.Enabled = False

    TSB_Modificar.Enabled = False

    If (Ver_reserva = 4) Then 'Si hemos seleccionado una fila en
el DataGrid DG_Reservas

        'Asignamos valores a las propiedades del los controles de
la pestaña "Reserva"

        'Informacion del hotel y de la habitacion reservada

        RB_Nombre_Hotel.Enabled = True

        RB_Localidad_Hotel.Enabled = True

        GB_Busqueda_Hotel.Enabled = True

        CB_Hotel1.Text = hotel1

        B_Reserva.Enabled = False

        TB_Localidad_Hotel.Text = ""
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
GB_Hotel.Enabled = True

TB_Nombre_Hotel.Text = Hotel.nombre_valor

TB_Categoria.Text = Hotel.categoria_valor

TB_Direccion_Hotel.Text = Hotel.direccion_valor

TB_Telefono_Hotel.Text = Hotel.telefono_valor

RTB_Descripcion_Hotel.Text = Hotel.descripcion_valor

RTB_Descripcion_Hotel.ReadOnly = True

GB_Habitacion.Enabled = True

CB_Tipo_Habitacion.Text = Objeto_Habitacion.Nombre_valor

TB_Capacidad.Text = Objeto_Habitacion.Capacidad_valor

TB_Importe_noche.Text = Objeto_Habitacion.Precio_valor

CB_Tipo_Habitacion.Enabled = False

RTB_Descripcion_Habitacion.ReadOnly = False

RTB_Descripcion_Habitacion.Text =
Objeto_Habitacion.Descripcion_valor

RTB_Descripcion_Habitacion.ReadOnly = True

End If

Panel_Usuario.Enabled = False 'Deshabilitamos el Panel de
usuario para que no se pueda modificar

editable = False ' Editable = false indica que no se puede
editar la informacion de usuario

End Sub

'Evento que se produce cuando el control "TP_List_Res_Enter" es el
elemento activo del formulario

' (cuando hemos selecciona la pestaña Listado Reservas)

Private Sub TP_List_Res_Enter(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TP_List_Res.Enter

Label7.Text = ""
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
'Habilitamos y deshabilitamos los botones correspondientes

TSB_Guardar.Enabled = False

TSB_Editar.Enabled = False

TSB_Baja.Enabled = False

list_reservas = True 'list_reservas = true indica que hemos
seleccionado la pestaña listado reservas

If (Ver_reserva = 2) Then 'Si hemos pulsado el Boton B_Reserva
y se ha mostrado el formulario Form_Reserva

    TSB_Modificar.Enabled = False 'Deshabilitamos el boton
Modificar

    TSB_Borrar.Enabled = False      'Deshabilitamos el boton
Borrar

Else

    If Me.DG_Reservas.RowCount = 0 Then

        TSB_Modificar.Enabled = False

        TSB_Borrar.Enabled = False

    Else ' Si hemos seleccionado una fila del DataGridView
DG_Reservas

        TSB_Modificar.Enabled = True 'Habilitamos el boton
Modificar

        TSB_Borrar.Enabled = True      'Habilitamos el boton
Borrar

    End If

End If

End If

B_Reserva.Enabled = False      'Deshabilitamos el boton
B_Reserva

Panel_Usuario.Enabled = False 'Deshabilitamos el Panel de
usuario para que no se pueda modificar

editable = False ' Editable = false indica que no se puede
editar la informacion de usuario

End Sub
```


Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
'Evento que se produce cuando el control "TP_Usuario" es el
elemento activo del formulario

' (cuando hemos selecciona la pestaña Usuario)

Private Sub TP_Usuario_Enter(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TP_Usuario.Enter

    'Habilitamos y deshabilitamos los botones correspondientes

    TSB_Guardar.Enabled = False

    TSB_Editar.Enabled = True

    TSB_Baja.Enabled = True

    TSB_Borrar.Enabled = False

    If editable = True Then ' Editable es true si hemos pulsado el
boton Editar

        TSB_Modificar.Enabled = True 'Habilitamos el boton
Modificar

    Else 'Editable es false sin no hemos pulsado el boton Editar

        TSB_Modificar.Enabled = False 'Deshabilitamos el boton
Modificar

    End If

    list_reservas = False 'list_reservas = false indica que no
hemos seleccionado la pestaña listado reservas

    Label7.Text = "" 'Borramos el contenido de la etiqueta de
información

End Sub

'Evento que se produce cuando la propiedad checked del RadioButton
"RB_Localidad_Hotel" cambia de valor

Private Sub RB_Localidad_Hotel_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RB_Localidad_Hotel.CheckedChanged

    B_Reserva.Enabled = False 'Deshabilitamos el boton B_Reserva

    'Modificamos las propiedades de color del boton B_Reserva
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
B_Reserva.BackColor = SystemColors.Control

B_Reserva.ForeColor = SystemColors.ControlText

If RB_Localidad_Hotel.Checked = True Then ' Si esta
seleccionado el RadioButton RB_Localidad_Hotel

    While CB_Hotel2.Items.Count <> 0 'Borramos todos los items
del ComboBox CB_Hotel2

        CB_Hotel2.Items.RemoveAt(0)

    End While

    CB_Hotel2.Text = "Lista de hoteles"

    TB_Localidad_Hotel.Enabled = True 'Habilitamos el TextBox
TB_Localidad_Hotel

    CB_Hotel2.Enabled = False 'Deshabilitamos el ComboBox
CB_Hotel2

    CB_Hotel1.Enabled = False 'Deshabilitamos el ComboBox
CB_Hotel1

    CB_Hotel1.Text = "Lista de hoteles"

    CB_Tipo_Habitacion.Enabled = False 'Dehabilitamos el
ComboBox CB_Tipo_Habitacion

    TB_Nombre_Hotel.Text = "" 'Borramos el contenido del
TextBox TB_Nombre_Hotel

    TB_Categoria.Text = ""

    TB_Direccion_Hotel.Text = ""

    TB_Telefono_Hotel.Text = ""

    RTB_Descripcion_Hotel.Text = ""

    TB_Capacidad.Text = ""

    TB_Importe_noche.Text = ""

    RTB_Descripcion_Habitacion.Text = ""

    While CB_Tipo_Habitacion.Items.Count <> 0 'Borramos todos
los items del ComboBox CB_Tipo_Habitacion

        CB_Tipo_Habitacion.Items.RemoveAt(0)

    End While

    CB_Tipo_Habitacion.Text = "Lista de tipos de habitaciones"
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Else ' Si no esta seleccionado el RadioButton
RB_Localidad_Hotel

    TB_Localidad_Hotel.Enabled = False 'Deshabilitamos el
TextBox TB_Localidad_Hotel

    CB_Hotel2.Enabled = False           'Deshabilitamos el
ComboBox CB_Hotel2

End If

End Sub

'Evento que se produce cuando la propiedad checked del RadioButton
"RB_Nombre_Hotel" cambia de valor

Private Sub RB_Nombre_Hotel_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RB_Nombre_Hotel.CheckedChanged

    B_Reserva.Enabled = False 'Deshabilitamos el boton B_Reserva

    'Modificamos las propiedades de color del boton B_Reserva

    B_Reserva.BackColor = SystemColors.Control

    B_Reserva.ForeColor = SystemColors.ControlText

    If RB_Nombre_Hotel.Checked = True Then ' Si esta seleccionado
el RadioButton RB_Nombre_Hotel

        CB_Hotel1.Enabled = True 'Habilitamos el ComboBox
CB_Hotel1

        TB_Localidad_Hotel.Enabled = False 'Deshabilitamos el
TextBox TB_Localidad_Hotel

        TB_Localidad_Hotel.Text = ""

        CB_Hotel2.Enabled = False 'Deshabilitamos el ComboBox
CB_Hotel2

        CB_Hotel2.Text = "Lista de hoteles"

        CB_Hotel1.Text = "Lista de hoteles"

        CB_Tipo_Habitacion.Enabled = False 'Deshabilitamos el
ComboBox CB_Tipo_Habitacion

        TB_Nombre_Hotel.Text = "" 'Borramos el contenido del
TextBox TB_Nombre_Hotel
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
TB_Categoria.Text = ""

TB_Direccion_Hotel.Text = ""

TB_Telefono_Hotel.Text = ""

RTB_Descripcion_Hotel.Text = ""

TB_Capacidad.Text = ""

TB_Importe_noche.Text = ""

RTB_Descripcion_Habitacion.Text = ""

While CB_Tipo_Habitacion.Items.Count <> 0 'Borramos todos
los items del ComboBox CB_Tipo_Habitacion

    CB_Tipo_Habitacion.Items.RemoveAt(0)

End While

CB_Tipo_Habitacion.Text = "Lista de tipos de habitaciones"

Else

    CB_Hotel1.Enabled = False 'Deshabilitamos el ComboBox
CB_Hotel1

End If

End Sub

' Evento que se produce cuando cambia el valor de la propiedad
Text del control "TB_Localidad_Hotel"

Private Sub TB_Localidad_Hotel_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TB_Localidad_Hotel.TextChanged

    localidad = TB_Localidad_Hotel.Text.Trim 'Elimina los espacios
en blanco al principio y al final de la cadena

    If localidad <> "" Then 'Si se ha escrito algo en el TextBox
TB_Localidad_Hotel

        Label7.Text = "Mueva el cursor hacia la lista de hoteles"
' Mostramos un mensaje

    End If
```

```
End Sub

' Evento que se produce cuando se hace click en el control
"TB_Localidad_Hotel"

Private Sub TB_Localidad_Hotel_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TB_Localidad_Hotel.Click

    TB_Localidad_Hotel.Text = "" ' Borramos el contenido del
TextBox TB_Localidad_Hotel

    CB_Hotel2.Enabled = False 'Deshabilitamos el ComboBox
CB_Hotel2

End Sub

' Evento que se produce cuando se selecciona un elemento de la
lista desplegable del ComboBox "CB_Hotel1" y se cierra

Private Sub CB_Hotel1_SelectionChangeCommitted(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CB_Hotel1.SelectionChangeCommitted

    Dim Datos As SqlDataReader

    hotell = CB_Hotel1.SelectedItem.ToString 'Guardamos el
elemento seleccionado de la lista desplegable del ComboBox CB_Hotel1

    While CB_Tipo_Habitacion.Items.Count <> 0 'Borramos todos los
items del ComboBox CB_Tipo_Habitacion

        CB_Tipo_Habitacion.Items.RemoveAt(0)

    End While

    Try

        'Obtenemos información del hotel

        ' La consulta "Consulta_SQL" selecciona todos los campos
de un registro de la tabla Hotel
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
' donde campo nombre es igual al nombre del Hotel
seleccionado

Consulta_SQL = "SELECT * FROM Hotel WHERE nombre =
@nombre"

Comando = New SqlCommand(Consulta_SQL, Conexion)

Comando.Parameters.Add(New SqlParameter("@nombre",
SqlDbType.VarChar))

Comando.Parameters("@nombre").Value = hotel1

Datos = Comando.ExecuteReader()

If Datos.Read() Then

    'Escribimos la infomación del hotel en los TextBox
correspondientes

    TB_Nombre_Hotel.Text = Datos("nombre")

    TB_Categoria.Text = Datos("categoria")

    TB_Telefono_Hotel.Text = Datos("telefono")

    RTB_Descripcion_Hotel.Text = Datos("descripcion")

    TB_Direccion_Hotel.Text = Datos("domicilio") & ", " &
Datos("localidad") _
    & " " & "(" & Datos("provincia") & ")" & ", " &
Datos("codpostal")

    idhotel1 = Datos("idhotel")

End If

Datos.Close()

Comando = Nothing

Datos = Nothing

Hotel.idhotel_valor = idhotel1

Hotel.nombre_valor = TB_Nombre_Hotel.Text

Hotel.descripcion_valor = RTB_Descripcion_Hotel.Text
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Hotel.categoria_valor = TB_Categoria.Text
Hotel.direccion_valor = TB_Direccion_Hotel.Text
Hotel.telefono_valor = TB_Telefono_Hotel.Text

Catch ex As SqlException
    ' Si no hay error en el acceso a la base de datos
    MessageBox.Show("ERROR 12:" & ControlChars.CrLf &
ex.Message & _
        ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")
    Form_acceso.Show() 'Mostramos el formulario Form_acceso
    Me.Close() ' Cerramos el formulario Form_usuario
End Try

Try
    ' Averiguamos los tipos de habitacion del hotel
seleccionado

    ' La consulta "Consulta_SQL" realiza la operacion INNER
JOIN, la cual realiza una combinacion con todos aquellos
    ' registros de las dos tablas (Hotel y TipoHabitacion) en
el que el campo comun de ambas (identificador de habitacion)
    ' tenga el mismo valor (el valor del identificador de
habitacion) y donde el campo hotel_id de la tabla
    ' TipoHabitacion sea igual al identificador del hotel
    ' Seleccionamos el campo nombre de la tabla Habitacion
    Consulta_SQL = "SELECT Habitacion.nombre FROM Habitacion
INNER JOIN TipoHabitacion ON " & _
        "Habitacion.idhabitacion =
TipoHabitacion.habitacion_id WHERE TipoHabitacion.hotel_id =
@hotel_id"
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Comando = New SqlCommand(Consulta_SQL, Conexion)

Comando.Parameters.Add(New SqlParameter("@hotel_id",
SqlDbType.BigInt))

Comando.Parameters("@hotel_id").Value = idhotel1

Datos = Comando.ExecuteReader()

While Datos.Read() 'mientras leamos datos de la tabla
    CB_Tipo_Habitacion.Items.Add(Datos("nombre"))
'Añadimos items al ComboBox CB_Tipo_Habitacion
End While

Datos.Close()

Comando = Nothing

Datos = Nothing

Catch ex As SqlException

    ' Si no hay error en el acceso a la base de datos
    MessageBox.Show("ERROR 13:" & ControlChars.CrLf &
ex.Message & _
        ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

    Form_acceso.Show()

    Me.Close()

End Try

CB_Tipo_Habitacion.Enabled = True 'Habilitamos el ComboBox
CB_Tipo_Habitacion

TB_Localidad_Hotel.Enabled = False 'Deshabilitamos el TextBox
TB_Localidad_Hotel

CB_Hotel2.Enabled = False
```


Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
CB_Hotel1.Enabled = False

RB_Localidad_Hotel.Checked = False 'Establece que el control
RB_Localidad_Hotel esta desactivado

RB_Nombre_Hotel.Checked = False      'Establece que el control
RB_Nomb_Hotel esta desactivado

B_Reserva.Enabled = True 'Habilitamos el Boton B_Reserva

'Modificamos las propiedades de color del boton B_Reserva

B_Reserva.BackColor = Color.White

B_Reserva.ForeColor = Color.Red

End Sub

' Evento que se produce cuando se selecciona un elemento de la
lista desplegable del ComboBox "CB_Hotel2" y se cierra

Private Sub CB_Hotel2_SelectionChangeCommitted(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CB_Hotel2.SelectionChangeCommitted

    Dim Datos As SqlDataReader

    hotel1 = CB_Hotel2.SelectedItem.ToString 'Guardamos el
elemento seleccionado de la lista desplegable del ComboBox CB_Hotel1

    While CB_Tipo_Habitacion.Items.Count <> 0 'Borramos todos los
items del ComboBox CB_Tipo_Habitacion

        CB_Tipo_Habitacion.Items.RemoveAt(0)

    End While

    Try

        'Obtenemos información del hotel

        ' La consulta "Consulta_SQL" selecciona todos los campos
de un registro de la tabla Hotel
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
' donde campo nombre es igual al nombre del Hotel
seleccionado

Consulta_SQL = "SELECT * FROM Hotel WHERE nombre =
@nombre"

Comando = New SqlCommand(Consulta_SQL, Conexion)

Comando.Parameters.Add(New SqlParameter("@nombre",
SqlDbType.VarChar))

Comando.Parameters("@nombre").Value = hotel1

Datos = Comando.ExecuteReader()

If Datos.Read() Then

    'Escribimos la infomación del hotel en los TextBox
correspondientes

    TB_Nombre_Hotel.Text = Datos("nombre")

    TB_Categoria.Text = Datos("categoria")

    TB_Telefono_Hotel.Text = Datos("telefono")

    RTB_Descripcion_Hotel.Text = Datos("descripcion")

    TB_Direccion_Hotel.Text = Datos("domicilio") & ", " &
Datos("localidad") _
    & " " & "(" & Datos("provincia") & ")" & ", " &
Datos("codpostal")

    idhotel1 = Datos("idhotel")

End If

Datos.Close()

Comando = Nothing

Datos = Nothing

Catch ex As SqlException

    ' Si no hay error en el acceso a la base de datos
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
        MessageBox.Show("ERROR 14:" & ControlChars.CrLf &
ex.Message & _
        ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

        Form_acceso.Show()

        Me.Close()

    End Try

    Try

        ' Averiguamos los tipos de habitacion del hotel
seleccionado

        ' La consulta "Consulta_SQL" realiza la operacion INNER
JOIN, la cual realiza una combinacion con todos aquellos

        ' registros de las dos tablas (Hotel y TipoHabitacion) en
el que el campo comun de ambas (identificador de habitacion)

        ' tenga el mismo valor (el valor del identificador de
habitacion) y donde el campo hotel_id de la tabla

        ' TipoHabitacion sea igual al identificador del hotel

        ' Seleccionamos el campo nombre de la tabla Habitacion

        Consulta_SQL = "SELECT Habitacion.nombre FROM Habitacion
INNER JOIN TipoHabitacion ON " & _

        "Habitacion.idhabitacion = TipoHabitacion.habitacion_id
WHERE TipoHabitacion.hotel_id = @hotel_id"

        Comando = New SqlCommand(Consulta_SQL, Conexion)

        Comando.Parameters.Add(New SqlParameter("@hotel_id",
SqlDbType.BigInt))

        Comando.Parameters("@hotel_id").Value = idhotel1

        Datos = Comando.ExecuteReader()

        While Datos.Read() 'mientras leamos datos de la tabla
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
        CB_Tipo_Habitacion.Items.Add(Datos("nombre"))
'Añadimos items al ComboBox CB_Tipo_Habitacion

    End While

    Datos.Close()

    Comando = Nothing

    Datos = Nothing

Catch ex As SqlException

    ' Si no hay error en el acceso a la base de datos
    MessageBox.Show("ERROR 15:" & ControlChars.CrLf &
ex.Message & _
        ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

    Form_acceso.Show()

    Me.Close()

End Try

    CB_Tipo_Habitacion.Enabled = True 'Habilitamos el ComboBox
CB_Tipo_Habitacion

    TB_Localidad_Hotel.Enabled = False 'Deshabilitamos el TextBox
TB_Localidad_Hotel

    CB_Hotel2.Enabled = False

    CB_Hotel1.Enabled = False

    RB_Localidad_Hotel.Checked = False 'Establece que el control
RB_Localidad_Hotel esta desactivado

    RB_Nombre_Hotel.Checked = False      'Establece que el control
RB_Nomb_Hotel esta desactivado

    B_Reserva.Enabled = True      'Habilitamos el Boton B_Reserva

    'Modificamos las propiedades de color del boton B_Reserva

    B_Reserva.BackColor = Color.White

    B_Reserva.ForeColor = Color.Red
```

```

End Sub

' Evento que se produce cuando se selecciona un elemento de la
lista desplegable del ComboBox "CB_Tipo_Habitacion" y se cierra

Private Sub CB_Tipo_Habitacion_SelectionChangeCommitted(ByVal
sender As System.Object, ByVal e As System.EventArgs) Handles
CB_Tipo_Habitacion.SelectionChangeCommitted

    Dim Datos As SqlDataReader

    Dim habitacion As String

    habitacion = CB_Tipo_Habitacion.SelectedItem.ToString
'Guardamos el elemento seleccionado de la lista desplegable del
ComboBox CB_Tipo_Habitacion

Try

    'Averiguamos información de la habitacion

    ' La consulta "Consulta_SQL" realiza la operacion INNER
JOIN, la cual realiza una combinacion con todos aquellos

    ' registros de las dos tablas (Hotel y TipoHabitacion) en
el que el campo comun de ambas (identificador de habitacion)

    ' tenga el mismo valor (el valor del identificador de
habitacion) y donde el campo nombre de la tabla Haabitacion es igual

    ' nombre de la habitacion seleccionada y el campo hotel_id
de la tabla TipoHabitacion es igual al identificador del hotel

    ' seleccionado.

