

# CASO PRÁCTICO 1

## ESTUDIOS DEL DESARROLLO DE UNA APLICACIÓN SOFTWARE

### Contexto

Trabajas en una empresa que se dedica al desarrollo de software, y un cliente os ha encargado un proyecto. La empresa es una gran compañía de gimnasios, cuenta con más de 100 centros deportivos en toda España, y una de sus principales actividades es el pádel. Debido a la gran demanda de este deporte, y para facilitar las reservas, os han encargado desarrollar una aplicación que permita reservar las pistas de pádel a través de Internet. La empresa está comprometida con el software libre, por lo que os piden que el entorno de desarrollo escogido cumpla con esa premisa. Además, especifican que los objetivos que persiguen con la aplicación son:

- Consultar el estado de cada pista de pádel (disponible o no).
- Posibilidad de realizar pagos con tarjeta.
- Controlar la necesidad de reservar la pista con o sin luz.
- Generar un listado de pistas reservadas por horas y clientes.

### Cuestiones a resolver

Debes realizar el diseño de la planificación del proyecto de desarrollo de este software pensando en el ciclo de vida del software. Para ello, debes:

- Planificar el análisis de los requerimientos.
- Planificar el diseño, indicando el lenguaje de programación que utilizarías, así como las herramientas para la obtención del código fuente.
- Planificar el resto de las etapas del desarrollo del software indicando en cada una de ellas el objetivo que se persigue.

Es importante que, en ningún caso, se realice ningún tipo de programación, ni tampoco escribir líneas de código.

---

## DESARROLLO DE APLICACIÓN DE RESERVAS DE PISTAS DE PÁDEL

---

### 1. Introducción y Contexto

La empresa de desarrollo de software en la que trabajamos ha recibido un encargo de una reconocida cadena de gimnasios que opera más de 100 centros deportivos en toda España. Esta empresa busca modernizar y optimizar su sistema de reservas de pistas de pádel, un deporte en creciente popularidad que atrae a un amplio público.

El objetivo principal de este proyecto es desarrollar una aplicación web que permita a los usuarios reservar pistas de pádel de manera sencilla y eficiente a través de Internet. Esto no solo facilitará la experiencia del usuario, sino que también mejorará la gestión interna del gimnasio al permitir un control más efectivo de la disponibilidad de las pistas y de las reservas realizadas.

Este documento detalla la planificación para el desarrollo de la aplicación, que seguirá un enfoque sistemático basado en el ciclo de vida del software. Este enfoque garantizará que cada fase del desarrollo se ejecute de manera organizada y eficiente. Un aspecto fundamental de este proyecto es el compromiso de la empresa con el software libre, lo que implica que se emplearán herramientas y lenguajes de código abierto. Esto permitirá al cliente tener una mayor flexibilidad y control sobre el software, facilitando futuras modificaciones y personalizaciones sin depender de proveedores externos.

La importancia de este proyecto radica no solo en la implementación de una herramienta tecnológica, sino también en la mejora de la experiencia del usuario y la optimización de los recursos del gimnasio. A medida que el pádel sigue creciendo en popularidad, contar con un sistema de reservas eficiente será clave para satisfacer la demanda y fomentar la lealtad de los clientes.

## 2. Análisis de Requerimientos

La fase de análisis de requerimientos es crucial para el éxito del proyecto, ya que aquí se identifican y documentan las necesidades y expectativas del cliente y de los usuarios finales. Este proceso involucra varias actividades que se llevarán a cabo en colaboración con el cliente, asegurando que la solución propuesta se alinee con sus objetivos.

