

ANTEPROYECTO

HOTEL RURAL RSF

Suárez Franco, Rafael

Proyecto Integrado Ciclo Superior DAM

Departamento de Informática – IES Torre del rey

Marzo de 2022

Descripción del proyecto a realizar y las tecnologías a emplear

1) Modelo de Datos:

- Descripción de necesidades (breve explicación o enunciado de los datos que queremos guardar):

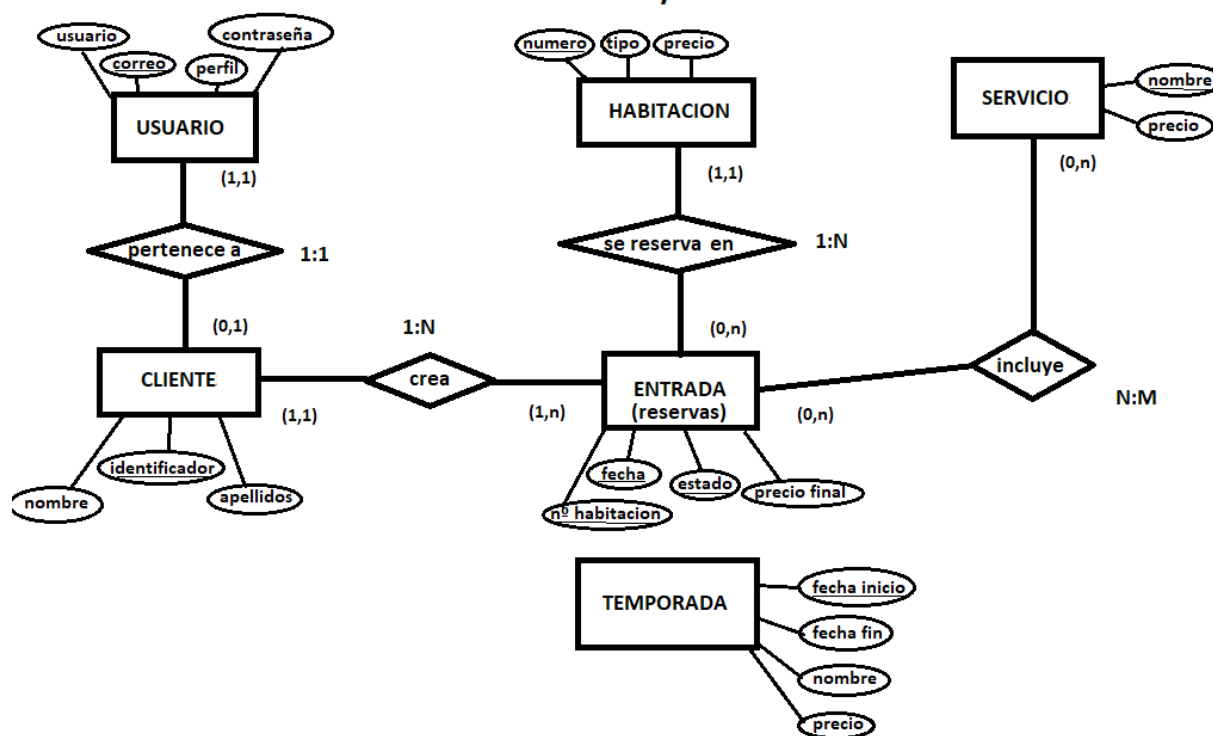
Necesitamos guardar las reservas y ocupaciones que se realizan en el hotel, de las cuales queremos guardar el nº de habitación, la fecha, el estado (si es reserva u ocupación), el cliente titular de la entrada y el precio que pagará o pagó (ya sea reserva u ocupación). Si una habitación pasa de reservada a ocupada, nos interesa guardar el registro de que estuvo previamente reservada (es decir, no modificamos o borramos la reserva, en ese caso guardamos los registros con los dos estados distintos). De las habitaciones, guardaremos su número, tipo y precio unitario por día. Una habitación puede estar reservada y/u ocupada un mismo día, como ya hemos dicho. Del cliente guardaremos el nombre, apellidos y un identificador (DNI por ejemplo) de momento nos ahorraremos el teléfono u otros datos personales. Los clientes pueden tener o no un usuario para esta aplicación (si no tiene, un administrador gestionará sus reservas y ocupaciones). Del usuario guardaremos un correo, nombre de usuario, contraseña y tipo de perfil (usuario o admin). Por otra parte, el hotel cuenta con servicios que se pueden contratar individualmente para cada día. De los servicios guardaremos el nombre y el precio. Debemos guardar qué servicios se contratan para qué días y qué habitaciones. Por último, hay que tener en cuenta las temporadas. Dependiendo de en qué fecha se reserva o se ocupa, el precio puede subir (temporada media o baja). Las temporadas las creará el administrador manualmente, dado que pueden variar de un año a otro. Nos interesa guardar su fecha de inicio y de fin, el precio adicional, y un nombre descriptivo (media, baja, alta, etc).

