

Universitat Oberta de Catalunya

Ingeniería Técnica de Informática de Gestión

UOC Resort

Gestión de reserva de actividades en un hotel utilizando la plataforma .NET

Estudiante

Consultor

Fecha de entrega

Javier Garcia-Tenorio Usano

Jordi Sánchez Cano

11 de Junio de 2010

Contenido

Ilustraciones	3
Resumen de la memoria	4
Introducción. Breve descripción del proyecto	5
Objetivos del proyecto	7
Arquitectura básica de la aplicación.	9
Planificación del proyecto.	11
Descripción de los requisitos.....	15
Requisitos relativos a la reserva de actividades.....	16
Requisitos asociados al terminal del cliente	17
Requisitos asociados al terminal del empleado.....	19
Análisis de los casos de uso.....	20
Caso de uso 1: Reservar Actividad	21
Caso de uso 2: Reservar Servicio.....	22
Caso de uso 3: Cancelar Reserva.....	23
Caso de uso 4: Consultar Agenda.....	24
Caso de uso 5: Alta Habitación.....	24
Caso de uso 6: Elaborar Informe	25
Diseño de la aplicación.....	26
Subsistema de Lógica de Negocio y Base de Datos.....	26
Subsistema Terminal de Empleado	30
Subsistema Terminal de Huésped	31
Implementación. Tecnologías usadas	33
Acceso a datos ADO.NET	33
Consultas integradas en el lenguaje (LINQ)	33
Windows Communication Foundation (WCF).....	34
Windows Presentation Foundation (WPF).....	35
Reporting Services.....	36
Productos entregados	37
Desviaciones del proyecto	47
Líneas abiertas del proyecto	49
Conclusiones	50
Bibliografía y recursos utilizados.....	52

Ilustraciones

Ilustración 1: Arquitectura de la solución	9
Ilustración 2: Planificación original	13
Ilustración 3: Modificaciones sobre la planificación original	14
Ilustración 4: Casos de uso de la aplicación	20
Ilustración 5: Diagrama estático de la lógica del negocio	28
Ilustración 6: Diagrama relacional de la base de datos.....	29
Ilustración 7: DespliegueServidor	37
Ilustración 8: EmpleadoUOC. Pantalla principal	38
Ilustración 9: EmpleadoUOC. Creación de una nueva estancia	38
Ilustración 10: EmpleadoUOC. Creación de una nueva reserva	39
Ilustración 11: EmpleadoUOC. Cancelación de una reserva	39
Ilustración 12: EmpleadoUOC. Informes.....	40
Ilustración 13: HuespedUOC. Ventana principal.....	41
Ilustración 14: HuespedUOC. Introducción del número de habitación	42
Ilustración 15: HuespedUOC. Introducción de la contraseña	42
Ilustración 16: HuespedUOC. Servicio de limpieza o de despertador	43
Ilustración 17: Ilustración 17: HuespedUOC. Servicio de habitaciones	43
Ilustración 18: HuespedUOC. Reserva de bicicletas.....	44
Ilustración 19: HuespedUOC. Reserva de bienestar	44
Ilustración 20: HuespedUOC. Confirmación de reserva.....	45
Ilustración 21: HuespedUOC. Pantalla final	45

Resumen de la memoria

En esta memoria se detalla el proceso de desarrollo de la solución *UOCResort*, una solución que gestionará las reservas de actividades de ocio de un hotel que posee spa e instalaciones deportivas.

La solución desarrollada permitirá a los huéspedes del hotel, así como a los empleados del mismo, realizar reservas tanto de actividades (excursiones en bicicleta o partidos de pádel, entre otros) como de servicios (servicios de habitaciones o llamada despertador). La solución desarrollada consistirá en tres aplicaciones, dos de ellas pertenecientes a la capa cliente y la tercera perteneciente a la capa servidor. La comunicación entre capas se llevará a cabo usando la tecnología WCF (Windows Communication Foundation). Las aplicaciones de la capa cliente se desarrollarán usando la tecnología WPF (Windows Presentation Foundation).

La aplicación se ha desarrollado completamente dentro de la plataforma .NET. Esta plataforma integra diversos lenguajes de programación orientados a objetos, así como tecnologías punteras de acceso a datos, servicios web, cifrado y seguridad, gráficos... Se ha usado el lenguaje C# para la capa servidor y Visual Basic para la capa cliente.

Introducción. Breve descripción del proyecto

El proyecto consiste en una aplicación que permitirá gestionar actividades de relajación y entretenimiento, así como de servicios básicos, dentro del hotel *UOC Resort*, que será el cliente del proyecto. Concretamente, la aplicación permitirá:

- Que los huéspedes del hotel, así como los empleados del mismo, puedan realizar reservas de actividades deportivas, de relajación o de ocio. Posteriormente se detallarán todas estas actividades.
- Que los huéspedes soliciten ciertos servicios propios de un hotel convencional, como por ejemplo servicio de habitaciones y servicio de limpieza. Posteriormente se detallarán estos servicios.
- Que los huéspedes puedan cancelar reservas de actividades anteriormente realizadas.
- Que los empleados del hotel conozcan estas solicitudes de reserva lo antes posible, pudiendo comenzar su gestión y preparación momentos después de realizar la reserva.
- Que los empleados del hotel puedan cancelar reservas en nombre de los huéspedes.

La aplicación consistirá en una serie de terminales a los que accederán los huéspedes del hotel para poder realizar las reservas. Dichos terminales se encontrarán en las habitaciones del hotel, así como en diversos puntos clave del mismo: la recepción, el bar, gimnasio... Los empleados del hotel contarán con terminales diferentes, desde los que podrán recibir las peticiones de los clientes, así como realizar o cancelar reservas en su nombre.

En esta memoria se detallarán todos los pasos seguidos en el proceso de creación de la herramienta. Los principales apartados de los que se compone son:

- Los objetivos principales del proyecto, sobre los que se asientan los demás elementos del mismo.
- La planificación seguida durante el proyecto, indicando los principales hitos del mismo.
- Los requisitos del proyecto, especificados por el cliente y que reflejan sus deseos respecto a la aplicación a desarrollar.
- Los casos de uso, que serán la especificación exacta de la manera en la que el usuario interactuará con la aplicación. Han de estar acordados previamente por cliente y desarrollador.
- El análisis del proyecto, partiendo de los casos de uso y produciendo una serie de diagramas estáticos de clase que serán la base de la implementación.

- La implementación, en la que los diagramas estáticos de clases y casos de uso se convierten en código ejecutable y en la aplicación en sí. Se mencionarán las tecnologías más relevantes y novedosas usadas en el desarrollo del proyecto.
- Se incluirán capturas de pantalla de la aplicación.
- También se incluirá un apartado donde se describirán las desviaciones respecto del plan original. Qué es lo que no ha dado tiempo o no ha sido posible hacer.
- Se dedicará un apartado para hablar de las líneas de mejora del proyecto: qué se le puede añadir y qué mejoras aportaría a la solución.
- La memoria finalizará con las conclusiones del proyecto.

Objetivos del proyecto

El objetivo básico del proyecto, sobre el que se derivarán los demás, es:

- Mejorar la experiencia de los huéspedes en el hotel.

Teniendo la satisfacción del huésped como meta principal, podemos empezar a desglosar la meta en objetivos más específicos. Estos serían:

- Mejorar la manera en la que el huésped contrata actividades y servicios en el hotel.
- Mejorar la gestión, por parte del personal del hotel, de las actividades y servicios solicitados por el cliente.

Especificando más en detalle los objetivos anteriores obtendremos los requisitos específicos de la aplicación:

- El huésped podrá contratar actividades y servicios de manera sencilla, cómoda, rápida e independiente del conocimiento informático que éste posea.
- El personal del hotel accederá instantáneamente a las peticiones del huésped, pudiendo comenzar a preparar la actividad -en términos de logística, personal...- lo antes posible.
- Todas las actividades solicitadas por los huéspedes estarán almacenadas de manera segura en un único lugar.
- Se podrá llevar un registro de las actividades contratadas por los huéspedes, de manera que la dirección del hotel pueda conocer qué actividades son las más solicitadas y cuáles lo son menos.

Posteriormente estos requisitos se desarrollarán dando lugar a los requisitos de la aplicación, que se especificarán posteriormente.

Igualados en importancia con los requisitos anteriores están los requisitos derivados del desarrollo de un proyecto de fin de carrera relacionado con las tecnologías asociadas a la plataforma .NET. He elegido este área dentro de todos los trabajos de fin de carrera por su alto interés y relevancia y porque puede ser importante en mi carrera profesional. Por tanto, también debe considerarse como objetivo, con el mismo nivel de prioridad que los anteriores, el aprender y dominar lo más profundamente posible la plataforma .NET, incluyendo sus lenguajes base, sus tecnologías de acceso a datos, programación orientada a servicios, diseño

de interfaces gráficas y todas aquellas tecnologías y recursos que puedan ser necesarias a lo largo del proyecto. Este tipo de objetivos son difíciles de medir, ya que no existe una marca que distinga a los que conocen .NET de los que no. Es necesario, por lo tanto, que tanto el desarrollador del proyecto como el tutor y el tribunal de evaluación decidan si el aprovechamiento del Proyecto de Fin de Carrera ha sido adecuado y si el objetivo anterior se ha cumplido.

Arquitectura básica de la aplicación.

La aplicación ha de permitir a los huéspedes solicitar actividades y servicios a través de terminales. Asimismo, la aplicación permitirá a los empleados del hotel gestionar las peticiones de los clientes con el fin de ponerlas en marcha, también desde terminales situados en las instalaciones del hotel. Por último, del lado del servidor se dispondrá de una base de datos que almacenará todos los datos de la aplicación. Todas las operaciones de gestión del negocio estarán situadas del lado del servidor.

Un esquema sencillo de la aplicación a desarrollar podría ser el siguiente:

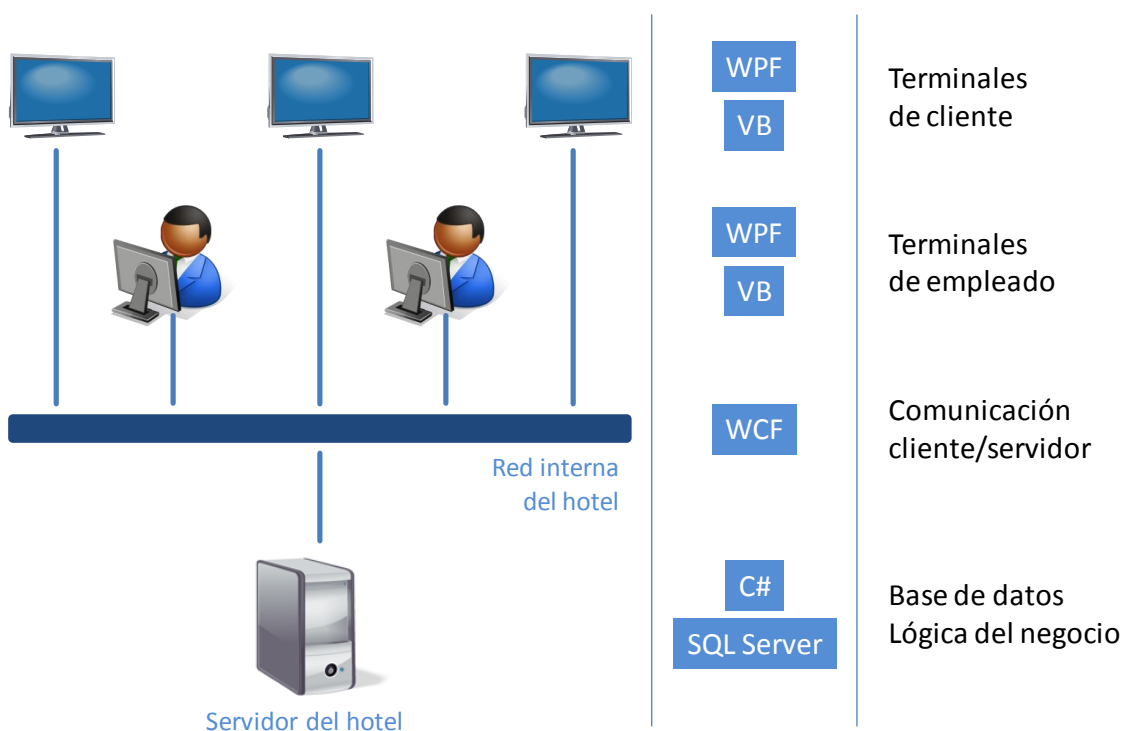


Ilustración 1: Arquitectura de la solución

De acuerdo con esta imagen, la aplicación se divide en cuatro subsistemas:

- **Lógica del negocio:** este subsistema se encargará de todas las actividades que conforman el núcleo de la aplicación, así como de acceder a la base de datos. Entre otras tareas, este subsistema se ocupará de:
 - Recibir peticiones de reserva de actividades.
 - Decidir si la actividad se puede llevar a cabo (p.ej. si hay plazas suficientes)
 - Efectuar las gestiones oportunas.

- Comunicar a otros subsistemas si la operación se ha efectuado de manera satisfactoria o no.
- **Terminal de cliente:** este subsistema representa la interfaz de usuario que se le mostrará al cliente del hotel. Realizará las siguientes tareas:
 - Presentar al cliente las diferentes actividades que propone el hotel.
 - Ofrecerle las opciones correspondientes a la actividad elegida.
 - Contactar con el subsistema de lógica del negocio para efectuar la reserva de la actividad.
- **Terminal de empleado:** este subsistema representa la interfaz de usuario que se le mostrará al empleado del hotel. Este subsistema se encargará de:
 - Presentar a los empleados del hotel las actividades solicitadas por los clientes.
 - Mostrar las diferentes opciones y tareas a completar para poder gestionar las actividades.
 - Contactar con el subsistema de lógica del negocio para efectuar las operaciones necesarias de gestión de las actividades.
- **Base de datos:** este subsistema contiene la base de datos con toda la información necesaria para la aplicación.

Como se verá en los siguientes apartados, toda la planificación del proyecto y las tareas relativas al mismo se han desarrollado teniendo en cuenta los anteriores subsistemas.

Planificación del proyecto.

