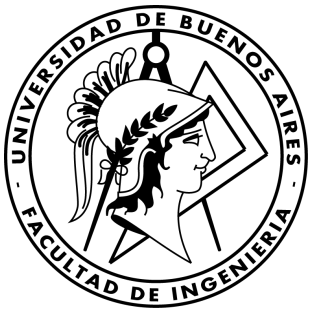

Construcción Colaborativa: Desarrollo de la Aplicación Web del Hotel Tenvagos

[TB022] INTRODUCCIÓN AL DESARROLLO DE SOFTWARE
Primer Cuatrimestre del 2024

2 Integrantes

ALUMNO	PADRÓN	MAIL
Bobadilla, Daniel	111879	dbobadilla@fi.uba.ar
Díaz, Bautista	107729	bdiaz@fi.uba.ar
Duarte, Lucas	112062	ldduarte@fi.uba.ar
Maguiña, Joaquin	106565	jmaguinap@fi.uba.ar
Montaño, Harold	106712	hmontano@fi.uba.ar
Proz, Gaston	105868	gproz@fi.uba.ar
Racioppi, Valentina	110998	vdiazr@fi.uba.ar
Riveros, Gustavo	112442	griveros@fi.uba.ar
Terrussi, Alexander	107212	lterrussi@fi.uba.ar

Índice

3 Palabras clave / Keywords

3.1 Español

Algunas de las palabras que permiten catalogar al trabajo realizado son las siguientes: HTML, hotel, reservas, habitaciones, usuarios, diseño, base de datos, frontend, backend, Github, Hotel lujoso, Reservas de hotel, Habitaciones lujosas, Iniciar sesión, Reserva en línea, Alojamiento de lujo, Hotel cinco estrellas, Suite presidencial, Servicios premium, Confort y elegancia, Portal de reservas, Base de datos de clientes, Página de perfil, Verificación de disponibilidad, Navegación intuitiva.

3.2 English

Some of the keywords that allow categorizing the work done are the following: HTML, hotel, reservations, rooms, users, design, database, frontend, backend, GitHub, Luxury hotel, Hotel reservations, Luxurious rooms, Log in, Online Booking, Luxury accommodation, Five-star hotel, Presidential suite, Premium services, Comfort and elegance, Booking portal, Customer database, Profile page, Availability check, Intuitive Navigation.

4 Introducción

Este Trabajo Práctico presenta el desarrollo de la aplicación web del Hotel Tenvagos, un proyecto enfocado en proporcionar a los usuarios una plataforma intuitiva y eficaz para explorar y reservar habitaciones de manera conveniente. El objetivo principal de esta aplicación es ofrecer una experiencia digital que refleje la excelencia y la calidez que caracterizan al Hotel Tenvagos.

A lo largo de este documento, se detallará el proceso de desarrollo que abarca desde la planificación inicial hasta la implementación final de la aplicación web. Se explorarán aspectos clave como la arquitectura del sistema, las funcionalidades implementadas, el diseño de la interfaz de usuario, y las tecnologías utilizadas para asegurar un producto final que cumpla con los estándares de calidad y satisfaga las necesidades de nuestros usuarios.

A lo largo de este Trabajo Práctico, se detallarán los desafíos encontrados, las soluciones implementadas y las lecciones aprendidas durante el proceso de desarrollo.

5 Solución propuesta

En esta sección, se describe la solución propuesta para el desarrollo de la aplicación web del Hotel Tenvagos. El objetivo principal es explicar las estructuras utilizadas, el flujo del programa, así como proporcionar una explicación detallada de las partes más importantes del código y las dificultades encontradas durante el desarrollo.

5.1 Estructura y Componentes Clave

La aplicación web del Hotel Tenvagos se desarrolló utilizando el framework Flask de Python, elegido por su simplicidad y flexibilidad para aplicaciones web. A continuación, se destacan algunos componentes clave:

Arquitectura MVC: Implementamos una arquitectura Modelo-Vista-Controlador (MVC) donde:

- **Modelo:** Define los modelos de datos utilizando SQLAlchemy para interactuar con la base de datos MySQL. Por ejemplo, el modelo Room representa las habitaciones disponibles.
- **Vista:** Utiliza plantillas HTML con Jinja2 para renderizar la interfaz de usuario y presentar información dinámica a los usuarios.
- **Controlador:** Implementado en las rutas de Flask, que manejan las solicitudes HTTP, procesan datos y actualizan la base de datos según sea necesario.