- Descripción del model E/R:

Como podemos ver en la imagen del modelo Entidad Relación, la entidad o tabla principal de este proyecto será la de entradas, que recogerá las reservas y ocupaciones del hotel. su clave principal será el número de habitación junto con la fecha y el estado, dado que una habitación en un mismo día puede estar reservada y más tarde ocupada. La relación

con las habitaciones es de uno a varios, dado que una habitación tendrá 0 o varias entradas, pero cada entrada es de una sola habitación. En cuanto a los clientes, la relación es la misma que con las habitaciones (no contamos con que hayan varias personas ocupando la habitación, solo tenemos en cuenta el titular de la reserva). Por lo que su clave principal se guardará en entradas para tener una referencia al cliente. Por otra parte, como ya hemos dicho, los clientes pueden tener un usuario, aunque no es obligatorio (el administrador puede dar de alta un cliente pero no tiene por qué crearle un usuario en el sistema) Por lo que la clave del cliente pasará también a la tabla de usuarios. En cuanto a los servicios, hay una relación de varios a varios dado que en una entrada se pueden contratar varios servicios y un mismo servicio puede estar contratado en varias entradas. En cuanto a las temporadas, no es necesario crear ninguna relación con alguna entidad del modelo, dado que simplemente usaremos esta tabla para consultar fechas en ellas. Su clave principal puede ser, por ejemplo, la fecha de inicio, dado que no debe repetirse (las temporadas no pueden solaparse en sus periodos).