Las tareas se han desarrollado ajustándose a las entregas de la asignatura. Teniendo en cuenta que la asignatura de Proyecto de Fin de Carrera se compone de 3 PEC's y una entrega adicional de memoria y presentación, se han agrupado las tareas en esos cuatro grupos. Las tareas que se han realizado son las siguientes:

- La primera fase **PEC 1** ha coincidido con la elaboración del Plan de Proyecto y se compuso de las siguientes tareas:
 - Análisis de la competencia.
 - Análisis de los requisitos de alto nivel.
 - Preparación de la planificación.
 - Preparación de la PEC 1.
- La **PEC 2** correspondía a las fases de análisis y diseño. Las tareas de esta fase se han agrupado según el subsistema al que pertenecen.
 - Hay tareas comunes a todos los subsistemas, en los que consideramos a la aplicación como un todo. Las tareas han sido:
 - Realización de un diagrama de secuencia.
 - Realización de un diagrama de actividad.
 - Realización de un diagrama de casos de uso.
 - Análisis de los requisitos de la aplicación.
 - Para el análisis y diseño de los subsistemas de Terminal de cliente, Terminal de empleado y Lógica del negocio fueron necesarias las siguientes tareas:
 - Diagrama estático (de clases).
 - Diagrama de casos de uso.
 - Boceto de la interfaz de usuario.
 - El subsistema de Base de Datos llevó asociadas las siguientes tareas:
 - Identificar entidades.
 - Realización del diagrama entidad-relación.
 - Desarrollo de un prototipo de Base de Datos, sobre el que se realizarán pruebas de inserción, borrado y actualización de datos.
 - Por último, se dedicó algo de tiempo a la confección de los entregables específicos de la PEC 2.

- La **PEC 3** estuvo asociada a la implementación de la aplicación. Las tareas se agruparon según el subsistema al que se encuentran asociadas. Para esta fase, los subsistemas de lógica del negocio y base de datos se han considerado uno sólo, debido a la fuerte relación que ambos poseen. Las tareas asociadas a cada uno de los subsistemas han sido:
 - Codificación del subsistema.
 - Testeo del subsistema.
 - Confección de los entregables específicos de la PEC 3.
- La **PEC 4** ha consistido en la confección de la memoria del proyecto y la presentación asociada al mismo. Las tareas de esta fase han sido:
 - Preparación de la memoria.
 - Preparación de la presentación.

La planificación del proyecto se ha confeccionado usando MS Project 2007.

La planificación sólo ha sufrido una desviación, que no ha supuesto ningún riesgo para el desarrollo del proyecto ya que se produjo al comienzo de la misma y la reorganización del proyecto fue muy sencilla. Al inicio del proyecto se consideró –como reflejan los diagramas de las páginas siguientes- implementar primero el terminal del cliente (la aplicación *HuespedUOC*), seguida del termina del empleado (la aplicación *EmpleadoUOC*), para acabar con la implementación de la Base de Datos y la lógica del negocio. Lógicamente, tiene mucho más sentido implementar primero la Base de Datos, posteriormente implementar la lógica del negocio (acceso y manejo de esos datos) para acabar con las dos aplicaciones cliente, que usarán las funcionalidades de la aplicación servidor de manera intensiva. Afortunadamente, el error era evidente y se subsanó rápidamente sin más que intercambiar los subsistemas para colocar primero los subsistemas 3 y 4 (Base de datos y lógica del negocio) y detrás los subsistemas relativos a la capa del cliente. En las páginas siguientes se encuentran capturas de pantalla del diagrama de Gantt del proyecto antes y después de la fase de implementación, y donde se puede ver la desviación antes mencionada. Para mejorar la legibilidad sólo se adjuntan los dos últimos hitos -PEC3 y PEC4- de la parte modificada.

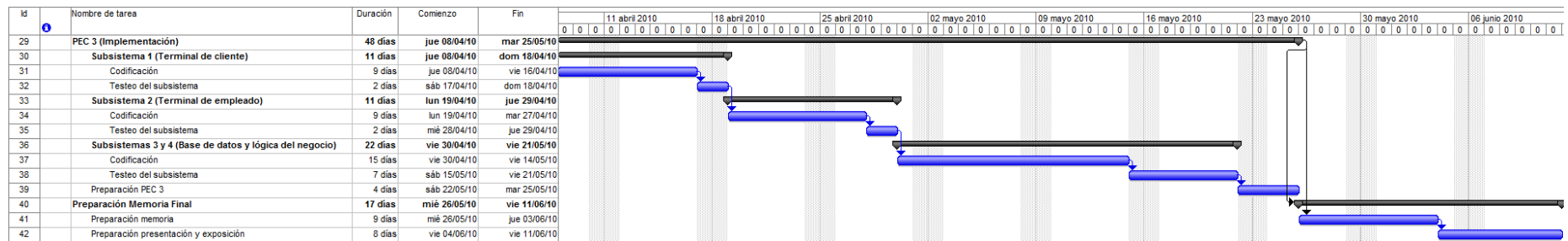
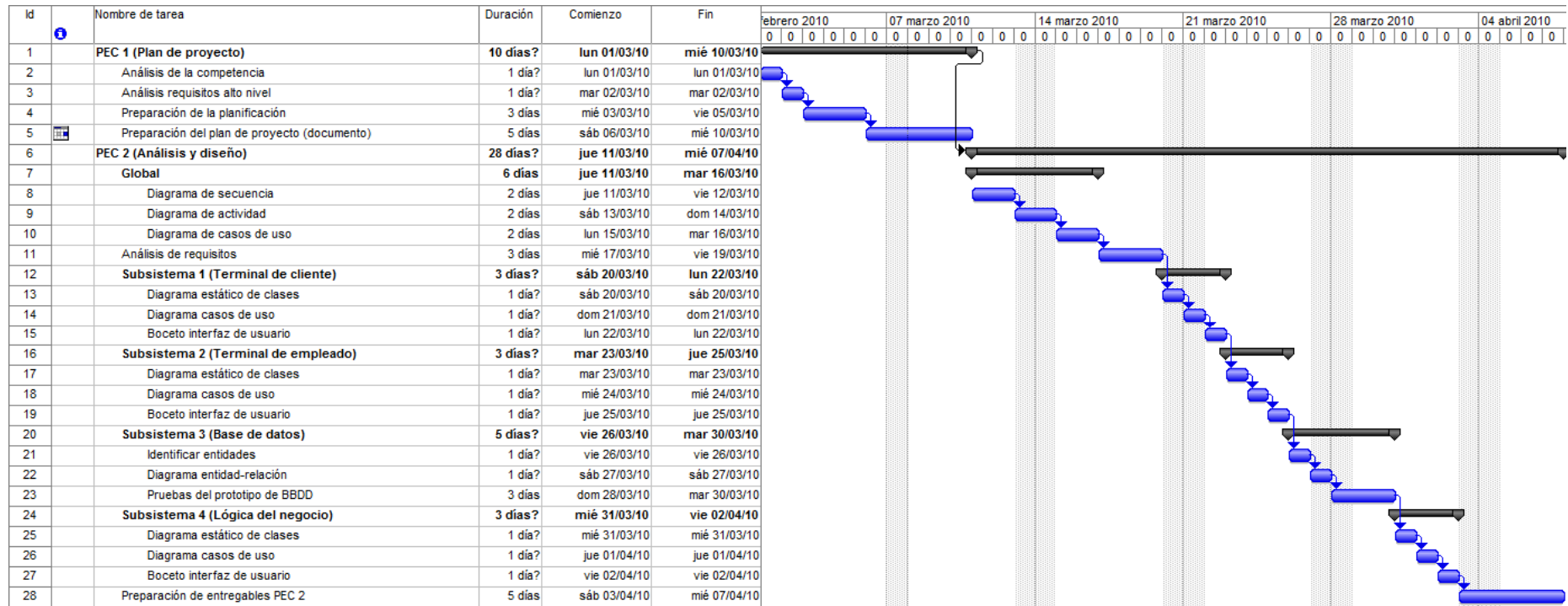


Ilustración 2: Planificación original

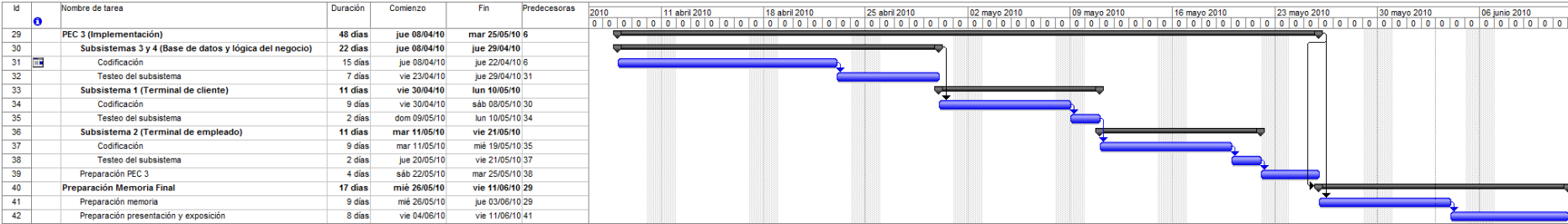


Ilustración 3: Modificaciones sobre la planificación original

Descripción de los requisitos

En esta sección se detallarán los requisitos de la aplicación. Los requisitos se presentan ordenados según la tarea que la aplicación desempeñe. Cada requisito consta de:

- Un identificador, que permitirá referenciar un requisito de manera unívoca.
- Una breve clasificación, que describirá la parte de la aplicación a la que el requisito referencia. De esa manera será más sencillo ordenar los requisitos y acceder a ellos.
- Una descripción del requisito.

Las características que le imponemos a los requisitos serán:

- Son unitarios: cada requisito especifica una única funcionalidad, de esta manera será más sencillo de testear y comprobar la satisfacción del cliente.
- Son medibles: cada requisito ha de ser especificado de manera que sea sencillo y claro comprobar si la aplicación desarrollada cumple con él o no. Un requisito ambiguo no podrá ser comprobado correctamente y no se sabrá a ciencia cierta si la aplicación lo cumple o no.

Los requisitos se han dividido en varias partes:

- Relativos a la reserva de actividades y servicios en general.
- Relativos a la reserva de actividades y servicios por parte del cliente.
- Relativos a la reserva de actividades y servicios por parte del empleado.

Ciertos requisitos no han podido llevarse a cabo, fundamentalmente por la falta de tiempo. Dichos requisitos se resaltan en rojo sobre los demás. En la sección de desviaciones del proyecto se explicará las razones por las cuales se han ignorado esos requisitos.

Requisitos relativos a la reserva de actividades

Id.	Tema	Descripción
RES_GEN_001	Reserva general	El usuario de la aplicación podrá realizar reservas a través de la aplicación.
RES_GEN_002	Reserva general	Las reservas podrán ser de dos tipos: de una actividad o de un servicio.
RES_GEN_010	Reserva general	En cada sesión sólo se permitirá reservar una actividad. Si se desean efectuar varias reservas para el mismo día, deberá ser en dos sesiones, comenzando de nuevo.
RES_GEN_020	Reserva general	Los clientes del hotel sólo podrán efectuar reservas los días que tengan contratada estancia en el hotel.
RES_GEN_030	Reserva general	Para formalizar una reserva será necesario adjuntar el número de habitación a la misma.
RES_GEN_040	Reserva general	Se deberá evitar que haya reservas fraudulentas – que alguien reserve actividades para otra habitación.
RES_GEN_050	Reserva general	El cliente puede cancelar reservas que haya hecho anteriormente.
RES_GEN_060	Reserva general	Se deberá evitar que haya cancelaciones fraudulentas –que alguien cancele actividades para otra habitación.
RES_GEN_070	Reserva general	El cliente puede consultar a través de los terminales qué actividades tiene reservadas para un día determinado.
RES_GEN_080	Reserva general	Por razones de auditoría, las reservas que se cancelen o se hayan realizado han de permanecer accesibles para el personal, y ser distinguibles del resto.
RES_GEN_090	Reserva general	La funcionalidad de editar reservas existentes no será necesaria. El procedimiento a seguir consistirá en cancelar la reserva y crear una nueva.
RES_GEN_100	Reserva general	Todas las actividades, salvo el servicio de habitaciones, van incluidas en el precio de la reserva, luego no se necesitará gestionar el precio de cada actividad.

Requisitos asociados al terminal del cliente

Id.	Tema	Descripción
RES_CLI_001	Reserva deportiva. Pádel	El cliente podrá reservar una pista de pádel de entre las dos existentes.
RES_CLI_010	Reserva deportiva. Pádel	Para reservar una pista de pádel será necesario especificar un día, una hora de inicio y un número de pista.
RES_CLI_020	Reserva deportiva. Pádel	El hotel cuenta con dos pistas de pádel. No se pueden reservar sesiones para más de esas dos pistas.
RES_CLI_030	Reserva deportiva. Pádel	La pista de pádel sólo se podrá reservar para una hora un determinado día. Una misma habitación no puede reservar dos horas de pádel un mismo día, ni alternas ni contiguas.
RES_CLI_040	Reserva deportiva. Bicicleta.	El hotel cuenta con 20 bicicletas para alquiler, todas del mismo tipo. No podrán reservarse más de ese número de bicicletas en un momento determinado.
RES_CLI_050	Reserva deportiva. Bicicleta.	Para efectuar la reserva de bicicletas el cliente ha de especificar el día de la reserva, la hora de inicio, la hora de finalización y el número de bicicletas que se desean reservar.
RES_CLI_060	Reserva de relajación. Masaje.	Dentro de las actividades de relajación, el cliente podrá reservar sesiones de masaje durante su estancia en el hotel.
RES_CLI_070	Reserva de relajación. Masaje.	Para reservar una sesión de masaje, será necesario adjuntar el día de la sesión, la hora de inicio de la sesión y la sala de masaje a la que se acudirá.
RES_CLI_080	Reserva de relajación. Masaje.	Todas las sesiones de masaje durarán una hora exactamente.
RES_CLI_090	Reserva de relajación. Masaje.	Existen tres salas de masaje. No se podrán reservar sesiones para salas adicionales.
RES_CLI_100	Reserva de relajación. Baños.	Dentro de las actividades de relajación, el cliente podrá reservar sesiones de baño relajante durante su estancia en el hotel.
RES_CLI_110	Reserva de relajación. Baños.	Para reservar una sesión de baño relajante, el cliente ha de facilitar el día de la sesión, la hora de inicio de la sesión y la sala de baños a la que se acudirá.
RES_CLI_120	Reserva de relajación. Baños.	Todas las sesiones de baño relajante durarán una hora exactamente.
RES_CLI_130	Reserva de relajación. Baños.	Existen 10 salas de baño disponibles. No se podrán reservar sesiones para salas adicionales.
RES_CLI_140	Reserva de ocio. Teatro.	El cliente del hotel podrá reservar un máximo de dos entradas para una sesión de teatro.
RES_CLI_150	Reserva de ocio. Teatro.	Para reservar una sesión de teatro será necesario un día, y un número de asientos.
RES_CLI_160	Reserva de ocio.	El cliente podrá decidir dónde se ubicarán los

	Teatro.	asientos reservados (fila y butaca) dentro del teatro.
RES_CLI_170	Reserva de ocio. Teatro.	Cada día habrá una única función. Se le informará al usuario de qué función teatral se realiza cada día.
RES_CLI_180	Cancelación de actividades	El cliente podrá cancelar las actividades que haya reservado anteriormente. Cancelar un servicio queda fuera del marco de la aplicación. Para hacerlo el cliente deberá ponerse en contacto con recepción.
RES_CLI_190	Reserva de servicios. Despertador	El cliente podrá solicitar a través de la aplicación una llamada despertador el día de su estancia y a la hora que se desee.
RES_CLI_200	Reserva de servicios. Servicio de habitaciones	El cliente podrá realizar pedidos de servicio de habitaciones. Cada pedido constará de uno o más elementos de la carta (comida o bebida).
RES_CLI_210	Reserva de servicios. Servicio de habitaciones	La aplicación ha de poder almacenar el precio de los pedidos de servicio de habitaciones realizados por el cliente, a fin de poder introducirlos en la factura.
RES_CLI_220	Reserva de servicios. Servicio de limpieza	El cliente podrá solicitar la limpieza de su habitación a través de la aplicación.
RES_CLI_230	Reserva de servicios. Servicio de limpieza	El personal del hotel deberá conocer si algún cliente solicita dos o más veces el servicio de limpieza en el transcurso de un mismo día.