Flujo del Programa: El flujo típico de la aplicación incluye:

- **Inicio y Autenticación:** Los usuarios pueden registrarse e iniciar sesión para acceder a funcionalidades exclusivas como la reserva de habitaciones.1
- **Exploración de Habitaciones:** Los huéspedes pueden ver la lista de habitaciones disponibles, sus características y precios.2
- **Reserva de Habitaciones:** Se permite a los usuarios seleccionar una habitación, elegir fechas de estadía y confirmar la reserva mediante un formulario seguro.3
- **Gestión de Reservas:** Los huéspedes registrados pueden ver y gestionar sus reservas actuales y pasadas desde su perfil.4

Configuración de la Base de Datos: Utilizamos SQLAlchemy para la abstracción de la base de datos. Aquí configuramos la conexión a MySQL y definimos modelos como User, Room, y Reservation.

Rutas de Flask: Definimos rutas que manejan las solicitudes HTTP y actualizan la base de datos según sea necesario.

5.2 Explicación sobre el código

A continuación se mostrarán algunas partes del código que nos parecieron interesantes o pensamos que requieren alguna explicación:

En habitaciones.html, linea 22, se puede ver que se le puede pasar como variable una ruta a filename de la siguiente manera.

```
<div class="front" style="background-image: url('{{ url_for('static', filename='')
    }}{{room.url_imagen}}')">
```

En home.html, para el formulario de contacto que se encuentra a partir de la linea 273 se utilizó una pagina externa Formsubmit, en el atributo *action* del form se debe poner la url a la página, con el email que se quiere que reciba la información del form como ruta, o un código que ellos te proporcionan para que el email no quede expuesto en el html. Esta pagina recibe el método POST y procesa los atributos *name* de los diferentes elementos de un form, luego envía un mail al correo proporcionado en la ruta con la información del formulario.

6 Pruebas y/o validación

Durante el desarrollo de este proyecto, llevamos a cabo pruebas exhaustivas utilizando la impresión de variables y datos para verificar su estado y asegurar la correcta transferencia de información a lo largo del proceso. Sin embargo, el principal método empleado para solucionar problemas y garantizar el correcto funcionamiento del proyecto fue la realización de pruebas manuales. Navegamos por el sitio y utilizamos las diversas funcionalidades disponibles para identificar y corregir cualquier inconveniente.

7 Plan de actividades

El proceso de construcción del software y, por ende, el desarrollo del proyecto fue sumamente interesante, especialmente debido al número de integrantes en nuestro grupo y a cómo nos organizamos para trabajar.

Inicialmente, dividimos al equipo en dos grupos principales: uno enfocado principalmente en el desarrollo del Backend del proyecto y otro encargado del desarrollo del Frontend. Sin embargo, para mantener a todos los integrantes informados sobre el progreso del trabajo, llevamos a cabo llamadas diarias en Discord. Durante estas reuniones, discutíamos los avances realizados y abordábamos cualquier error encontrado. Este enfoque nos permitió que todos los miembros estuvieran al tanto de los avances del proyecto y contribuyeran activamente.

Desde los primeros pasos del proyecto, se incluyeron activamente propuestas sobre la estética de la página, paletas de colores, estructuración y modelos de navegación.5 El equipo de Frontend trabajó arduamente para obtener resultados visuales que facilitaran la navegación intuitiva del usuario.

En cuanto al plan de actividades, el equipo se enfocó en tres pasos básicos para crear el producto propuesto en el primer encuentro en nuestra mesa de trabajo:

Modelado por Figma: Se inició con el diseño y modelado de la página utilizando Figma.8 Concepción de las acciones de la página: En esta etapa, se definieron y estructuraron las acciones de la página. Alcances del cliente mediante peticiones: Se implementaron las funcionalidades de interacción del usuario mediante peticiones GET y POST. Una vez satisfecho el primer punto, se repartieron

las actividades necesarias para lograr la estructura básica de la página utilizando tecnologías como HTML, CSS, Bootstrap y JavaScript. Esta fase quedó reflejada en la rama ”front-branch”de nuestro GitHub.

En el segundo punto, se comenzó a añadir dinamismo a la página con Flask, aprovechando su versatilidad para implementar funcionalidades que permitieran al usuario realizar reservas y cancelarlas con la menor cantidad de clics posibles, utilizando estrategias de çall to action”.

Finalmente, en el tercer punto, se mejoró la lógica de la aplicación para asegurar que el usuario pudiera realizar y cancelar reservas correctamente, respetando las fechas de entrada y salida. Esto se logró mediante la implementación de scripts en JavaScript.