MODELO E/R HOTEL

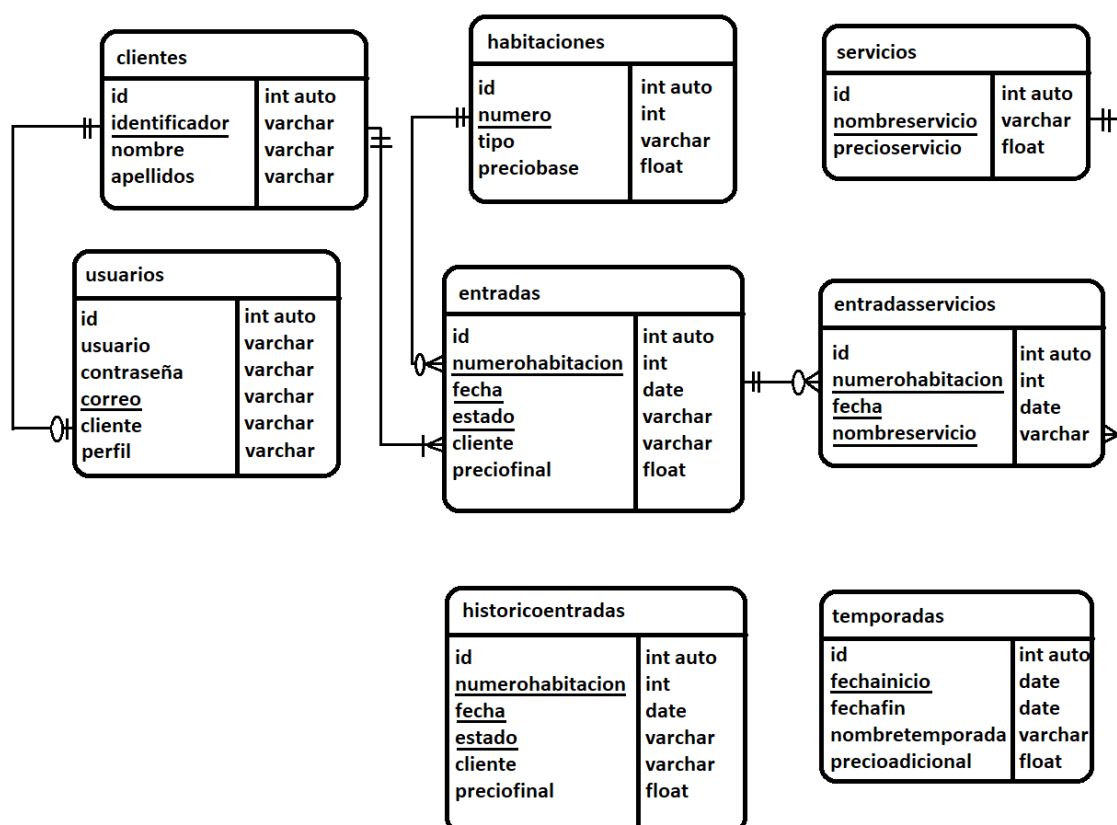


- Modelo patas de gallo:

Por último para tener una vista de las tablas que vamos a crear y utilizar, podemos hacer uso de un diagrama de patas de gallo. Cada entidad que hemos mencionado tendrá su propia tabla. La única relación que generará una tabla nueva será la que hay entre entradas y servicios, que nombraremos "entradaservicios", la cual se compone de las claves primarias de las otras dos. Sin embargo no contamos con el estado de la tabla entradas. Se entiende que los servicios contratados, por ejemplo en una reserva, serán los que se prestarán si esa reserva se convierte en ocupación. Si en el programa es necesario diferenciar entre servicios contratados (reserva) y servicios prestados (ocupación) siempre se puede hacer un join a la tabla entradas y comprobar el estado. Por otra parte la clave de habitaciones se propaga a entradas, para pasar a ser parte de

la clave principal de dicha tabla. Sin embargo la clave "identificador" de clientes se queda con el rol de clave foránea, dado que no es necesaria para diferenciar las distintas reservas/ocupaciones. Como hemos dicho, consideramos que una reserva/ocupación tiene un único cliente. Hay que destacar que, dado que no guardamos cuando una habitación está libre, se entiende que si dicha habitación para un día concreto no tiene registro en la tabla, se considera libre. Esto quiere decir que anular una reserva u ocupación implica eliminar registros de la tabla entradas, dado que no consideramos un tipo de estado "libre" o "liberado". Para no perder información sobre las anulaciones, tendremos una tabla (historicoentradas) idéntica a la tabla de entradas en la que guardaremos todos los registros borrados de dicha tabla. La misma lógica podría aplicar a la tabla entradasservicios, pero consideramos que guardar los servicios que se han cancelado no es tan relevante. En cualquier caso esta es la planificación inicial de la base de datos del programa, por lo que no estamos cerrados a hacer cambios. En cuanto a la tabla de temporadas, como hemos dicho, solo la usaremos para consultar fechas en sus registros, no hay relación directa con otras tablas. Su lógica es parecida a la de la tabla entradas: si una fecha no cae en un registro, quiere decir que cae en temporada baja, y que por tanto, no se sube el precio.

HOTELRSF MODELO DE PATAS DE GALLO



2) Funcionalidades del programa:

Dado que contaremos con dos tipos de perfiles, en primer lugar enumeraremos las posibilidades que tiene un usuario cliente, para más tarde presentar las opciones exclusivas de los administradores.

Un cliente podrá:

- Registrarse en el sistema y logearse. Será necesario iniciar sesión para acceder al restoe del sistema, ya sea cliente o administrador. Las contraseñas se guardarán encriptadas en md5.
- Visualizar la pantalla principal, donde se encuentran las habitaciones disponibles. En dicha pantalla se puede ver el estado de todas las habitaciones en un día concreto.
- Navegar entre fechas. Para visualizar el estado de las habitaciones en otros días. Para los clientes es probable que esté restringido a fechas presentes y futuras.
- Reservar una habitación en un día concreto, o en un periodo de tiempo. El cliente podrá acceder al formulario de administración diario de cada día, siempre y cuando dicho día no esté reservado u ocupado por otro cliente. Las opciones del cliente se limitarán a elegir qué días y qué servicios reservar, será el administrador (repcionista) quien confirme las ocupaciones y tenga la postestad de manipular el precio y otros aspectos.
- Acceder al historial, presupuesto y facturas correspondientes al cliente.
- Anular reservas que estén a su nombre.
- Visualizar la pantalla mensual, para ver los estados futuros de las habitaciones de manera más generalizada.