    ' Seleccionamos los campos descripcion, precio de la tabla
TipoHabitacion y capacidad de la tabla Habitacion

    Consulta_SQL = "SELECT
TipoHabitacion.descripcion,TipoHabitacion.precio,Habitacion.capacidad
" & _

                                "FROM Habitacion INNER JOIN TipoHabitacion
ON " & _

```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
"TipoHabitacion.habitacion_id = Habitacion.idhabitacion
WHERE Habitacion.nombre = @nombre and TipoHabitacion.hotel_id =
@hotel_id "
```

```
'Consulta_SQL = "SELECT * FROM TipoHabitacion WHERE
hotel_id = @hotel_id and nombre = @nombre"
```

```
Comando = New SqlCommand(Consulta_SQL, Conexion)
```

```
Comando.Parameters.Add(New SqlParameter("@hotel_id",
SqlDbType.BigInt))
```

```
Comando.Parameters.Add(New SqlParameter("@nombre",
SqlDbType.VarChar))
```

```
Comando.Parameters("@hotel_id").Value = idhotel1
```

```
Comando.Parameters("@nombre").Value = habitacion
```

```
Datos = Comando.ExecuteReader()
```

```
If Datos.Read() Then
```

```
'Escribimos la información de la habitacion en los
TextBox correspondientes
```

```
RTB_Descripcion_Habitacion.Text = Datos("descripcion")
```

```
TB_Importe_noche.Text = Datos("precio")
```

```
TB_Capacidad.Text = Datos("capacidad")
```

```
End If
```

```
Datos.Close()
```

```
Comando = Nothing
```

```
Datos = Nothing
```

```
Catch ex As SqlException
```

```
' Si no hay error en el acceso a la base de datos
```

```
MessageBox.Show("ERROR 16:" & ControlChars.CrLf &
ex.Message & _
```

```
ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Form_acceso.Show()

Me.Close()

End Try

End Sub

'Evento que se produce cuando el raton ya no esta en la parte
visible del control "TB_Localidad_Hotel"

Private Sub TB_Localidad_Hotel_MouseLeave(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TB_Localidad_Hotel.MouseLeave

    Dim Datos As SqlDataReader

    Label7.Text = "" 'Borramos el contenido de la etiqueta de
información

    While CB_Hotel2.Items.Count <> 0 'Borramos todos los items del
ComboBox CB_Hotel2

        CB_Hotel2.Items.RemoveAt(0)

    End While

    Try

        If localidad <> "" Then ' Si se ha escrito algo en el
TextBox TB_Localidad_Hotel

            'Averiguamos los hoteles que se encuentra en la
localidad escrita

            ' La consulta "Consulta_SQL" selecciona el campo
nombre de un registro de la tabla Hotel

            ' donde el campo localidad es igual a la localidad
escrita en el TextBox TB_Localidad_Hotel

            Consulta_SQL = "SELECT nombre FROM Hotel WHERE
localidad = @localidad"

            Comando = New SqlCommand(Consulta_SQL, Conexion)

            Comando.Parameters.Add(New SqlParameter("@localidad",
SqlDbType.VarChar))
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Comando.Parameters("@localidad").Value =
TB_Localidad_Hotel.Text

Datos = Comando.ExecuteReader()

While Datos.Read() 'mientras leamos datos de la tabla
    CB_Hotel2.Items.Add(Datos("nombre")) 'Añadimos
items al ComboBox CB_Hotel2
End While

Datos.Close()

Comando = Nothing

Datos = Nothing

If CB_Hotel2.Items.Count = 0 Then 'Si no se ha
encontrado ningun hotel
    MessageBox.Show("En la localidad " &
TB_Localidad_Hotel.Text & " no hay hoteles")

    CB_Hotel2.Enabled = False

    Exit Sub

End If

Else

    MessageBox.Show("Introduzca localidad para busqueda")

    Exit Sub

End If

Catch ex As SqlException

    ' Si no hay error en el acceso a la base de datos

    MessageBox.Show("ERROR 17:" & ControlChars.CrLf &
ex.Message & _

    ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2 Express.
Miguel Monzo Pastor ")

    Form_acceso.Show()

Me.Close()
```



```
End Try

    CB_Hotel2.Enabled = True

End Sub

' Evento que se produce cuando se muestra el formulario
"Form_usuario" por primera vez.

Private Sub Form_usuario_Shown(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Shown

    If Registrarse = True Then ' Si estamos en el modo registro ->
usuario no registrado

        'Habilitamos y deshabilitamos los botones correspondientes

        TSB_Editar.Enabled = True

        TSB_Guardar.Enabled = False

        TSB_Salir.Enabled = True

        TSB_Modificar.Enabled = False

        TSB_Baja.Enabled = False

        TSB_Borrar.Enabled = False

    Else 'si NO estamos en el modo registro -> usuario ya
registrado

        'Habilitamos y deshabilitamos los botones correspondientes

        TSB_Modificar.Enabled = False

        TSB_Salir.Enabled = True

        TSB_Baja.Enabled = True

        TSB_Editar.Enabled = True

        TSB_Guardar.Enabled = False

        TSB_Borrar.Enabled = False

    End If

End Sub

' Evento que se produce al pulsar el boton borrar de la barra de
herramientas
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Private Sub TSB_Borrar_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TSB_Borrar.Click

    Dim Datos As SqlDataReader

    If Me.DG_Reservas.CurrentRow.Selected = False Then ' Si no
hemos seleccionado una fila del DataGridView DG_Reservas

        MessageBox.Show("Porfavor seleccione una reserva para
modificar o borrar")

        Exit Sub

    ElseIf DateDiff(DateInterval.Day, Now.Date,
CDate(DG_Reservas.CurrentRow.Cells(1).Value)) <= 0 Then ' Si hemos
seleccionado una reserva con fecha anterior a la fecha de hoy

        MessageBox.Show("Seleccione reserva para borrar con fecha
de inicio posterior a la fecha de hoy")

        Exit Sub

    Else

        'Guardamos la informacion de la reserva en el Objeto
Reserva

        Objeto_Reserva.idreserva_valor =
Me.DG_Reservas.CurrentRow.Cells(0).Value

        Objeto_Reserva.fechainicio_valor =
Me.DG_Reservas.CurrentRow.Cells(1).Value

        Objeto_Reserva.fechafin_valor =
Me.DG_Reservas.CurrentRow.Cells(2).Value

        Objeto_Reserva.precio_valor =
CSng(Me.DG_Reservas.CurrentRow.Cells(3).Value)

        Objeto_Reserva.ocupacion_valor =
Me.DG_Reservas.CurrentRow.Cells(4).Value

        Objeto_Reserva.nombretomador_valor =
Me.DG_Reservas.CurrentRow.Cells(5).Value

        Objeto_Reserva.cantidad_valor =
Me.DG_Reservas.CurrentRow.Cells(6).Value
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Objeto_Reserva.cliente_id_valor =
Me.DG_Reservas.CurrentRow.Cells(7).Value

Objeto_Reserva.hotel_id_valor =
Me.DG_Reservas.CurrentRow.Cells(8).Value

Objeto_Reserva.habitacion_id_valor =
Me.DG_Reservas.CurrentRow.Cells(9).Value

Try

    'Averiguamos informacion sobre el tipo de habitacion
de la reserva

    ' La consulta "Consulta_SQL" realiza la operacion
INNER JOIN, la cual realiza una combinacion con todos aquellos

    ' registros de las dos tablas (Habitacion y
TipoHabitacion) en el que el campo comun de ambas (identificador de
habitacion)

    ' tenga el mismo valor (el valor del identificador de
la habitacion) y donde el campo habitacion_id de la tabla

    ' TipoHabitacion tenga el valor del indentificador de
habitacion de la reserva y el hotel_id tenga el valor del hotel de la
reserva

    ' Seleccionamos los campos de la tabla Habitacion y
TipoHabitacion.

    Consulta_SQL = "SELECT
TipoHabitacion.hotel_id,TipoHabitacion.habitacion_id,TipoHabitacion.de
scripcion,TipoHabitacion.precio," & _

"TipoHabitacion.numhabitaciones,Habitacion.nombre,Habitacion.capacidad
FROM Habitacion INNER JOIN TipoHabitacion ON " & _

"TipoHabitacion.habitacion_id = Habitacion.idhabitacion
WHERE TipoHabitacion.habitacion_id = @idhabitacion and
TipoHabitacion.hotel_id = @idhotel"

    Comando = New SqlCommand(Consulta_SQL, Conexion)

    Comando.Parameters.Add(New
SqlParameter("@idhabitacion", SqlDbType.BigInt))

    Comando.Parameters.Add(New SqlParameter("@idhotel",
SqlDbType.BigInt))
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Comando.Parameters("@idhabitacion").Value =
Objeto_Reserva.habitacion_id_valor

Comando.Parameters("@idhotel").Value =
Objeto_Reserva.hotel_id_valor

Datos = Comando.ExecuteReader()

If Datos.Read() Then

    'Guardamos informacion de la habitacion en el
Objeto Habitacion

    Objeto_Habitacion.Hotel_id_valor =
Datos("hotel_id")

    Objeto_Habitacion.Idtipohabitacion_valor =
Datos("habitacion_id")

    Objeto_Habitacion.Descripcion_valor =
Datos("descripcion")

    Objeto_Habitacion.Precio_valor = Datos("precio")

    Objeto_Habitacion.Numhabitaciones_valor =
Datos("numhabitaciones")

    Objeto_Habitacion.Nombre_valor = Datos("nombre")

    Objeto_Habitacion.Capacidad_valor =
Datos("capacidad")

End If

Datos.Close()

Comando = Nothing

Datos = Nothing

Catch ex As SqlException

    ' Si no hay error en el acceso a la base de datos

    MessageBox.Show("ERROR 18:" & ControlChars.CrLf &
ex.Message & _
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

MessageBox.Show("ERROR EN EL ACCESO A LA BASE DE
DATOS, INTENTELO DE NUEVO")

Exit Sub 'Salimos del procedimiento

'Form_acceso.Show()

'Me.Close()

End Try

Try

'Averiguamos informacion sobre el hotel de la reserva

' La consulta "Consulta_SQL" realiza la operacion
INNER JOIN, la cual realiza una combinacion con todos aquellos

' registros de las dos tablas (Hotel y TipoHabitacion)
en el que el campo comun de ambas (identificador de hotel)

' tenga el mismo valor (el valor del identificador de
hotel de la reserva)

' Seleccionamos los campos de la tabla hotel.

Consulta_SQL = "SELECT Hotel.idhotel,Hotel.nombre
,Hotel.descripcion,Hotel.categoria," & _

"Hotel.domicilio,Hotel.localidad,Hotel.provincia,Hotel.codpostal,Hotel
.telefono FROM Hotel INNER JOIN TipoHabitacion ON " & _

"Hotel.idhotel = TipoHabitacion.hotel_id WHERE
TipoHabitacion.hotel_id = @hotel_id"

Comando = New SqlCommand(Consulta_SQL, Conexion)

Comando.Parameters.Add(New SqlParameter("@hotel_id",
SqlDbType.BigInt))

Comando.Parameters("@hotel_id").Value =
Objeto_Habitacion.Hotel_id_valor

Datos = Comando.ExecuteReader()

If Datos.Read() Then
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
'Guardamos los datos del hotel en el objeto Hotel
Hotel.idhotel_valor = Datos("idhotel")
Hotel.nombre_valor = Datos("nombre")
Hotel.descripcion_valor = Datos("descripcion")
Hotel.categoria_valor = Datos("categoria")
Hotel.direccion_valor = Datos("domicilio") & ", "
& Datos("localidad") _
& " " & "(" & Datos("provincia") & ")" & ", " &
Datos("codpostal")
Hotel.telefono_valor = Datos("telefono")

End If

hotel2 = Hotel.nombre_valor
idhotel1 = Hotel.idhotel_valor
Datos.Close()
Comando = Nothing
Datos = Nothing

Catch ex As SqlException
' Si no hay error en el acceso a la base de datos
MessageBox.Show("ERROR 19:" & ControlChars.CrLf &
ex.Message & _
ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")
MessageBox.Show("ERROR EN EL ACCESO A LA BASE DE
DATOS, INTENTELO DE NUEVO")

Exit Sub 'Salimos del procedimiento
'Form_acceso.Show()
'Me.Close()

End Try
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Ver_reserva = 3 'Indica se muestra el formulario
Form_Reserva pulsando el boton TSB_Borrar

'Habilitamos y deshabilitamos los botones correspondientes
en el formulario Form_Reserva

Form_Reserva.TSB_Guardar_Reserva.Enabled = False

Form_Reserva.TSB_Modificar_Reserva.Enabled = False

Form_Reserva.TSB_Borrar_Reserva.Enabled = True

'Mostramos el formulario Form_Reserva

Form_Reserva.Visible = True

Form_Reserva.Show()

End If

End Sub

' Evento que se produce al hacer click en el boton B_Reserva de la
pestaña reserva

Private Sub B_Reserva_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles B_Reserva.Click

    GB_Busqueda_Hotel.Enabled = True 'Habilitar el GroupBox
GB_Busqueda_Hotel

    CB_Hotel1.Text = "Lista de hoteles"

    CB_Hotel2.Text = "Lista de hoteles"

    TB_Localidad_Hotel.Text = ""

    RB_Nombre_Hotel.Checked = False 'Establece que el control
RB_Nombre_Hotel esta desactivado

    RB_Localidad_Hotel.Checked = False 'Establece que el control
RB_Localidad_Hotel esta desactivado

    GB_Busqueda_Hotel.Enabled = False 'Deshabilitar el GroupBox
GB_Busqueda_Hotel

'Modificamos las propiedades de color del boton B_Reserva

B_Reserva.BackColor = SystemColors.Control
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
B_Reserva.ForeColor = SystemColors.ControlText

Ver_reserva = 2 'Indica que se muestra el formulario
Form_Reserva pulsando el boton B_Reserva

'Habilitamos y deshabilitamos los botones correspondientes en
el formulario Form_Reserva

Form_Reserva.TSB_Guardar_Reserva.Enabled = False

Form_Reserva.TSB_Modificar_Reserva.Enabled = False

Form_Reserva.TSB_Borrar_Reserva.Enabled = False

MessageBox.Show("REALICE UNA CONSULTA PARA VER DISPONIBILIDAD
DEL HOTEL")

If Descarga_reserva = False Then 'Si el formulario
Form_Reserva esta cargado

'Mostramos el formulario Form_Reserva y hacemos que sea el
formualrio activo

Form_Reserva.Visible = True

Form_Reserva.Show()

Form_Reserva.Activate()

Else 'Si el formulario Form_Reserva no esta esta cargado

Form_Reserva.Visible = True

Form_Reserva.Show()

End If

B_Reserva.Enabled = False 'Deshabilitamos el boton B_Reserva

End Sub
```


Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

'Evento que se produce cuando se hace click en cualquier parte de una la celda del Control DataGrid DG_Reservas

```
Private Sub DG_Reservas_CellClick(ByVal sender As System.Object,  
ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles  
DG_Reservas.CellClick
```

```
Dim Datos As SqlDataReader
```

'Guardamos la informacion de la reserva en el Objeto Reserva

```
Objeto_Reserva.idreserva_valor =  
Me.DG_Reservas.CurrentRow.Cells(0).Value
```

```
Objeto_Reserva.fechainicio_valor =  
Me.DG_Reservas.CurrentRow.Cells(1).Value
```

```
Objeto_Reserva.fechafin_valor =  
Me.DG_Reservas.CurrentRow.Cells(2).Value
```

```
Objeto_Reserva.precio_valor =  
CSng(Me.DG_Reservas.CurrentRow.Cells(3).Value)
```

```
Objeto_Reserva.ocupacion_valor =  
Me.DG_Reservas.CurrentRow.Cells(4).Value
```

```
Objeto_Reserva.nombretomador_valor =  
Me.DG_Reservas.CurrentRow.Cells(5).Value
```

```
Objeto_Reserva.cantidad_valor =  
Me.DG_Reservas.CurrentRow.Cells(6).Value
```

```
Objeto_Reserva.cliente_id_valor =  
Me.DG_Reservas.CurrentRow.Cells(7).Value
```

```
Objeto_Reserva.hotel_id_valor =  
Me.DG_Reservas.CurrentRow.Cells(8).Value
```

```
Objeto_Reserva.habitacion_id_valor =  
Me.DG_Reservas.CurrentRow.Cells(9).Value
```

```
Try
```

'Averiguamos informacion sobre el tipo de habitacion de la reserva

' La consulta "Consulta_SQL" realiza la operacion INNER JOIN, la cual realiza una combinacion con todos aquellos



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
' registros de las dos tablas (Habitacion y
TipoHabitacion) en el que el campo comun de ambas (identificador de
habitacion)

' tenga el mismo valor (el valor del identificador de la
habitacion) y donde el campo habitacion_id de la tabla

' TipoHabitacion tenga el valor de indentificador de tipo
de habitacion de la reserva y el campo hotel_id tenga el valor del

' hotel de la reserva

' Seleccionamos los campos de la tabla Habitacion y
TipoHabitacion.

Consulta_SQL = "SELECT
TipoHabitacion.hotel_id,TipoHabitacion.habitacion_id,TipoHabitacion.de
scripcion,TipoHabitacion.precio," & _

"TipoHabitacion.numhabitaciones,Habitacion.nombre,Habitacion.capacidad
FROM Habitacion INNER JOIN TipoHabitacion ON " & _

"TipoHabitacion.habitacion_id =
Habitacion.idhabitacion WHERE TipoHabitacion.habitacion_id =
@idhabitacion and TipoHabitacion.hotel_id = @idhotel"

Comando = New SqlCommand(Consulta_SQL, Conexion)

Comando.Parameters.Add(New SqlParameter("@idhabitacion",
SqlDbType.BigInt))

Comando.Parameters.Add(New SqlParameter("@idhotel",
SqlDbType.BigInt))

Comando.Parameters("@idhabitacion").Value =
Objeto_Reserva.habitacion_id_valor

Comando.Parameters("@idhotel").Value =
Objeto_Reserva.hotel_id_valor

Datos = Comando.ExecuteReader()

If Datos.Read() Then

'Guardamos informacion de la habitacion en el Objeto
Habitacion
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Objeto_Habitacion.Hotel_id_valor = Datos("hotel_id")

Objeto_Habitacion.Idtipohabitacion_valor =
Datos("habitacion_id")

Objeto_Habitacion.Descripcion_valor =
Datos("descripcion")

Objeto_Habitacion.Precio_valor = Datos("precio")

Objeto_Habitacion.Numhabitaciones_valor =
Datos("numhabitaciones")

Objeto_Habitacion.Nombre_valor = Datos("nombre")

Objeto_Habitacion.Capacidad_valor = Datos("capacidad")

End If

Datos.Close()

Comando = Nothing

Datos = Nothing

Catch ex As SqlException

    ' Si no hay error en el acceso a la base de datos

    MessageBox.Show("ERROR 20:" & ControlChars.CrLf &
ex.Message & _

        ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

    Form_acceso.Show()

    Me.Close()

End Try

Try

    'Averiguamos informacion sobre el hotel de la reserva

    ' La consulta "Consulta_SQL" realiza la operacion INNER
JOIN, la cual realiza una combinacion con todos aquellos
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
' registros de las dos tablas (Hotel y TipoHabitacion) en
el que el campo comun de ambas (identificador de hotel)

' tenga el mismo valor (el valor del identificador de
hotel de la reserva)

' Seleccionamos los campos de la tabla hotel.

Consulta_SQL = "SELECT Hotel.idhotel,Hotel.nombre
,Hotel.descripcion,Hotel.categoria," & _

"Hotel.domicilio,Hotel.localidad,Hotel.provincia,Hotel.codpostal,Hotel
.telefono FROM Hotel INNER JOIN TipoHabitacion ON " & _

"Hotel.idhotel = TipoHabitacion.hotel_id WHERE
TipoHabitacion.hotel_id = @hotel_id"

Comando = New SqlCommand(Consulta_SQL, Conexion)

Comando.Parameters.Add(New SqlParameter("@hotel_id",
SqlDbType.BigInt))

Comando.Parameters("@hotel_id").Value =
Objeto_Habitacion.Hotel_id_valor

Datos = Comando.ExecuteReader()

If Datos.Read() Then

'Guardamos los datos del hotel en el objeto Hotel

Hotel.idhotel_valor = Datos("idhotel")

Hotel.nombre_valor = Datos("nombre")

Hotel.descripcion_valor = Datos("descripcion")

Hotel.categoria_valor = Datos("categoria")

Hotel.direccion_valor = Datos("domicilio") & ", " &
Datos("localidad") _
& " " & "(" & Datos("provincia") & ")" & ", " &
Datos("codpostal")

Hotel.telefono_valor = Datos("telefono")

End If

hotel2 = Hotel.nombre_valor

Datos.Close()
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Comando = Nothing

Datos = Nothing

hotell = Hotel.nombre_valor

Catch ex As SqlException

    ' Si no hay error en el acceso a la base de datos

    MessageBox.Show("ERROR 21:" & ControlChars.CrLf &
ex.Message & _

        ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

    Form_acceso.Show()

    Me.Close()

End Try

Ver_reserva = 4 'Se ha selccionado una fila del DataGrid
DG_Reservas y podemos ver infrmacion de la reserva en la pestaña
reserva

End Sub

' Evento que se produce al pulsar el boton info de la barra de
herramientas

Private Sub TSB_Info_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TSB_Info.Click

    Sobre.Show()

End Sub

End Class
```



FORMULARIO INFO_RESERVA

```
Imports System.Data.SqlClient
```

```
Public Class Form_Reserva
```

```
    Private contador1 As Integer
```

```
    Private contador2 As Integer
```

```
    Private THabitacion As String 'Variable para almacenar el tipo de habitacion
```

```
    Private IHabitacion As Int64 'Variable para almacenar el identificador del tipo de habitacion
```

```
    ' Evento que se produce al pulsar sobre el boton cerrar de la barra de herramientas
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Private Sub TSB_Cerrar_Reserva_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TSB_Cerrar_Reserva.Click

    'Habilitamos y deshabilitamos los comobox y radiobutton
correspondientes

    Form_usuario.GB_Busqueda_Hotel.Enabled = True

    Form_usuario.RB_Localidad_Hotel.Enabled = True

    Form_usuario.RB_Nombre_Hotel.Enabled = True

    Form_usuario.CB_Hotel2.Enabled = False

    Form_usuario.CB_Hotel1.Enabled = False

    'Ponemos valores a las propiedades de los controles del
formulario Form_usuario, pestaña reserva

    Form_usuario.CB_Hotel1.Text = "Lista de hoteles"
    Form_usuario.CB_Hotel2.Text = "Lista de hoteles"
    Form_usuario.TB_Localidad_Hotel.Text = ""
    Form_usuario.RB_Nombre_Hotel.Checked = False
    Form_usuario.RB_Localidad_Hotel.Checked = False
    Form_usuario.GB_Hotel.Enabled = True
    Form_usuario.TB_Nombre_Hotel.Text = ""
    Form_usuario.TB_Categoria.Text = ""
    Form_usuario.TB_Direccion_Hotel.Text = ""
    Form_usuario.TB_Telefono_Hotel.Text = ""
    Form_usuario.RTB_Descripcion_Hotel.Text = ""
    Form_usuario.GB_Habitacion.Enabled = True
    Form_usuario.CB_Tipo_Habitacion.Text = ""
    Form_usuario.TB_Capacidad.Text = ""
    Form_usuario.TB_Importe_noche.Text = ""
    Form_usuario.RTB_Descripcion_Habitacion.Text = ""
```



```
Descarga_reserva = True ' Variable Descarga_reserva = true  
indica que el formulario Form_Reserva no esta cargado  
  
Ver_reserva = 0 ' Ver_reserva = 0 indica que no se muestra el  
formulario reserva
```

```
'Mostramos y hacemos activo el formulario Form_usuario  
Form_usuario.Visible = True  
Form_usuario.Show()  
Form_usuario.Activate()
```

```
Me.Close() 'Cerramos formulario Form_Reserva
```

```
End Sub
```

```
'Evento que se produce cuando un formulario Form_Reserva pasa a  
estar en primer plano.
```

```
Private Sub Form_Reserva_Activated(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles MyBase.Activated
```

```
Dim dias_reserva As Integer
```

```
CB_THabitaciones.Text = "Tipos de habitaciones disponibles"
```

```
If Ver_reserva = 1 Then 'Hemos accedido al formulario de  
informacion de reserva a traves del boton "modificar" de la barra de  
herramientas
```

