



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Hotel Serrucho

Proyecto

Administración y diseño de bases de datos.

Lucas Pérez Rosario
(alu0101399928)

Saúl Sosa Díaz
(alu0101404141)

Kilian González Rodríguez
(alu0101222325)

La Laguna, 20 de Noviembre de 2023



Índice:

Escenario para la base de datos elegida.	2
Objetivos del proyecto	2
Especificaciones de requisitos del proyecto	3
Especificación de la API	5



Escenario para la base de datos elegida.

El Serrucho es un prestigioso hotel que se encuentra en una famosa ciudad turística junto al mar. Cada habitación, desde las acogedoras estándar hasta las lujosas suites, está diseñada meticulosamente, ofreciendo grandes vistas y una decoración exquisita, adaptándose a cada tipo de viajero. Entre sus servicios, se pueden encontrar los esenciales para una cómoda estancia, entre ellos se pueden destacar: Un restaurante con buena comida, un spa relajante, gimnasio completo, piscina y opciones de masajes. Sin embargo, el hotel quiere ofrecer servicios exclusivos para los clientes más fieles como: transporte desde y hacia el aeropuerto, niñeras de confianza, salas de cine privadas y taxis a disposición.

Objetivos del proyecto

Los objetivos principales del proyecto son los siguientes:

- **Gestión Integral de Reservas y Servicios.**

Nuestro sistema innovador gestionará de manera integral las reservas de habitaciones, vinculándose con una diversidad de servicios personalizados. Cada huésped disfrutará de una experiencia a medida, desde opciones de ocio hasta servicios exclusivos, todo gestionado de forma eficiente y digital.

- **Relaciones con Proveedores bajo Control Riguroso.**

El módulo de gestión de proveedores es clave para mantener la excelencia en el servicio. Este sistema nos permitirá controlar la duración de los contratos y garantizar el cumplimiento legal, asegurando así un suministro constante y sin interrupciones para el hotel. Esta herramienta será esencial para anticipar renovaciones contractuales y mantener relaciones fluidas y beneficiosas con nuestros proveedores.

- **Estandarización de la Limpieza y Mantenimiento**

Un sistema de seguimiento detallado para la limpieza y el mantenimiento de las habitaciones asegurará que nuestros altos estándares de higiene y confort se mantengan en todo momento. Con este módulo, nuestro equipo de limpieza podrá operar con mayor eficiencia y efectividad, asegurando que cada habitación esté impecable para nuestros huéspedes.



- **Gestión Avanzada de Recursos Humanos**

El módulo de gestión de personal ofrecerá una visión completa de nuestros empleados, desde los cursos de formación realizados hasta sus jornadas laborales y responsabilidades departamentales. Esta herramienta no sólo optimizará la administración de recursos humanos, sino que también promoverá el desarrollo profesional y la satisfacción de nuestro equipo, elementos cruciales para el éxito de nuestro hotel.

Especificaciones de requisitos del proyecto

Para el desarrollo de la base de datos nos centramos en añadir los modelos de datos que fuimos conociendo a lo largo de la asignatura. Estos son los siguientes:

- Para el uso de la herencia hemos diferenciado entre dos tipos de clientes, los premium y los estándar. Los premium serán capaces de reservar ciertos servicios que los estándar no podrán.
- Como entidad débil decidimos añadir un registro de mantenimiento para las habitaciones. Lo que hace a esta entidad débil es que si se eliminan en algún momento las habitaciones, el registro de mantenimiento pierde totalmente su utilidad
- Para la relación triple hemos decidido crear la relación reserva, donde un cliente, al realizar una reserva, puede reservar varias habitaciones, además de múltiples servicios. Así tendremos una tabla para relacionar una reserva con el cliente que la ha realizado teniendo también otros aspectos como la fecha o los días de reserva y luego tendremos otras tablas con las que podemos relacionar una reserva concreta con la habitación y también otra que relaciona los servicios. Hemos tenido en cuenta que un cliente no puede reservar una habitación que ya esté reservada en un tiempo estipulado.
- De las tantas relaciones añadidas, tenemos de los distintos tipos, como puede ser una relación muchos a muchos entre empleados y departamentos, donde un empleado puede trabajar en varios departamentos y un departamento puede tener a su vez muchos empleados. También tenemos la relación del tipo 1 a muchos, como puede ser la relación entre departamentos y cursos, donde un departamento puede tener varios cursos pero un curso solo puede pertenecer a un departamento
- Para el caso de inclusividad hemos obligado a los empleados a realizar al menos un curso asociado al departamento en el que vayan a trabajar.