Requisitos asociados al terminal del empleado

Id.	Tema	Descripción
RES_GES_010	Gestión de reservas	El empleado del hotel podrá efectuar reservas en nombre de los clientes, adjuntando los mismos datos que éstos, excepto la contraseña personal asociada a la habitación.
RES_GES_020	Gestión de reservas	El empleado del hotel podrá efectuar cancelaciones de reservas en nombre de los clientes. No será necesario adjuntar contraseña.
RES_GES_025	Gestión de reservas	El empleado del hotel utilizará la aplicación sólo para cancelar actividades, no servicios. Los servicios se cancelarán por los cauces habituales.
RES_GES_030	Gestión de reservas	El empleado del hotel podrá elaborar un informe automático en el que, para una actividad determinada, se listen las habitaciones que han reservado dicha actividad, en qué días y a qué horas.
RES_GES_040	Gestión de reservas	El empleado del hotel podrá elaborar un informe automático en el que, para una habitación determinada, se listen las actividades que han reservado, en qué días y a qué horas.
RES_GES_050	Gestión de reservas	El empleado del hotel podrá elaborar un informe automático en el que, para un determinado intervalo de días, se listen las habitaciones que han reservado alguna actividad determinada, qué actividad han reservado, en qué días y a qué horas.
RES_GES_060	Gestión de reservas	Las solicitudes de limpieza de habitación han de resolverse lo antes posible; los empleados deben recibir notificación de la solicitud en 10 minutos como máximo para poder limpiar la habitación en un tiempo razonable.
RES_GES_070	Gestión de reservas	Las solicitudes de servicio de habitaciones han de resolverse lo antes posible; los empleados deben recibir notificación de la solicitud en 5 minutos como máximo para poder limpiar la habitación en un tiempo razonable.

Análisis de los casos de uso.

En esta sección se describen los casos de uso de la aplicación, es decir, la manera en la que los usuarios finales interactuarán con la aplicación. Distinguiremos dos tipos de usuarios:

- Clientes, que serán los huéspedes del hotel.
- Empleados, que serán los empleados del hotel.

El diagrama de casos de uso de la aplicación es el siguiente:

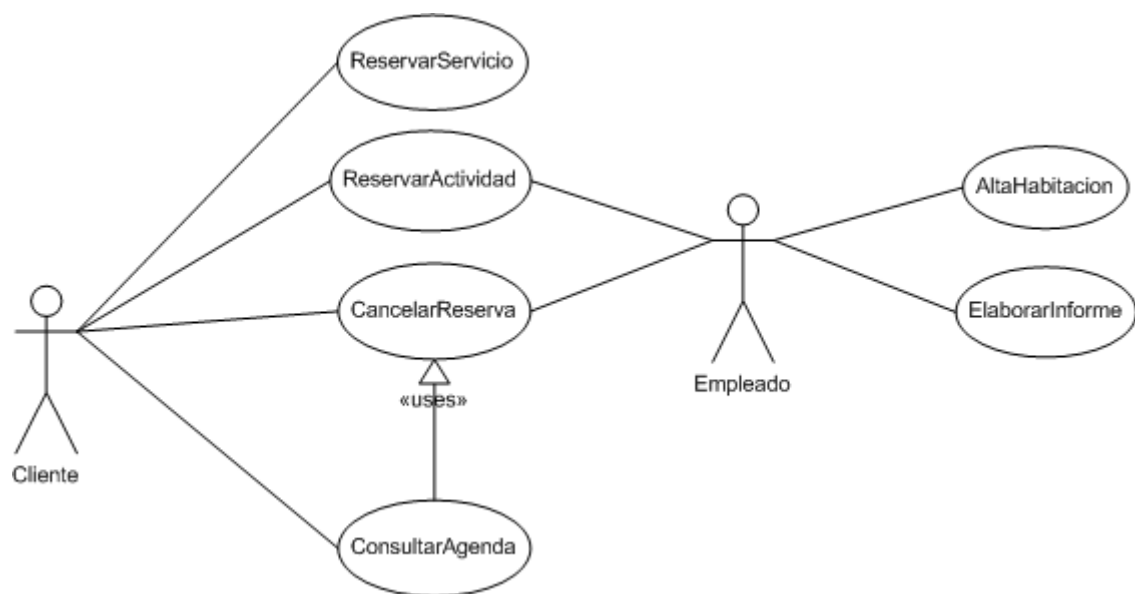


Ilustración 4: Casos de uso de la aplicación

Como se puede ver en el diagrama, tanto huéspedes como empleados comparten dos casos de uso, ya que ambos pueden tanto reservar nuevas actividades como cancelar reservas existentes.

En el siguiente apartado se explica en detalle cada caso de uso. Para más información sobre algunos términos el lector puede irse al Glosario de términos. La descripción de los casos de uso no ha cambiado desde el documento de análisis y diseño, pero su escritura se ha adaptado para no hacer referencia a los bocetos de la interfaz de usuario, como se hacía en aquel documento.

Caso de uso 1: Reservar Actividad

- **Resumen de la funcionalidad:** Se reserva una nueva actividad y se almacena en la Base de Datos.
- **Actores que intervienen:** Cliente y Empleado. Las acciones a desarrollar son las mismas, aunque la interfaz de usuario varíe según el actor.
- **Precondición:** Se conoce la actividad a reservar, el número de habitación correspondiente. Si el que reserva es un cliente, además se tiene la contraseña correspondiente a la habitación. Hay una conexión disponible con la Base de Datos del servidor.
- **Postcondición:** La Base de Datos contiene registrada la actividad que se acaba de reservar.
- **Flujo principal:** Si el que reserva la habitación es el cliente, la lista de acciones a realizar será:
 - El cliente indica que desea reservar una actividad
 - El cliente selecciona el tipo de actividad que desea reservar: deporte, bienestar u ocio. Si el cliente ha optado por deporte o bienestar, tendrá que elegir qué tipo de actividad deportiva o de bienestar elige.
 - El cliente ha de identificarse apropiadamente facilitando el número de habitación y la contraseña personal.
 - El cliente elige los datos adicionales de la reserva:
 - Si se trata de una actividad de pádel, indicará la hora a la que desee jugar y la pista de pádel en la que desee hacerlo (siempre que esté libre)
 - Si se trata de una actividad de bicicleta, indicará el número de bicicletas que desee usar y las horas que las reservará.
 - Se le solicitará la confirmación al usuario. Tras obtenerla la reserva se hará efectiva y se informará del resultado al usuario de la aplicación.
- **Flujo alternativo:** Si es un empleado del hotel quién efectúa la reserva en nombre del cliente, no será necesario introducir la contraseña de la aplicación. La interfaz de usuario será diferente de la mostrada al cliente del hotel.
- **Observaciones:** Se aplican las siguientes restricciones:
 - El cliente no puede reservar actividades fuera de su período de estancia en el hotel.

- El cliente no puede reservar más de una hora por día para las actividades de pádel y bienestar.
- El cliente no puede reservar más de 5 bicicletas para un mismo día.

Caso de uso 2: Reservar Servicio

- **Resumen de la funcionalidad:** Se reserva un nuevo servicio y se almacena en la Base de Datos.
- **Actores que intervienen:** Cliente. El empleado del hotel no puede usar la aplicación para reservar servicios en nombre del cliente.
- **Precondición:** Se conoce el servicio a reservar, el número de habitación y la contraseña correspondiente. Hay una conexión disponible con la Base de Datos del servidor.
- **Postcondición:** La Base de Datos contiene registrado el servicio que se acaba de reservar.
- **Flujo principal:** Se le presentarán las siguientes ventanas al cliente:
 - Posteriormente el usuario indicará que quiere reservar un servicio.
 - El cliente selecciona el servicio que desea reservar: servicio de llamada despertador, servicio de limpieza de habitaciones o servicio de habitaciones.
 - El cliente ha de identificarse apropiadamente facilitando el número de habitación y la contraseña personal.
 - El cliente elige los detalles de la reserva del servicio:
 - Si se trata de servicio de habitaciones, elegirá el contenido del pedido (comida y bebida).
 - Si se trata de una llamada despertador, elegirá el día y la hora a la que esa llamada va a sonar.
 - Si se trata de una solicitud de limpieza de habitación, elegirá el día y la hora a la que ese servicio se va a efectuar.
 - Se le solicitará confirmación y posteriormente se le confirmará que la reserva se ha efectuado satisfactoriamente.
- **Observaciones:** Se aplican las siguientes restricciones:
 - El cliente no puede reservar servicios fuera de su período de estancia en el hotel.

Caso de uso 3: Cancelar Reserva

- **Resumen de la funcionalidad:** El cliente cancela una actividad anteriormente reservada.
- **Actores que intervienen:** Cliente y Empleado, aunque a través de interfaces de usuario diferentes.
- **Precondición:** El cliente ha realizado alguna reserva previamente y quiere cancelarla. El cliente conoce su número de habitación y su contraseña asociada. Hay una conexión disponible con la Base de Datos del servidor.
- **Postcondición:** La reserva aparece como cancelada, aunque no se borra del sistema, ya que el hotel quiere auditar las actividades canceladas.
- **Flujo principal:** Si es el cliente quién realiza la operación, los pasos que se siguen son:
 - El cliente selecciona la opción “Cancelar actividad” en la ventana correspondiente a la Ilustración 2.
 - El cliente ha de identificarse apropiadamente facilitando el número de habitación y la contraseña personal.
 - Se mostrarán las actividades asociadas a la habitación correspondiente, y el usuario podrá seleccionar la actividad/es a cancelar.
 - Se le pedirá confirmación al usuario sobre la cancelación.
 - Se le mostrará una confirmación al usuario de la cancelación efectuada.
- **Flujo alternativo:** Si es el empleado el que realiza la operación, no será necesario introducir la contraseña. La interfaz de usuario que se mostrará al empleado será diferente de la mostrada al cliente.
- **Observaciones:** Se avisará al usuario si no hay actividades reservadas que se puedan cancelar.

Caso de uso 4: Consultar Agenda

- **Resumen de la funcionalidad:** El usuario consulta las actividades que ha reservado.
- **Actores que intervienen:** Cliente. El empleado puede acceder a la misma información a través de la generación de informes.
- **Precondición:** El cliente ha realizado como mínimo una reserva previamente y quiere consultarla/s. El cliente conoce su número de habitación y su contraseña asociada. Hay una conexión disponible con la Base de Datos del servidor.
- **Postcondición:** El cliente obtiene una lista de actividades reservadas por él.
- **Flujo principal:**
 - El cliente indicará que desea consultar la lista de todas las actividades reservadas durante su estancia.
 - El cliente ha de identificarse apropiadamente facilitando el número de habitación y la contraseña personal.
 - Se muestran las actividades reservadas por el cliente en el período de su estancia. Cuando el cliente desee, se volverá a la pantalla principal.
- **Observaciones:** Las actividades se presentarán ordenadamente, según se haga efectiva la reserva antes o después.

Caso de uso 5: Alta Habitación

- **Resumen de la funcionalidad:** Se da de alta en la base de datos a un nuevo huésped del hotel.
- **Actores que intervienen:** Empleado.
- **Precondición:** Se conoce la habitación donde el huésped o los huéspedes van a alojarse y el período de tiempo en el que lo van a hacer. Hay una conexión disponible con la Base de Datos del servidor.
- **Postcondición:** El cliente está dado de alta en el sistema y puede efectuar reservas por sí mismo a través de un terminal. El cliente posee una contraseña numérica que le permitirá identificarse de manera segura.
- **Flujo principal:**
 - El empleado del hotel introduce el número de habitación de la reserva.
 - El empleado genera un número aleatorio que servirá de contraseña.
 - El empleado introducirá el período de estancia de los huéspedes, a fin de que se puedan reservar actividades sólo en el período en el que estén alojados.
 - El empleado valida los datos y los envía al servidor.

- **Observaciones:** La contraseña quedará almacenada en la Base de Datos. Al usuario se le suministrará junto con la llave de su habitación.