### 2.1 Requerimientos Funcionales

1. **Consulta de Disponibilidad de Pistas:** Los usuarios podrán consultar en tiempo real el estado de cada pista de pádel, identificando si está disponible o ya ha sido reservada. Esto se logrará mediante una interfaz intuitiva que muestre un calendario de disponibilidad.  
*Ejemplo:* Un usuario interesado en jugar el próximo sábado puede acceder a la aplicación, seleccionar la fecha y visualizar qué pistas están libres en su horario preferido. Esto evitará frustraciones al momento de realizar una reserva.
2. **Pagos en Línea:** La aplicación deberá incorporar un sistema de pagos seguro que permita a los usuarios realizar transacciones con tarjetas de crédito y débito. La implementación de protocolos de seguridad como HTTPS y la utilización de servicios de pago confiables (como *Stripe* o *PayPal*) garantizará que las transacciones sean seguras.  
*Ejemplo:* Después de seleccionar una pista y el horario deseado, el usuario podrá ser redirigido a una pasarela de pago. Una vez que introduzca los detalles de su tarjeta y complete el pago, la aplicación confirmará la reserva automáticamente.
3. **Opciones de Reserva con Iluminación:** Los usuarios tendrán la opción de seleccionar si desean reservar la pista con, o sin iluminación, lo que será especialmente relevante para aquellos que juegan en horarios nocturnos. Esto permitirá al sistema gestionar mejor las reservas y las necesidades de los clientes.  
*Ejemplo:* Un usuario que juega al pádel regularmente puede preferir siempre reservar pistas con iluminación para asegurarse de que su partido no se vea afectado por la falta de luz natural.
4. **Listados de Reservas:** La aplicación generará listados detallados de las reservas realizadas, que podrán ser filtrados por hora y cliente. Esto ayudará a la administración del gimnasio a realizar un seguimiento de las reservas y optimizar la asignación de recursos.  
*Ejemplo:* Un administrador del gimnasio podría acceder a un reporte que muestre todas las reservas realizadas en un día específico, permitiendo evaluar la demanda y ajustar horarios o disponibilidades según sea necesario.

### 2.2 Requerimientos No Funcionales

1. **Seguridad:** La aplicación debe garantizar la seguridad de la información personal y financiera de los usuarios. Esto incluye el cifrado de datos sensibles y la implementación de medidas de protección contra ataques cibernéticos.  
*Ejemplo:* Las transacciones de pago estarán protegidas por cifrado SSL, y la información del usuario se almacenará de manera segura, cumpliendo con normativas como el RGPD.
2. **Usabilidad:** Se priorizará una interfaz sencilla e intuitiva, con un diseño que facilite la navegación y el acceso a las funcionalidades principales. Esto puede incluir menús claros, botones visibles y un flujo de usuario lógico.  
*Ejemplo:* Se realizarán pruebas de usabilidad con usuarios reales para asegurarse de que la aplicación sea fácil de entender y utilizar, reduciendo así la curva de aprendizaje.

3. **Compatibilidad con Software Libre:** Se emplearán lenguajes y herramientas de código abierto para garantizar que el software pueda ser modificado y distribuido sin restricciones. Esto permitirá al cliente y a la comunidad realizar actualizaciones y mejoras.  
*Ejemplo:* Utilizar herramientas como GitHub para el control de versiones facilitará la colaboración entre desarrolladores y permitirá que el código sea revisado y mejorado por la comunidad.

Este análisis de requerimientos se realizará a través de reuniones periódicas y sesiones de feedback con el cliente, donde se documentarán todas las especificaciones y se validará que las expectativas sean correctamente entendidas y reflejadas en el desarrollo del software.

### 3. Diseño del Proyecto

La fase de diseño del proyecto es fundamental para establecer cómo se implementarán los requerimientos identificados en la fase anterior. Durante esta etapa, se definirán las tecnologías a utilizar, la arquitectura del sistema y la interfaz de usuario.

#### 3.1 Lenguaje y Herramientas de Desarrollo

Para cumplir con los objetivos del proyecto y el compromiso con el software libre, se ha decidido utilizar las siguientes tecnologías:

- **Backend:** Se optará por **Python** como lenguaje de programación, gracias a su claridad y facilidad de uso. **Django**, un potente framework para desarrollo web en Python, será la herramienta elegida para la creación de la lógica del backend, ya que permite un desarrollo rápido y cuenta con un conjunto robusto de características para la seguridad y gestión de bases de datos.
- **Frontend:** Para la interfaz de usuario se utilizarán **HTML**, **CSS** y **JavaScript**. Se evaluarán bibliotecas de código abierto como **React** o **Vue.js**, que permiten construir interfaces de usuario modernas y reactivas, mejorando la experiencia del usuario al interactuar con la aplicación.