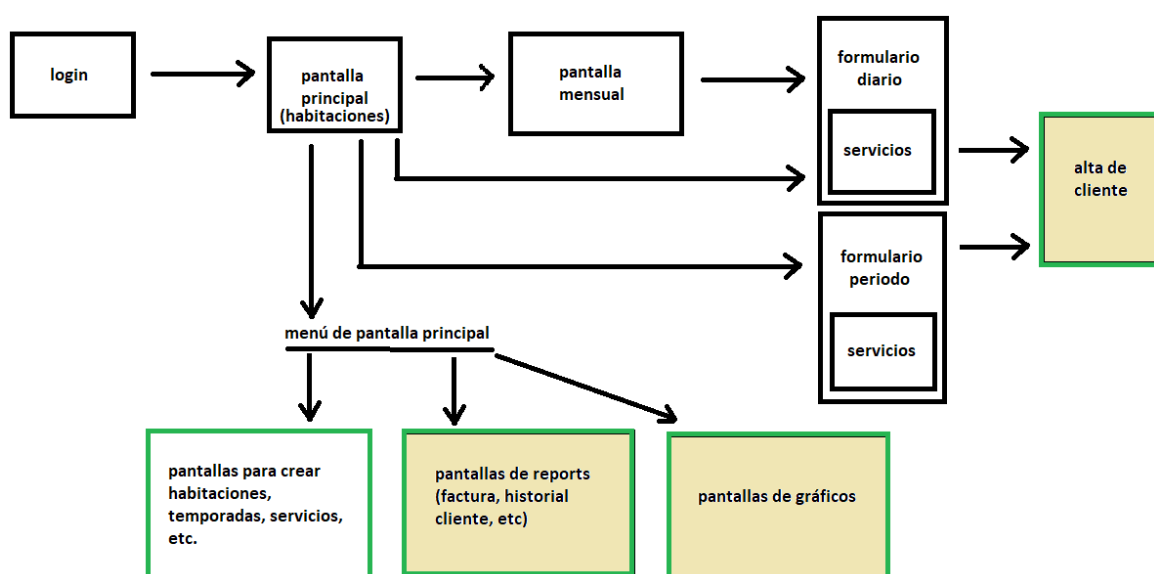
Un administrador podrá, además de todo lo anterior:

- Crear habitaciones, temporadas, servicios, clientes y usuarios. Podrá crear clientes sin usuarios, desde los formularios de reserva. Es decir, un cliente puede hacer una reserva y que la gestión la realice un administrador, en cuyo caso, puede introducir los datos del nuevo cliente a la vez que realiza la reserva. Posiblemente hayan opciones para editar también los elementos mencionados en este punto, por ejemplo, para cambiar los precios de un servicio o una habitación.
- Informes: Crear facturas o presupuestos, ya sea por periodos para una habitación determinada o por clientes concretos. Imprimir un itinerario de servicios, para informar al personal del hotel de los servicios que deben prestar a cada habitación en un día concreto. Inprimir un listado sobre cualquier tabla y cualquier campo de la base de datos. Por ejemplo, una lista de habitaciones o de clientes. Los listados también se podrán ordenar por cualquier campo.
- Imprimir gráficos de las reservas u ocupaciones en un periodo concreto, así como los ingresos esperados o conseguidos en dicho periodo. Del mismo modo, imprimir

un gráfico de la rentabilidad de los servicios del hotel, para una habitación concreta o para todas, en un periodo determinado.

- Administración total de las reservas y ocupaciones, capacidad de manipular precio y estado de las habitaciones (respetando unas reglas que enunciaremos más tarde), así como los servicios asociados dichas reservas.

A continuación, un esquema simple de las pantallas que formarán parte del programa. Las pantallas en amarillo son aquellas que en principio sólo estarán disponibles en la versión de escritorio. Las pantallas con los marcos verdes son aquellas a las que sólo tendrán acceso los administradores (o cuyo acceso por parte de clientes será restringido).