Caso de uso 6: Elaborar Informe

- **Resumen de la funcionalidad:** El empleado del hotel elabora un informe que contiene datos de reserva de actividades por parte de los clientes del hotel.
- **Actores que intervienen:** Empleado
- **Precondición:** Hay una conexión disponible con la Base de Datos.
- **Postcondición:** Se obtiene un informe con la información deseada por el empleado.
- **Flujo principal:**
 - El empleado selecciona “Consultas” desde el menú principal de la aplicación.
 - El empleado elegirá que consulta desea hacer en función de sus deseos: puede elegir un listado de:
 - Las actividades.
 - Las actividades finalizadas.
 - Las actividades canceladas.
 - Se mostrará el informe al empleado.

Diseño de la aplicación

En esta aplicación se describen las principales decisiones de diseño de los tres subsistemas, prestando especial atención al subsistema de Base de Datos y lógica de negocio, que es el que contiene las principales responsabilidades de toda la aplicación.

Subsistema de Lógica de Negocio y Base de Datos.

Este subsistema representará la capa servidor y se ha de comunicar con las capas cliente usando la tecnología WCF (*Windows Communication Foundation*). Esta tecnología -que se describirá posteriormente con más detalle-, implica que debe existir una interfaz que contenga todas las operaciones que la capa cliente puede solicitar y una clase que implemente dicha interfaz. La clase y la interfaz son, respectivamente *ServidorUOC* e *IServidorUOC*. La clase *ServidorUOC* usará a las diversas clases gestoras para realizar las reservas y devolver las reservas existentes. Para ello, se recurre a dos clases asociativas que representarán los datos que “viajarán” del cliente al servidor y del servidor al cliente. Cuando la capa cliente solicita información sobre reservas existentes, o cuando la misma capa solicita efectuar una reserva, la información estará incluida en una de estas clases: son las que comienzan por la palabra *contrato*.

Cada una de las clases gestoras recurrirá a la base de datos para obtener o actualizar información. El acceso a la base de datos se realiza mediante una clase llamada *GestorBBDD*, que implementa dos patrones:

- El patrón **Fachada**, que simplifica la interfaz de una clase o librería de clases compleja. La tecnología ADO.NET proporciona multitud de modos de acceso a los datos (procedimientos almacenados en T-SQL, sentencias SQL, objetos *DataReader*, *DataAdapter*...) que pueden acabar haciendo el acceso a los datos complejo, especialmente si trabajan varios desarrolladores en el subsistema y cada uno gestiona los datos de manera diferente. Mediante la clase gestora, tendremos garantizado que el acceso a los datos se hará de manera homogénea y así el testeo y posterior mantenimiento será más sencillo.
- El patrón **Singleton**, que asegura que sólo habrá una conexión abierta a la base de datos, independientemente de la clase que acceda a ella.

Al no disponer de interfaz gráfica, la capa servidor cuenta con un fichero de registro -log- donde se registran los accesos, y que proporciona información importante en caso de errores.

También se registra la fecha del acceso y de cada petición. Esta clase también implementa el patrón Singleton para asegurar que sólo hay una instancia de la clase en memoria.

En la siguiente página se incluye el diagrama de clases del subsistema de acceso a datos y lógica del negocio. No se muestran las clases GestorBBDD ni Log porque dificultarían la legibilidad del diagrama.

Posteriormente se ha incluido un diagrama relacional de la Base de datos, donde se pueden apreciar:

- Todas las tablas.
- Los atributos de cada tabla.
- Las relaciones entre tablas.
- La clave principal de cada tabla, simbolizada por la imagen de una llave amarilla.

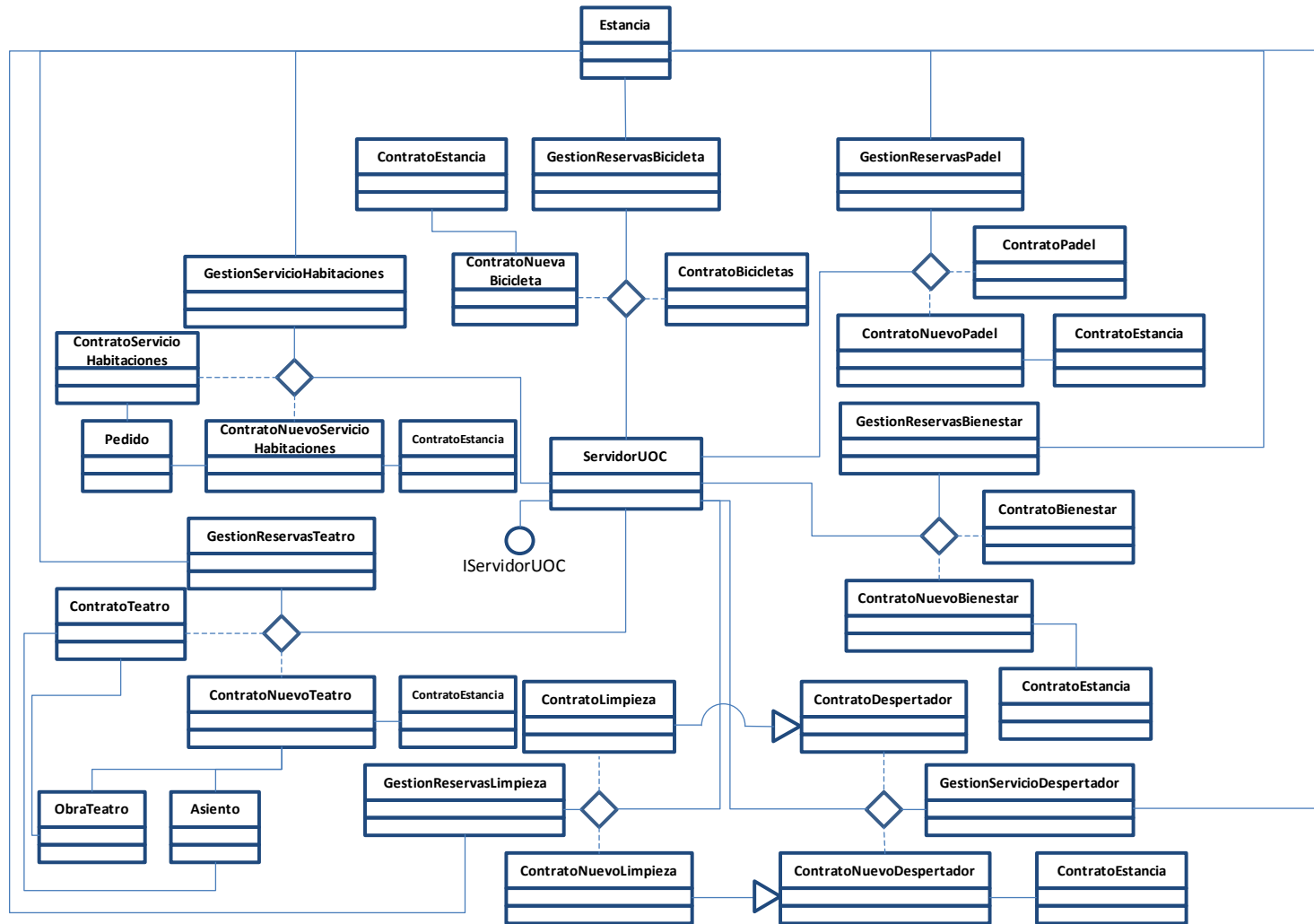


Ilustración 5: Diagrama estático de la lógica del negocio

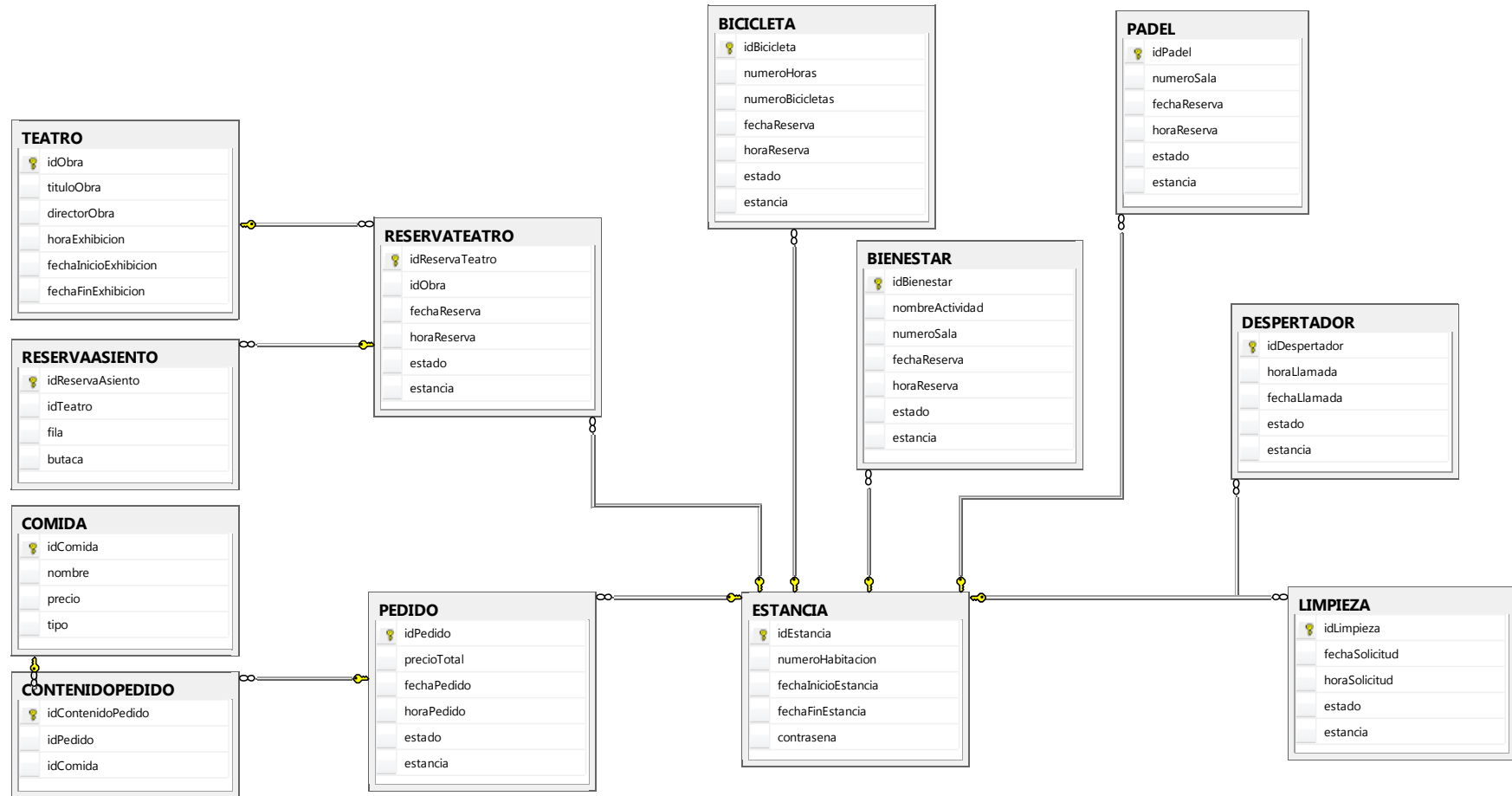


Ilustración 6: Diagrama relacional de la base de datos

Subsistema Terminal de Empleado

Este subsistema engloba la aplicación que usarán los empleados del hotel para gestionar las reservas de actividades de los huéspedes del hotel. Debido a que este subsistema pertenece a la capa cliente, tiene una estructura sencilla, ya que todas las tareas complejas son realizadas en la capa de servidor: de ahí que las clases sean todas de interfaz. En la sección de Productos entregados se incluyen capturas de todas estas pantallas. Este subsistema consta de las siguientes clases:

- **Window1:** representa la pantalla que se muestra al usuario cuando se ejecuta la aplicación. Contiene un menú con todas las funcionalidades y tres listas donde se muestran los servicios contratados por los clientes y que van a hacerse efectivos próximamente.
- **NuevaActividad:** representa la pantalla que permite al usuario reservar nuevas actividades en nombre de los huéspedes.
- **CancelarReserva:** representa la pantalla que permite al empleado del hotel cancelar reservas en nombre de los huéspedes.
- **NuevaEstancia:** contiene la pantalla que permite crear nuevas estancias -nuevos huéspedes alojándose en una habitación durante un determinado periodo de tiempo.
- **VentanaInforme:** esta clase contiene la funcionalidad de mostrar informes con datos procedentes de la base de datos del servidor. En la sección “Tecnologías usadas” se describirá brevemente cómo se generan este tipo de informes.

Subsistema Terminal de Huésped

Este subsistema engloba la aplicación que usarán los huéspedes del hotel para gestionar reservas de actividades y servicios. Al igual que el anterior subsistema, el subsistema de Terminal de Huésped tiene una estructura sencilla, ya que todas las tareas complejas son realizadas en la capa de servidor: de ahí que las clases sean todas de interfaz. Hay numerosas clases en este subsistema, pero la mayoría sólo sirven para mostrar unas pocas opciones al usuario, y su implementación es muy sencilla. Las clases que realizan tareas de más envergadura son:

- **frmNumHab:** esta clase representa la pantalla para solicitar el número de habitación del huésped, y contacta con el servidor para obtener la contraseña asociada a dicha estancia.
- **frmContrasena:** esta clase contiene la lógica asociada a la obtención de la contraseña por parte del cliente y su comparación con la contraseña almacenada en la Base de Datos. Si la información es correcta, mostrará la ventana de detalles de la reserva solicitada.
- **frmReservaBicicleta, frmReservaBienestar y frmReservaPadel:** representan las pantallas para solicitar los datos relativos a las reservas de actividades de bicicleta, bienestar y pádel, respectivamente. Contienen toda la lógica necesaria para chequear la consistencia de los datos, y solicita al servidor información relativa a otras reservas realizadas en las mismas fechas.
- **frmConfirmacionReserva:** esta clase solicita al servidor que realice la reserva con los datos recabados en alguna de las reservas anteriores. Esta clase usa un delegado para llamar al método correcto en función de la reserva solicitada por el usuario.
- **frmCancelacion:** esta clase muestra las actividades reservadas por el usuario durante su estancia, bien porque el usuario desea conocer las reservas que tiene pendientes o bien porque desea cancelar una de éstas. En el caso de que desee cancelar una actividad, las reservas de la lista llevarán asociados *checkboxes* para que el usuario de la aplicación marque qué reservas desea cancelar.
- **frmLimpiezaDespertador:** esta clase gestiona todo lo necesario para reservar servicios de limpieza o de llamada despertador. Almacena toda la información necesaria y después solicita al servidor que efectúe la reserva.