```
dias_reserva = DateDiff(DateInterval.Day,  
CDate(TB_Fecha_entrada.Text), CDate(TB_Fecha_salida.Text)) + 1
```


Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
TB_Total.Text = CStr((CSng(TB_Importe.Text) *
dias_reserva) * CInt(TB_NHab.Text)) 'Calculamos el total del importe
de la reserva

End If

If Ver_reserva = 2 Then 'Hemos accedido al formulario de
informacion de reserva a traves del boton "consultar disponibilidad"
en pestaña reserva

' Inicializamos las propiedad de los controles del
formulario "Form_Reserva"

TB_Hab.Text = ""

TB_HLibres.Text = ""

TB_Importe.Text = "0"

TB_CHabitacion.Text = ""

NUD_Nhab.Value = 1

NUD_OHabitacion.Value = 1

TB_NHab.Text = CStr(NUD_Nhab.Value) 'Convertimos a string
el valor del control NumericUpDown "NUD_Nhab"

TB_OHab.Text = CStr(NUD_OHabitacion.Value)

NUD_Nhab.Value = 1

NUD_OHabitacion.Value = 1

TB_Total.Text = "0"

NUD_Nhab.Enabled = False

NUD_OHabitacion.Enabled = False

End If

End Sub
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
' Evento que se produce cuando se selecciona un elemento de la
lista desplegable del ComboBox "CB_THabitaciones" y se cierra

Private Sub CB_THabitaciones_SelectionChangeCommitted(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
CB_THabitaciones.SelectionChangeCommitted

    Dim dias_reserva As Integer

    'Convertimos a cadena el valor del item seleccionado en el
ComboBox "CB_THabitaciones"

    'y lo asignamos a la variable THabitacion de tipo string
    THabitacion = CB_THabitaciones.SelectedItem.ToString

    TB_Hab.Text = CB_THabitaciones.SelectedItem.ToString

    'Calculamos los dias reservados

    dias_reserva = DateDiff(DateInterval.Day,
CDate(TB_Fecha_entrada.Text), CDate(TB_Fecha_salida.Text)) + 1

    NUD_Nhab.Enabled = True 'Habilitamos el control NumericUpDown
"NUD_Nhab"

    NUD_OHabitacion.Enabled = True

    If Ver_reserva = 2 Then 'Hemos accedido al formulario de
informacion de reserva a traves del boton "consultar disponibilidad"
de la pestaña Reserva

        For contador1 = 0 To (V_Disponibilidad.Length - 1) 'Para
cada tipo de habitacion del hotel de la reserva

            If
V_Disponibilidad(contador1).Habitacion_valor.Nombre_valor =
THabitacion Then ' Si es el tipo de habitacion seleccionada
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
'Actualizamos los controles del formulario
Form_Reserva con los datos de la habitacion y la reserva

    TB_HLibres.Text =
V_Disponibilidad(contador1).Num_disponibles_valor

    TB_Importe.Text =
V_Disponibilidad(contador1).Habitacion_valor.Precio_valor

    TB_CHabitacion.Text =
V_Disponibilidad(contador1).Habitacion_valor.Capacidad_valor

'NumericUpDown ocupacion de las habitaciones
NUD_OHabitacion.Value = 1

NUD_OHabitacion.Maximum =
CInt(TB_CHabitacion.Text)

NUD_OHabitacion.Minimum = 1

'NumericUpDown numero de habitaciones reservadas
NUD_Nhab.Value = 1

NUD_Nhab.Maximum = CInt(TB_HLibres.Text)

NUD_Nhab.Minimum = 1

'Actulizamos los TextBox con los valores obtenidos
en los NumericUpDown

    TB_OHab.Text = CStr(NUD_OHabitacion.Value)

    TB_NHab.Text = CStr(NUD_Nhab.Value)

'Actualizamos propiedad Text del control TextBox
"TB_Total"

    'Importe total = (dias reservados *importe por
dia)*Numero de habitaciones reservadas

    TB_Total.Text = CStr((CSng(TB_Importe.Text) *
dias_reserva) * NUD_Nhab.Value)

Exit For

End If

Next
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
    If Cint(TB_HLibres.Text) = 0 Then 'Si no quedan
habitaciones libres

        MessageBox.Show("No quedan habitaciones libres del
tipo: " & THabitacion)

        CB_THabitaciones.Text = "Tipos de habitaciones
disponibles"

        'Deshabilitamos los NumeriUpDown "NUD_Nhab" y
"NUD_OHabitacion"

        NUD_Nhab.Enabled = False

        NUD_OHabitacion.Enabled = False

    Else ' Si hay habitaciones libres

        'Habilitamos los NumeriUpDown "NUD_Nhab" y
"NUD_OHabitacion"

        NUD_Nhab.Enabled = True

        NUD_OHabitacion.Enabled = True

    End If

End If

    If Ver_reserva = 1 Then 'Hemos accedido al formulario de
informacion de reserva a traves del boton "modificar" de la barra de
herramientas

        If THabitacion <> Objeto_Habitacion.Nombre_valor Then ' Si
la habitacion seleccionada es distinta a la reservada

            For contador1 = 0 To (V_Disponibilidad.Length - 1)
'Para cada tipo de habitacion del hotel de la reserva

                If

V_Disponibilidad(contador1).Habitacion_valor.Nombre_valor =
THabitacion Then ' Si es el tipo de habitacion seleccionada

                    'Actualizamos los controles del formulario
Form_Reserva con los datos de la habitacion y la reserva

                    TB_HLibres.Text =
V_Disponibilidad(contador1).Num_disponibles_valor
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
TB_Importe.Text =
V_Disponibilidad(contador1).Habitacion_valor.Precio_valor

TB_CHabitacion.Text =
V_Disponibilidad(contador1).Habitacion_valor.Capacidad_valor

'NumericUpDown ocupacion de las habitaciones
NUD_OHabitacion.Value = 1
NUD_OHabitacion.Maximum =
CInt(TB_CHabitacion.Text)
NUD_OHabitacion.Minimum = 1

'NumericUpDown numero de habitaciones
reservadas
NUD_Nhab.Value = 1
NUD_Nhab.Maximum = CInt(TB_HLibres.Text)
NUD_Nhab.Minimum = 1

'Actulizamos los TextBox con los valores
obtenidos en los NumericUpDown
TB_OHab.Text = CStr(NUD_OHabitacion.Value)
TB_NHab.Text = CStr(NUD_Nhab.Value)

'Actualizamos propiedad Text del control
TextBox "TB_Total"

'Importe total = (dias reservados *importe por
dia)*Numero de habitaciones reservadas
TB_Total.Text = CStr((CNg(TB_Importe.Text) *
dias_reserva) * NUD_Nhab.Value)

Exit For
End If

Next

If CInt(TB_HLibres.Text) = 0 Then 'Si no quedan
habitaciones libres

MessageBox.Show("No quedan habitaciones libres del
tipo: " & THabitacion)
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
CB_THabitaciones.Text = "Tipos de habitaciones
disponibles"

'Deshabilitamos los NumeriUpDown "NUD_Nhab" y
"NUD_OHabitacion"

NUD_Nhab.Enabled = False

NUD_OHabitacion.Enabled = False

Else ' Si hay habitaciones libres

'Habilitamos los NumeriUpDown "NUD_Nhab" y
"NUD_OHabitacion"

NUD_Nhab.Enabled = True

NUD_OHabitacion.Enabled = True

End If

Else ' Si la habitacion seleccionada es igual a la
reservada

For contador1 = 0 To (V_Disponibilidad.Length - 1)
'Para cada tipo de habitacion del hotel de la reserva

If
V_Disponibilidad(contador1).Habitacion_valor.Nombre_valor =
THabitacion Then ' Si es el tipo de habitacion seleccionada

'Actualizamos los controles del formulario
Form_Reserva con los datos de la habitacion y la reserva

'utilizamos los datos del objeto habitacion
reservada y el objeto reserva

TB_HLibres.Text =
V_Disponibilidad(contador1).Num_disponibles_valor

TB_NHab.Text =
Objeto_Reserva.cantidad_valor.ToString

TB_Importe.Text =
CStr(Objeto_Habitacion.Precio_valor)

TB_CHabitacion.Text =
CStr(Objeto_Habitacion.Capacidad_valor)

TB_OHab.Text =
CStr(Objeto_Reserva.ocupacion_valor)

'El maximo de habitaciones para reservar será
igual al numero de habitaciones reervadas más las hbaitaciones libres
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
        NUD_Nhab.Maximum =
Objeto_Reserva.cantidad_valor + CInt(TB_HLibres.Text)

        NUD_Nhab.Minimum = 1

        NUD_Nhab.Value = Objeto_Reserva.cantidad_valor

        NUD_OHabitacion.Value =
Objeto_Reserva.ocupacion_valor

        NUD_OHabitacion.Maximum =
CInt(TB_CHabitacion.Text)

        NUD_OHabitacion.Minimum = 1

        'Actualizamos propiedad Text del control
TextBox "TB_Total"

        'Importe total = (dias reservados *importe por
dia)*Numero de habitaciones reservadas

        TB_Total.Text = CStr((CSng(TB_Importe.Text) *
dias_reserva) * NUD_Nhab.Value)

        Exit For

    End If

Next

    If CInt(TB_HLibres.Text) = 0 Then 'Si no quedan
habitaciones libres

        MessageBox.Show("No quedan habitaciones libres del
tipo: " & THabitacion)

        CB_THabitaciones.Text = "Tipos de habitaciones
disponibles"

        'Deshabilitamos los NumeriUpDown "NUD_Nhab" y
"NUD_OHabitacion"

        NUD_Nhab.Enabled = False

        NUD_OHabitacion.Enabled = False

    Else ' Si hay habitaciones libres
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
        'Habilitamos los NumeriUpDown "NUD_Nhab" y
"NUD_OHabitacion"

        NUD_Nhab.Enabled = True

        NUD_OHabitacion.Enabled = True

    End If

End If

End If

End Sub

' Evento que se produce antes de cerrar el formulario
"Form_Reserva"

Private Sub Form_Reserva_FormClosing(ByVal sender As
System.Object, ByVal e As System.Windows.Forms.FormClosingEventArgs)
Handles MyBase.FormClosing

    'Habilitamos y deshabilitamos los comobox y radiobutton
correspondientes

    Form_usuario.GB_Busqueda_Hotel.Enabled = True

    Form_usuario.RB_Localidad_Hotel.Enabled = True

    Form_usuario.RB_Nombre_Hotel.Enabled = True

    Form_usuario.CB_Hotel2.Enabled = False

    Form_usuario.CB_Hotel1.Enabled = False

    'Ponemos valores a las propiedades de los controles del
formulario Form_usuario, pestaña reserva

    Form_usuario.CB_Hotel1.Text = "Lista de hoteles"

    Form_usuario.CB_Hotel2.Text = "Lista de hoteles"

    Form_usuario.TB_Localidad_Hotel.Text = ""

    Form_usuario.RB_Nombre_Hotel.Checked = False

    Form_usuario.RB_Localidad_Hotel.Checked = False

    Form_usuario.GB_Hotel.Enabled = True
```


Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Form_usuario.TB_Nombre_Hotel.Text = ""
Form_usuario.TB_Categoria.Text = ""
Form_usuario.TB_Direccion_Hotel.Text = ""
Form_usuario.TB_Telefono_Hotel.Text = ""
Form_usuario.RTB_Descripcion_Hotel.Text = ""
Form_usuario.GB_Habitacion.Enabled = True
Form_usuario.CB_Tipo_Habitacion.Text = ""
Form_usuario.TB_Capacidad.Text = ""
Form_usuario.TB_Importe_noche.Text = ""
Form_usuario.RTB_Descripcion_Habitacion.Text = ""
```

```
Ver_reserva = 0 ' Ver_reserva = 0 indica que no se muestra el
formulario reserva
```

```
'Mostramos y hacemos activo el formulario Form_usuario
```

```
Form_usuario.Visible = True
```

```
Form_usuario.Show()
```

```
Form_usuario.Activate()
```

```
End Sub
```

```
Private Sub TSB_Imprimir_Reserva_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TSB_Imprimir_Reserva.Click
```

```
If seleccionarImpresora() = True Then ' Si hemos seleccionado
impresora
```

```
'Configuramos el Form_Reserva para imprimir -> cambiamos
las propiedades de los controles para una mejor impresion
```

```
'No mostramos los siguientes controles:
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
ToolStrip1.Visible = False  
TB_HLibres.Visible = False  
Label11.Visible = False  
TB_HLibres.Visible = False  
GB_D_obligatorios.Visible = False  
GB_Disponibilidad.Visible = False
```

impresion 'Cambiamos las propiedades de los controles para una mejor

```
Me.BackColor = Color.White  
TB_Cliente_Reserva.BackColor = Color.White  
TB_Hotel_Reserva.BackColor = Color.White  
TB_Direccion_Reserva.BackColor = Color.White  
  
TB_CHabitacion.BackColor = Color.White  
TB_Importe.BackColor = Color.White  
TB_Tomador.BackColor = Color.White  
TB_Fecha_entrada.BackColor = Color.White  
TB_Fecha_salida.BackColor = Color.White  
TB_Total.BackColor = Color.White  
  
TB_Hab.BackColor = Color.White  
TB_NHab.BackColor = Color.White  
TB_OHab.BackColor = Color.White  
  
Me.FormBorderStyle = Windows.Forms.FormBorderStyle.None  
TB_Cliente_Reserva.BorderStyle = BorderStyle.None  
TB_Hotel_Reserva.BorderStyle = BorderStyle.None  
TB_Direccion_Reserva.BorderStyle = BorderStyle.None  
TB_CHabitacion.BorderStyle = BorderStyle.None  
TB_Importe.BorderStyle = BorderStyle.None
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
TB_Tomador.BorderStyle = BorderStyle.None

TB_Fecha_entrada.BorderStyle = BorderStyle.None

TB_Fecha_salida.BorderStyle = BorderStyle.None

TB_Total.BorderStyle = BorderStyle.None

TB_Hab.BorderStyle = BorderStyle.None

TB_NHab.BorderStyle = BorderStyle.None

TB_OHab.BorderStyle = BorderStyle.None

Panell.ForeColor = Color.Black

PrintForm1.Print (Me,
PowerPacks.Printing.PrintForm.PrintOption.ClientAreaOnly) 'Imprimimos
el formulario formateado parta impresion

    End If

End Sub

' Evento quese produce cuando se ha terminado la impresion del
formulario Form_Reserva

Private Sub PrintForm1_EndPrint (ByVal sender As Object, ByVal e As
System.Drawing.Printing.PrintEventArgs) Handles PrintForm1.EndPrint

    'Volvemos a poner las propiedades originales a los controles
de formulario Form_Reserva

    ToolStrip1.Visible = True

    TB_HLibres.Visible = True

    Label11.Visible = True

    TB_HLibres.Visible = True

    GB_D_obligatorios.Visible = True

    GB_Disponibilidad.Visible = True

    Me.BackColor = SystemColors.ControlLight

    TB_Cliente_Reserva.BackColor = SystemColors.ControlLight

    TB_Hotel_Reserva.BackColor = SystemColors.ControlLight

    TB_Direccion_Reserva.BackColor = SystemColors.ControlLight
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
CB_THabitaciones.BackColor = Color.White
TB_CHabitacion.BackColor = SystemColors.ControlLight
TB_Importe.BackColor = SystemColors.ControlLight
TB_Tomador.BackColor = SystemColors.ControlLight
TB_Fecha_entrada.BackColor = SystemColors.ControlLight
TB_Fecha_salida.BackColor = SystemColors.ControlLight
TB_Total.BackColor = SystemColors.ControlLight
TB_Hab.BackColor = SystemColors.ControlLight
TB_NHab.BackColor = SystemColors.ControlLight
TB_OHab.BackColor = SystemColors.ControlLight

Me.FormBorderStyle = Windows.Forms.FormBorderStyle.Sizable
TB_Cliente_Reserva.BorderStyle = BorderStyle.Fixed3D
TB_Hotel_Reserva.BorderStyle = BorderStyle.Fixed3D
TB_Direccion_Reserva.BorderStyle = BorderStyle.Fixed3D
TB_CHabitacion.BorderStyle = BorderStyle.Fixed3D
TB_Importe.BorderStyle = BorderStyle.Fixed3D
TB_Tomador.BorderStyle = BorderStyle.Fixed3D
TB_Fecha_entrada.BorderStyle = BorderStyle.Fixed3D
TB_Fecha_salida.BorderStyle = BorderStyle.Fixed3D
TB_Total.BorderStyle = BorderStyle.Fixed3D
TB_Hab.BorderStyle = BorderStyle.Fixed3D
TB_NHab.BorderStyle = BorderStyle.Fixed3D
TB_OHab.BorderStyle = BorderStyle.Fixed3D
```

End Sub

' Evento que se produce cuando pulsamos el boton "guardar reserva"
de la barra de herramientas



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Private Sub TSB_Guardar_Reserva_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TSB_Guardar_Reserva.Click

    Dim Resultado As Integer

    Dim oDataAdapter As SqlDataAdapter

    Dim oDataSet As DataSet

    Dim oCB As SqlCommandBuilder

    If (TB_Hab.Text = "") Or (TB_NHab.Text = "0") Or (TB_OHab.Text
= "0") Then ' Si no hemos seleccionado tipo de habitacion, numero de
habitaciones y ocupacion

        MessageBox.Show("Porfavor rellene todo los campos
obligatorios")

        Exit Sub

    Else

        Try

            'Alamcenamos en la base de datos la informacion de la
reserva

            'Crear comando escalar

            ' La consulta "Consulta_SQL" inserta en la tabla
Reserva un registro con los valores obtenidos en el formulario

            Consulta_SQL = "INSERT INTO " & _

            "Reserva(fechainicio, fechafin, precio, ocupacion, nombretomador, cliente_i
d, hotel_id, habitacion_id, cantidad) " & _

            "VALUES
(@fechainicio, @fechafin, @precio, @ocupacion, @nombretomador, @cliente_id,
@hotel_id, @habitacion_id, @cantidad) "
```



```

'Crear comando

Comando = New SqlCommand(Consulta_SQL, Conexion)

' Añadir parametros al comando

Comando.Parameters.Add(New
SqlParameter("@fechainicio", SqlDbType.DateTime))

Comando.Parameters.Add(New SqlParameter("@fechafin",
SqlDbType.DateTime))

Comando.Parameters.Add(New SqlParameter("@precio",
SqlDbType.Money))

Comando.Parameters.Add(New SqlParameter("@ocupacion",
SqlDbType.TinyInt))

Comando.Parameters.Add(New
SqlParameter("@nombretomador", SqlDbType.VarChar))

Comando.Parameters.Add(New SqlParameter("@cliente_id",
SqlDbType.BigInt))

Comando.Parameters.Add(New SqlParameter("@hotel_id",
SqlDbType.BigInt))

Comando.Parameters.Add(New
SqlParameter("@habitacion_id", SqlDbType.BigInt))

Comando.Parameters.Add(New SqlParameter("@cantidad",
SqlDbType.Int))

' Actualizamos el objeto Reserva

Objeto_Reserva.fechainicio_valor = F_entrada

Objeto_Reserva.fechafin_valor = F_salida

Objeto_Reserva.precio_valor = CSng(TB_Total.Text)

Objeto_Reserva.ocupacion_valor = CInt(TB_OHab.Text)

Objeto_Reserva.cliente_id_valor =
Objeto_Usuario.Id_valor

```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
        For contador1 = 0 To (V_Disponibilidad.Length - 1)
'Para cada tipo de habitacion del hotel de la reserva

            If
V_Disponibilidad(contador1).Habitacion_valor.Nombre_valor =
THabitacion Then ' Si es el tipo de habitacion seleccionada

                Objeto_Reserva.habitacion_id_valor =
V_Disponibilidad(contador1).Habitacion_valor.Idtipohabitacion_valor

                Exit For

            End If

        Next

        Objeto_Reserva.hotel_id_valor = idhotel1

        Objeto_Reserva.cantidad_valor = CInt(TB_NHab.Text)

        Objeto_Reserva.nombretomador_valor = TB_Tomador.Text

        'Añadir parametro al comando

        Comando.Parameters("@fechainicio").Value =
Objeto_Reserva.fechainicio_valor

        Comando.Parameters("@fechafin").Value =
Objeto_Reserva.fechafin_valor

        Comando.Parameters("@precio").Value =
Objeto_Reserva.precio_valor

        Comando.Parameters("@ocupacion").Value =
Objeto_Reserva.ocupacion_valor

        Comando.Parameters("@nombretomador").Value =
TB_Tomador.Text

        Comando.Parameters("@cliente_id").Value =
Objeto_Reserva.cliente_id_valor

        Comando.Parameters("@habitacion_id").Value =
Objeto_Reserva.habitacion_id_valor

        Comando.Parameters("@hotel_id").Value =
Objeto_Reserva.hotel_id_valor

        Comando.Parameters("@cantidad").Value =
Objeto_Reserva.cantidad_valor
```



```

        Resultado = Comando.ExecuteNonQuery() 'Ejecutar
comando . La sentencia a ejecutar no devuelve un conjunto de
registros

        ' devuelve un entero indicando el numero de filas que
se han visto afectadas por la ejecucion del objeto commmand

Comando = Nothing

Catch ex As SqlException

    ' Si no hay error en el acceso a la base de datos
    MessageBox.Show("ERROR 22:" & ControlChars.CrLf &
ex.Message & _
        ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")
    MessageBox.Show("NO SE HA PODIDO GUARDAR LA RESERVA")

    Exit Sub 'Salimos del procedimiento

End Try

MessageBox.Show("RESERVA GUARDADA CORRECTAMENTE")

Try

    ' Actualizamos el DataGridView "DG_Reservas"

    Consulta_SQL = "SELECT * FROM Reserva WHERE cliente_id
= @cliente_id"

    Comando = New SqlCommand(Consulta_SQL, Conexion)

    Comando.Parameters.Add(New SqlParameter("@cliente_id",
SqlDbType.BigInt))

