



NOMBRE DEL PROYECTO	SPACEFY
EMPRESA	
GITHUB	
NOMBRES DE LOS PARTICIPANTES	Marcos Muñoz Noelia Sánchez Oscar Monforte Pilar Denia
LOCALIDAD	Madrid
ITINERARIO	1.- Acuerdo y conformidad de idea de Proyecto. 2.- Análisis y definición de las funcionalidades de la aplicación. 3.- Definición y confección de las diferentes páginas de la capa cliente. 4.- Determinación de las técnicas a aplicar. 5.- Ejecución de la Base de Datos. 6.- Desarrollo del Modelo Lógico. 7.- Integración de las diferentes partes del proyecto. 8.- Pruebas y verificación de la funcionalidad de la aplicación desarrollada. 9.- Ejecución de Memoria.

INDICE

1. OBJETIVO DEL PROYECTO.....	3
2. METODOLOGÍA DE TRABAJO.....	4
3. ESQUEMA TÉCNICO.....	5
3.1. MODELO.....	6
3.2. VISTA.....	8
3.3. CONTROLADOR	8
3.4. BASE DE DATOS	9
3.5. DIAGRAMA GENERAL DE LA APLICACIÓN	10
4. EL PROYECTO	11

1. OBJETIVO DEL PROYECTO

En el mundo actual cada vez con mejor y más rápido acceso a la tecnología, se tiene la posibilidad mediante herramientas web de automatizar y agilizar multitud de procesos y gestiones del día a día.

El objetivo del proyecto SPACEFY es desarrollar una aplicación web que cubra la necesidad de reservar espacios de uso común a un conjunto de usuarios, facilitando la gestión de reservas y cancelaciones de esos espacios comunes de una forma fiable e inmediata.

A partir de esta principal funcionalidad de la aplicación, se ha observado que se le puede dotar de otros procesos adicionales, como son:

- Control de agenda y del uso de los espacios, por parte de un perfil que sería tipo administrador. Descartado de la planificación inicial del proyecto.
- Notificación automática de la reserva (día y hora) al usuario, mediante mensaje al móvil o correo electrónico.
- Siguiendo la tendencia actual de “sharing” y de aprovechamiento de espacios privados, proporcionar a la aplicación el acceso, previo registro, a usuarios externos para la explotación de esos espacios en los periodos ociosos. Obteniendo ingresos extras para financiar el mantenimiento de las instalaciones.

Para el proyecto se ha tomado como muestra, a la hora de desarrollarlo, una comunidad de propietarios que disponen de una pista de pádel y una sala de uso común. Aunque su funcionalidad sirve para cualquier instalación que disponga de espacios compartidos: zonas deportivas dentro de clubes para sus asociados, conservatorios de música con salas a disposición de sus alumnos, centros culturales, coworking, etc.

2. METODOLOGÍA DE TRABAJO

Para la implementación/desarrollo de la aplicación se ha trabajado de diversas formas, conjugando desde reuniones presenciales hasta la implementación de metodologías ágiles en el desarrollo del proyecto.

El desarrollo del mismo se inició tras varias reuniones, en las cuales se consiguió llegar rápidamente a un consenso en lo referente al proyecto a ejecutar y en las que se hizo un reparto inicial del mismo en cuatro vistas para la aplicación; una vista para el login, una para el administrador, una para contener el perfil del usuario y finalmente otra donde se pudiera llevar a cabo la reserva requerida. Además, se vio la necesidad de estructurar y crear la base de datos para darle funcionalidad a la aplicación.

En el reparto inicial no hubo ninguna incidencia, cada miembro del equipo cogió la parte que prefirió para desarrollarla. Esto se hizo a pocas semanas de iniciar el curso, solo habiéndose impartido la parte de Front de la formación.

La parte del Back resultó más complicada de organizar/iniciar, ya que la propia estructura de la formación recibida no permitía establecer de forma concisa un punto claro de inicio para el desarrollo de la misma. Según se avanzaba en la formación se veían más facilidades para la implementación de esta parte, por lo que no se tenía claro cuál era el momento ideal para comenzar a trabajar en ella.