- **frmServicioHabitaciones.** Esta clase gestiona todo lo necesario para reservar servicios de servicio de habitaciones. Almacena toda la información necesaria y después solicita al servidor que efectúe la reserva.

Implementación. Tecnologías usadas

En esta sección se detallan las principales tecnologías usadas en el desarrollo del proyecto. No se pretende describirlas exhaustivamente, sino sólo dar una descripción general y su aplicación en el proyecto.

Acceso a datos ADO.NET

ADO.NET es la tecnología de Microsoft para acceso y modificación de datos almacenados en un Sistema Gestor de Bases de Datos (SGBD). Permite acceder a diversas fuentes de datos, como Microsoft SQL Server, Oracle, Microsoft Access o MySQL. El lector puede obtener una visión general de la tecnología ADO.NET en este [enlace](#).

El proyecto usa ADO.NET para consultar y actualizar datos de la Base de Datos. Para ello es necesario usar un objeto SqlConnection que nos permita conectarnos a la Base de Datos. Posteriormente, usando un objeto DataAdapter –objeto que nos permitirá ejecutar comandos sobre una Base de Datos- podremos introducir los datos desde la Base de Datos en un objeto DataSet –objeto que representa una base de datos en miniatura, incluyendo tablas y relaciones entre ellas-. Las inserciones se efectuarán usando objetos SqlCommand.

ADO.NET emplea diversos modos de acceder y tratar los datos. En este proyecto se ha usado el lenguaje SQL para efectuar las consultas y realizar las inserciones y actualizaciones.

Consultas integradas en el lenguaje (LINQ)

En la versión 3.5 de la plataforma .NET se incluyó la tecnología LINQ -Language INtegrated Query- que añade consultas semejantes a las SQL estándar a los lenguajes de la plataforma. De esta manera, es posible realizar consultas a:

- Cualquier colección que implemente la interfaz IEnumerable: listas, arrays, diccionarios...
- Documentos XML.
- Bases de datos.
- Sistemas de archivos.

Se puede encontrar más información sobre LINQ en este [enlace](#). El proyecto utiliza LINQ dentro del subsistema Terminal de cliente, para procesar los datos devueltos por el servidor que incluyen reservas ya realizadas. Cuando el huésped desea realizar una reserva, es necesario

conocer qué otras reservas hay realizadas para esas mismas fechas, con el fin de no reservar dos actividades para el mismo día a la misma hora. El servidor devuelve una lista de reservas, y es necesario saber qué reservas hay el mismo día que selecciona el cliente para efectuar la reserva. Sin usar LINQ, sería necesario recorrer toda la lista devuelta por el servidor, interrogando a cada elemento reserva de la lista por su fecha: si la fecha coincide con el día seleccionado por el cliente, se procesará posteriormente. Usando LINQ solamente es necesaria una sola sentencia.

Windows Communication Foundation (WCF)

WCF es una tecnología nueva de Microsoft que aúna todas las tecnologías anteriores de Microsoft para conectar equipos conectados en red, y es la tecnología base para crear aplicaciones orientadas a servicios. Se puede encontrar más información de WCF en el siguiente [enlace](#). Para comprender mejor WCF es necesario definir primero algunos términos:

- **Endpoint.** Elemento que proporciona los detalles del servicio que las aplicaciones cliente requieren para poder comunicarse con él. Un endpoint engloba tres conceptos secundarios, a los que se conoce como el ABC de WCF: Address, Binding y Contract.
- **Address.** Responde a la pregunta ¿dónde se encuentra el servicio? Representa la URI donde el cliente puede encontrar el servicio, por ejemplo <http://localhost/miServicio>.
- **Binding.** Responde a la pregunta ¿cómo me comunico con el servicio? Engloba aspectos como el protocolo de comunicación, codificación o el cifrado.
- **Contract.** Responde a la pregunta ¿qué me ofrece el servicio? Engloba las operaciones que el servicio proporciona, y los tipos de datos que el servidor devuelve. Hay varios tipos de Contract:
 - **Service Contract.** Es la interfaz que engloba todas las operaciones que expone el servicio.
 - **Operation Contract.** Es una operación que el servicio ofrece al cliente, y que se encuentra declarada dentro del Service Contract.
 - **Data Contract.** Es una clase que será devuelta por el servicio como resultado de una llamada a una de sus operaciones.
- **Host Process.** Es necesario que el servicio WCF esté alojado en un proceso huésped. Este proceso puede ser una aplicación de Windows (WinForms o WPF) o incluso un servicio de Windows.

Una vez el servicio esté alojado, las aplicaciones que deseen comunicarse con él deberán emplear la dirección –Address- especificada en el mismo. En el proyecto, todas las operaciones relacionadas con reservas de actividades, cancelación de reservas u obtención de reservas se ofrecen como operaciones, estando disponibles para los clientes que lo soliciten.

Una vez desarrollado el servicio, es necesario alojarlo en una aplicación huésped. Para hacerlo más sencillo se ha elegido una aplicación WinForms, que es capaz de alojar el servicio y de ponerlo en marcha para que escuche peticiones externas. Una vez puesto en marcha, las aplicaciones cliente pueden acceder y operar con él.

Desde el punto de vista del desarrollo, el entorno Visual Studio facilitaba enormemente el enlace entre cliente y servidor, siendo necesario solamente poner en marcha el servicio e indicar al cliente que el servicio deseado se encontraba en la dirección determinada por la propiedad Address. En ese momento, el explorador de objetos de Visual Studio mostraba las operaciones y las clases expuestas por el servicio, y el Intellisense proporcionaba ayudas a la escritura de código relacionadas con la llamada a operaciones del mismo.

Windows Presentation Foundation (WPF)

WPF es una tecnología nueva de Microsoft que redefine la manera de elaborar interfaces de usuario. Engloba interfaces de usuario, gráficos en 2D y 3D, documentos y multimedia. Se puede encontrar más información de WPF en este [enlace](#). Algunas características de esta tecnología son:

- **Separación de la apariencia y el comportamiento.** Ahora, la apariencia de la interfaz gráfica (tipo de letra, tamaño de los controles, colores...) está especificada en el fichero XAML (Extensible Application Markup Language) y el comportamiento se especifica en un fichero de código C# o Visual Basic. Ambas partes permanecen unidas a través de eventos y enlace de datos *data binding*.
- **Controles compuestos.** Se pueden crear controles de casi cualquier tipo, y casi cualquier control puede otro. Ahora es posible crear una lista desplegable *ComboBox* que contenga elementos consistentes en una imagen, un *Checkbox* y una etiqueta de texto.
- **Apariencia personalizable.** La apariencia de los controles es mucho más flexible que antes. El uso de estilos –*styles*- y plantillas –*templates*- la hacen todavía más flexible.
- **Independiente de la resolución.** Los controles podrán ser del tamaño que se desee, y como están basados en gráficos vectoriales, mantendrán intacta su apariencia.

Toda la capa cliente está programada usando WPF: el subsistema Terminal de Empleado y el subsistema Terminal de Cliente. Cada clase de interfaz consta de dos ficheros, el fichero XAML, que define la apariencia, y el fichero de código, llamado *code behind*, que implementa la lógica de la clase. El fichero XAML lleva información de los métodos (implementados en el fichero de código) que se ejecutarán cuando el usuario interactúe con los controles de la pantalla.

Visual Studio facilitaba enormemente la edición y creación de las interfaces de usuario, siendo posible editar directamente el fichero XAML, viendo los cambios al momento en pantalla, o a través de las tablas de propiedades, como era normal en WinForms.

Reporting Services

Reporting Services es una tecnología de Microsoft que permite elaborar informes de fuentes de datos de manera sencilla. Ofrece menos posibilidades que Crystal Reports, pero en cambio ofrece mucha más sencillez, pudiendo crear un informe completo de gran apariencia en pocos minutos. Se puede encontrar más información en este [enlace](#).

Para crear el informe, se necesita una fuente de datos, que puede ser:

- Una base de datos.
- Un objeto.
- Un servicio.

Posteriormente es necesario crear la apariencia del informe, indicando qué campos del origen de datos aparecen en el mismo.

Todos estos pasos pueden hacerse a través de un asistente, que facilita la creación del origen de datos y la apariencia del informe.

En tiempo de ejecución el informe se muestra en una ventana que posibilita navegar por los diferentes registros, exportar los contenidos a Excel o a pdf o imprimir el informe. En la sección de productos entregados se incluye una captura de pantalla con un informe ya creado.

Productos entregados

En esta sección se incluyen capturas de pantalla de las aplicaciones desarrolladas en el marco del proyecto. No se pretende incluir todas las pantallas de las aplicaciones, sino sólo las más importantes.

Los productos entregados en el marco de este proyecto son tres:

- **DespliegueServidor:** esta aplicación contiene el servicio y la aplicación WinForms que hace de huésped.
- **EmpleadoUOC:** esta aplicación contiene el código del subsistema Terminal de Empleado.
- **HuespedUOC:** esta aplicación contiene el código del subsistema Terminal de Cliente.

La aplicación DespliegueServidor sólo contiene una pantalla, necesaria para arrancar y detener el servicio:

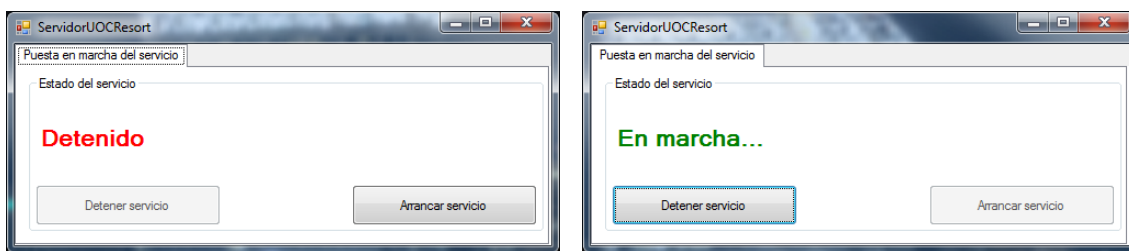


Ilustración 7: DespliegueServidor

Una vez el servidor esté en marcha, las aplicaciones EmpleadoUOC y HuespedUOC pueden ejecutarse sin problemas.

La ventana principal de la aplicación EmpleadoUOC se puede ver en la página siguiente. En ella se pueden observar las reservas de servicios que los clientes han solicitado para los próximos cuatro días. Actualmente el margen de cuatro días no puede ser modificado, pero en la sección de Líneas abiertas se contempla esta mejora. En la imagen siguiente se puede observar que hay una reserva pendiente del servicio de limpieza para el 8 de Junio a las 8 de la mañana. Sin embargo, no hay reservas pendientes de servicio de habitaciones ni de servicio de llamada despertador.

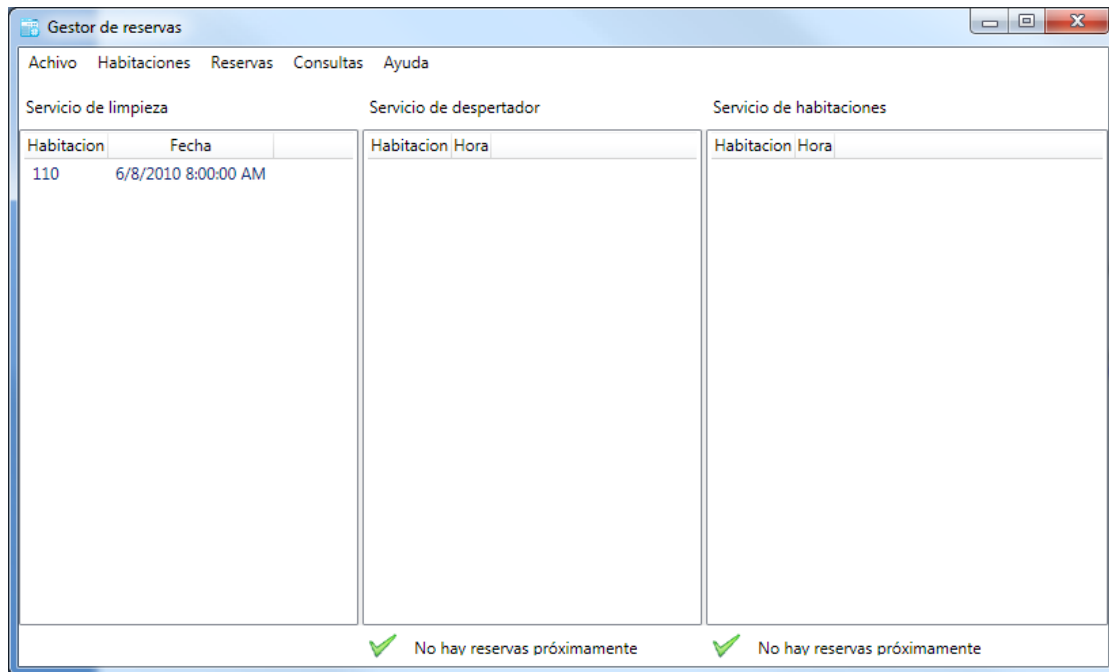


Ilustración 8: EmpleadoUOC. Pantalla principal

La pantalla para crear una nueva estancia es la siguiente:

Numero de habitacion: 108

Contraseña: 5110

Duracion de la estancia

Desde: 16/06/2010

Hasta: 25/06/2010

Aceptar Cancelar

Ilustración 9: EmpleadoUOC. Creación de una nueva estancia

En esta pantalla el empleado del hotel puede crear una nueva estancia para permitir que nuevos huéspedes puedan reservar actividades usando la aplicación HuespedUOC.