#### 3.2 Justificación

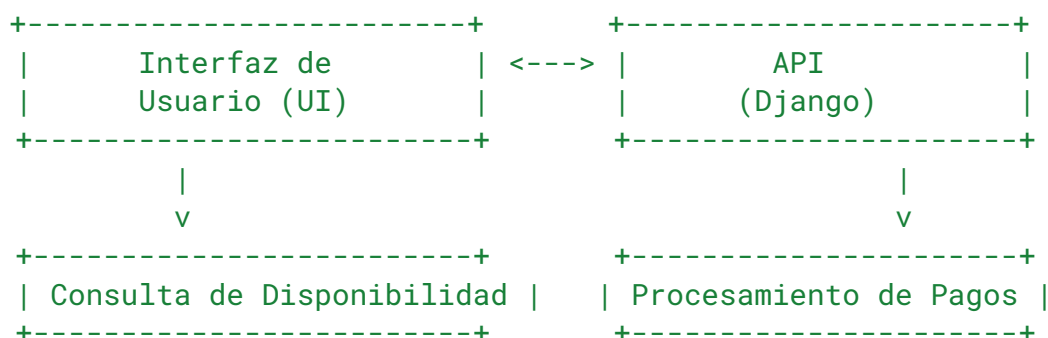
La elección de Python y Django se fundamenta en varias razones clave:

1. **Facilidad de Aprendizaje:** Python es conocido por su sintaxis sencilla y legible, lo que facilita la incorporación de nuevos desarrolladores al proyecto.
2. **Comunidad Activa:** Django cuenta con una comunidad amplia y activa que contribuye con recursos, documentación y soporte, lo que puede ser invaluable durante el desarrollo.
3. **Seguridad Integrada:** Django proporciona muchas características de seguridad listas para usar, como protección contra ataques de inyección SQL, Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF).
4. **Escalabilidad:** Django está diseñado para facilitar la escalabilidad, permitiendo que la aplicación crezca en funcionalidad y rendimiento a medida que aumenta la demanda.

En cuanto al frontend, React y Vue.js son populares por su capacidad para crear aplicaciones interactivas y dinámicas. Estas bibliotecas permiten construir componentes reutilizables, lo que mejora la eficiencia del desarrollo y la mantenibilidad del código.

### 3.3 Diagrama Conceptual

El siguiente diagrama conceptual ilustra la arquitectura de la aplicación, mostrando cómo interactúan los diferentes módulos:



Este diagrama representa la relación entre la interfaz de usuario y la API, que actuará como intermediaria para procesar las solicitudes del usuario y gestionar la lógica del negocio, así como el acceso a la base de datos.

## 4. Planificación de las Etapas del Desarrollo

La planificación de las etapas del desarrollo es un aspecto esencial para garantizar que el proyecto avance de manera ordenada y que se cumplan los plazos establecidos. A continuación, se presentan las fases del ciclo de vida del desarrollo de la aplicación:

### ● Análisis

- **Validación de Requisitos:** Se llevarán a cabo reuniones regulares con el cliente para revisar y validar los requisitos. Estas sesiones se documentarán y se utilizarán para ajustar el desarrollo en función de la retroalimentación recibida.
- **Documentación de Especificaciones:** Se elaborará un documento detallado que incluya descripciones de cada funcionalidad, diagramas de flujo y casos de uso. Esto servirá como referencia durante el desarrollo y ayudará a todos los miembros del equipo a estar alineados.

### ● Diseño

- **Arquitectura de la Aplicación:** Durante esta fase, se diseñará la arquitectura general del sistema. Esto incluirá la definición de los modelos de datos, las relaciones entre entidades y los puntos finales de la API.
- **Diseño de la Interfaz de Usuario:** Se crearán prototipos de la interfaz utilizando herramientas de diseño como Figma o Adobe XD. Estos prototipos serán revisados por el cliente y los usuarios finales para recibir retroalimentación antes de la implementación.

### ● Implementación (Sin programación en esta fase)

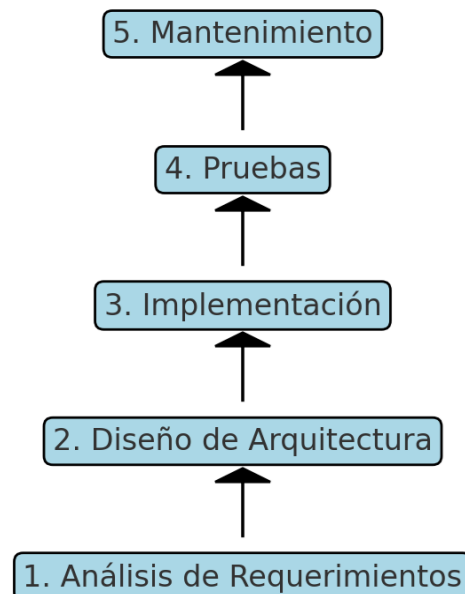
- **Estructura del Código:** Se establecerá una estructura clara para el código, con convenciones de nomenclatura y organización de archivos que faciliten la comprensión y mantenimiento del mismo.
- **Selección de Bibliotecas y Herramientas:** Se investigarán y seleccionarán las bibliotecas necesarias para el proyecto, garantizando que sean de código abierto y estén bien mantenidas. Esto incluye herramientas para gestión de estado, enrutamiento y optimización del rendimiento.