    Comando.Parameters("@cliente_id").Value =
Objeto_Usuario.Id_valor

```


Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
'crear adaptador
oDataAdapter = New SqlDataAdapter(Comando)

'crear commandbuilder
oCB = New SqlCommandBuilder(oDataAdapter)

'crear dataset
oDataSet = New DataSet()
oDataAdapter.Fill(oDataSet, "Reservas")

Form_usuario.DG_Reservas.DataSource = oDataSet
Form_usuario.DG_Reservas.DataMember = "Reservas"

'Modificamos propiedades por codigo del DataGrid
DG_Reservas.Un DataGrid es un control nos permite realizar

'enlace complejo de datos con ADO.NET
Form_usuario.DG_Reservas.Columns(0).Visible = False
'Ocultar la columna 0 del DataGrid DG_Reservas Identificador de reserva
Form_usuario.DG_Reservas.Columns(7).Visible = False
Form_usuario.DG_Reservas.Columns(8).Visible = False
Form_usuario.DG_Reservas.Columns(9).Visible = False
Form_usuario.DG_Reservas.Columns(3).MinimumWidth = 120
'Ancho minimo de la columna: 120 pixeles
Form_usuario.DG_Reservas.Columns(5).MinimumWidth = 230
Form_usuario.DG_Reservas.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill 'Ajustar ancho de la columna

Catch ex As SqlException

' Si no hay error en el acceso a la base de datos
MessageBox.Show("ERROR 23:" & ControlChars.CrLf &
ex.Message & _
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

'Cerramos el formulario y volvemos al formulario de
acceso

Form_acceso.Show()

Me.Close()

End Try

Comando = Nothing

End If

'Cerramos el formulario Form_Reseva y mostramos el formulario
Form_usuario

Descarga_reserva = True

Form_usuario.Visible = True

Form_usuario.Show()

Form_usuario.Activate()

Me.Close()

End Sub

' Evento que se produce cuando pulsamos el boton "modificar
reserva" de la barra de herramientas

Private Sub TSB_Modificar_Reserva_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TSB_Modificar_Reserva.Click

    Dim Resultado As Integer

    Dim oDataAdapter As SqlDataAdapter

    Dim oDataSet As DataSet

    Dim oCB As SqlCommandBuilder
```

```

        If DateDiff(DateInterval.Day, DateTime.Today, F_entrada) <= 0
        Then 'Si la fecha de entrada es posterior a la fecha de hoy

            MessageBox.Show("Introduzca fecha de entrada posterior a
            fecha de hoy")

            Exit Sub

        ElseIf (TB_Hab.Text = "") Or (TB_NHab.Text = "0") Or
        (TB_OHab.Text = "0") Then ' Si no hemos introducido tipo de
        habitacion, numero de habitaciones y ocupacion

            MessageBox.Show("Porfavor rellene todo los campos
            obligatorios")

            Exit Sub

        Else

            Try

                ' Actualizamos la base de datos con la informacion de
                la reserva

                'Crear comando escalar

                ' La consulta "Consulta_SQL" actualiza el registro
                cuyo campo idreserva es igual al identificador de la reserva

                ' de la tabla Reserva con los valores obtenidos en el
                formulario Form_Reserva

                Consulta_SQL = "UPDATE Reserva " & _

                "SET fechainicio = @fechainicio, fechafin =
                @fechafin, precio = @precio, ocupacion = @ocupacion," & _

                "nombretomador = @nombretomador, cliente_id =
                @cliente_id, hotel_id = @hotel_id, habitacion_id=
                @habitacion_id, cantidad = @cantidad " & _
    
```



```

        "WHERE idreserva = @idreserva;"

'Crear comando
Comando = New SqlCommand(Consulta_SQL, Conexion)

' Añadir parametros al comando
Comando.Parameters.Add(New
SqlParameter("@fechainicio", SqlDbType.DateTime))

Comando.Parameters.Add(New SqlParameter("@fechafin",
SqlDbType.DateTime))

Comando.Parameters.Add(New SqlParameter("@precio",
SqlDbType.Money))

Comando.Parameters.Add(New SqlParameter("@ocupacion",
SqlDbType.TinyInt))

Comando.Parameters.Add(New
SqlParameter("@nombretomador", SqlDbType.VarChar))

Comando.Parameters.Add(New SqlParameter("@cliente_id",
SqlDbType.BigInt))

Comando.Parameters.Add(New SqlParameter("@hotel_id",
SqlDbType.BigInt))

Comando.Parameters.Add(New
SqlParameter("@habitacion_id", SqlDbType.BigInt))

Comando.Parameters.Add(New SqlParameter("@cantidad",
SqlDbType.Int))

Comando.Parameters.Add(New SqlParameter("@idreserva",
SqlDbType.BigInt))

' Actualizamos el objeto Reserva
Objeto_Reserva.fechainicio_valor = F_entrada.Date
Objeto_Reserva.fechafin_valor = F_salida.Date
    
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Objeto_Reserva.precio_valor = CSng(TB_Total.Text)

Objeto_Reserva.ocupacion_valor = CInt(TB_OHab.Text)

Objeto_Reserva.cliente_id_valor =
Objeto_Usuario.Id_valor

For contador1 = 0 To (V_Disponibilidad.Length - 1)

    If
V_Disponibilidad(contador1).Habitacion_valor.Nombre_valor =
THabitacion Then

        Objeto_Reserva.habitacion_id_valor =
V_Disponibilidad(contador1).Habitacion_valor.Idtipohabitacion_valor

        Exit For

    End If

Next

Objeto_Reserva.cantidad_valor = CInt(TB_NHab.Text)

Objeto_Reserva.nombretomador_valor = TB_Tomador.Text

Comando.Parameters("@fechainicio").Value =
Objeto_Reserva.fechainicio_valor

Comando.Parameters("@fechafin").Value =
Objeto_Reserva.fechafin_valor

Comando.Parameters("@precio").Value =
Objeto_Reserva.precio_valor

Comando.Parameters("@ocupacion").Value =
Objeto_Reserva.ocupacion_valor

Comando.Parameters("@nombretomador").Value =
TB_Tomador.Text

Comando.Parameters("@cliente_id").Value =
Objeto_Reserva.cliente_id_valor

Comando.Parameters("@hotel_id").Value =
Objeto_Reserva.hotel_id_valor

Comando.Parameters("@habitacion_id").Value =
Objeto_Reserva.habitacion_id_valor
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Comando.Parameters("@cantidad").Value =
Objeto_Reserva.cantidad_valor

Comando.Parameters("@idreserva").Value =
Objeto_Reserva.idreserva_valor

Resultado = Comando.ExecuteNonQuery()

Comando = Nothing

Catch ex As SqlException
    ' Si no hay error en el acceso a la base de datos
    MessageBox.Show("ERROR 24:" & ControlChars.CrLf &
ex.Message & _
        ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")
    MessageBox.Show("NO SE HA PODIDO MODIFICAR LA
RESERVA")

Exit Sub ' Salimos del procedimeinto

End Try

MessageBox.Show("RESERVA MODIFICADA CORRECTAMENTE")

Try
    'Averiguamos los datos de la reserva actualizada para
actualizar el DataGridView "DG_Reservas"

    ' La consulta "Consulta_SQL" selecciona todos los
campos de un registro de la tabla Reserva donde

    ' el campo cliente_id es igual al identificador del
cliente que realiza la reserva

    Consulta_SQL = "SELECT * FROM Reserva WHERE cliente_id
= @cliente_id"
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Comando = New SqlCommand(Consulta_SQL, Conexion)

Comando.Parameters.Add(New SqlParameter("@cliente_id",
SqlDbType.BigInt))

Comando.Parameters("@cliente_id").Value =
Objeto_Usuario.Id_valor

'crear adaptador

oDataAdapter = New SqlDataAdapter(Comando)

'crear commandbuilder

oCB = New SqlCommandBuilder(oDataAdapter)

'crear dataset

oDataSet = New DataSet()

oDataAdapter.Fill(oDataSet, "Reservas")

Form_usuario.DG_Reservas.DataSource = oDataSet

Form_usuario.DG_Reservas.DataMember = "Reservas"

'Modificamos propiedades por codigo del DataGridView
DG_Reservas.Un DataGridView es un control nos permite realizar

'enlace complejo de datos con ADO.NET

Form_usuario.DG_Reservas.Columns(0).Visible = False
'Ocultar la columna 0 del DataGridView DG_Reservas

Form_usuario.DG_Reservas.Columns(7).Visible = False

Form_usuario.DG_Reservas.Columns(8).Visible = False

Form_usuario.DG_Reservas.Columns(3).MinimumWidth = 120
'Ancho minimo de la columna: 120 pixeles

Form_usuario.DG_Reservas.Columns(5).MinimumWidth = 230

Form_usuario.DG_Reservas.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill 'Ajustar ancho de la columna

Catch ex As SqlException
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
' Si no hay error en el acceso a la base de datos
MessageBox.Show("ERROR 25:" & ControlChars.CrLf &
ex.Message & _
ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

' Cerramos el formulario Form_Reserva y mostramos el
formulariode acceso

Form_acceso.Show()

Me.Close()

End Try

Comando = Nothing

'Cerramos el formulario Form_Reseva y mostramos el
formulario Form_usuario

Descarga_reserva = True

Form_usuario.Visible = True

Form_usuario.Show()

Me.Close()

End If

End Sub

' Evento que se produce cuando pulsamos el boton "borrar reserva"
de la barra de herramientas

Private Sub TSB_Borrar_Reserva_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TSB_Borrar_Reserva.Click

Dim Resultado As Integer

Dim oDataAdapter As SqlDataAdapter

Dim oDataSet As DataSet

Dim oCB As SqlCommandBuilder
```


Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Try

    'Borramos dela base de datos una reserva

    'Crear comando escalar

    ' La consulta "Consulta_SQL" borra un registro de la tabla
Reserva donde

    ' el campo idreserva es igual al identificador de la
reserva

    Consulta_SQL = "DELETE FROM Reserva WHERE idreserva =
@idreserva"

    'Crear comando

    Comando = New SqlCommand(Consulta_SQL, Conexion)

    ' Añadir parametros al comando

    Comando.Parameters.Add(New SqlParameter("@idreserva",
SqlDbType.BigInt))

    Comando.Parameters("@idreserva").Value =
Objeto_Reserva.idreserva_valor

    Resultado = Comando.ExecuteNonQuery() ' Ejecutamos comando

    Comando = Nothing

Catch ex As SqlException

    ' Si no hay error en el acceso a la base de datos

    MessageBox.Show("ERROR 26:" & ControlChars.CrLf &
ex.Message & _
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

MessageBox.Show("ERROR: NO SE HA PODIDO BORRAR LA
RESERVA")

Exit Sub ' Salimos del procedimiento

End Try

MessageBox.Show("RESERVA BORRADA CORRECTAMENTE")

Try

'Actualizamos el DataGridView "DG_Reservas"

' La consulta "Consulta_SQL" selecciona todos los
registros de la tabla Reserva donde

' el campo cliente_id es igual al identificador del
cliente que realiza la reserva

Consulta_SQL = "SELECT * FROM Reserva WHERE cliente_id =
@cliente_id"

Comando = New SqlCommand(Consulta_SQL, Conexion)

Comando.Parameters.Add(New SqlParameter("@cliente_id",
SqlDbType.BigInt))

Comando.Parameters("@cliente_id").Value =
Objeto_Usuario.Id_valor

'crear adaptador

oDataAdapter = New SqlDataAdapter(Comando)

'crear commandbuilder

oCB = New SqlCommandBuilder(oDataAdapter)

'crear dataset

oDataSet = New DataSet()

oDataAdapter.Fill(oDataSet, "Reservas")

Form_usuario.DG_Reservas.DataSource = oDataSet
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Form_usuario.DG_Reservas.DataMember = "Reservas"

'Modificamos propiedades por codigo del DataGrid
DG_Reservas.Un DataGrid es un control nos permite realizar

'enlace complejo de datos con ADO.NET

Form_usuario.DG_Reservas.Columns(0).Visible = False
'Ocultar la columna 0 del DataGrid DG_Reservas

Form_usuario.DG_Reservas.Columns(7).Visible = False

Form_usuario.DG_Reservas.Columns(8).Visible = False

Form_usuario.DG_Reservas.Columns(3).MinimumWidth = 120
'Ancho minimo de la columna: 120 pixeles

Form_usuario.DG_Reservas.Columns(5).MinimumWidth = 230

Form_usuario.DG_Reservas.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill 'Ajustar ancho de la columna

Catch ex As SqlException

' Si no hay error en el acceso a la base de datos
MessageBox.Show("ERROR 27:" & ControlChars.CrLf &
ex.Message & _
ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

' Cerramos el formulario Form_Reserva y mostramos el
formulariode acceso

Form_acceso.Show()

Me.Close()

End Try

Comando = Nothing

'Cerramos el formulario Form_Reserva y mostramos el formulario
Form_usuario

Descarga_reserva = True

Form_usuario.Visible = True

Form_usuario.Show()

Form_usuario.Activate()

Me.Close()
```



```
End Sub

'Evento que se ejecuta antes de que se muestre un formulario por
primera vez.

Private Sub Form_Reserva_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load

    'Actualizamos los controles del formulario Form_Reserva con
los datos correspondientes

    'dependiendo de coo se ha accedido al formulario

Try

    Descarga_reserva = False 'Descarga_reserva = False indica
que el formulario Form_Reserva se ha cargado

    If Ver_reserva = 1 Then 'Hemos accedido al formulario de
informacion de reserva a traves del boton "modificar" de la barra de
herramientas

        'Actualizamos los controles del formulario con los
valores de los obkjetos Reserva y Hotel

        TB_Cliente_Reserva.Text =
Objeto_Reserva.nombretomador_valor

        TB_Tomador.Text = Objeto_Reserva.nombretomador_valor

        TB_Hotel_Reserva.Text = Hotel.nombre_valor

        TB_Direccion_Reserva.Text = Hotel.direccion_valor

        TB_Hab.Text = Objeto_Habitacion.Nombre_valor

        TB_CHabitacion.Text =
Objeto_Habitacion.Capacidad_valor

        TB_Importe.Text = Objeto_Habitacion.Precio_valor

        TB_Fecha_entrada.Text =
Objeto_Reserva.fechainicio_valor.ToLongDateString
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
TB_Fecha_salida.Text =
Objeto_Reserva.fechafin_valor.ToLongDateString

DTP_F_entrada.Value =
Objeto_Reserva.fechainicio_valor.ToLongDateString

DTP_F_salida.Value =
Objeto_Reserva.fechafin_valor.ToLongDateString

TB_NHab.Text = Objeto_Reserva.cantidad_valor.ToString
TB_OHab.Text = Objeto_Reserva.ocupacion_valor
TB_Total.Text = Objeto_Reserva.precio_valor.ToString

GB_D_obligatorios.Enabled = False

ElseIf (Ver_reserva = 2) Then 'Hemos accedido al
formulario de informacion de reserva a traves del boton "consultar
disponibilidad" en pestaña reserva

TB_Cliente_Reserva.Text = Objeto_Usuario.Nombre_valor
& " " & _
Objeto_Usuario.Apellidos_valor
TB_Tomador.Text = TB_Cliente_Reserva.Text
TB_Hotel_Reserva.Text = hotell
TB_Direccion_Reserva.Text =
Form_usuario.TB_Direccion_Hotel.Text
TB_Fecha_entrada.Text =
DTP_F_entrada.Value.ToLongDateString
TB_Fecha_salida.Text =
DTP_F_salida.Value.ToLongDateString

GB_D_obligatorios.Enabled = False
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
ElseIf (Ver_reserva = 3) Then 'Hemos accedido al  
formulario de informacion de reserva a traves del boton "borrar" de la  
barra de herramientas con la pestaña listado reservas seleccionada
```

```
'Actualizamos los controles del formulario con los  
valores de los objetos Reserva y Hotel
```

```
TB_Cliente_Reserva.Text =  
Objeto_Reserva.nombretomador_valor
```

```
TB_Tomador.Text = Objeto_Reserva.nombretomador_valor
```

```
TB_Hotel_Reserva.Text = Hotel.nombre_valor
```

```
TB_Direccion_Reserva.Text = Hotel.direccion_valor
```

```
TB_Hab.Text = Objeto_Habitacion.Nombre_valor
```

```
TB_CHabitacion.Text =  
Objeto_Habitacion.Capacidad_valor
```

```
TB_Importe.Text = Objeto_Habitacion.Precio_valor
```

```
TB_Fecha_entrada.Text =  
Objeto_Reserva.fechainicio_valor.ToLongDateString
```

```
TB_Fecha_salida.Text =  
Objeto_Reserva.fechafin_valor.ToLongDateString
```

```
DTP_F_entrada.Value =  
Objeto_Reserva.fechainicio_valor.ToLongDateString
```

```
DTP_F_salida.Value =  
Objeto_Reserva.fechafin_valor.ToLongDateString
```

```
TB_Total.Text = Objeto_Reserva.precio_valor.ToString
```

```
TB_NHab.Text = Objeto_Reserva.cantidad_valor.ToString
```

```
TB_OHab.Text = Objeto_Reserva.ocupacion_valor
```

```
GB_D_obligatorios.Enabled = False
```

```
GB_Disponibilidad.Enabled = False
```

```
End If
```

```

Catch ex As Exception

    ' Si no hay error en el acceso a la base de datos
    MessageBox.Show("ERROR 28:" & ControlChars.CrLf &
ex.Message & _
        ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor")

    'Cerramos el formulario Form_Reserva y mostramos el
formulario Form_usuario

    Descarga_reserva = True

    Form_usuario.Visible = True

    Form_usuario.Show()

    Form_usuario.Activate()

    Me.Close()

End Try

End Sub

' Evento que se produce cuando el valor del control NumericUpDown
"NUD_Nhab" cambia

Private Sub NUD_Nhab_ValueChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles NUD_Nhab.ValueChanged

    Dim Dias_reserva As Integer

    'Calculamos la cantidad de dias reservados

    Dias_reserva = DateDiff(DateInterval.Day,
CDate(TB_Fecha_entrada.Text), CDate(TB_Fecha_salida.Text)) + 1

    TB_NHab.Text = CStr(NUD_Nhab.Value) 'Actualizamos propiedad
Text del control TextBox "TB_NHab"

    'Actualizamos propiedad Text del control TextBox "TB_Total"

    'Importe total = (dias reservados *importe por dia)*Numero de
habitaciones reservadas

```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
TB_Total.Text = CStr((CSng(TB_Importe.Text) * Dias_reserva) *  
NUD_Nhab.Value)
```

```
End Sub
```

```
'Evento que se produce cuando hacemos click en el boton  
"B_Consulta"
```

```
Private Sub B_Consulta_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles B_Consulta.Click
```

```
Dim D_entrada As Integer ' variable para almacenar la  
diferencia de dias entre hoy y la fecha de entrada
```

```
Dim dias As Integer ' variable para almacenar la diferencia de  
dias entre la fecha de entrada y la fecha de salida
```

```
Dim Sum As Integer ' variable para almacenar el numero de  
habitaciones reservadas de un tipo de habitacion
```

```
Dim n As Integer ' variable para almacenar el numero de tipos  
de habitaciones del hotel de la reserva
```

```
Dim contador As Integer ' variable para utilizar en el bucle  
for
```

```
Dim Habitacion As Clase_Tipo_Habitacion ' Habitacion ->  
variable para almacenar un objeto "Tipo_Habitacion"
```

```
Dim Disp_habitacion As Disponibilidad ' Disp_habitacion ->  
variable para almacenar un objeto "Disponibilidad"
```

```
'Variables para la conexión y consulta a la base de datos
```

```
Dim Datos As SqlDataReader
```

```
CB_THabitaciones.Items.Clear() 'Borramos todos los items del  
ComboBox "CB_THabitaciones"
```

```
F_entrada = DTP_F_entrada.Value.Date ' Guardamos fecha de  
entrada
```

```
F_salida = DTP_F_salida.Value.Date ' Guardamos fecha de  
salida
```


Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
D_entrada = DateDiff(DateInterval.Day, DateTime.Today,
F_entrada)

dias = DateDiff(DateInterval.Day, F_entrada, F_salida)

If (D_entrada <= 0) Then 'Si hemos seleccionado una fecha de
entrada anterior a la fecha de hoy

    MessageBox.Show("Introduzca fecha de entrada posterior a
la fecha de hoy")

    Exit Sub

ElseIf dias < 0 Then ' Si hemos seleccionado una fecha de
salida anterior a la fecha de entrada

    MessageBox.Show("Introduzca fecha de salida igual o
posterior a la fecha de entrada")

    Exit Sub

Else 'si hemos seleccionado correctamente la fecha de entrada
y de salida

    GB_D_obligatorios.Enabled = True 'Habilitamos el GroupBox
"GB_D_obligatorios"

    'Modificamos las propiedades de color del boton B_Consulta
    B_Consulta.BackColor = SystemColors.ControlLight
    B_Consulta.ForeColor = SystemColors.ControlText

    'Modificamos las propiedades de color de primer plano
    CB_THabitaciones.ForeColor = Color.Red
    NUD_Nhab.ForeColor = Color.Red
    NUD_OHabitacion.ForeColor = Color.Red

    ReDim V_Disponibilidad(0) ' Creamos vector con 0 elementos
    n = -1 'inicializamos n

    Try

        'Averiguamos los tipos de habitacion del hotel donde
se quiere realizar la reserva
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
' La consulta "Consulta_SQL" realiza la operacion  
INNER JOIN, la cual realiza una combinacion con todos aquellos  
  
' registros de las dos tablas (Habitacion y  
TipoHabitacion) en el que el campo comun de ambas (identificador de  
habitacion)  
  
' tenga el mismo valor (el valor del identificador de  
la habitacion) y donde el campo hotel_id de la tabla  
  
' TipoHabitacion tenga el valor del indentificador del  
hotel donde se realiza la reserva  
  
' Seleccionamos los campos de la tabla Habitacion y  
TipoHabitacion.
```

```
Consulta_SQL = "SELECT  
TipoHabitacion.hotel_id,TipoHabitacion.habitacion_id,TipoHabitacion.de  
scripcion,TipoHabitacion.precio," & _
```

```
"TipoHabitacion.numhabitaciones,Habitacion.nombre,Habitacion.capacidad  
FROM Habitacion INNER JOIN TipoHabitacion ON " & _
```

```
"TipoHabitacion.habitacion_id =  
Habitacion.idhabitacion WHERE TipoHabitacion.hotel_id = @hotel_id"
```

```
Comando = New SqlCommand(Consulta_SQL, Conexion)  
  
Comando.Parameters.Add(New SqlParameter("@hotel_id",  
SqlDbType.BigInt))  
  
Comando.Parameters("@hotel_id").Value = idhotel1  
  
Datos = Comando.ExecuteReader()
```

```
While Datos.Read()  
  
'Guardamos la informacion de cada tipo de  
habitacion en un objeto habitacion  
  
Habitacion = New Clase_Tipo_Habitacion  
  
Habitacion.Hotel_id_valor = Datos("hotel_id")  
  
Habitacion.Idtipohabitacion_valor =  
Datos("habitacion_id")
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Habitacion.Descripcion_valor =
Datos("descripcion")

Habitacion.Precio_valor = Datos("precio")

Habitacion.Numhabitaciones_valor =
Datos("numhabitaciones")

Habitacion.Nombre_valor = Datos("nombre")

Habitacion.Capacidad_valor = Datos("capacidad")

'Guardamos informacion de la disponibilidad de
cada tipo de habitacion

Disp_habitacion = New Disponibilidad(Habitacion,
Habitacion.Numhabitaciones_valor)

n = n + 1 'incrementamos variable que cuenta el
numero de tipos de habitacion

ReDim Preserve V_Disponibilidad(n)
'Redimensionamos el vector manteniendo los datos

V_Disponibilidad(n) = Disp_habitacion 'Almacenamos
el objeto disponibilidad en el vector "V_Disponibilidad"

End While

Datos.Close()

Datos = Nothing

Comando = Nothing

Catch ex As SqlException

' Si no hay error en el acceso a la base de datos
MessageBox.Show("ERROR 29:" & ControlChars.CrLf &
ex.Message & _
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

Exit Sub 'Salimos del procedimiento

End Try

For contador = 0 To (V_Disponibilidad.Length - 1) 'Para
cada objeto Disponibilidad almacenado en el vector "V_Disponibilidad"

Try

'Averiguamos el numero de habitaciones reservadas
de cada tipo de habitacion del hotel de la reserva

' entre las fechas F_entrada y F_salida

'La funcion CONVERT(datetime,fecha,103) convierte
una expresión fecha de tipo string a un tipo de datos

'datetime con formato dd/mm/aa

' La consulta "Consulta_SQL" realiza el suma de
todos los valores contenidos en el campo cantidad

' de la tabla reserva donde el campo
tipohabitacion_id es igual al tipo de habitacion del hotel de reserva

' y fechainicio esta comprendida entre la fecha de
inicio y la fecha de fin de la reserva o

' fechafin esta comprendida entre la fecha de
inicio y la fecha de fin de la reserva o

' fechainicio es anterior a la fecha de inicio y
fechafin es posterior a la fecha de fin de la reserva

Consulta_SQL = "SELECT SUM(cantidad)AS sum FROM
Reserva WHERE habitacion_id = @habitacion_id " & _

"AND ((CONVERT(datetime,fechainicio,103) BETWEEN
CONVERT(datetime,@inicio,103) AND CONVERT(datetime,@fin,103)) OR " & _

"(CONVERT(datetime,fechafin,103) BETWEEN
CONVERT(datetime,@inicio,103) AND CONVERT(datetime,@fin,103)) OR " & _

"((CONVERT(datetime,fechainicio,103)
<=CONVERT(datetime,@inicio,103)) AND
(CONVERT(datetime,fechafin,103)>=CONVERT(datetime,@fin,103)))"
```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Comando = New SqlCommand(Consulta_SQL, Conexion)

Comando.Parameters.Add(New
SqlParameter("@habitacion_id", SqlDbType.BigInt))

Comando.Parameters.Add(New SqlParameter("@inicio",
SqlDbType.DateTime))

Comando.Parameters.Add(New SqlParameter("@fin",
SqlDbType.DateTime))

Comando.Parameters("@habitacion_id").Value =
V_Disponibilidad(contador).Habitacion_valor.Idtipohabitacion_valor

Comando.Parameters("@inicio").Value = F_entrada

Comando.Parameters("@fin").Value = F_salida

Datos = Comando.ExecuteReader() 'ExecuteReader
devuelve un conjunto de registros

If Datos.Read() Then ' Si hay un registro

    If Not (Convert.IsDBNull(Datos("sum"))) Then
'Si no nulo el campo sumatorio

        Sum = Datos("sum") 'Almacenamos el
sumatorio

        'Almacenamos el numero de habitaciones
diponibles de cada tipo de habitacion

V_Disponibilidad(contador).Num_disponibles_valor =
V_Disponibilidad(contador).Habitacion_valor.Numhabitaciones_valor _
- Sum

    Else 'Si es nulo el campo sumatorio -> no hay
habitaciones reservadas

V_Disponibilidad(contador).Num_disponibles_valor =
V_Disponibilidad(contador).Habitacion_valor.Numhabitaciones_valor

    End If

End If
```



```

        Datos.Close()

        Comando = Nothing

        Datos = Nothing

    Catch ex As SqlException

        ' Si no hay error en el acceso a la base de datos
        MessageBox.Show("ERROR 30:" & ControlChars.CrLf &
ex.Message & _
        ControlChars.CrLf & " SERVIDOR:SQL SERVER 2008 R2
Express. Miguel Monzo Pastor ")

        Exit Sub 'Salimos del procedimiento

    End Try

Next

        GB_D_obligatorios.Enabled = True 'Habilitamos el GroupBox
"GB_D_obligatorios"

        'Actualizamos los TextBox de fecha de entrada y fecha de
salida

        TB_Fecha_entrada.Text = F_entrada.ToLongDateString

        TB_Fecha_salida.Text = F_salida.ToLongDateString

        MessageBox.Show("CONSULTA REALIZADA CORRECTAMENTE")

    End If

        'Agregamos items al ComboBox CB_Thabitaciones

        'El ComboBox CB_Thabitaciones contiene la lista de tipos de
habitacion

        For contador = 0 To (V_Disponibilidad.Length - 1)