Si lo que desea el empleado del hotel es reservar una actividad en nombre de algún huésped, tendrá que usar la siguiente pantalla:

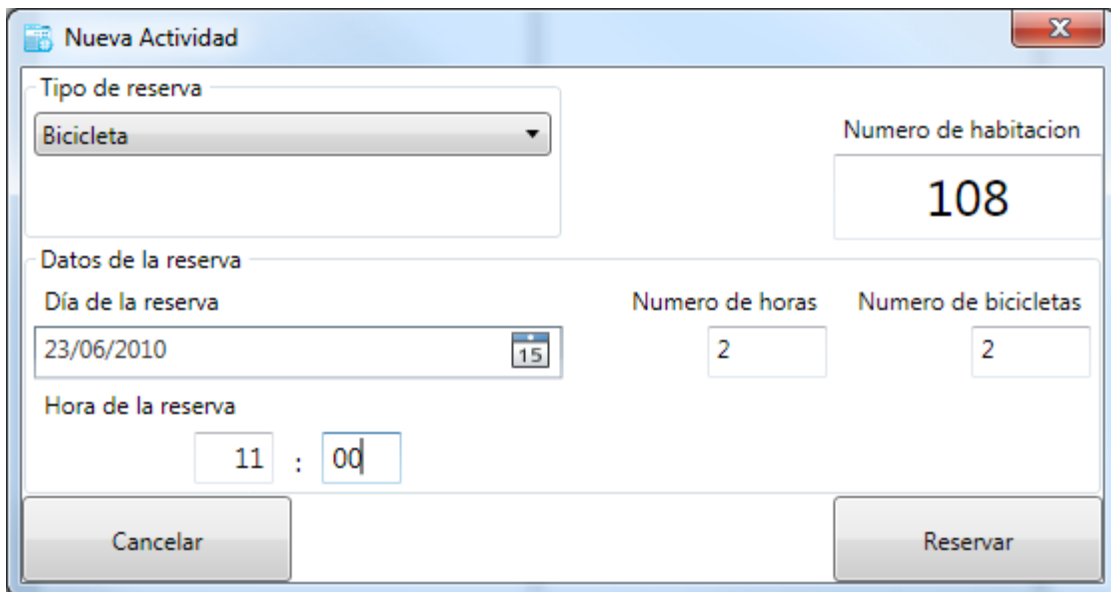


Ilustración 10: EmpleadoUOC. Creación de una nueva reserva

Todas las actividades se reservan desde la misma pantalla: los controles variarán dinámicamente según la actividad escogida para así solicitar la información adecuada a la reserva.

Si por el contrario lo que se desea es cancelar una reserva, la ventana donde se realiza esta acción es:

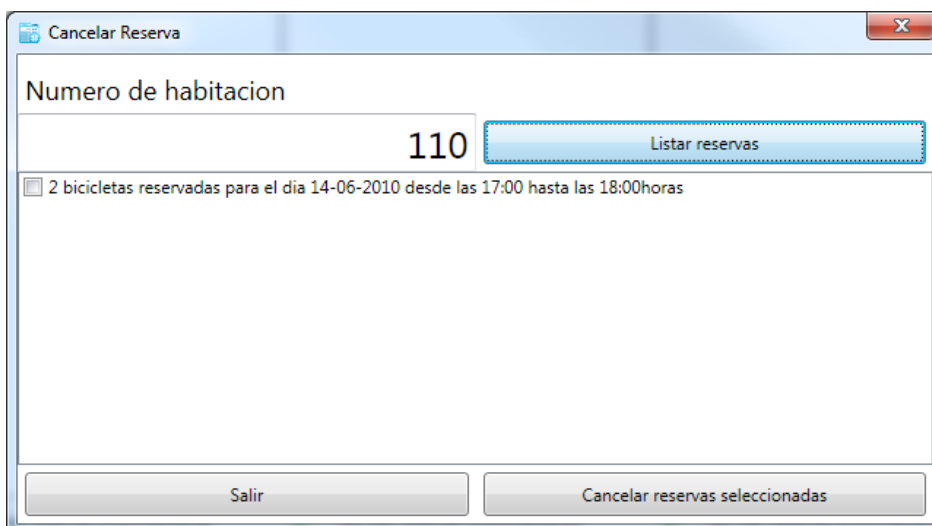
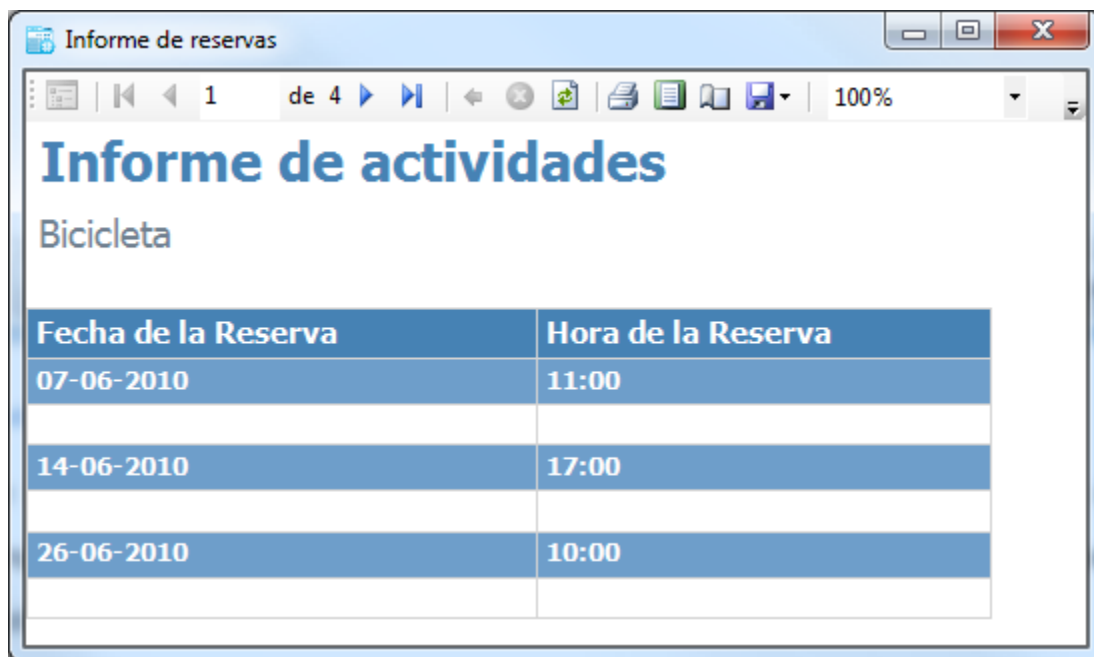


Ilustración 11: EmpleadoUOC. Cancelación de una reserva

La ventana muestra una lista de las actividades reservadas por una habitación determinada; el empleado del hotel sólo tendrá que seleccionar la actividad que se desee cancelar y pulsar el botón *Cancelar reservas seleccionadas*.

Para obtener informes, el usuario ha de seleccionar una cualquiera de las opciones del menú *Consultas*. Actualmente sólo hay tres opciones disponibles: listar las actividades pendientes, listar las actividades realizadas y listar las actividades canceladas. Eligiendo cualquiera de ellas se mostrará la ventana de los informes:



Fecha de la Reserva	Hora de la Reserva
07-06-2010	11:00
14-06-2010	17:00
26-06-2010	10:00

Ilustración 12: EmpleadoUOC. Informes

Desde esta ventana el usuario puede navegar por el informe para ver cuáles han sido las actividades que se han reservado y se han realizado, cuáles se han cancelado y cuáles están todavía pendientes de realizarse. Asimismo, el usuario puede:

- Imprimir el informe, pudiendo incluso configurar la página para ello.
- Exportar el informe en formato Microsoft Excel o Adobe PDF
- Cambiar el zoom para aumentar o disminuir el tamaño de la tabla.

La aplicación HuespedUOC está diseñada y concebida de manera diferente a la de EmpleadoUOC. HuespedUOC ha de ser sumamente sencilla de usar, y solamente podrá usarse un ratón para seleccionar sus diferentes opciones, ya que en el futuro se prevé usar pantallas táctiles para esta aplicación, a fin de hacerla aún más sencilla de usar.

La ventana principal de la aplicación es:

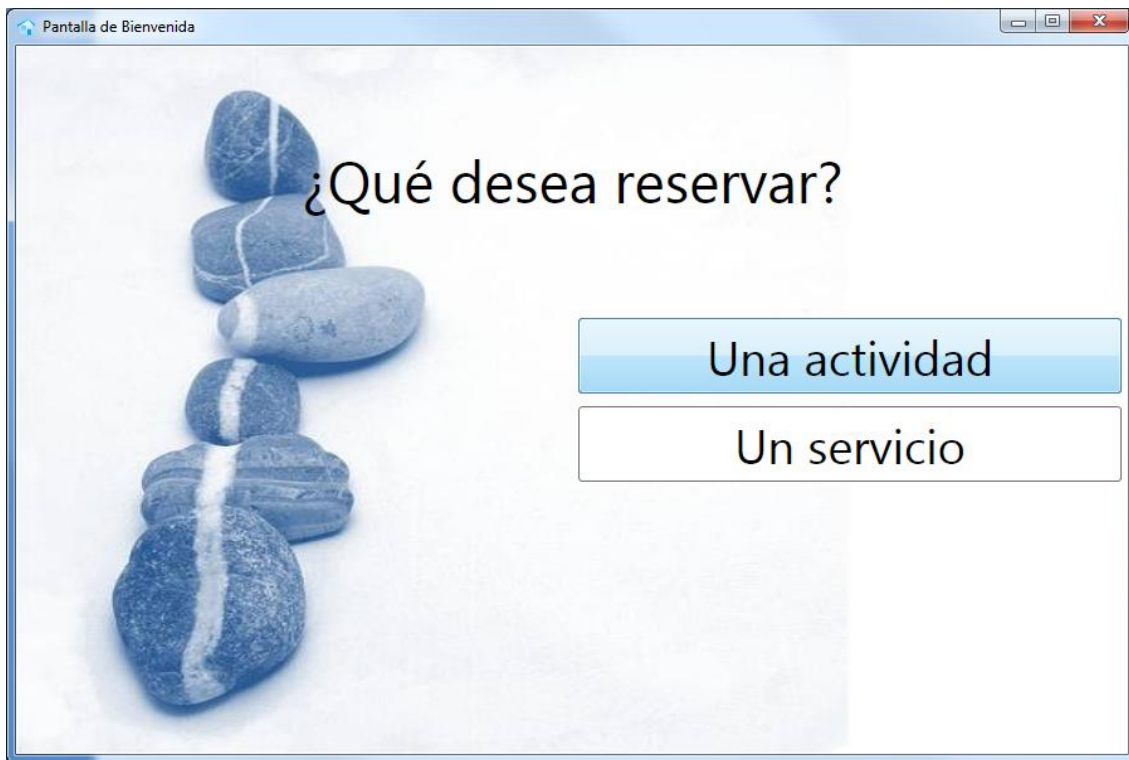


Ilustración 13: HuespedUOC. Ventana principal

Desde esta ventana el usuario podrá navegar por todas las opciones que le ofrece la aplicación: reservar y cancelar actividades, consultar las actividades ya reservadas y reservar servicios. En cualquier momento, si se ha equivocado de opción o ha cambiado de opinión, el usuario podrá volver atrás tantas veces como quiera hasta llegar a la ventana principal.

Sea lo que sea lo que el huésped desee reservar, tendrá que identificarse apropiadamente introduciendo su número de habitación y su contraseña. Las ventanas para realizar estas tareas se muestran en la siguiente página. Aunque solicitar una contraseña puede parecer engorroso, es necesario para evitar que se hagan reservas no deseadas en nombre de otras habitaciones.