Finalmente, se inició el desarrollo de la parte de Back justo cuando se consideró por parte de todos los miembros del equipo, que era el momento ideal para ello. Por su complejidad, nos organizamos dos a dos en equipos, encargándose uno de ellos de todo lo relacionado con el usuario de la aplicación y el otro de la parte relacionada con la gestión de reservas.

A escasas tres semanas de finalizar la formación y el proyecto transversal, se detectó un punto crítico en cuanto al desarrollo del proyecto, cuando se tuvo que unir la parte de Front con la de Back, ya que las vistas elegidas para la parte de Front resultaban complejas de vincular con la parte de Back, con la formación que se había recibido. Este punto crítico se pudo solventar con la eliminación de una de las vistas planteadas inicialmente y gracias a uno de los miembros del equipo, con mayores conocimientos en programación se llevó a cabo la integración de ambas capas.

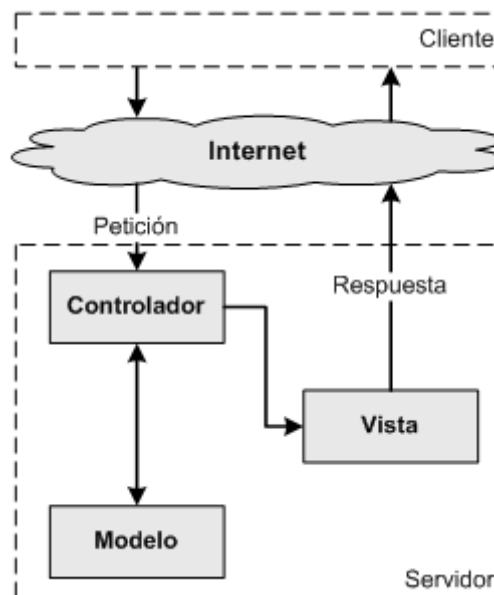
Para concluir, habría que reseñar que el grupo de trabajo ha mantenido más de siete reuniones de carácter presencial para la resolución de problemas y consultas técnicas, además de mantener el contacto a través de diferentes mecanismos para hacer más ágil la resolución y finalización de la aplicación.

3. ESQUEMA TÉCNICO

La aplicación Web SPACEFY esta desarrollada mediante Java Enterprise Edition (JEE) utilizando el IDE Eclipse y desarrollado bajo el patrón MVC (Modelo–Vista–Controlador).

Este patrón fue inventado para realizar una separación entre la interfaz gráfica y el código soporte del funcionamiento de la aplicación.

Una de las principales ventajas de trabajar con el patrón MVC es que la implementación se realiza de forma modular. Sus vistas muestran información actualizada siempre. Cualquier modificación que afecte al dominio como aumentar métodos, implica una modificación solo en el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos.



Para la gestión de dependencias se ha optado por utilizar la herramienta MAVEN ya que tiene un modelo de configuración de construcción basado en XML y utiliza un Project Object Model (POM).

También la aplicación realiza la gestión de la utilización de Cookies para así tener un manejo de la sesión del navegador y de esta manera ofrecer al usuario una mejor experiencia de navegación.

3.1. MODELO

El modelo es un conjunto de clases que representan la información y es la parte encargada del diseño de representar la Lógica de Negocio de la aplicación.

En esta aplicación lo conforman las siguientes clases:

modelo

DaoUsuarios

```
public boolean autenticar(String mail, String pass)
public int ObtenerIdUsuario(String mail, String pass)
public Usuario ObtenerUsuario(int numUser)
public boolean actualizarUsuario(Usuario user)
```

DaoReserva

```
public boolean cancelarReserva(int idReserva, int idUsuario, String idTypeUser)
public List<Reserva> mostrarSala(Date fecha, int idSala)
public List<Reserva> mostrarUsuario(Date fecha, int idUsuario)
public boolean altaReserva(Date fecha, int idSala, int idUsuario)
```