```

Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
CB_THabitaciones.Items.Add(V_Disponibilidad(contador).Habitacion_valor  
.Nombre_valor)
```

```
Next
```

```
If Ver_reserva = 1 Then 'Hemos accedido al formulario de  
informacion de reserva a traves del boton "modificar" de la barra de  
herramientas
```

```
TSB_Modificar_Reserva.Enabled = True
```

```
TSB_Guardar_Reserva.Enabled = False
```

```
TSB_Borrar_Reserva.Enabled = False
```

```
ElseIf Ver_reserva = 2 Then 'Hemos accedido al formulario de  
informacion de reserva a traves del boton "consultar disponibilidad"  
en pestaña reserva
```

```
TSB_Modificar_Reserva.Enabled = False
```

```
TSB_Guardar_Reserva.Enabled = True
```

```
TSB_Borrar_Reserva.Enabled = False
```

```
End If
```

```
End Sub
```

```
' Evento que se produce cuando el valor del control  
"DTP_F_entrada" cambia
```

```
Private Sub DTP_F_entrada_ValueChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
DTP_F_entrada.ValueChanged
```

```
F_entrada = DTP_F_entrada.Value ' Almacenamos el valor en la  
variable F_entrada
```

```
'Modificamos las propiedades de color
```

```
GB_D_obligatorios.BackColor = SystemColors.ControlLight 'Color  
de fondo
```

```
CB_THabitaciones.ForeColor = SystemColors.ControlText 'Color  
de primer plano
```

```
NUD_Nhab.ForeColor = SystemColors.ControlText
```

```
NUD_OHabitacion.ForeColor = SystemColors.ControlText
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
        GB_D_obligatorios.Enabled = False 'Deshabilitamos el GroupBox
        "GB_D_obligatorios"

    End Sub

    ' Evento que se produce cuando el valor del control DateTimePicker
    "DTP_F_salida" cambia

    Private Sub DTP_F_salida_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
DTP_F_salida.ValueChanged

        F_salida = DTP_F_salida.Value ' Almacenamos el valor en la
variable F_salida

        'Modificamos las propiedades de color

        GB_D_obligatorios.BackColor = SystemColors.ControlLight 'Color
de fondo

        CB_THabitaciones.ForeColor = SystemColors.ControlText 'Color
de primer plano

        NUD_Nhab.ForeColor = SystemColors.ControlText

        NUD_OHabitacion.ForeColor = SystemColors.ControlText

        GB_D_obligatorios.Enabled = False 'Deshabilitamos el GroupBox
        "GB_D_obligatorios"

    End Sub

    ' Evento que se produce cuando el valor del control NumericUpDown
    "NUD_OHabitacion" cambia

    Private Sub NUD_OHabitacion_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NUD_OHabitacion.ValueChanged

        'Actualizamos el valor de la propiedad Text del TextBox
        "TB_OHab"

        TB_OHab.Text = NUD_OHabitacion.Value.ToString

    End Sub
```



```
End Class
```

FORMULARIO SOBRE

```
Public NotInheritable Class Sobre
```

```
    Private Sub Sobre_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
End Sub
```

```
    Private Sub OKButton_Click(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles OKButton.Click
```

```
        Me.Close()
```



```
End Sub
```

```
End Class
```

CLASES

```
Public Class Clase_Usuario  
    Private Id As Int64  
    Private Tipo_Usuario As String  
    Private Login As String  
    Private Password As String  
    Private Email As String  
    Private Fechaalta As DateTime  
    Private Ultimoacceso As DateTime  
    Private Nombre As String  
    Private Apellidos As String  
    Private NIF As String
```

```
Private Domicilio As String

Private Localidad As String

Private Provincia As String

Private Codpostal As String

Private Telefono As String

Public Property Nombre_valor() As String

    Get

        Return Nombre

    End Get

    Set(ByVal value As String)

        Nombre = value

    End Set

End Property

Public Property Apellidos_valor() As String

    Get

        Return Apellidos

    End Get

    Set(ByVal value As String)

        Apellidos = value

    End Set

End Property

Public Property NIF_valor() As String

    Get

        Return NIF

    End Get

    Set(ByVal value As String)
```

```
NIF = value

End Set

End Property

Public Property Domicilio_valor() As String

    Get

        Return Domicilio

    End Get

    Set(ByVal value As String)

        Domicilio = value

    End Set

End Property

Public Property Localidad_valor() As String

    Get

        Return Localidad

    End Get

    Set(ByVal value As String)

        Localidad = value

    End Set

End Property

Public Property Provincia_valor() As String

    Get

        Return Provincia

    End Get

    Set(ByVal value As String)

        Provincia = value

    End Set

End Property

Public Property Codpostal_valor() As String
```

```
Get
    Return Codpostal
End Get

Set(ByVal value As String)
    Codpostal = value
End Set
End Property

Public Property Telefono_valor() As String
Get
    Return Telefono
End Get

Set(ByVal value As String)
    Telefono = value
End Set
End Property

Public Property Id_valor() As Int64

Get
    Return Id
End Get

Set(ByVal value As Int64)
    Id = value
End Set
End Property

Public Property Tipo_Usuario_valor() As String
Get
    Return Tipo_Usuario
End Get
```

```
Set (ByVal value As String)
    Tipo_Usuario = value
End Set
End Property
Public Property Login_valor() As String
    Get
        Return Login
    End Get
    Set (ByVal value As String)
        Login = value
    End Set
End Property
Public Property Password_valor() As String
    Get
        Return Password
    End Get
    Set (ByVal value As String)
        Password = value
    End Set
End Property
Public Property Email_valor() As String
    Get
        Return Email
    End Get
    Set (ByVal value As String)
        Email = value
    End Set
End Property
Public Property Fechaalta_valor() As DateTime
```

```
Get
    Return Fechaalta
End Get

Set(ByVal value As DateTime)
    Fechaalta = value
End Set
End Property

Public Property Ultimoacceso_valor() As DateTime
Get
    Return Ultimoacceso
End Get

Set(ByVal value As DateTime)
    Ultimoacceso = value
End Set
End Property

Public Sub New(ByVal Login_valor As String, ByVal Password_valor
As String)
    Login = Login_valor
    Password = Password_valor
End Sub

End Class

Public Class Clase_Tipo_Habitacion
    Private Idtipohabitacion As Int64
    Private Nombre As String
    Private Descripcion As String
    Private Capacidad As Int16
```



```
Private Precio As Single

Private Numhabitaciones As Int16

Private Hotel_id As Int64

Public Property Idtipohabitacion_valor() As Int64
    Get
        Return Idtipohabitacion
    End Get
    Set(ByVal value As Int64)
        Idtipohabitacion = value
    End Set
End Property

Public Property Nombre_valor() As String
    Get
        Return Nombre
    End Get
    Set(ByVal value As String)
        Nombre = value
    End Set
End Property

Public Property Descripcion_valor() As String
    Get
        Return Descripcion
    End Get
    Set(ByVal value As String)
        Descripcion = value
    End Set
```



```
End Property

Public Property Capacidad_valor() As String

    Get

        Return Capacidad

    End Get

    Set(ByVal value As String)

        Capacidad = value

    End Set

End Property

Public Property Precio_valor() As Single

    Get

        Return Precio

    End Get

    Set(ByVal value As Single)

        Precio = value

    End Set

End Property

Public Property Numhabitaciones_valor() As Int16

    Get

        Return Numhabitaciones

    End Get

    Set(ByVal value As Int16)

        Numhabitaciones = value

    End Set

End Property

Public Property Hotel_id_valor() As Int64

    Get

        Return Hotel_id

    End Get
```

```
        Set(ByVal value As Int64)
            Hotel_id = value
        End Set
    End Property
End Class

Public Class Disponibilidad
    Private Habitacion As Clase_Tipo_Habitacion
    Private Num_disponibles As Int16

    Public Property Habitacion_valor() As Clase_Tipo_Habitacion
        Get
            Return Habitacion
        End Get
        Set(ByVal value As Clase_Tipo_Habitacion)
            Habitacion = value
        End Set
    End Property

    Public Property Num_disponibles_valor() As Int16
        Get
            Return Num_disponibles
        End Get
        Set(ByVal value As Int16)
            Num_disponibles = value
        End Set
    End Property

    Public Sub New(ByVal THabitacion As Clase_Tipo_Habitacion, ByVal
Disponibles As Integer)
        Habitacion = THabitacion
        Num_disponibles = Disponibles
    End Sub
End Class
```

```
End Sub

End Class

Public Class Clase_Reserva

    Private idreserva As Int64

    Private fechainicio As DateTime

    Private fechafin As DateTime

    Private precio As Single

    Private ocupacion As Integer

    Private cliente_id As Int64

    Private habitacion_id As Int64

    Private hotel_id As Int64

    Private cantidad As Integer

    Private nombretomador As String

    Public Property idreserva_valor() As Int64

        Get

            Return idreserva

        End Get

        Set(ByVal value As Int64)

            idreserva = value

        End Set

    End Property

    Public Property fechainicio_valor() As DateTime

        Get

            Return fechainicio

        End Get

        Set(ByVal value As DateTime)
```

```
        fechainicio = value
    End Set
End Property

Public Property fechafin_valor() As DateTime
    Get
        Return fechafin
    End Get
    Set(ByVal value As DateTime)
        fechafin = value
    End Set
End Property

Public Property precio_valor() As Single
    Get
        Return precio
    End Get
    Set(ByVal value As Single)
        precio = value
    End Set
End Property

Public Property ocupacion_valor() As Integer
    Get
        Return ocupacion
    End Get
    Set(ByVal value As Integer)
        ocupacion = value
    End Set
End Property
```

```
End Property
```

```
Public Property cliente_id_valor() As Int64
```

```
Get
```

```
Return cliente_id
```

```
End Get
```

```
Set(ByVal value As Int64)
```

```
cliente_id = value
```

```
End Set
```

```
End Property
```

```
Public Property habitacion_id_valor() As Int64
```

```
Get
```

```
Return habitacion_id
```

```
End Get
```

```
Set(ByVal value As Int64)
```

```
habitacion_id = value
```

```
End Set
```

```
End Property
```

```
Public Property hotel_id_valor() As Int64
```

```
Get
```

```
Return hotel_id
```

```
End Get
```

```
Set(ByVal value As Int64)
```

```
hotel_id = value
```

```
End Set
```

```
End Property
```

```
Public Property cantidad_valor() As Int16
    Get
        Return cantidad
    End Get
    Set(ByVal value As Int16)
        cantidad = value
    End Set
End Property

Public Property nombretomador_valor() As String
    Get
        Return nombretomador
    End Get
    Set(ByVal value As String)
        nombretomador = value
    End Set
End Property

End Class

Public Class Clase_Hotel
    Private idhotel As Int64
    Private nombre As String
    Private direccion As String
    Private categoria As Integer
    Private telefono As String
    Private descripcion As String
```

```
Public Property idhotel_valor() As Int64
    Get
        Return idhotel
    End Get
    Set(ByVal value As Int64)
        idhotel = value
    End Set
End Property
```

```
Public Property nombre_valor() As String
    Get
        Return nombre
    End Get
    Set(ByVal value As String)
        nombre = value
    End Set
End Property
```

```
Public Property direccion_valor() As String
    Get
        Return direccion
    End Get
    Set(ByVal value As String)
        direccion = value
    End Set
End Property
```

```
Public Property categoria_valor() As Integer
```



```
Get
    Return categoria
End Get

Set(ByVal value As Integer)
    categoria = value
End Set
End Property

Public Property telefono_valor() As String
    Get
        Return telefono
    End Get
    Set(ByVal value As String)
        telefono = value
    End Set
End Property

Public Property descripcion_valor() As String
    Get
        Return descripcion
    End Get
    Set(ByVal value As String)
        descripcion = value
    End Set
End Property
```



```
End Class
```

MODULOS

```
Imports System.Drawing.Printing
```

```
Imports System.Data.SqlClient
```

```
Module Module1
```

```
    Public Usuario As String
```

```
    Public Password As String
```

```
    Public Registrarse As Boolean
```

```
    Public Objeto_Usuario As Clase_Usuario
```

```
    Public F_entrada As DateTime
```

```
    Public F_salida As DateTime
```

```
    Public Fecha_entrada As DateTime
```

```
    Public Fecha_salida As DateTime
```



Aplicación en Visual Basic .NET: Reservas de Hotel 1.0

```
Public hotel_consulta1 As String = ""

Public hotel2 As String = ""

Public V_Disponibilidad() As Disponibilidad = {} ' Vector que
almacena objetos Disponibilidad

Public Descarga_reserva As Boolean = True ' Variable
Descarga_reserva = true indica que el formulario Form_Reserva no esta
cargado

Public prtSettings As PrinterSettings

Public Objeto_Reserva As Clase_Reserva = New Clase_Reserva

Public Ver_reserva As Integer = 0

Public Hotel As Clase_Hotel = New Clase_Hotel

Public Objeto_Habitacion As Clase_Tipo_Habitacion = New
Clase_Tipo_Habitacion

Public hotell As String = ""