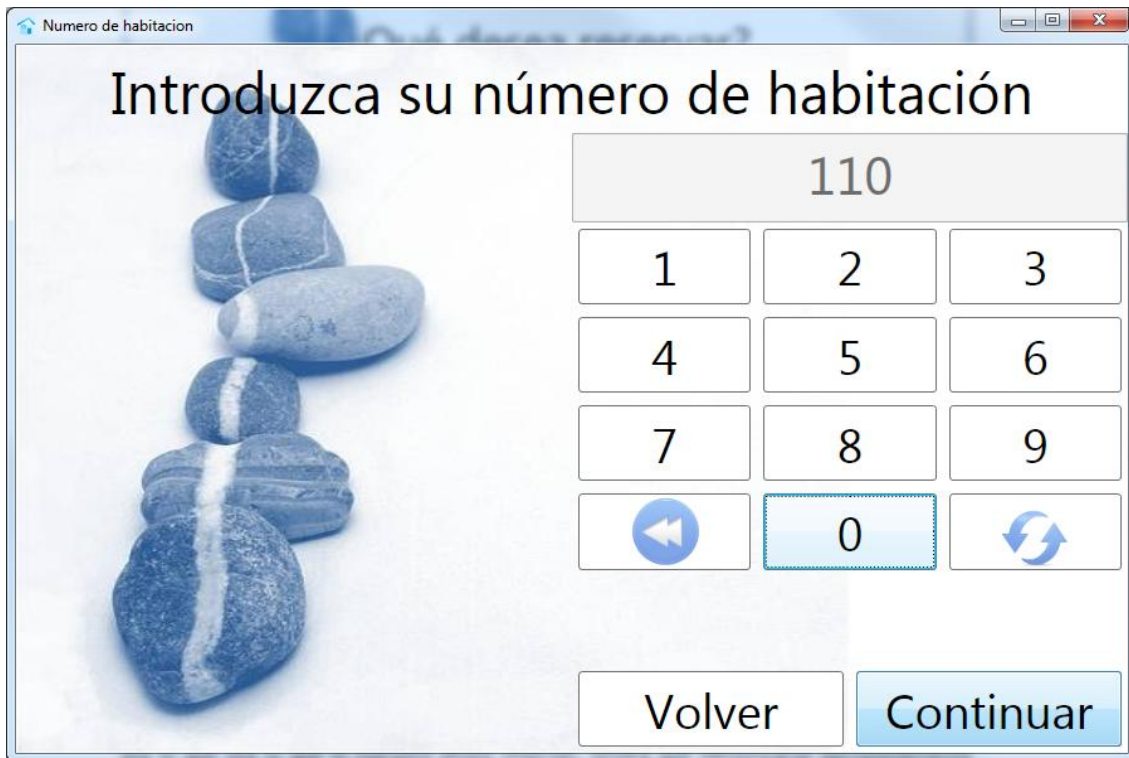


Ilustración 14: HuespedUOC. Introducción del número de habitación

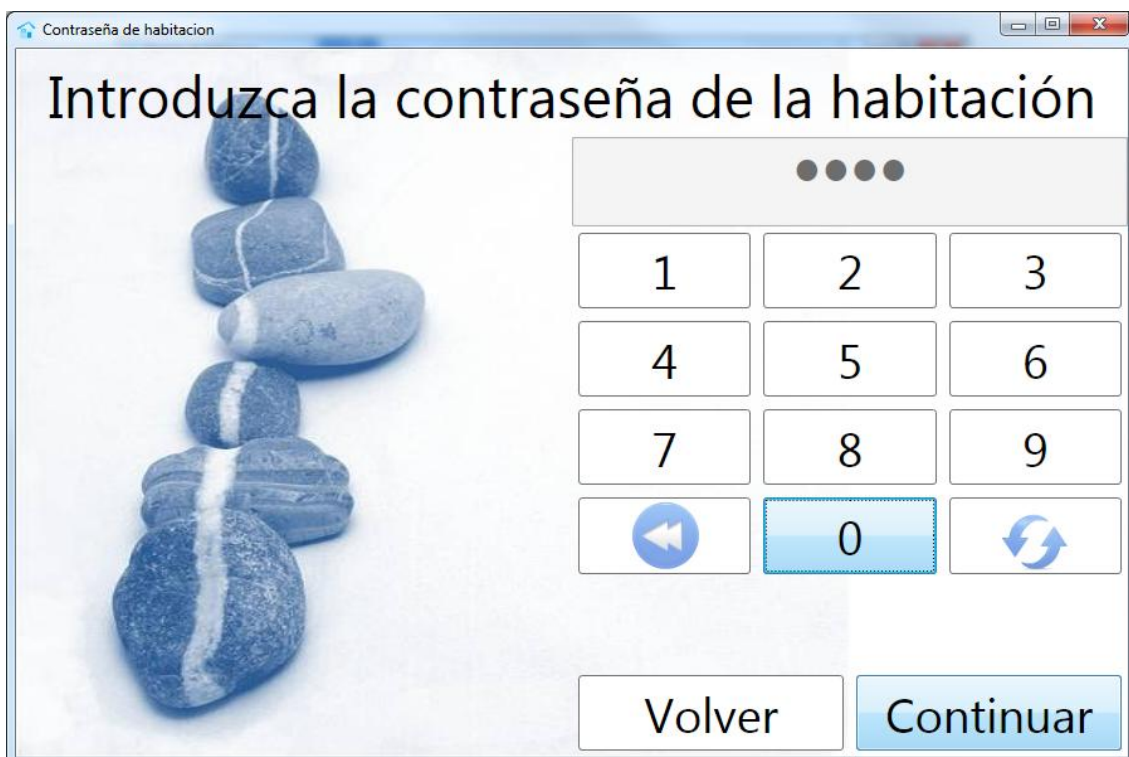
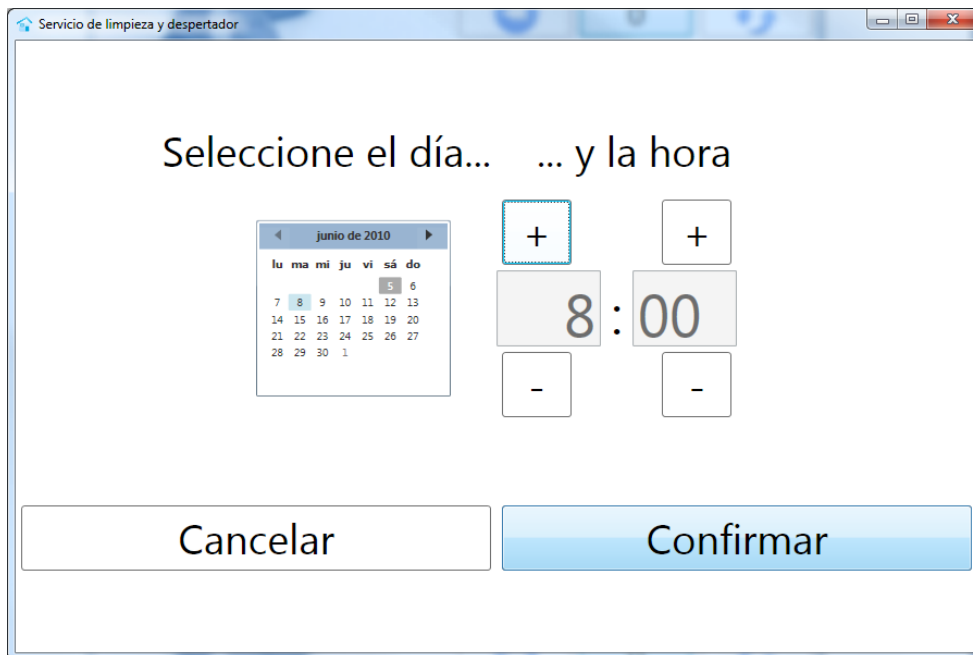


Ilustración 15: HuespedUOC. Introducción de la contraseña

Una vez identificado apropiadamente, el usuario tendrá que seleccionar las diferentes opciones que cada actividad o servicio conlleve consigo. Por ejemplo, para el caso de servicio de limpieza de habitación o servicio de llamada despertador, la ventana que aparece es:



Servicio de limpieza y despertador

Seleccione el día... y la hora

junio de 2010

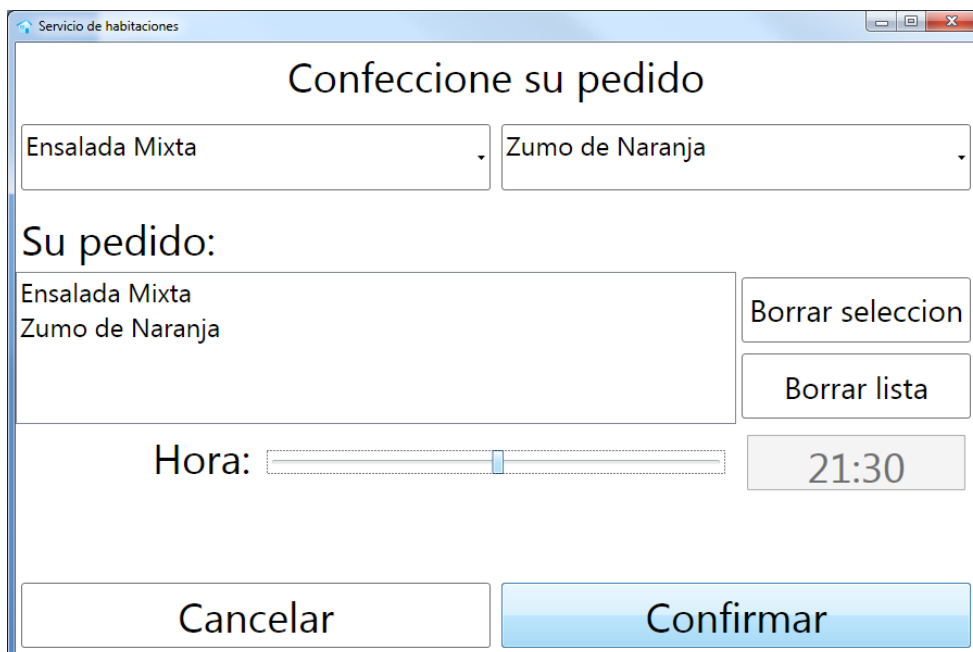
lu	ma	mi	ju	vi	sá	do
					5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1			

8:00

Cancelar Confirmar

Ilustración 16: HuespedUOC. Servicio de limpieza o de despertador

Si ha seleccionado servicio de habitaciones, la ventana será la siguiente:



Servicio de habitaciones

Confeccione su pedido

Ensalada Mixta Zumode Naranja

Su pedido:

Ensalada Mixta
Zumode Naranja

Borrar seleccion

Borrar lista

Hora: 21:30

Cancelar Confirmar

Ilustración 17: Ilustración 17: HuespedUOC. Servicio de habitaciones

Si ha solicitado reservar una actividad, las ventanas tendrán una apariencia ligeramente diferente. Por ejemplo, para reservar una excursión en bicicleta se mostrara:

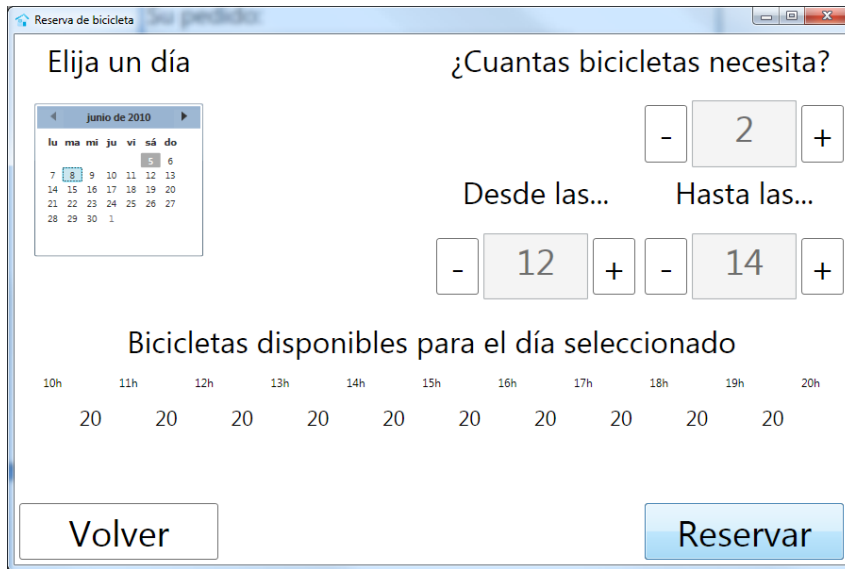


Ilustración 18: HuespedUOC. Reserva de bicicletas

En esta pantalla el usuario puede seleccionar y fijar todos los parámetros relativos a la reserva. Asimismo, al seleccionar un día en el calendario, en la tabla central se muestran las bicicletas libres a lo largo del día. Si la reserva es de pádel o de bienestar, la pantalla es casi idéntica, y en la tabla central se muestran las salas libres a los largo del día, en las que el huésped deberá reservar.

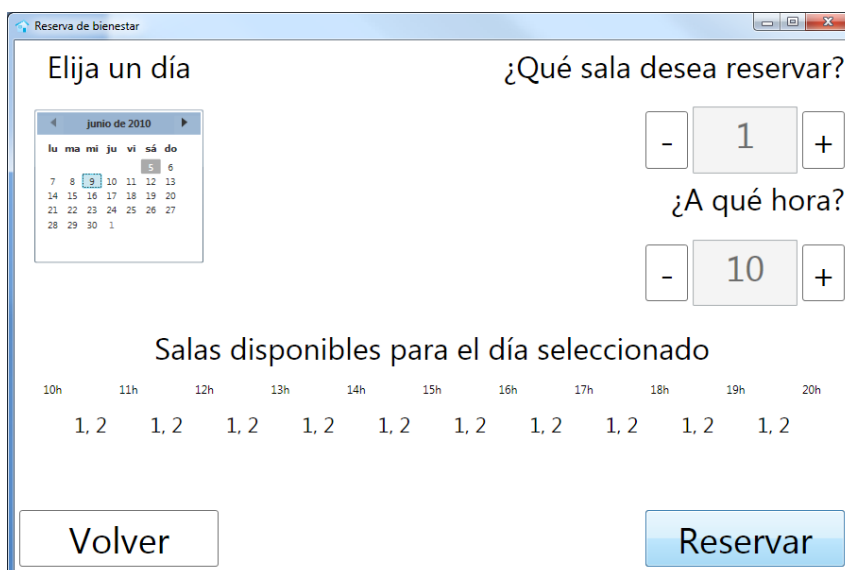


Ilustración 19: HuespedUOC. Reserva de bienestar

Una vez configurada la reserva, en el caso de las actividades se le requerirá confirmación al usuario mediante la siguiente ventana:

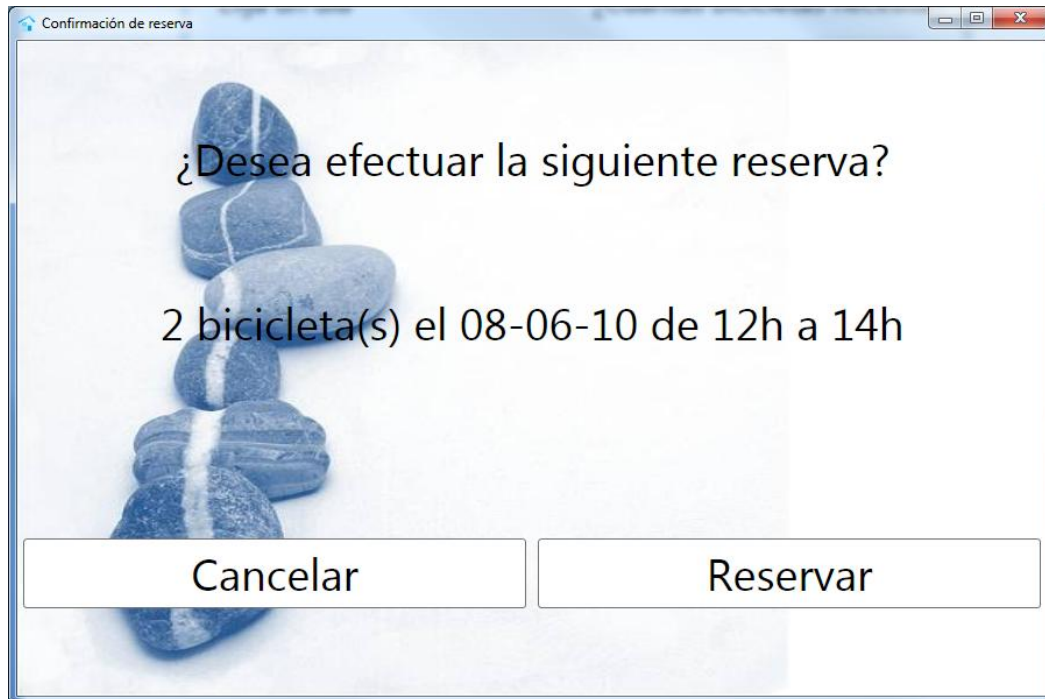


Ilustración 20: HuespedUOC. Confirmación de reserva

Una vez dada la confirmación, se mostrará una ventana de despedida, indicando que la reserva se ha realizado satisfactoriamente.