La necesidad de vincular objetos de Bases de Datos relacionales a objetos Java para su manejo en la aplicación nos hizo introducir una Capa de Persistencia que se desarrolló mediante Java Persistence API (JPA) con el motor de persistencia Hibernate. De esta manera los datos son expuestos a la aplicación en forma de objetos y la Lógica de Negocio es el encargado de manejarlos mediante las siguientes clases:

entidades

Reserva

```
public Reserva()
public Reserva(Date dateBooked, Date dateRequest, int idSpace, int idUser)
public int getIdBooking()
public void setIdBooking(int idBooking)
public Date getDateBooked()
public void setDateBooked(Date dateBooked)
public Date getDateRequest()
public void setDateRequest(Date dateRequest)
public int getIdSpace()
public void setIdSpace(int idSpace)
public int getIdUser()
public void setIdUser(int idUser)
```

Usuario

```
public Usuario()
public Usuario(String bankAccountUser, String dni, String email, String
firstName, int idCommunity, String idTypeUser, String lastName, String password,
int phone)
public int getIdUser()
public void setIdUser(int idUser)
public String getBankAccountUser()
public void setBankAccountUser(String bankAccountUser)
public String getDni()
public void setDni(String dni)
public String getEmail()
public void setEmail(String email)
public String getFirstName()
public void setFirstName(String firstName)
public int getIdCommunity()
public void setIdCommunity(int idCommunity)
public String getIdTypeUser()
public void setIdTypeUser(String idTypeUser)
public String getLastName()
public void setLastName(String lastName)
public String getPassword()
public void setPassword(String password)
public int getPhone()
public void setPhone(int phone)
```

3.2. VISTA

Las Vistas son las encargadas de la representación de los datos contenidos en el Modelo al usuario. La relación entre las vistas y el modelo son de muchas a uno, es decir cada vista se asocia a un modelo, pero pueden existir muchas vistas asociadas al mismo modelo.

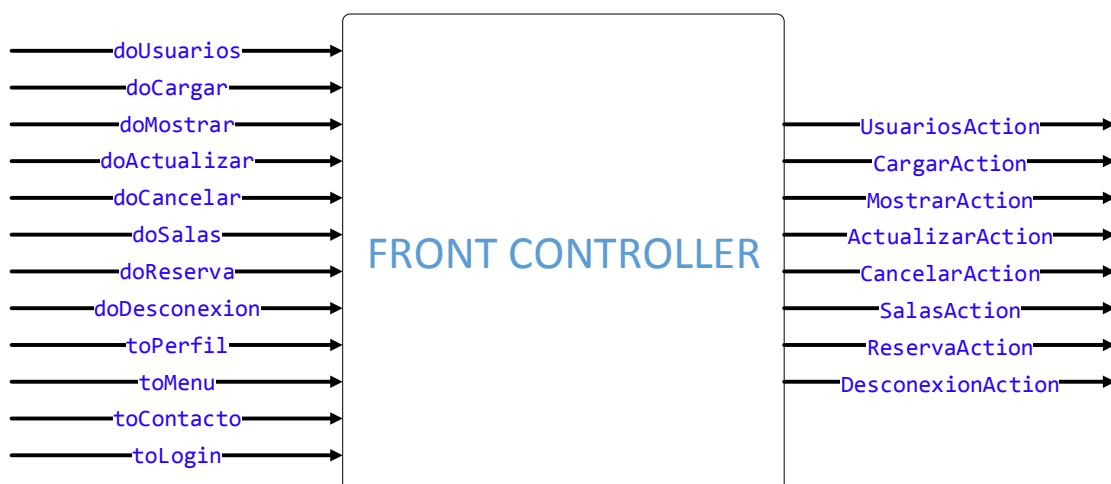
La vista solo necesita la información requerida del modelo para realizar un despliegue. Cada vez que se realiza una actuación que implica una modificación del modelo de dominio, la vista cambia a través de notificaciones generadas por el modelo de la aplicación. Sencillamente es la representación visual del modelo que redibuja las partes necesarias cuando se produce una modificación del mismo.

En esta aplicación se ha utilizado la técnica de programación AJAX (Asynchronous JavaScript And XML) en JSP y Servlets. AJAX lanza peticiones al servidor y genera como respuesta la información que necesitas en la página mediante JSON.

Todas las llamadas al motor AJAX son realizadas mediante JQuery y la representación de los datos a la interfaz de usuario mediante HTML 5 y CSS 3.