Public idhotell As Int64

Public list_reservas As Boolean = False

Public editable As Boolean = False

Public Conexion As SqlConnection = New SqlConnection ' Creo el
objeto Conexion

Public Consulta_SQL As String

Public Comando As System.Data.SqlClient.SqlCommand

Public Function seleccionarImpresora() As Boolean

    Dim prtDialog As New PrintDialog

    If prtSettings Is Nothing Then

        prtSettings = New PrinterSettings
```

```
End If
```

```
With prtDialog
```

```
    .AllowPrintToFile = True
```

```
    .AllowSelection = True
```

```
    .AllowSomePages = True
```

```
    .PrintToFile = True
```

```
    .ShowHelp = True
```

```
    .ShowNetwork = True
```

```
    .PrinterSettings = prtSettings
```

```
If .ShowDialog() = DialogResult.OK Then
```

```
    prtSettings = .PrinterSettings
```

```
Else
```

```
    Return False
```

```
End If
```

```
End With
```

```
Return True
```

```
End Function
```

```
End Module
```



2.6 Pruebas

Las pruebas se realizaron al finalizar la codificación de cada funcionalidad (alta, baja, creación de una reserva, modificación de una reserva etc.). Consistía en la comprobación de la correcta implementación de la interfaz gráfica, la inserción de información a través de dicha interfaz gráfica en la Base de datos (actualización, eliminación, creación de registros) y posterior comprobación de las tablas correspondientes y por último la refactorización del código para optimizarlo.

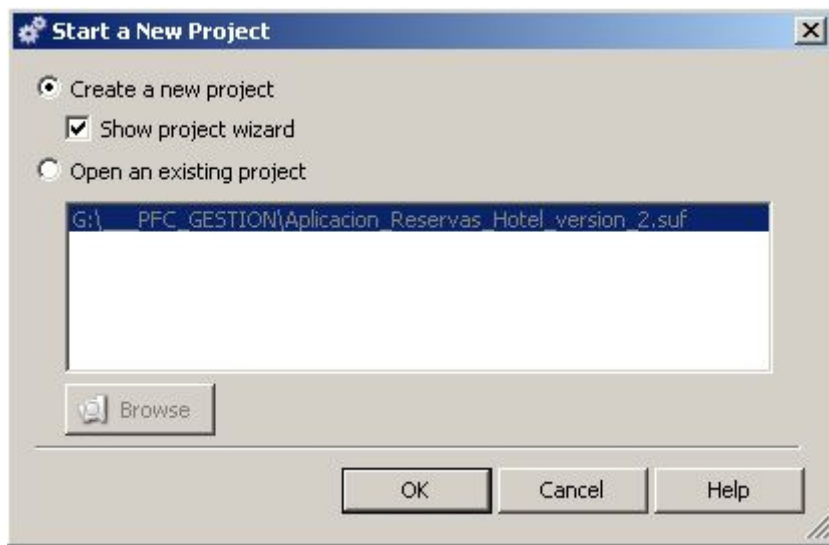
En todo momento hacía uso del depurador (debbuger) de Visual Studio, que es una utilidad esencial para detectar errores de ejecución.

2.7 Instalación

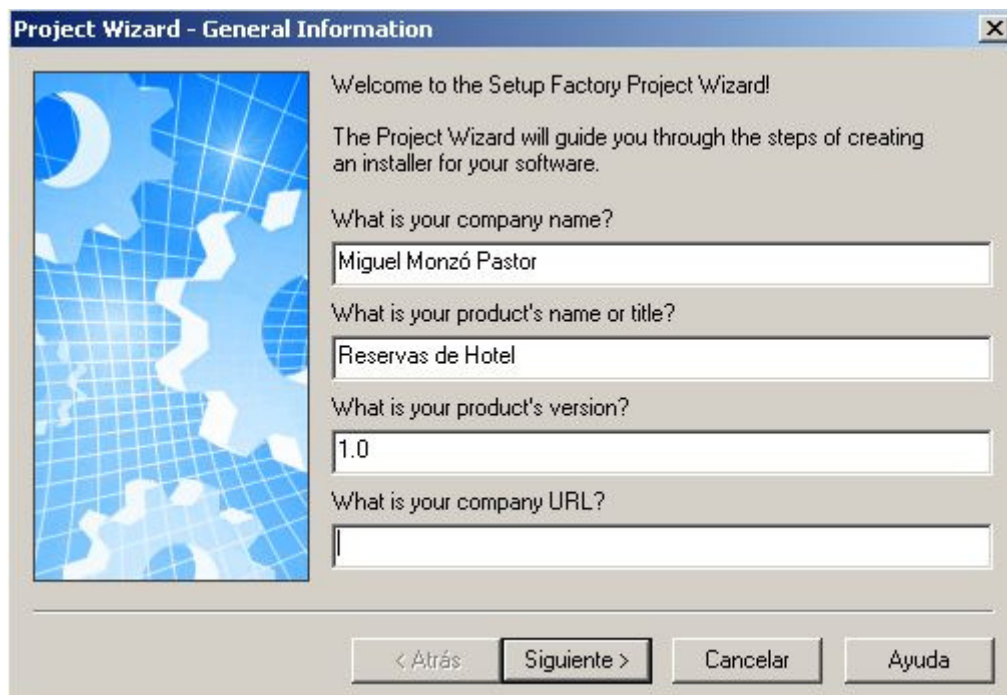
La instalación de la aplicación Reservas de hotel 1.0 se realiza de forma automatizada utilizando para ello el programa Setup Factory 9 (versión de evaluación).

Pasos para crear el fichero setup de instalación automatizada (instalador):

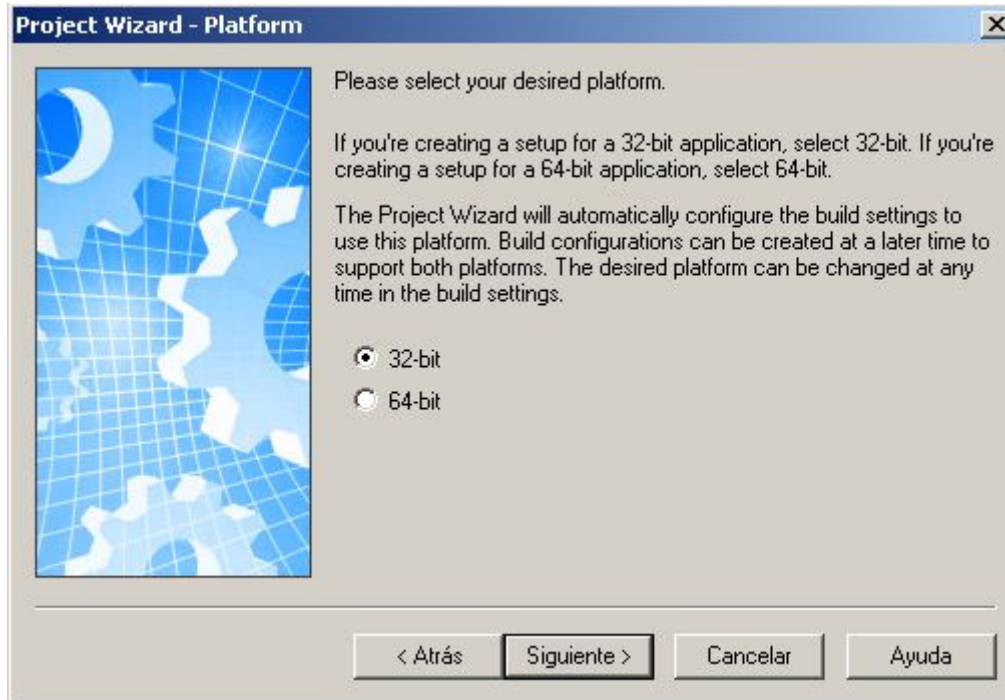
1. Crear Nuevo proyecto con el asistente de proyectos: File -> New Project y a continuación seleccionar Show project wizard



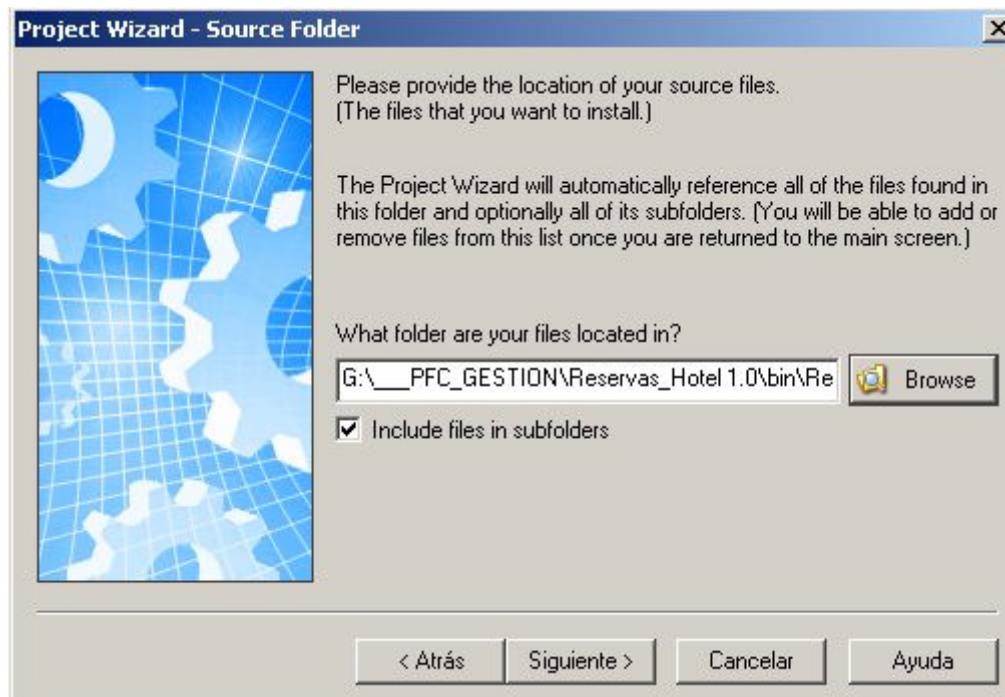
2. Pantalla 1 del asistente de proyectos: Información general



3. Pantalla 2 del asistente de proyectos: Selección de plataforma 32 bits o 64 bits.



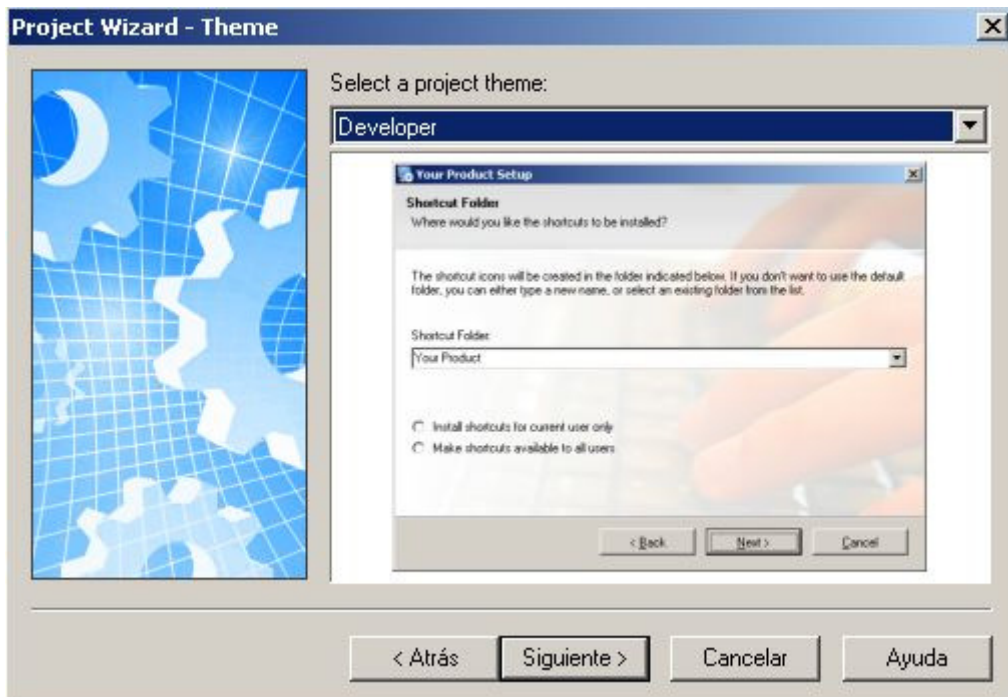
4. Pantalla 3 del asistente de proyectos: Selección de carpeta fuente.



5. Pantalla 4 del asistente de proyectos: Configuración del instalador. Seleccionar Wizard (asistente)



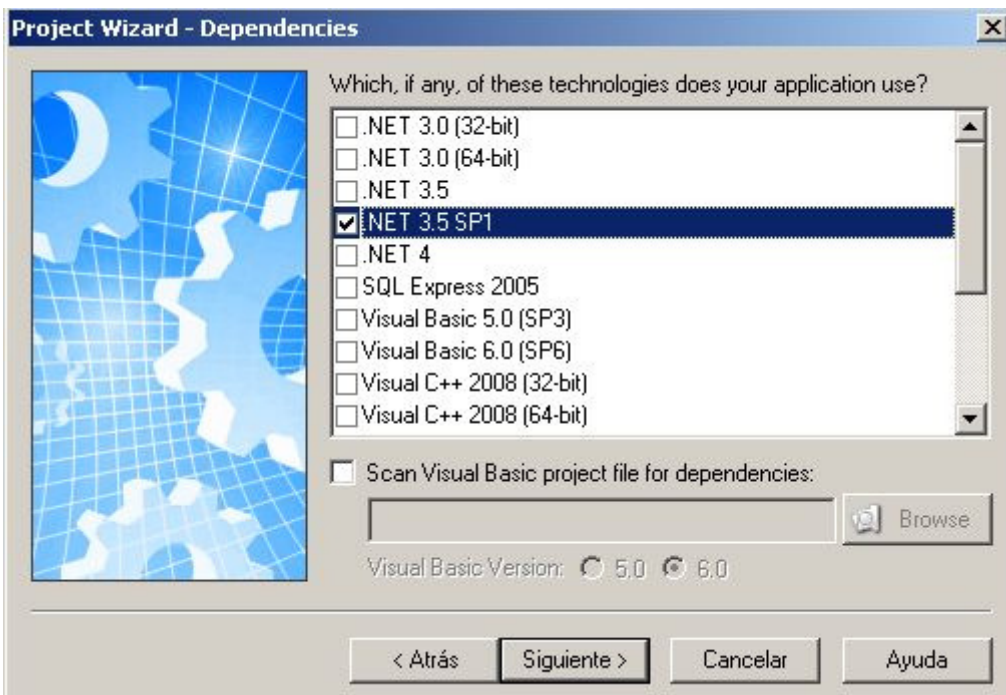
6. Pantalla 5 del asistente de proyectos: Selección de Tema del instalador.