- **Pruebas**

- **Plan de Pruebas:** Se desarrollará un plan de pruebas que incluya pruebas unitarias para verificar que cada componente funciona correctamente, pruebas de integración para asegurar que los módulos interactúan como se espera y pruebas de aceptación del usuario para validar que la aplicación cumple con los requisitos establecidos.
- **Pruebas de Usabilidad:** Se organizarán sesiones de prueba con usuarios reales para evaluar la interfaz y la facilidad de uso de la aplicación. Esto permitirá identificar áreas de mejora antes de la implementación final.

- **Mantenimiento**

- **Planificación de Actualizaciones:** Se establecerán procedimientos para recopilar retroalimentación de los usuarios y realizar actualizaciones regulares. Esto asegurará que la aplicación se mantenga relevante y siga satisfaciendo las necesidades de los usuarios.
- **Estrategia de Seguridad:** Se desarrollará un plan de seguridad que incluya políticas de copia de seguridad y protocolos de respuesta ante incidentes. La seguridad de la aplicación será una prioridad en todas las fases del desarrollo.



## 5. Cumplimiento con Software Libre

El compromiso con el software libre es un aspecto fundamental de este proyecto. La filosofía del software libre promueve la colaboración, la transparencia y la libertad para modificar y redistribuir el software. Este enfoque no solo beneficia al cliente, sino que también permite a la comunidad de desarrolladores contribuir al proyecto y mejorar la aplicación con el tiempo.

Durante el desarrollo de esta aplicación, se emplearán lenguajes y herramientas de código abierto, como Python y Django, así como bibliotecas frontend de código abierto como React o Vue.js. Esto garantiza que el software sea accesible y modificable por la comunidad, permitiendo a otros desarrolladores aprender y aportar a su evolución.

Además, se documentarán todas las decisiones tomadas durante el desarrollo, asegurando que cualquier persona interesada en colaborar en el futuro pueda comprender la arquitectura y el diseño de la aplicación. Esta transparencia es esencial para fomentar un ecosistema de desarrollo saludable y colaborativo.

El uso de software libre también permite que el cliente tenga un mayor control sobre su software, pudiendo realizar modificaciones y adaptaciones según sus necesidades sin depender de proveedores externos. Esto es especialmente importante en un entorno empresarial donde la adaptabilidad es clave para mantener la competitividad.

## 6. Conclusión

La planificación presentada establece una estructura clara y detallada para el desarrollo de una aplicación web de reservas de pistas de pádel. Este enfoque sistemático asegurará que se aborden adecuadamente todos los aspectos del desarrollo, desde la recolección de requerimientos hasta la implementación y mantenimiento del software.

La implementación de esta aplicación no solo cumple con los requerimientos del cliente, sino que también respeta el compromiso con el software libre, lo que permitirá un crecimiento y evolución continua de la solución. Se anticipa que esta herramienta no solo mejorará la experiencia del usuario y la gestión interna del gimnasio, sino que también proporcionará una base sólida para futuras innovaciones y mejoras.

A medida que el pádel sigue ganando popularidad, contar con un sistema de reservas eficiente y moderno será clave para satisfacer la creciente demanda y fomentar la lealtad de los clientes. Estamos comprometidos a ofrecer una solución que no solo cumpla con los estándares actuales, sino que también esté preparada para adaptarse a los cambios y tendencias del futuro.

Con un enfoque colaborativo y transparente, se espera que la aplicación no solo satisfaga las necesidades actuales del cliente, sino que también se convierta en una herramienta valiosa para la comunidad de jugadores de pádel, permitiendo que disfruten de su deporte favorito de manera más accesible y organizada.

## 7. Referencias

- **Cesur.** Unidad Didáctica 1 de Entornos de Desarrollo. Material educativo sobre entornos de desarrollo.
- **Youtube**
- **Udemy:** curso introductorio a Python