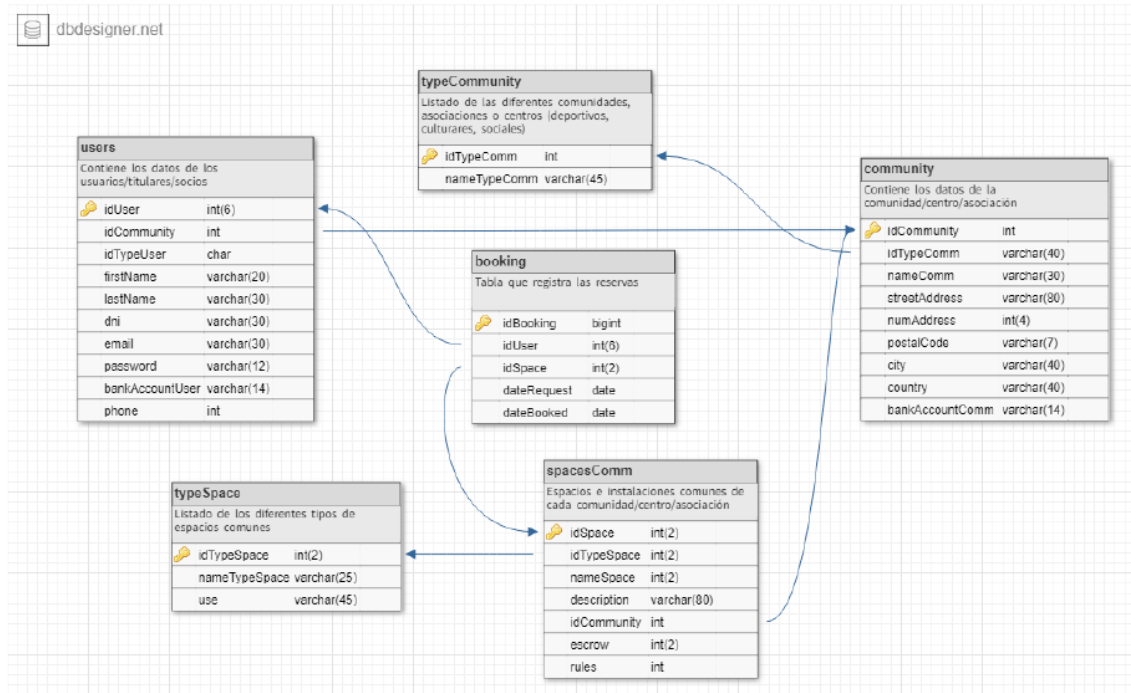
3.3. CONTROLADOR

El Controlador es el encargado de interpretar y dar sentido a las instrucciones que realiza el usuario, realizando actuaciones sobre el Modelo. Si se realiza algún cambio, comienza a actuar, tanto la modificación se produzca en una vista o en el modelo. Interactúa con el Modelo a través de una referencia al propio Modelo. Se comporta con respecto al siguiente esquema:

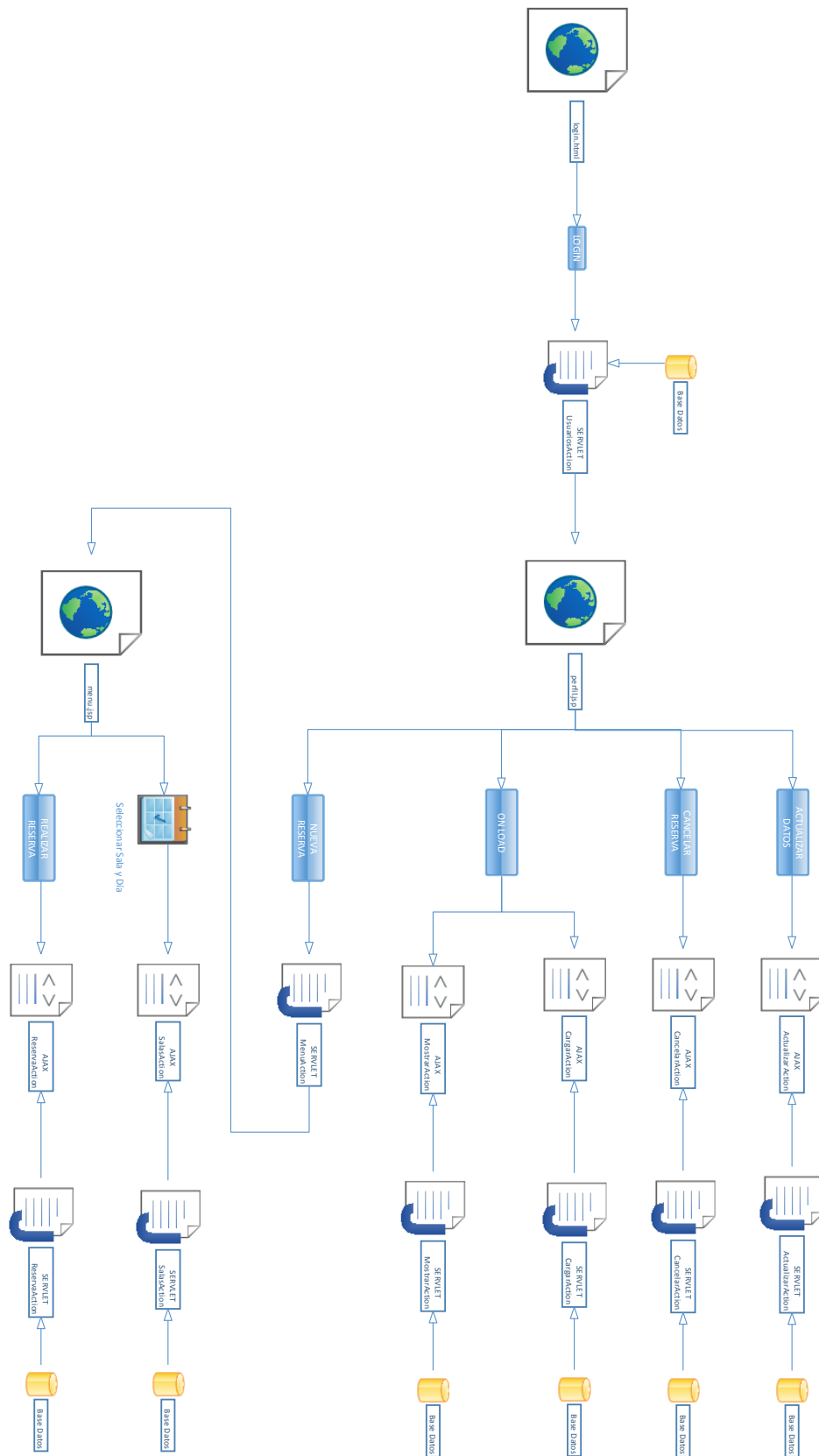


3.4. BASE DE DATOS

Dicha aplicación se conecta con una Base de Datos relacional en la que están registrados los usuarios, sus reservas y las distintas comunidades con sus espacios. Se ha realizado mediante MySQL de Oracle. Está formado por el siguiente esquema:



3.5. DIAGRAMA GENERAL DE LA APLICACIÓN



4. EL PROYECTO

La principal funcionalidad de la aplicación es el control de usuarios, comunidades de vecinos y espacios libres de uso comunitario presentes en cada comunidad, institución o empresa.

Concretamente, la funcionalidad del proyecto puede dividirse en tres, coincidiendo estrictamente con las tres vistas diseñadas como interfaz de usuario.

- Inicio de sesión:

En esta vista, se recoge el nombre de usuario y la contraseña con la que se da acceso a la plataforma a los usuarios registrados, con el fin de autorizarlos a interactuar con la aplicación y derivarlos a las siguientes opciones de navegación.

Estas son, la actualización de los datos de usuarios y el control de las reservas activas actualmente por cada usuario de ese usuario. La presentación de esta vista, esta soportada por un formulario de inicio de sesión, un prototipo del logo de la aplicación y una slider con fotografías de las posibles instalaciones accesibles a los usuarios a través de la aplicación. A continuación, se muestra una captura de pantalla sobre la vista “Inicio de sesión”.



Una vez el usuario se ha autenticado, entra en la vista perteneciente a su perfil.

- Perfil de usuario:

Aquí, el usuario puede modificar sus datos personales, como nombre, apellidos o teléfono, además de consultar las reservas pasadas y futuras realizadas por este.