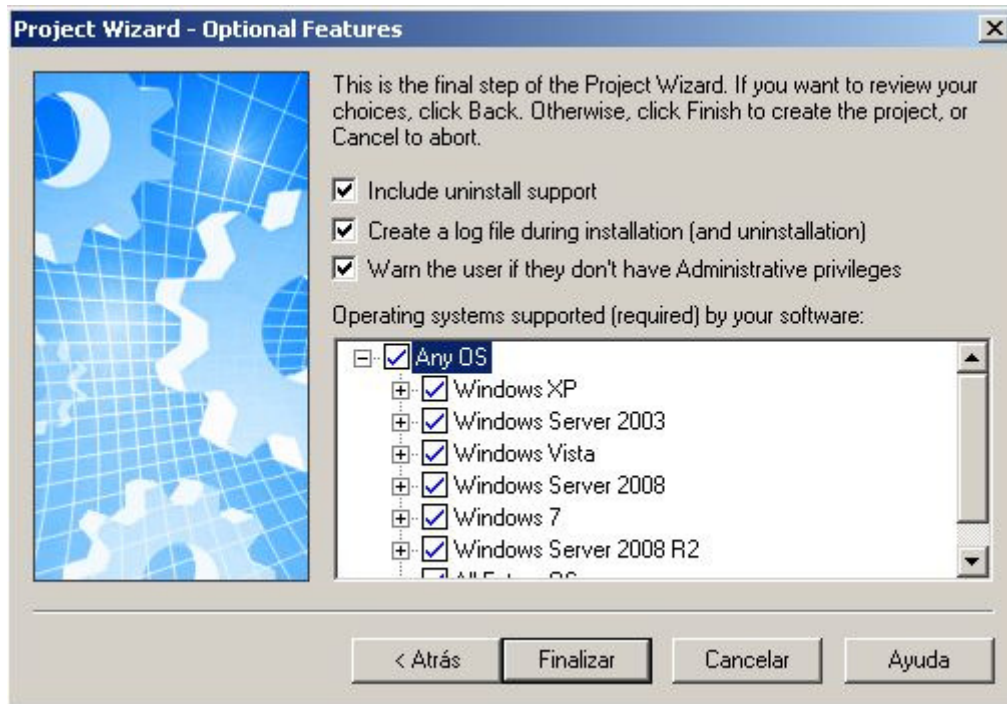
7. Pantalla 6 del asistente de proyectos: Selección de idioma del instalador.



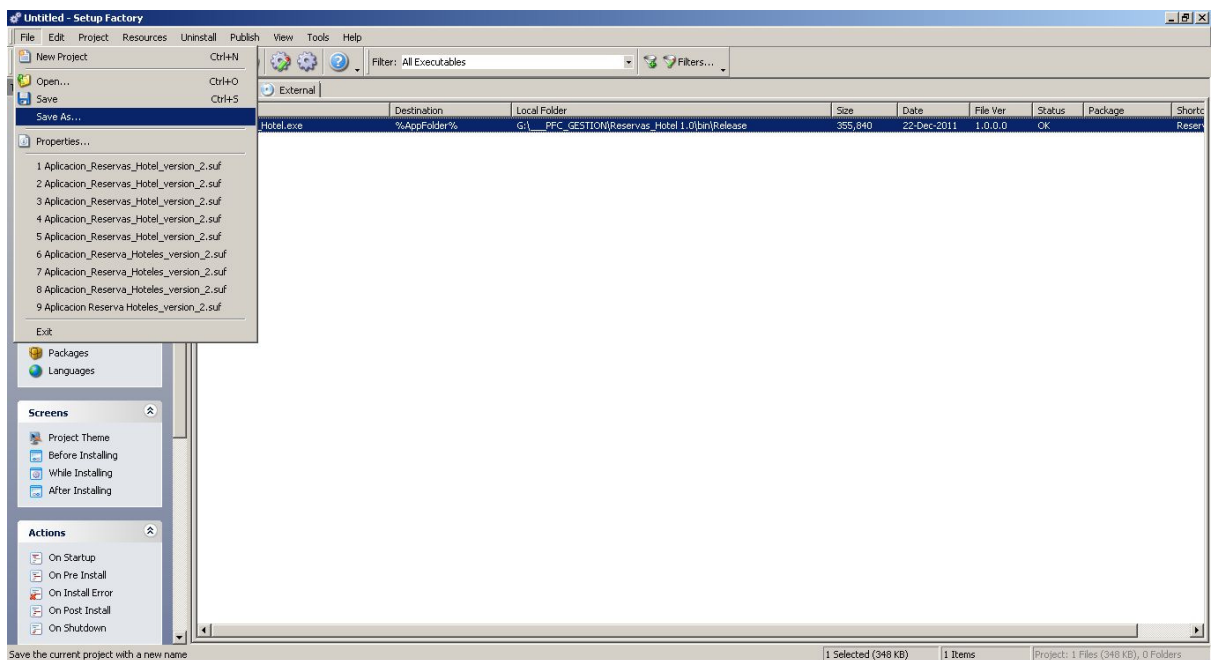
8. Pantalla 7 del asistente de proyectos: Selección de dependencias. Seleccionamos .NET 3.5 SP1



9. Pantalla 8 del asistente de proyectos: Características opcionales. Seleccionamos todas la características y en sistemas operativos soportados seleccionamos "Any OS". Pulsar Finalizar



10. Guardamos el proyecto: File -> Save As...



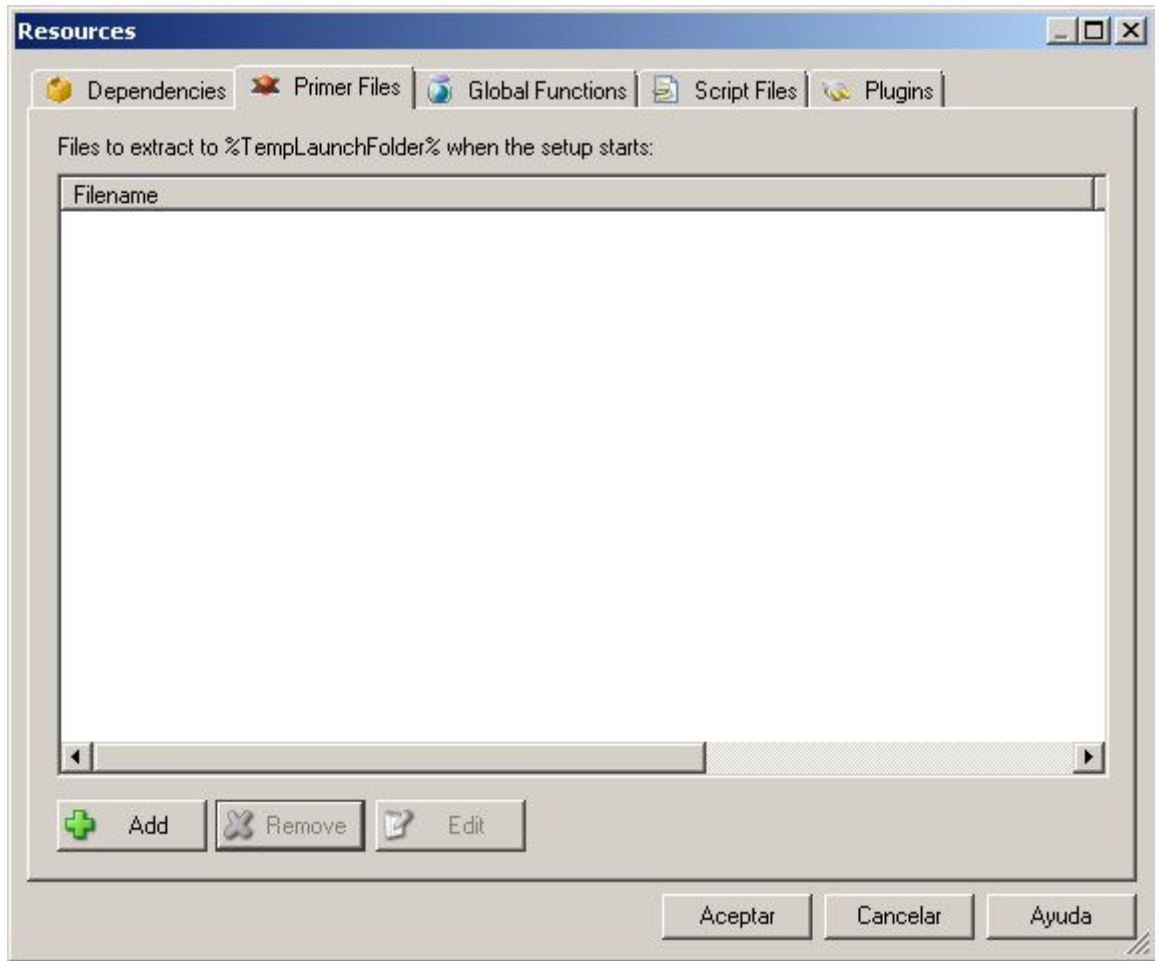
11. Añadimos archivos que necesitamos para ejecutar correctamente la aplicación Reservas de hotel 1.0 y que tienen que ser identificados y configurados por el programa Setup Factory (Primer files). Son los siguientes:

Visual Basic 2005 Power Packs 3 (utilidad de impresión)

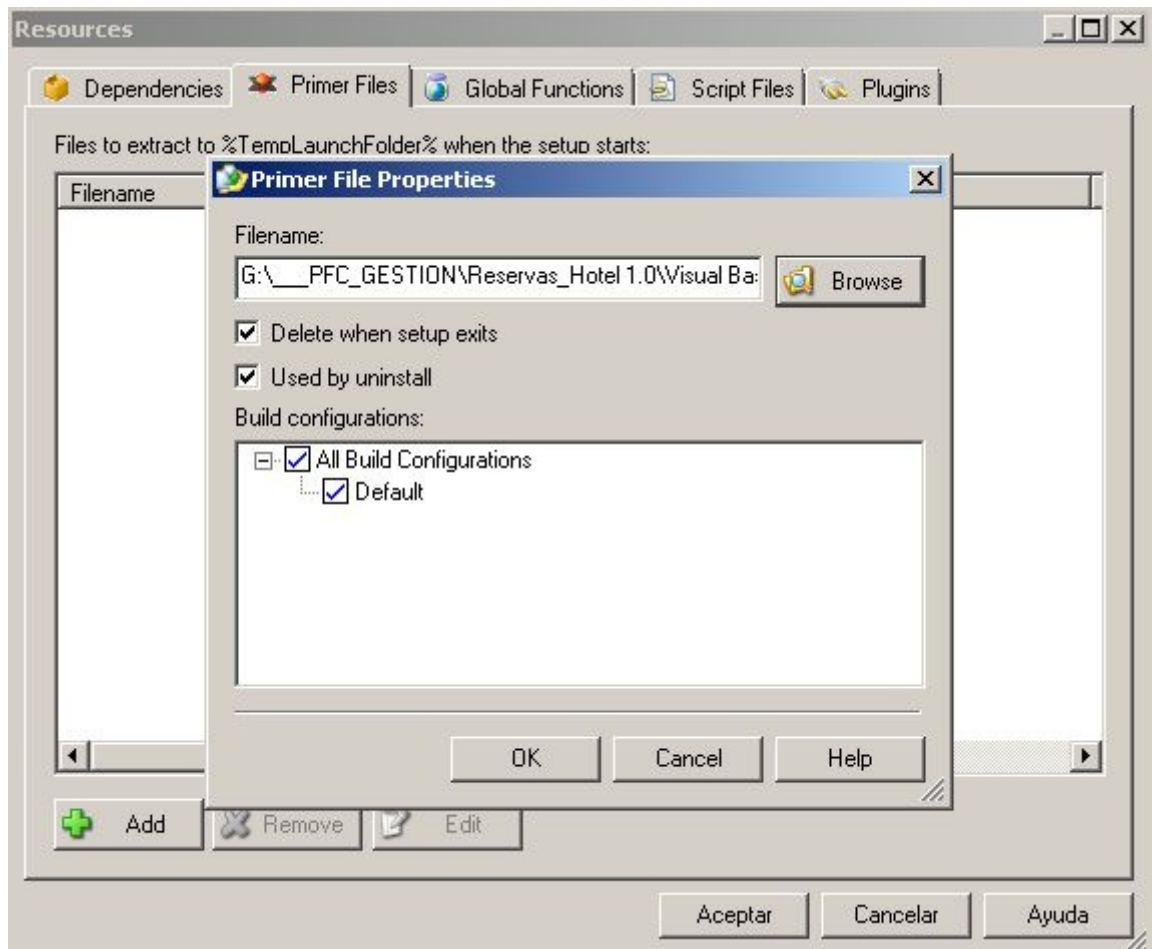
R-HMC-HOTEL.ico (icono de la aplicación)

Pasos:

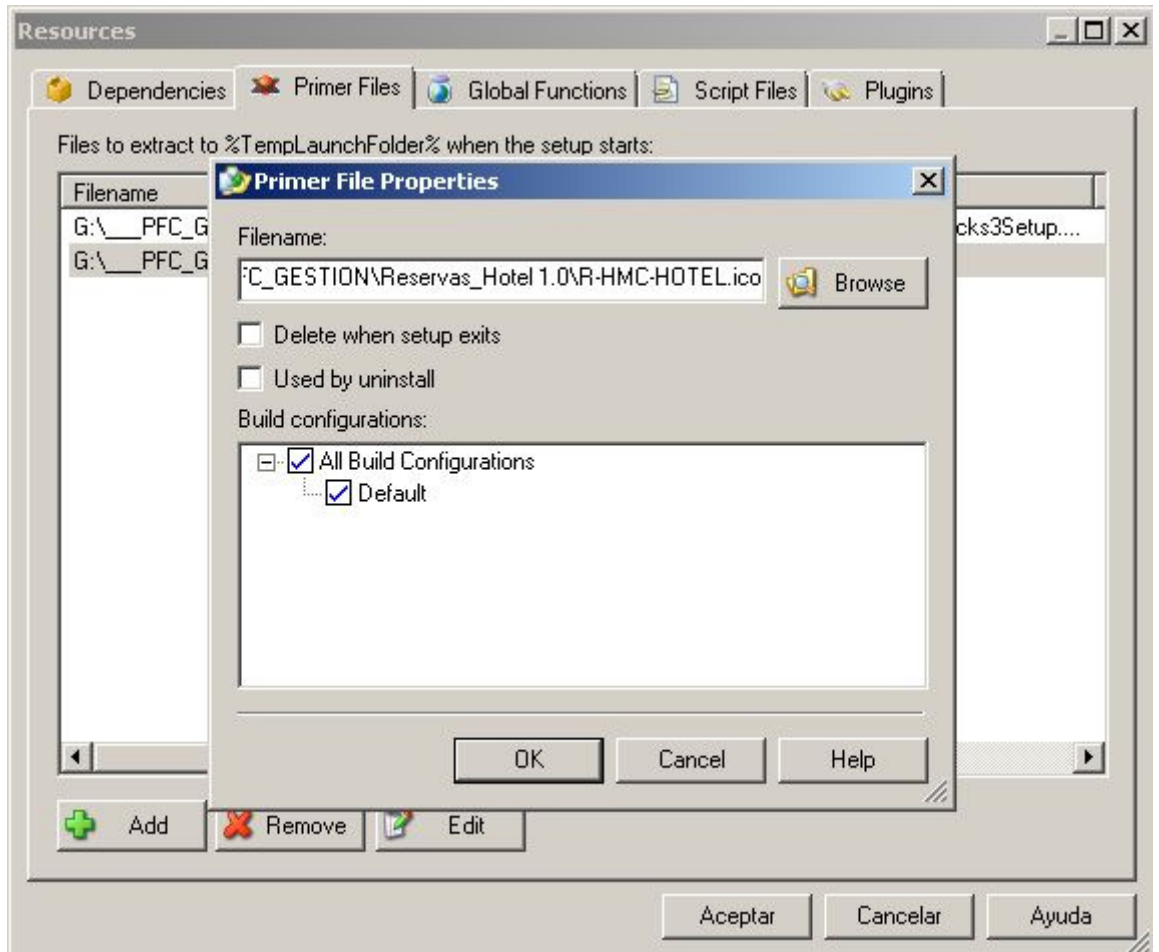
- a) Seleccionamos en la opción Resources la pestaña Primer files



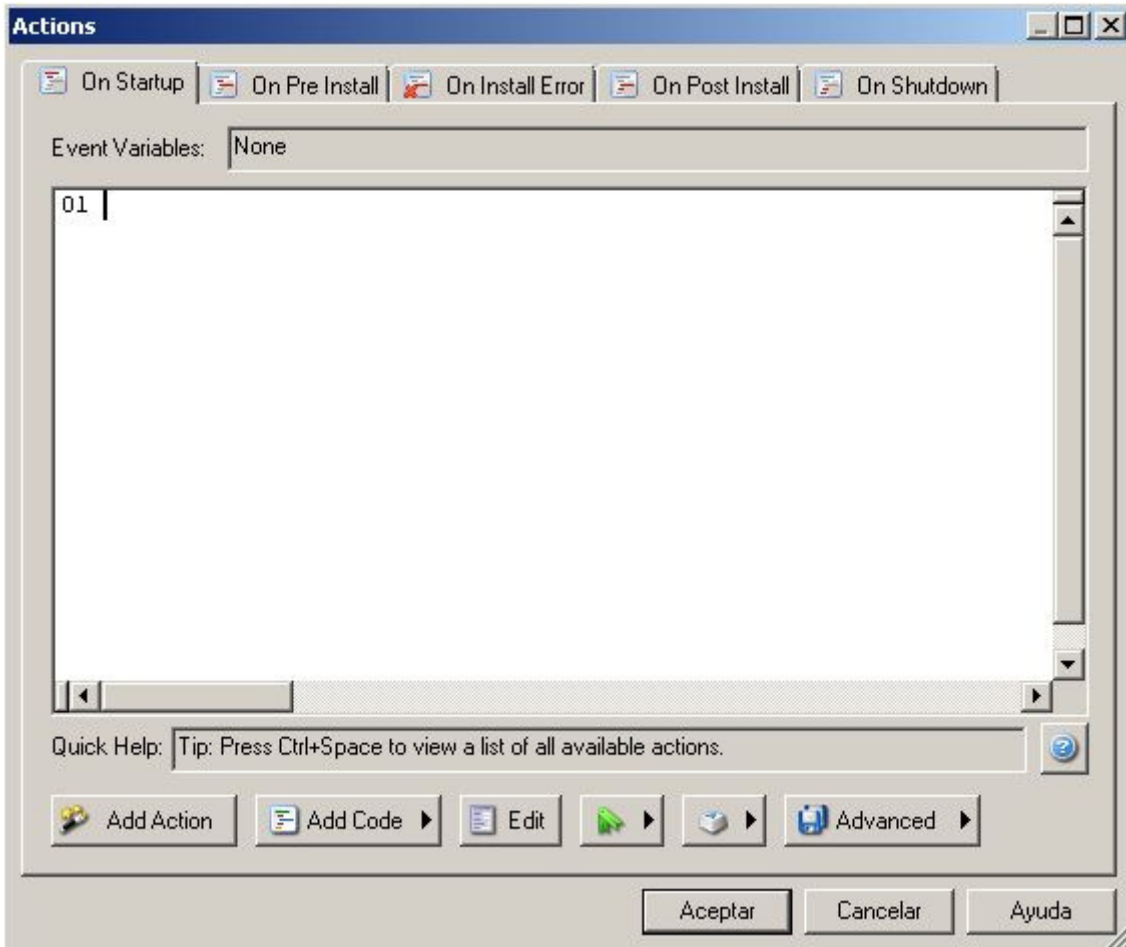
- b) Añadimos el archivo "VisualBasicPowerPacks3Setup.exe". Pulsamos el botón Add y a continuación buscamos el archivo y seleccionamos las opciones "Delete when setup exits" y "Used by uninstall". En Build configurations seleccionamos "All Build Configurations" y "Default"



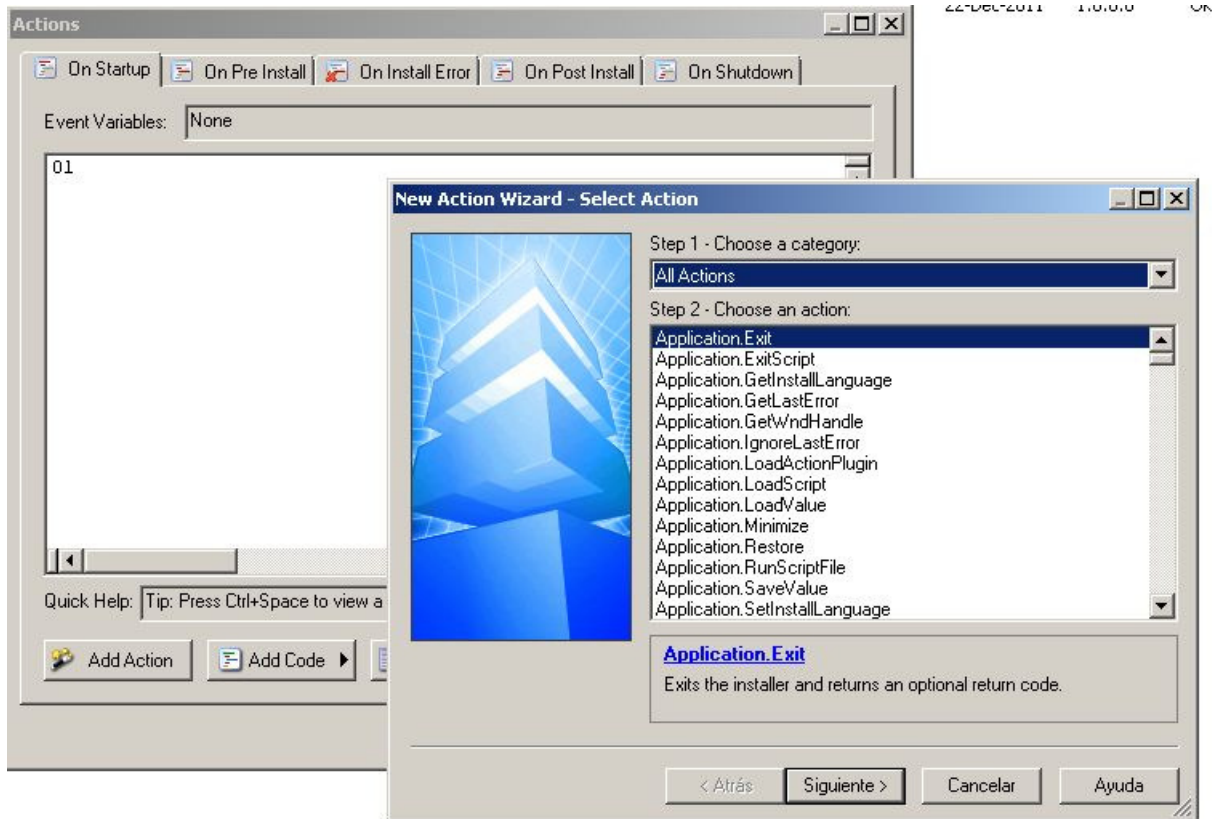
- c) Añadimos el archivo "R-HMC-HOTEL.ico". Pulsamos el botón Add y a continuación buscamos el archivo. En Build configurations seleccionamos "All Build Configurations" y "Default".



12. Configuramos el instalador de Reservas de Hotel 1.0 para que ejecute el instalador de la utilidad Visual Basic 2005 Power Packs 3 (VisualBasicPowerPacks3Setup.exe) antes de proseguir con la instalación de Reservas de Hotel 1.0. Para ello seleccionamos en **ACTIONS** la pestaña **On Startup**



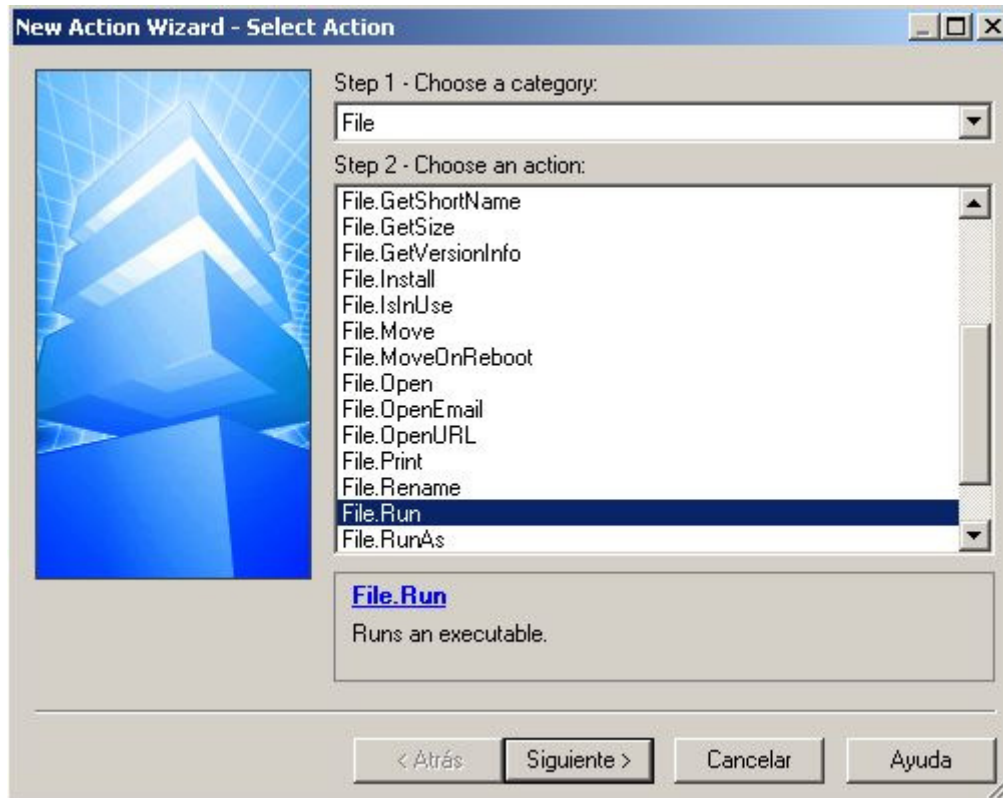
Añadimos acción pulsando el botón "Add Action". Se nos abrirá una ventana de asistente de nueva acción.



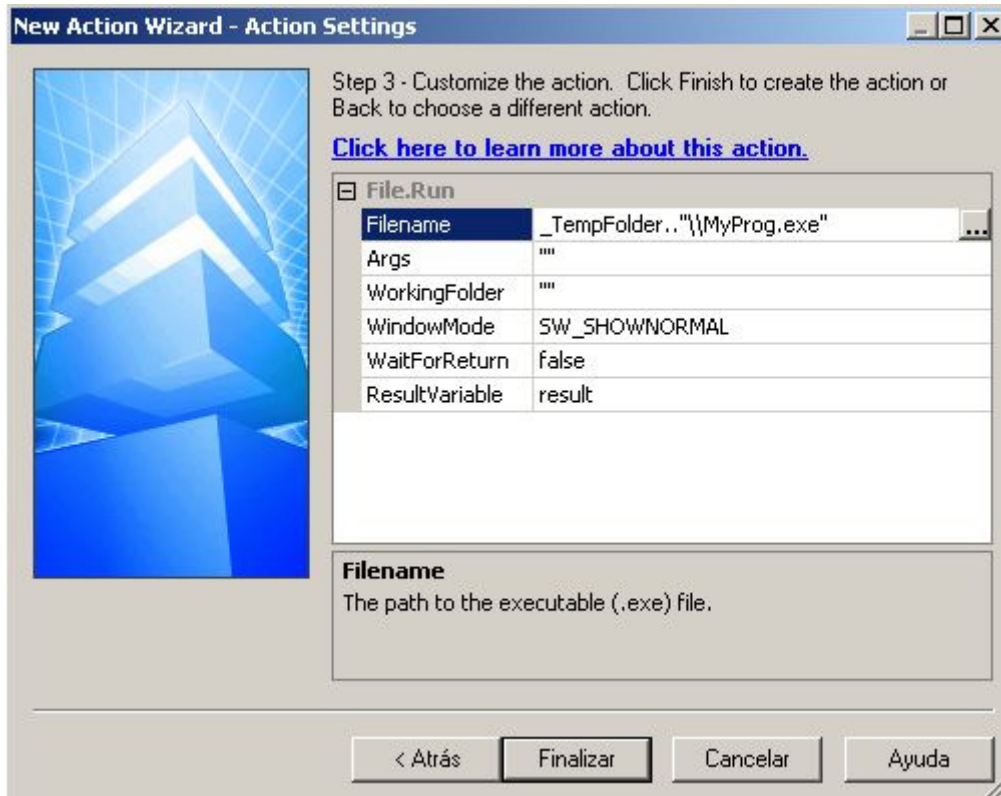
Asistente de Nueva Acción pantalla 1: Selección de acción

En Step 1 – Choose a category seleccionamos “File”

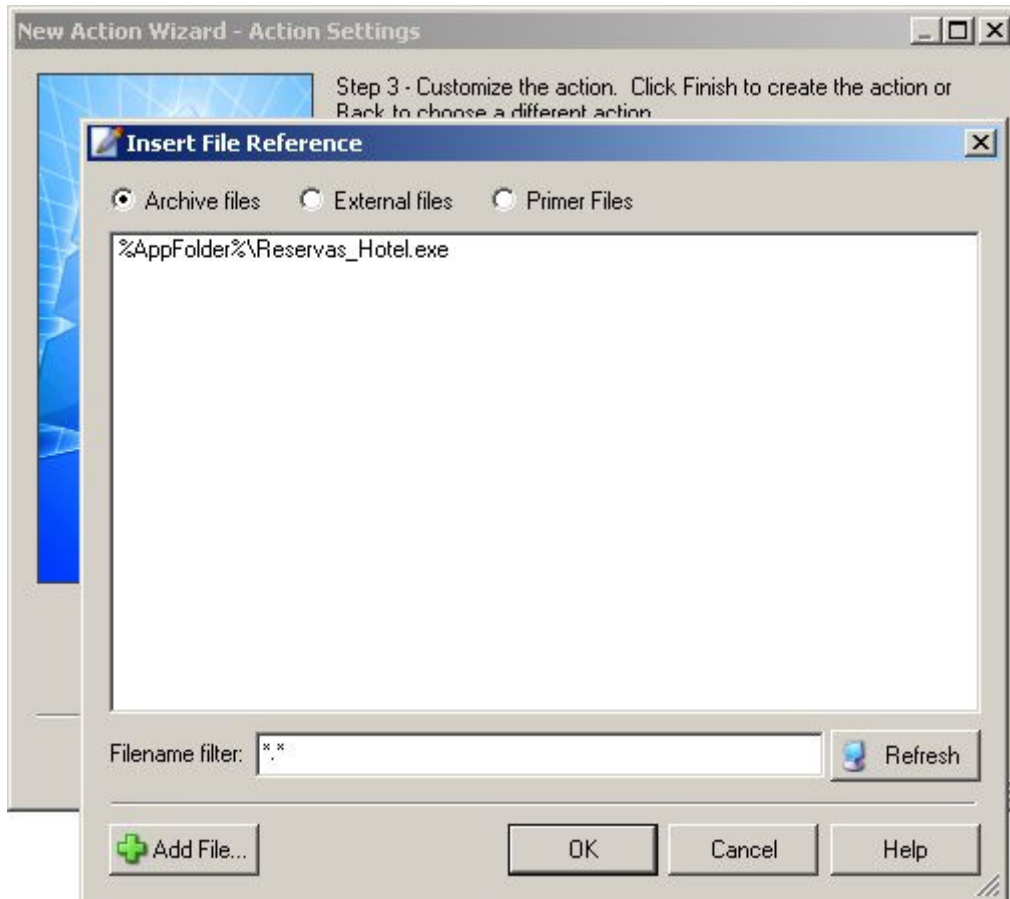
En Step 2 – Choose an action seleccionamos “File.Run”



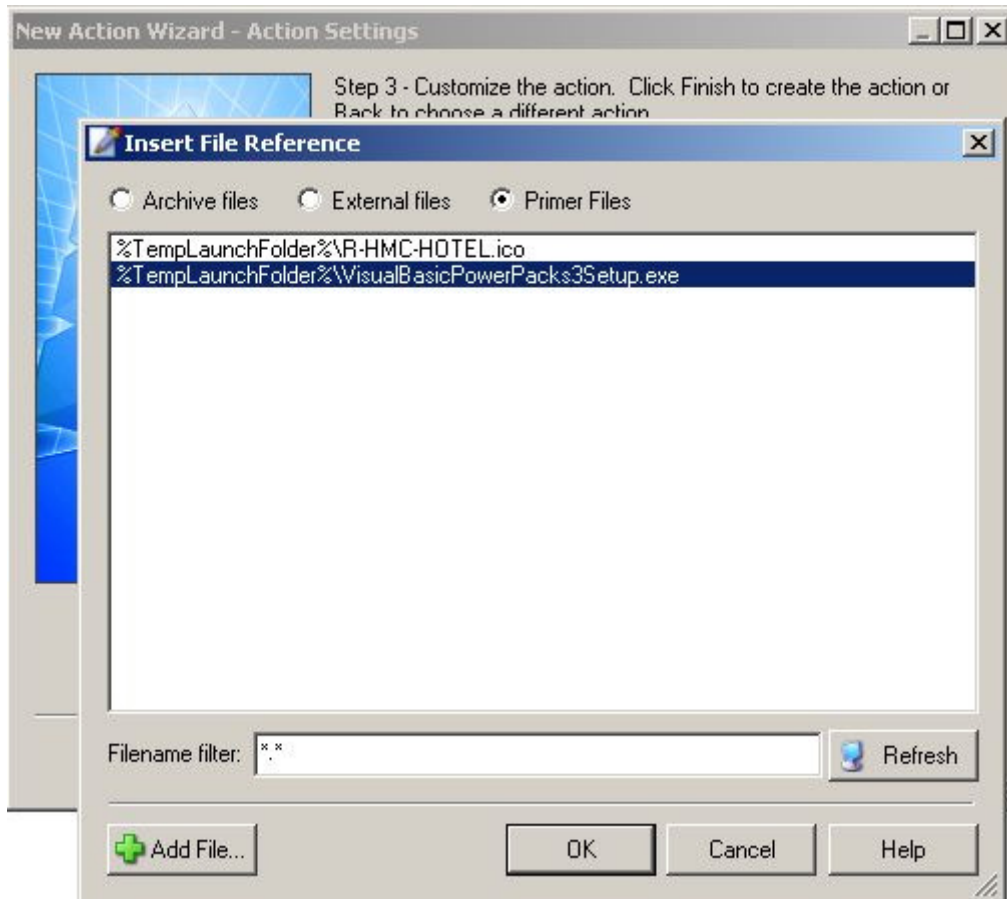
Asistente de Nueva Acción pantalla 2: Configuración de acción



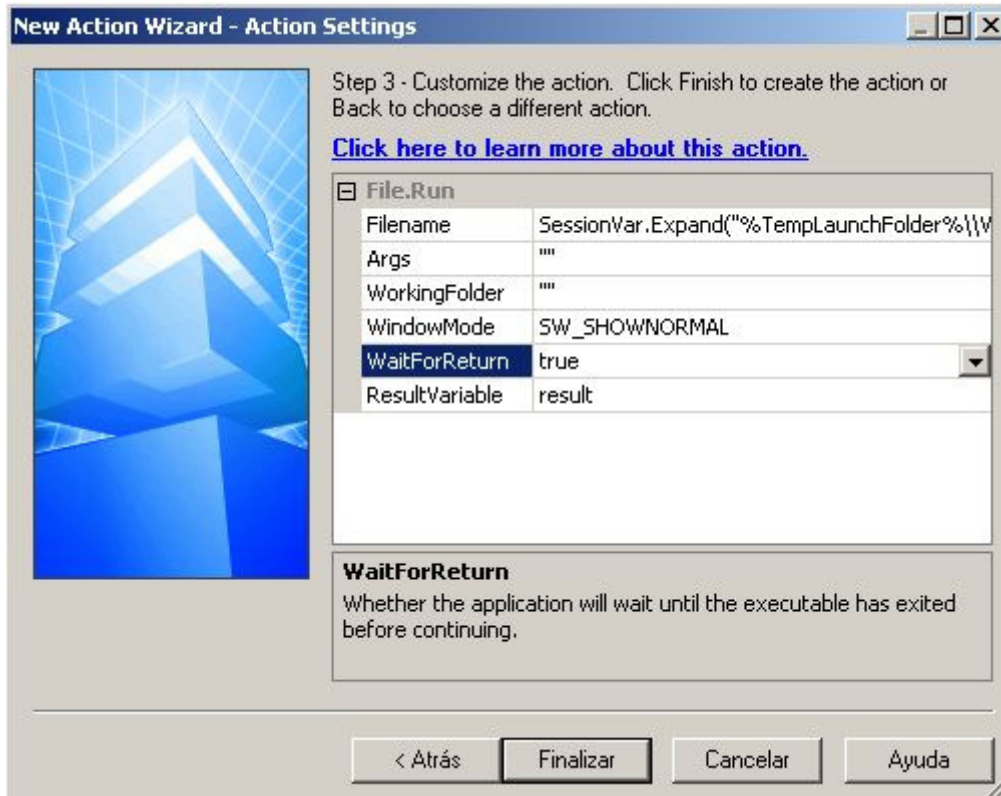
Seleccionamos Filename, pulsamos en el recuadro de puntos suspensivos y se nos abre una ventana de inserción de archivo referenciado.



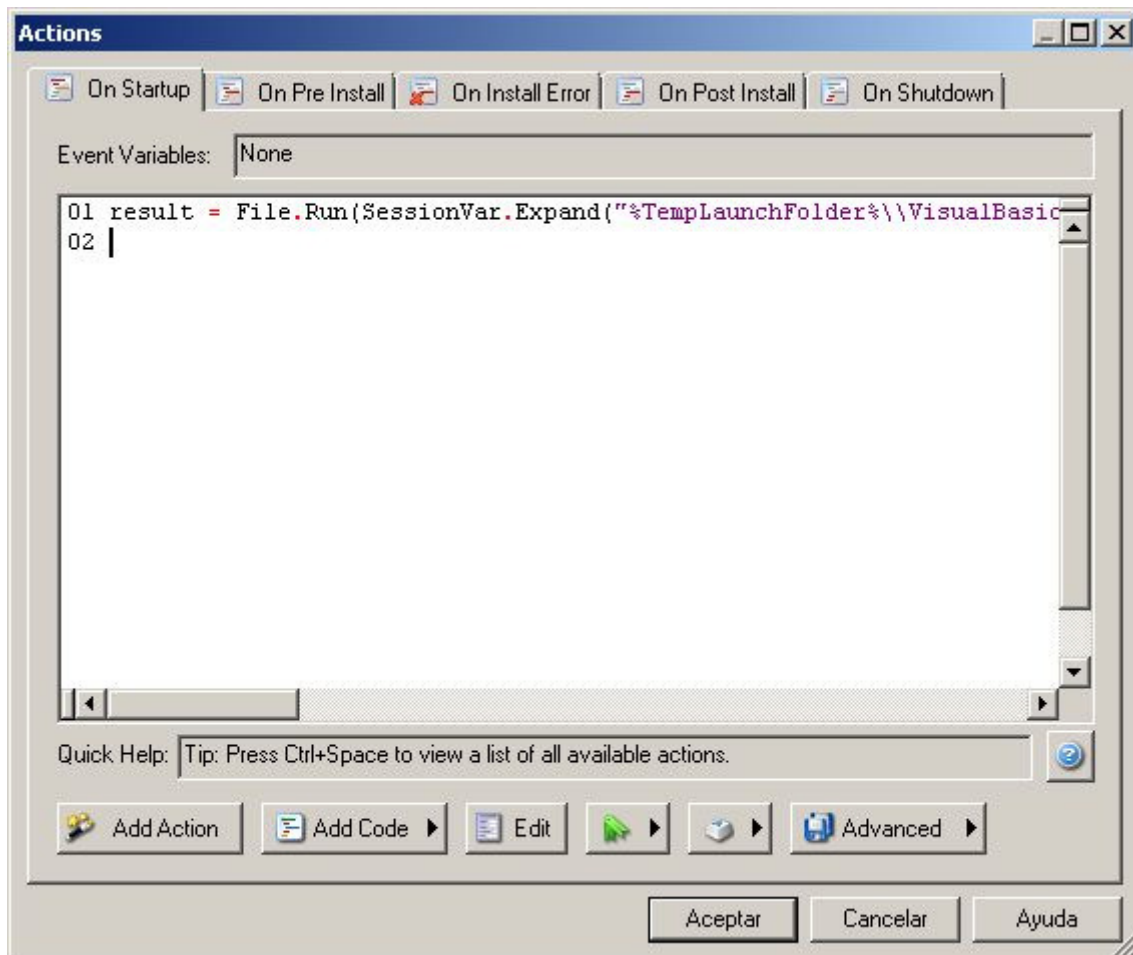
Seleccionamos la opción Primer Files y a continuación