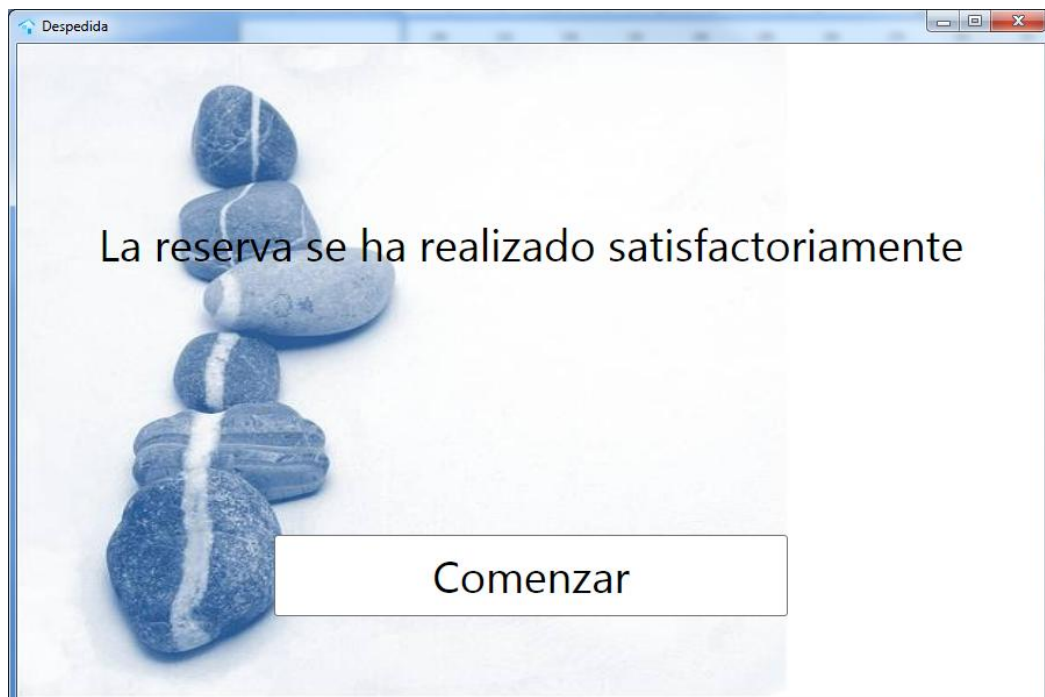


Ilustración 21: HuespedUOC. Pantalla final

Desde esta pantalla final, si el usuario pulsa el botón “Comenzar” se volverá a la pantalla principal. No tiene sentido cerrar y volver a abrir la aplicación, ya que estará instalada y accesible en lugares como la recepción o el restaurante del hotel. De esta forma quedará disponible para el siguiente huésped que desee reservar. Esta aplicación ha sido diseñada para ocupar toda la pantalla una vez ejecutada, sin mostrar siquiera los botones de minimizar, maximizar y cerrar, ya que el propósito de los terminales donde van instaladas es única y exclusivamente el de llevarlas instaladas. Sin embargo, para hacer más sencillo el proceso de prueba, se ha mantenido la aplicación en ventanas más pequeñas que se pueden cerrar igual que cualquier otra ventana.

Desviaciones del proyecto

En esta sección se detallan las partes que, principalmente por falta de tiempo no se han podido completar: representan cambios respecto de la planificación inicial. En esta sección se enumerarán dichas desviaciones, indicando las causas de la misma y unas líneas generales acerca de cómo hubiera sido su implementación en la solución final.

Las funcionalidades de las aplicaciones desarrolladas se han decidido en base a una especificación de requisitos. Por lo tanto, cambios en la planificación habrán causado requisitos no implementados en la solución final. Aquellos requisitos que no se han podido satisfacer aparecen en rojo en este documento en la sección de requisitos.

La principal desviación supone la no posibilidad de reservar entradas de teatro. Inicialmente los huéspedes y los empleados iban a ser capaces de poder reservar entradas para una función teatral determinada. Los huéspedes podrían decidir la obra, el día e incluso los asientos: fila y butaca.

La principal causa para la no implementación de esta funcionalidad ha sido la falta de tiempo. La funcionalidad estaba implementada en el servidor, e incluso había sido testada. Faltaba la implementación en las aplicaciones EmpleadoUOC y HuespedUOC. Dicha funcionalidad requería nuevos formularios para decidir qué asientos del teatro se reservan en la aplicación EmpleadoUOC, pero no aportaban nada diferente a otro tipo de reservas. Las entradas de teatro eran el último tipo de reservas por implementar, y se decidió prescindir de dicha funcionalidad.

La siguiente desviación comprende la generación de informes. Inicialmente, el empleado del hotel iba a ser capaz de generar tres tipos de informes:

- Habitaciones que han reservado una actividad determinada.
- Actividades que se han reservado a través de una estancia determinada.
- Actividades que se han reservado en un determinado intervalo de días.

Para generar estos informes, era necesario crear objetos dentro del servidor que contuvieran los datos a mostrar. Posteriormente, sólo era necesario hacer que esos objetos fueran orígenes de datos del informe; el informe se generaría automáticamente mostrando la información deseada. Lamentablemente, complicaciones a la hora de crear los instalables de

las aplicaciones hicieron que no hubiese tiempo para hacerlo. En su lugar se tuvo que optar por informes más sencillos, que son los que se pueden consultar en la aplicación actualmente.

Una última desviación es la no comprobación de la repetición de reservas de servicios de limpieza en un mismo día: un cliente no debería poder reservar dos servicios de limpieza en el mismo día. Para cumplir con este requisito se podría haber añadido una función de comprobación en el lado del servidor: alertaría en caso de efectuar una hipotética segunda reserva de servicio de limpieza. Una vez más, este requisito se desestimó por falta de tiempo y por considerarlo de menor importancia respecto de los demás.

Estas han sido las principales desviaciones del proyecto. Es importante tener en cuenta los estrechos márgenes de tiempo y la lista de requisitos y funcionalidades de la solución a implementar. Si es necesario desestimar un requisito o funcionalidad, primero se ha de tener en cuenta el impacto de ese requisito en la aplicación final. Ese requisito puede ser trascendente para la aplicación o ser simplemente una funcionalidad deseable. Asimismo, se deben tener en cuenta los objetivos del proyecto, siendo un objetivo principal el trabajar con un campo lo más amplio posible de tecnologías pertenecientes a la plataforma .NET. Si el requisito no es importante para la solución final, o si su desarrollo no supone el uso de tecnologías .NET novedosas, o que no se hayan usado aún en el proyecto, el requisito se puede desestimar. Si, por el contrario, el requisito es de alta importancia, o si su desarrollo supone un reto, es conveniente asumirlo.

Líneas abiertas del proyecto

En esta sección se incluyen sugerencias o mejoras para la solución, que pueden ser tenidas en cuenta para futuras versiones o para juzgar el trabajo realizado para la primera versión. La lista de mejoras se propone sin dar prioridad alguna a unos elementos sobre otros.

- Añadir la posibilidad de reservar entradas para actuaciones de teatro, tal y como recoge la planificación original.
- Ampliar el número de informes que genera la aplicación EmpleadoUOC, incluyendo aquellos recogidos en la planificación inicial.
- Actualmente ni la base de datos ni el servicio soportan concurrencia, es decir, varios accesos al servidor que se soliciten al mismo tiempo. En un hotel es muy probable que se dé la situación anterior, con lo que tratar la concurrencia desde WCF parece ser un tema prioritario.
- La aplicación HuespedUOC debe ejecutarse en pantalla completa, sin posibilidad de redimensionar la ventana.
- Para hacerla aún más sencilla de usar, la aplicación HuespedUOC puede ser adaptada para terminales de pantalla táctil.
- La ruta del servidor ha de poder ser cambiada desde la aplicación una vez instalada.
- Sería muy útil crear un cuadro de opciones para la aplicación EmpleadoUOC. Por el momento las siguientes opciones podrían ser configurables:
 - El número de días de margen que se usará para mostrar los servicios reservados próximamente.
 - La periodicidad con la que las reservas que se hayan efectuado se marcarán como finalizadas. Actualmente esta operación se realiza cada vez que la aplicación se ejecuta.
 - Como se ha indicado en una sugerencia anterior, se puede almacenar la ruta de comunicación con el servidor, por si este cambia de emplazamiento.
- En caso de ser necesario por motivos de facturación, la aplicación EmpleadoUOC debería poder ser capaz de elaborar facturas por los servicios o actividades reservados por un huésped durante su estancia.
- Actualmente las comunicaciones entre clientes y servidor se realizan a través del protocolo http, por el puerto 8080, y sin ninguna seguridad habilitada. Sería oportuno habilitar una comunicación segura entre las capas cliente y servidor.

Conclusiones

Trabajar en este proyecto me ha servido para aprender gran cantidad de aspectos de la plataforma .NET. A lo largo de todo el desarrollo he podido entrar en contacto con las siguientes tecnologías, de las que no sabía nada o casi nada:

- Acceso a datos ADO.NET. He aprendido a extraer datos, actualizarlos y a insertar datos nuevos. Asimismo, he creado una nueva base de datos desde cero en SQL Server Express 2005, incluyendo tablas, claves primaria y foránea y relaciones entre tablas.
- Consultas integradas en el lenguaje LINQ. He aprendido a interrogar a colecciones enumerables para extraer elementos que cumplan una serie de requisitos. Me hubiera gustado también poder extraer información de documentos XML e incluso de la Base de datos, pero por falta de tiempo no pude hacerlo.
- Interfaces de usuario. He usado ampliamente WPF, para desarrollar dos aplicaciones completas: he usado una amplia gama de controles: menús, botones, etiquetas, imágenes, listas desplegables, listas, controles *slider*. La aplicación que aloja el servicio WCF está basada en WinForms: es una aplicación muy sencilla, pero me ha valido como primera aproximación. Hubiera deseado aprender más sobre estilos (*styles*) y plantillas (*templates*), que es la aproximación de WPF a la herencia visual existente en WinForms, pero de nuevo el tiempo me lo impidió.
- Aplicaciones cliente-servidor. Sin tener ninguna experiencia previa, he conseguido comunicar dos aplicaciones usando Windows Communication Foundation (WCF), logrando transmitir datos a través de protocolo TCP. Creo que esta es la tecnología en la que menos he podido indagar, ya que los recursos disponibles en Internet normalmente consistían en ejemplos que no explican la tecnología en sí, sino solamente ese ejemplo. Personalmente creo que es una tecnología que se aprende a base de práctica y de prueba y error, siendo difícil dominarla en ese tiempo. Asimismo, no he usado IIS para alojar el servicio y me hubiera gustado usarlo para conocer todas sus posibilidades.
- Generación automática de informes. La generación de informes es una tecnología bastante amplia, sin contar con Crystal Reports, y sólo se ha usado una pequeña parte. Lo habitual hubiera sido usar la propia base de datos como fuente de datos, pero eso hubiera causado problemas en el despliegue, así que opté por la solución actual.

- Despliegue de aplicaciones. He creado instaladores para las tres aplicaciones, y dos de ellos, los relativos a las aplicaciones cliente, pueden ser considerados como complejos, ya que han requerido de una nueva ventana en el proceso de instalación para solicitar la dirección IP del servidor al usuario, además de una clase específica para poder actualizar la dirección IP en los ficheros de configuración de las aplicaciones. De igual manera, la desinstalación del programa también tuvo que ser intervenida para garantizar la correcta eliminación de los archivos de la aplicación.
- Lenguajes .NET: la capa servidor está desarrollada en C#.NET, y la capa cliente en Visual Basic .NET. Ha sido una gran oportunidad para comprobar cómo dos lenguajes de la plataforma .NET son perfectamente capaces de trabajar juntos. Ambos lenguajes son sencillos de aprender y potentes. El entorno de desarrollo Visual Studio es con diferencia el mejor entorno de desarrollo que he probado, por potencia, facilidad de uso y cantidad de posibilidades.

Desde mi perspectiva personal, el desarrollo de este proyecto ha sido una experiencia muy interesante: desarrollar una solución completa consistente en tres aplicaciones independientes ha sido un reto muy importante, al igual que hacerlo en un entorno en el que no tengo mucha experiencia y usando tecnologías completamente nuevas para mí. He procurado aprender todo lo posible en este semestre y el resultado ha sido muy enriquecedor. La plataforma todavía plantea multitud de posibilidades, así que seguiré aprendiendo de ella cuando el tiempo me lo permita.

Quiero también señalar que la realización de este proyecto ha sido un punto final casi perfecto a la carrera de Ingeniería Técnica de Informática de Gestión. Enfrentarte a un proyecto de desarrollo Software desde cero es un reto importante en el que puedes poner en práctica todos los conocimientos que has aprendido durante la carrera. He tenido que recuperar apuntes de Ingeniería del Software, Programación Orientada a Objetos y Bases de Datos.

También es necesario destacar el apartado de planificación del proyecto, correspondiente a las dos primeras PEC's: es algo sobre lo que no hemos aprendido nada en la carrera y sin embargo el saber planificar y elaborar un proyecto -de cualquier tipo- es algo primordial en la vida laboral actual.

Bibliografía y recursos utilizados

Sharp, John (2007). *Windows Communication Foundation Step by Step*. EEUU: Microsoft Press.

Petroutsos, Evangelos (2008). *La Biblia de Microsoft Visual Basic 2008*. Madrid: Anaya Multimedia.

Vrat Agarwal, Vidya; Huddleston, James (2008). *Beginning C# 2008 Databases: From Novice to Professional*. EEUU: Springer.

Mayo, Joe (2008). *C# Unleashed With the .NET Framework 3.5*. EEUU: Pearson Education.

Nathan, Adam (2007). *Windows Presentation Foundation Unleashed*. EEUU: Sams publishing

Christian Mosers WPF Tutorial.net. <http://www.wpftutorial.net/>

Página principal MSDN España: <http://msdn.microsoft.com/es-es/default.aspx>

Página Principal Windows Communication Foundation. <http://msdn.microsoft.com/en-us/netframework/aa663324.aspx>

Leveraging the ReportViewer control in WPF [entrada de blog].

<http://dotbay.blogspot.com/2009/03/leveraging-reportviewer-control-in-wpf.html>

Como hacer un setup personalizado en 15min con Visual Studio .NET [entrada de blog].

<http://geeks.ms/blogs/ilguerrero/archive/2010/04/18/como-hacer-un-setup-personalizado-en-15min-con-visual-studio-net.aspx>