La funcionalidad más destacada de esta vista, es la consulta de espacios reservados por el usuario, la cual puede usarla como simple consulta, o para cancelar reservas futuras en función de sus necesidades.

En el mismo calendario donde se pueden visualizar y cancelar las reservas, se encuentra el botón que le brinda acceso al usuario a la vista encargada de la realización de nuevas reservas. A continuación, se muestra una captura de pantalla sobre la vista “Perfil de usuario”.

localhost:8082/Spacefy/Controller?op=doUsuarios

RESERVA PERFIL CONTACTO CERRAR SESION

DATOS PERSONALES • DATOS BANCARIOS • AGENDA

DATOS PERSONALES *Rosa Flor*

NOMBRE [Rosa] [Edit]

APELLIDOS [Flor] [Edit]

DOCUMENTO DE IDENTIDAD [08359774F] [Edit]

E-MAIL [r.flor@gmail.com] [Edit]

NÚMERO DE TELÉFONO [609231553] [Edit]

localhost:8082/Spacefy/Controller?op=doUsuarios

RESERVA PERFIL CONTACTO CERRAR SESION

r.flor@gmail.com

NÚMERO DE TELÉFONO [609231553] [Edit]

PASSWORD [1234] [Edit]

GUARDAR CAMBIOS [Save]

DATOS BANCARIOS *ES7821006785892341676453*

CUENTA BANCARIA [ES7821006785892341676453] [Edit]

GUARDAR CAMBIOS [Save]

AGENDA [Calendar Icon]

08359774F

RESERVA PERFIL CONTACTO CERRAR SESION

E-MAIL
r.fior@gmail.com

NÚMERO DE TELÉFONO
609231553

PASSWORD
1234

GUARDAR CAMBIOS

CUENTA BANCARIA
ES11222233334455555555

Aceptar Cancelar

DATOS BANCARIOS
ES7821006785892341676453

RESERVA PERFIL CONTACTO CERRAR SESION

r.fior@gmail.com

NÚMERO DE TELÉFONO
609231553

PASSWORD
1234

GUARDAR CAMBIOS

GUARDAR CAMBIOS

Aceptar Cancelar

DATOS BANCARIOS

CUENTA BANCARIA
ES11222233334455555555

GUARDAR CAMBIOS

RESERVA PERFIL CONTACTO CERRAR SESION

(2019)