"%TempLaunchFolder%\VisualBasicPowerPacks3Setup.exe" y OK



El resto de las opciones de configuración de la acción se quedan como están exceptuando la opción WaitForReturn que es true (esperamos a que finalice el ejecutable VisualBasicPowerPacks3Setup.exe para seguir con la instalación de Reservas de Hotel 1.0).

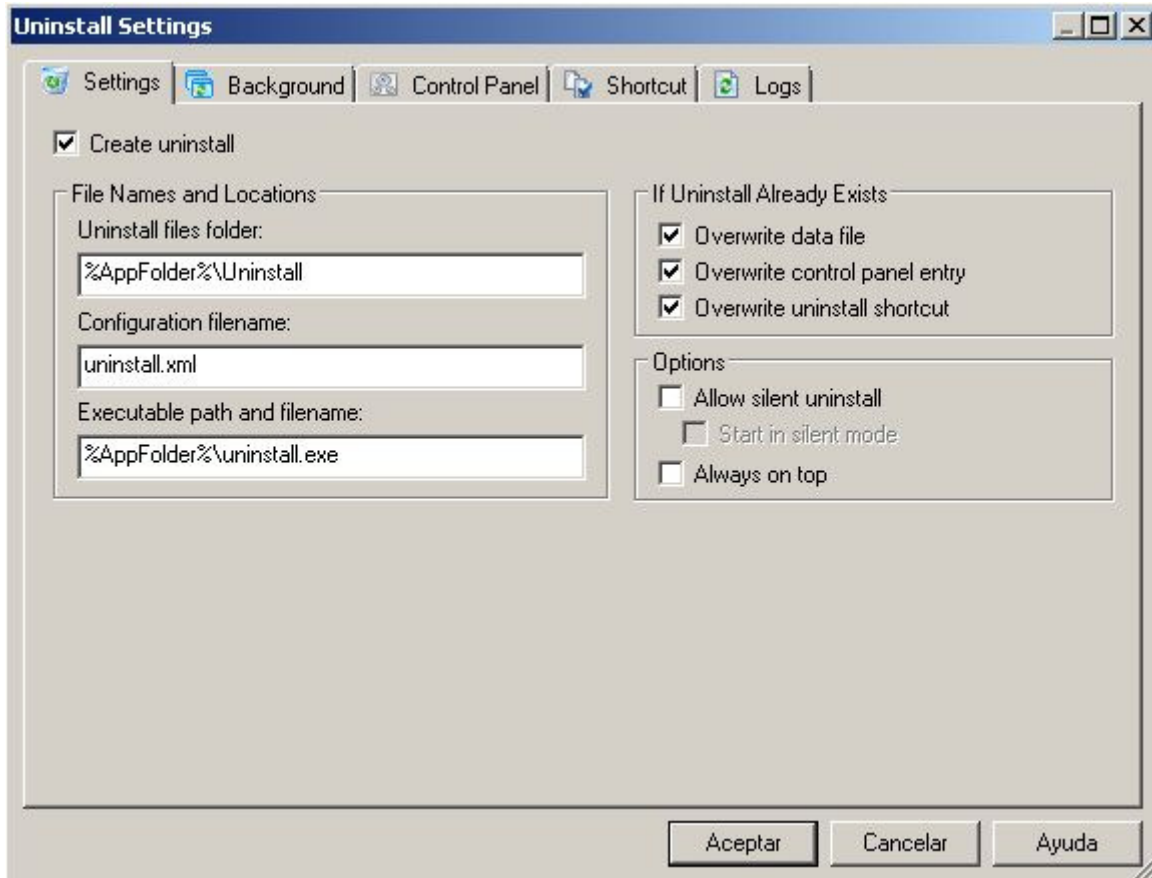


Resumiendo, hemos añadido una acción que es ejecutar una aplicación al inicio del instalador de Reservas de Hotel 1.0 (On Startup) que es VisualBasicPowerPacks3Setup.exe y el instalador no continuará hasta que no finalice dicha aplicación.

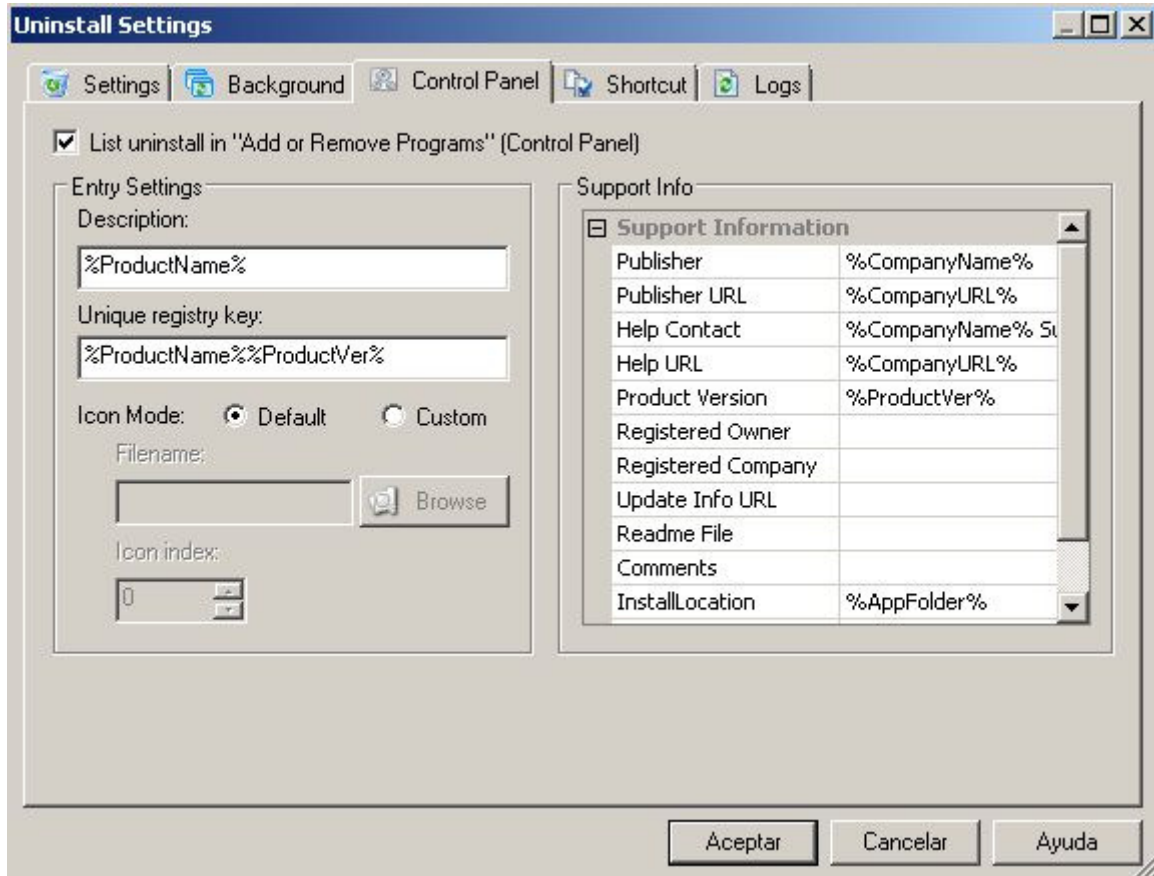


13. Configuramos la opción de Desinstalación. Para ello vamos a la opción Uninstall.

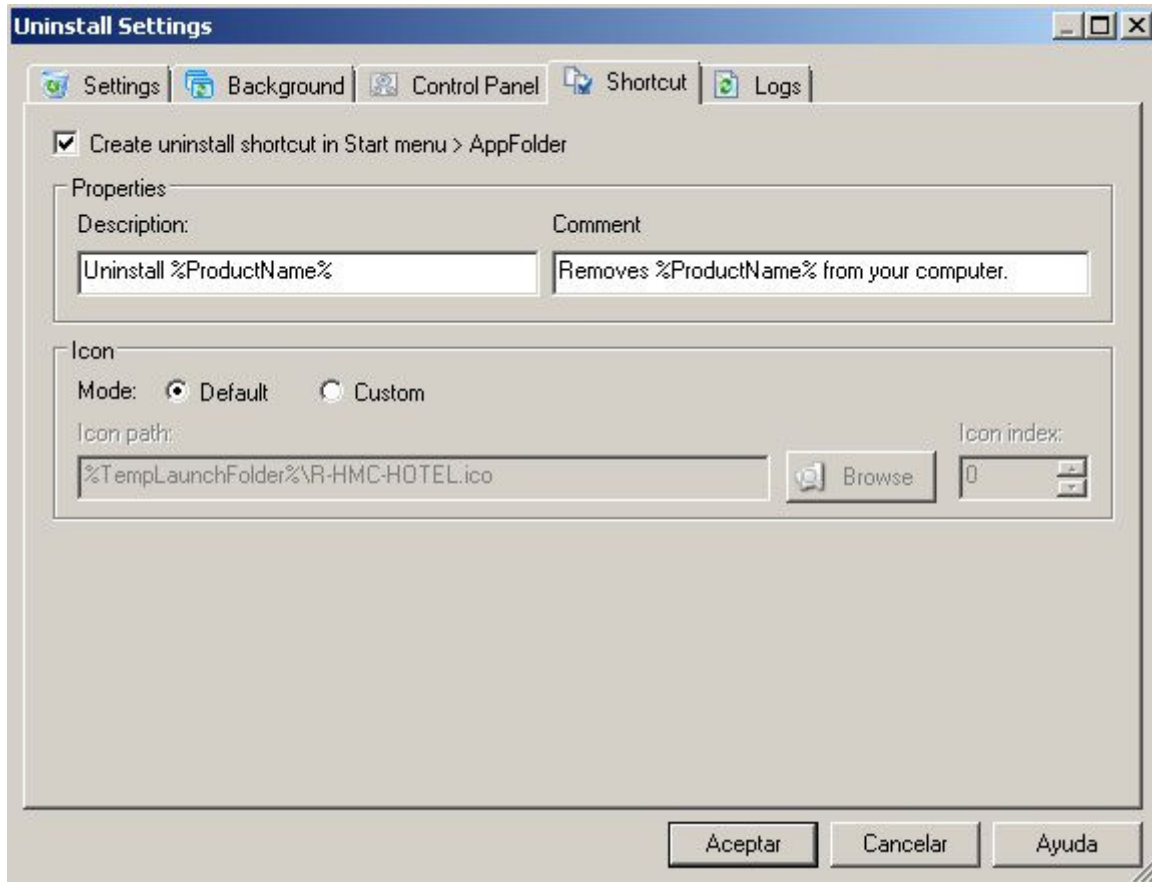
En la pestaña Settings habilitamos la opción Create uninstall (dejamos todas las opciones como están por defecto).



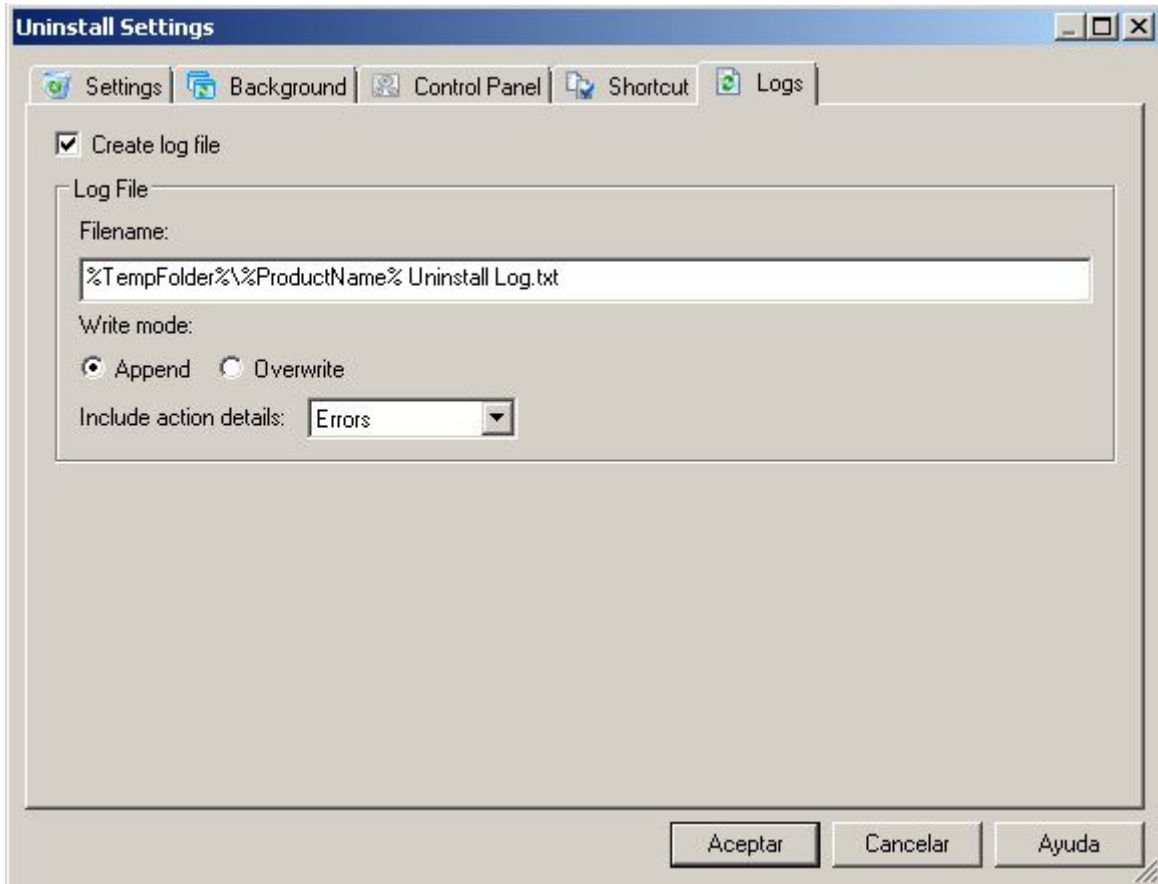
En la pestaña Control Panel habilitamos la opción "List uninstall in Add or Remove Programs (Control Panel)" (dejamos todas las opciones como están por defecto).




En la pestaña Shortcut habilitamos la opción "Create uninstall shortcut in Start menú > App Folder"
(dejamos todas las opciones como están por defecto).



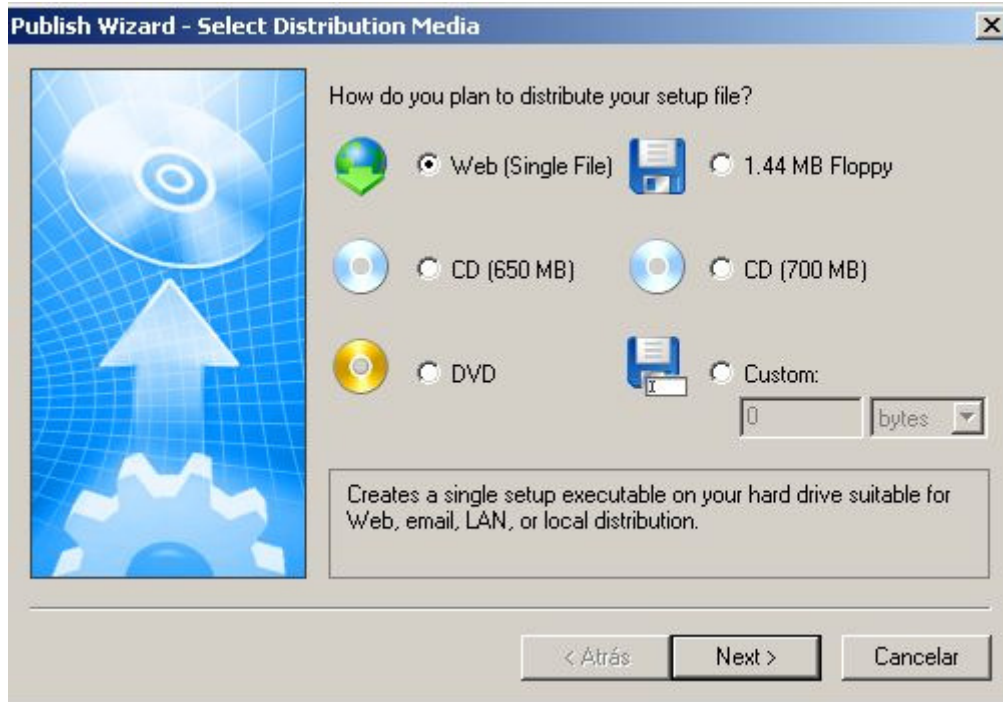
En la pestaña Logs habilitamos la opción "Create log file" (dejamos todas las opciones como están por defecto).



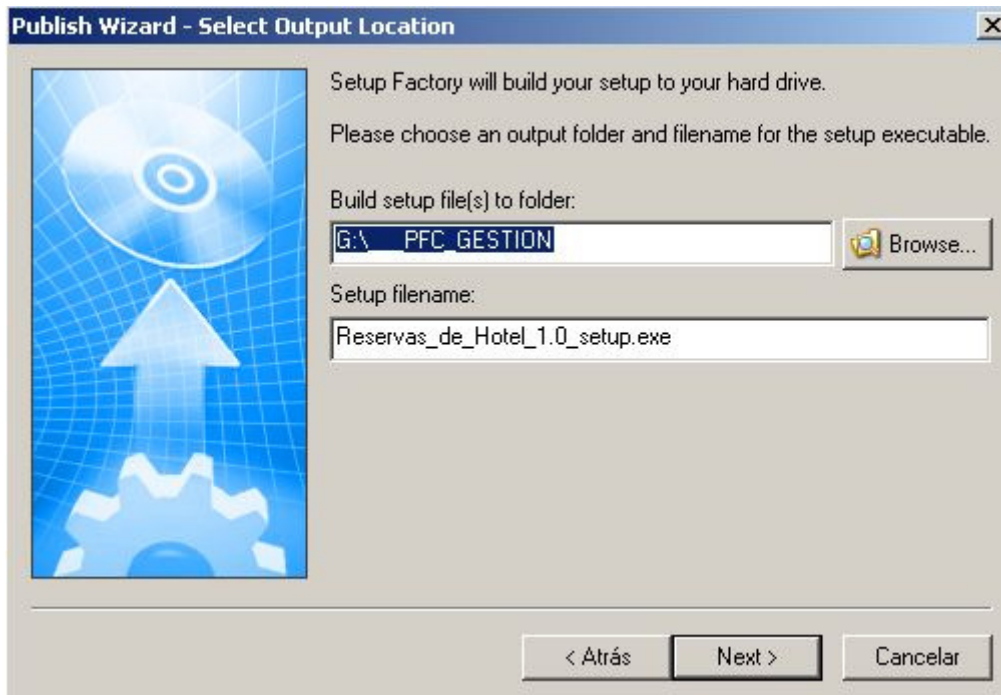
13. El último paso es publicar el proyecto (generar el instalador con sus dependencias y archivos necesarios). Para ello seleccionamos Publish → Build o pulsamos el botón  de la barra de herramientas.

Se nos abrirá una ventana de asistente de publicación:

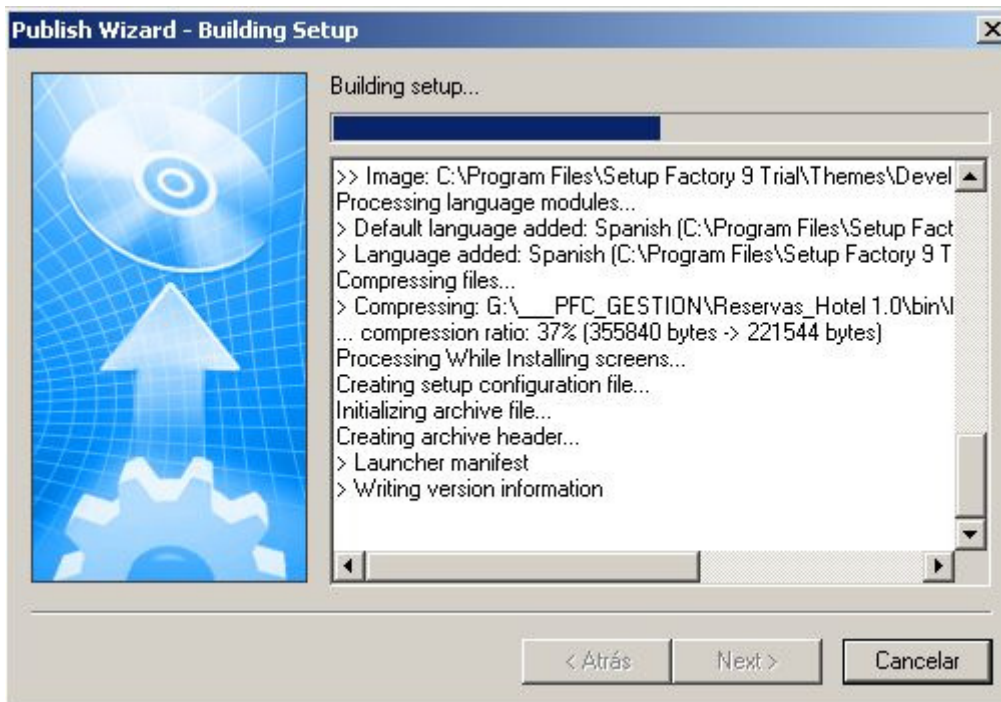
1. Seleccionaremos medio de distribución (Web single file).



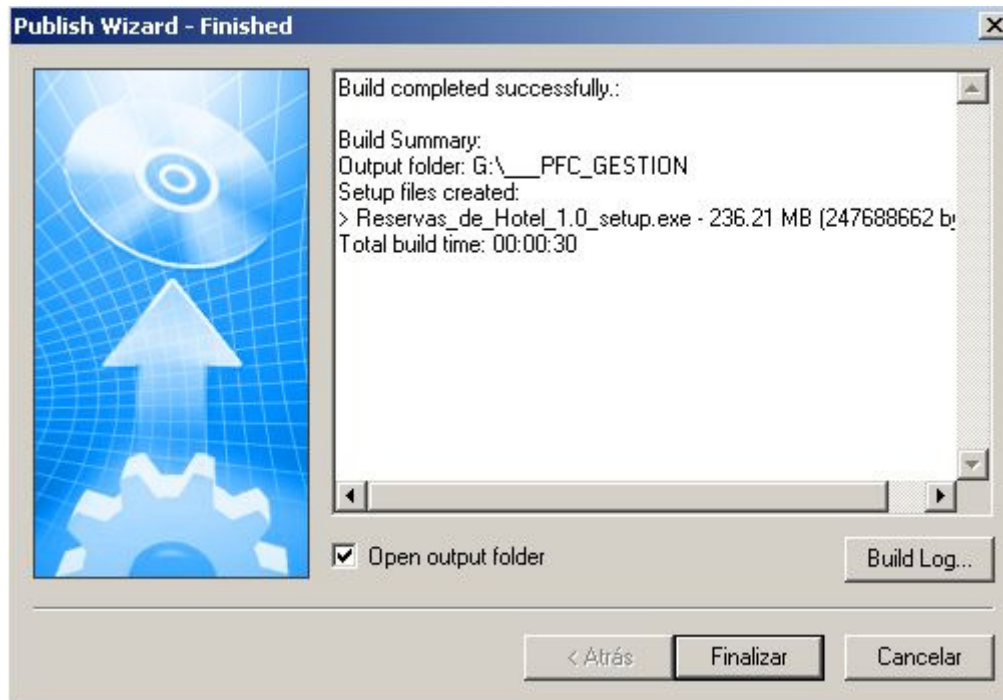
2. Indicaremos carpeta de destino y nombre del archivo instalador.



3. Se creará el archivo instalador con compresión.



4. Una vez finalizado podremos abrir la carpeta de destino del archivo instalador.



2.8 Bibliografía

- Apuntes de Ingeniería del Software de Sistemas
- Libro-Apunte nº 10 Bases de Datos Relacionales.UPV. Matilde Celma Giménez
- Libro Bases de Datos Relacionales. Pearson Prentice Hall 2003.Matilde Celda Giménez
- Libro Ingeniería del Software. Un enfoque práctico. McGraw Hill 2002 5º Edición. Roger S. Pressman
- Libro Diseño de Bases de Datos Relacionales. Rama 1999.Adoración de Miguel
- Libro programación en Visual Basic .NET. Grupo EIDOS.Luis Miguel Blanco
- Diversos manuales de SQL Server y SQL de autores desconocidos.
- MSDN Library 2008: Documentación de Visual Studio .NET