- Hemos usado una serie de funciones triggers para controlar el correcto funcionamiento y comportamiento de la base de datos por ejemplo en estos casos.
 - Validar el formato del correo electrónico, concretamente debe seguir el estándar RFC 5322.
 - Calcular el importe de la reserva en función de los servicios y habitaciones reservadas.
 - Verificar que no se solapen reservas sobre una misma habitación en una fecha ya ocupada.
 - Que un mismo empleado no puede tener más de una jornada que se solape.
 - Realizar el conteo de horas trabajadas de cada empleado en función al total de jornadas de ese empleado.
 - Obligar que si un empleado es gerente de un departamento trabaje en él, aunque no tenga cursos de dicho departamento.
 - Verificar que un empleado solo pueda trabajar en un departamento si posee los cursos indicados.
 - Los proveedores no pueden tener más de un contrato activo para un mismo servicio simultáneamente.
 - Verificar que los clientes estándar no pueden reservar servicios premium.



Especificación de la API

Como objetivo final del desarrollo del proyecto teníamos el de implementar una API que permitiera realizar ciertas operaciones con la base de datos. Para la realización de la misma hemos seguido el esquema que se muestra en la siguiente tabla, indicando, para cada ruta posible, sus verbos disponibles, los parámetros necesarios para realizar la operación y una descripción de lo que se podrá hacer en cada uno de los endpoints.

Método	URL	Acción
GET	/api/servicios	Consultar el listado de servicios disponibles.
GET	/api/habitaciones	Consultar el listado de habitaciones disponibles al momento de realizar la reserva.
GET	/api/reservas/?dni_cliente=...	Consultar el listado de reservas asociadas al cliente con DNI proporcionado.
POST	/api/reservas/?dni_cliente=...	Crear una reserva para el cliente con DNI dado y para rellenar los campos de la reserva estos se deben pasar en el body, en formato JSON, indicando los servicios, las habitaciones, la fecha y los días. Devolverá la reserva creada.
PUT	/api/reservas/?dni_cliente=...&codigo_reserva=...	Actualizará los servicios de la reserva indicada, siempre que pertenezca al DNI del usuario y no haya comenzado ya la reserva. Para eliminar o añadir servicios será necesario pasar un JSON en el body donde se indica <i>anadir</i> = true o false (añadir o quitar servicios respectivamente) y el campo <i>codigos_servicios</i> con aquellos servicios a modificar. Finalmente se devolverá la reserva con los servicios modificados y el nuevo importe.
DELETE	/api/reservas/DNI=...&codigo_reserva=...	Eliminar la reserva con <i>codigo_reserva</i> dado (siempre que el dni que se pasa corresponde con el asociado a la reserva y siempre que no haya comenzado aún la misma)



Para finalizar, debemos indicar que la implementación de la API ha estado enfocada en ser utilizada por un cliente del hotel, es por ello que se definen las entidades a las que puede o no acceder. En este caso, el cliente podrá ver los servicios y las habitaciones disponibles en el hotel y además de esto tendrá la posibilidad de ver las reservas asociadas a su DNI, crear una reserva y actualizar o eliminar alguna reserva asociada también a su DNI.

En cuanto a las entidades a las que no les hemos dado accesibilidad desde la API, como pueden ser los empleados, los contratos, los proveedores..., lo hemos hecho porque consideramos que en el escenario planteado, el de la API enfocada al cliente, estas entidades son internas del hotel y, por este motivo, queda restringido su acceso desde la API utilizada por el cliente.