ENE FEB MAR ABR MAY JUN JUL AGO SEP OCT NOV DIC

L	M	X	J	V	S	D
1	2	3	4	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	31				

Añadir Reserva

SALA COMUN CANCELADO

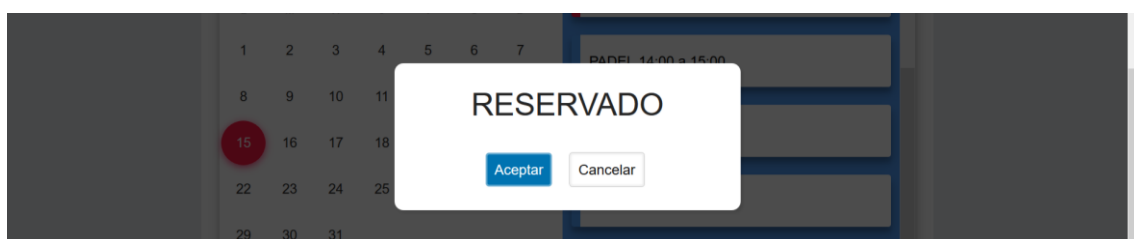
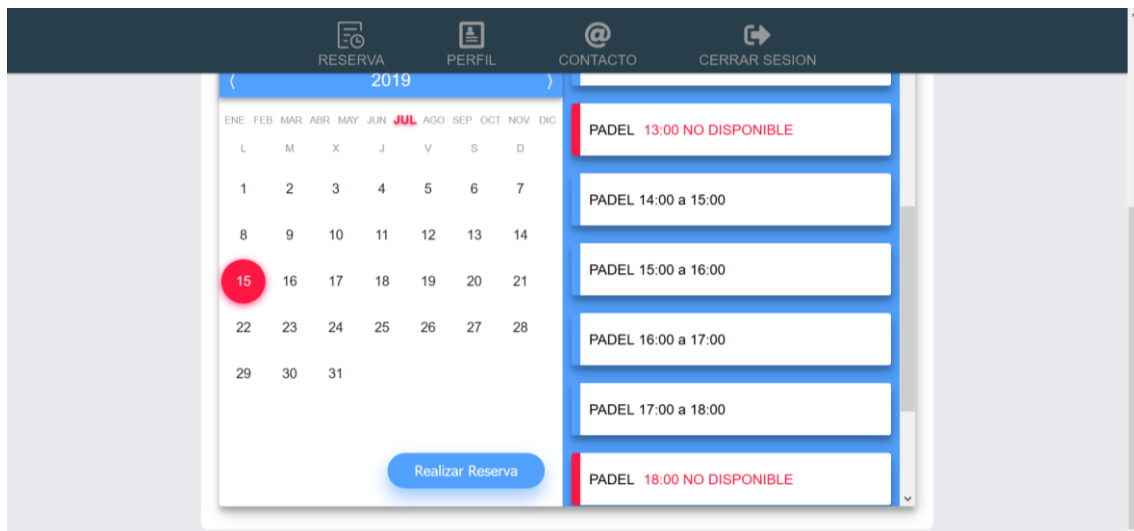
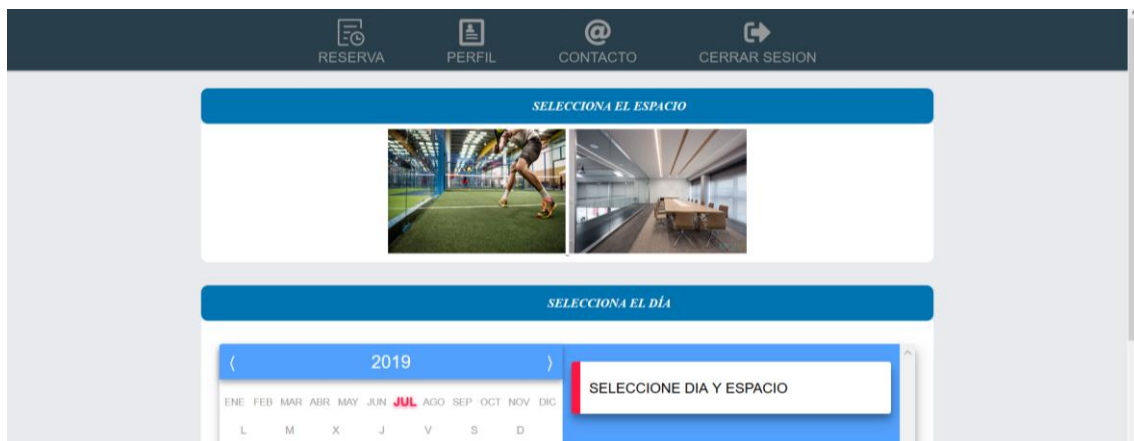
SALA COMUN 19:00

PADEL 10:00

- Realización de reservas:

Vista clave de la aplicación. Aquí, el usuario puede seleccionar el espacio en el que está interesado y, a partir de el, buscar los horarios que están libres para ser reservados. Las horas que estén previamente reservadas, se mostrarán en rojo y no podrán ser manipuladas por el usuario.

Una vez seleccionadas las opciones, se guardará en base de datos una reserva con el usuario, el tipo de sala y el horario seleccionado. A continuación, se muestra una captura de pantalla sobre la vista “Perfil de usuario”.





Por último, destacar las mejoras que recibirá la aplicación en el futuro:

- Activación de un usuario tipo administrador, que pueda cancelar reservas de otros usuarios, así como enviar comentarios y alertas a los usuarios pertenecientes la comunidad.
- Inclusión de usuarios externos a las comunidades, con los que la comunidad pueda aprovechar y monetizar sus espacios vacíos durante las horas que desee.
- Capacidad de la aplicación para incluir en su funcionalidad, nuevos espacios ajenos a las comunidades de vecinos, como por ejemplo conservatorios, bibliotecas públicas o salas de reuniones en empresas.