Nota: Cada pantalla será un formulario de Delphi (Unit1, Unit2, ...). Esto aplica en ambas versiones, y también a los reports (un formulario por cada report).

Como podemos ver, desde la pantalla principal se podrán reservar días individuales o periodos. Desde la pantalla mensual será más facil escoger los días concretos que se quieren reservar, o comprobar en qué periodos de tiempo están libres ciertas habitaciones para poder reservarlas con el formulario de periodo. En ambas pantallas se seguirá un código de colores para identificar rápidamente el estado de las habitaciones:

Rojo: ocupado.

Amarillo: reservado.

Verde claro: libre - temporada baja.

Verde: libre - temporada media.

Verde oscuro: libre - temporada alta.

Por otra parte, éstas son las reglas para cambiar los estados de las habitaciones:

Reservar/anular en Formulario de Periodo: solo se pueden hacer en días futuros, incluyendo el día de hoy. Anular también anula ocupaciones.

Formulario Diario:

- De libre a reservada : Día de hoy o futuro.
- De libre a ocupada : Día de hoy o en los 7 días pasados.
- De reservada a ocupada: Día de hoy o en el pasado
- De reservada a libre : Cualquier día.
- De ocupada a libre : Día de hoy (o futuro, pero no se puede marcar como ocupado en el futuro).
- De ocupada a reservada: Nunca.

Reglas adicionales para clientes (no administradores):

- Solo se pueden manipular habitaciones futuras y que no estén reservadas por otros clientes. Si por ejemplo habíamos contratado servicios para hoy, ya no podemos echarnos atrás.
- Solo se pueden reservar, como hemos dicho, la ocupación la confirmará un administrador.

Dicho esto, deberíamos implantar un mecanismo que impida que un usuario abuse del sistema, si por ejemplo, decidiera reservar todas las habitaciones para todo el año, impidiendo que otros usuarios puedan hacer sus reservas (y probablemente anulando las reservas que no pretende pagar). Por ejemplo podríamos hacer que un administrador revise las reservas que se realizan y que tenga que confirmarlas de algún modo antes de que se hagan efectivas, pudiendo contactar y fiscalizar al usuario que haga una cantidad anormal de reservas o alguna otra acción sospechosa.

3) Programación/desarrollo de la aplicación:

Como ya hemos dicho, este proyecto se construirá utilizando Delphi (lenguaje pascal). Nos centramos en la programación por eventos, dado que el usuario interactúa con los componentes que hay en la interfaz gráfica. Debido a que las posibles opciones dependen del tipo de perfil del usuario, debemos contar con sesiones que controlen dicho perfil para determinar qué acciones se pueden realizar. Para este propósito, podemos hacer uso de variables y funciones/procedimientos públicos, es decir, serán visibles y accesibles desde otros formularios. Esto nos permitirá pasar información entre distintas pantallas y ahorrar código, pudiendo contar con un formulario en el que situemos las

funciones, tablas de BBDD, etc a las que queramos acceder desde cualquier otra parte del programa. Otro aspecto a recalcar de este proyecto es que las características del hotel son ampliables, y por tanto, los componentes que representan las entidades de la base de datos también deben serlo. Esto se traduce en que algunas partes del programa no estarán predefinidas o construidas en su diseño inicial, sino que se generarán en tiempo de ejecución, dependiendo de los datos almacenados. Por ejemplo, si tenemos 10 habitaciones en principio pero vamos añadiendo otras nuevas, no nos servirá tener 10 botones en la pantalla, sino que debemos crear más.

En cuanto al lenguaje de base de datos, como ya hemos dicho, hemos escogido MySQL, dado que es una opción familiar y asequible para la envergadura de este proyecto. Las aplicaciones móviles en Delphi normalmente están pensadas para el uso de BBDD internas con SQLite, pero dado que deseamos tener una base de datos común a la que podamos acceder desde distintos dispositivos, optaremos por contar con un servidor MySQL. En la app móvil necesitaremos MyDAC, un componente de terceros que permite trabajar con MySQL (homólogo de FireDAC en la app de escritorio).