En lo que se refiere mas especifico a la parte de Backend del proyecto. Se organizo de la siguiente manera. Para comenzar, el grupo de back realizaba llamadas independientes al equipo de front, que se llevaban adelante luego de terminada la reunión daily con todo el equipo de trabajo. En estas reuniones, se dividían y organizaban tareas, se trataban problemas, y se veía el avance enfocado a lo que consta la parte Backend del proyecto. Además, una de las primeras actividades que se llevo a cabo fue la organización y el debate sobre como estarían compuestas las tablas de datos,y como se organizarían las mismas. Una de las partes mas importantes que llevo adelante este equipo fue la validación de todas las posibles interacciones y errores que pudiera llegar a tener el usuario al recorrer la pagina, con el objetivo de lograr una experiencia de usuario más agradable e interactiva.

Además, utilizamos la aplicación Trello para organizar de manera eficiente las tareas pendientes, establecer plazos estimados para su finalización y asegurarnos de que estuvieran alineadas con los entregables programados a lo largo del desarrollo. Estas sesiones fueron virtuales y proporcionaron a nuestro corrector una visión detallada de los avances. Recibimos orientación, comentarios y sugerencias sobre cómo avanzar, así como la oportunidad de discutir dudas y problemas que surgieron desde el último entregable.

Para garantizar la calidad, reducir los riesgos y facilitar el desarrollo con una menor probabilidad de errores, organizamos el proyecto en 4 ramas principales dentro de Github. En el lado del Frontend, utilizamos las ramas Front-branch y Front-develop. La primera contenía las vistas de la página web con sus respectivos HTML, mientras que la segunda incluía estas vistas integradas con Flask, permitiendo la navegación completa por el sitio. En cuanto al Backend, trabajamos en la rama Back-branch, donde se implementó la lógica para la conexión con la base de datos y el procesamiento de datos para mostrarlos en las vistas del Frontend. Finalmente, consolidamos las ramas Front-develop y Back-branch en la rama Main, donde se subió la versión final del proyecto integrado.

Esta estructura y metodología nos permitieron gestionar eficazmente el desarrollo del proyecto, asegurando un progreso continuo y una colaboración efectiva entre todos los miembros del equipo.

8 Hipótesis y Supuestos

Se utilizo una pagina Keto como referencia y guia de como era el objetivo final respecto al trabajo. El principal objetivo respecto al alcance de este trabajo era crear una pagina en la cual se pueda crear

usuarios, que los mismos puedan realizar reservas y gestionarlas.

9 Enlaces de interés

FIGMA
TENVAGOS
REPOSITORIO DE GITHUB
TRELLO

10 Resumen / Abstract

10.1 Español

El proyecto del Hotel Tenvagos se centró en desarrollar una aplicación web que proporcionara una plataforma intuitiva para la reserva de habitaciones. Desde el inicio, el equipo se ocupó de la estética de la página, la estructuración y los modelos de navegación. Utilizando HTML, CSS, Bootstrap, JavaScript y Flask, se creó una interfaz visualmente atractiva y fácil de usar.

La arquitectura Modelo-Vista-Controlador (MVC) fue fundamental en el desarrollo. SQLAlchemy se utilizó para interactuar con la base de datos MySQL, mientras que Jinja2 se empleó para renderizar las vistas dinámicas. El equipo de Frontend trabajó en el modelado inicial con Figma y la implementación de funcionalidades clave, mientras que el Backend gestionó la lógica de la aplicación y las peticiones al servidor.

Se llevaron a cabo pruebas exhaustivas, tanto manuales como automáticas, para garantizar el funcionamiento correcto de la aplicación. Las pruebas manuales implicaron la navegación por el sitio y el uso de distintas funcionalidades para identificar y resolver problemas. La organización del equipo se facilitó mediante reuniones diarias y el uso de herramientas como GitHub y Trello para coordinar las tareas y el progreso del proyecto.

El resultado fue una página web eficiente que permite a los usuarios registrarse, iniciar sesión, y gestionar sus reservas de forma sencilla. La optimización de la experiencia del usuario se centró en reducir el número de clics necesarios para completar acciones importantes. La metodología y la estructura de trabajo empleadas aseguraron una colaboración efectiva y un producto final de alta calidad que cumple con las expectativas y necesidades de los usuarios.

10.2 English

The Hotel Tenvagos project focused on developing an intuitive web application for room surfing and booking. From the outset, the team addressed page aesthetics, navigation structures, and user interface models using HTML, CSS, Bootstrap, JavaScript, and Flask to create a visually appealing and user-friendly interface.

The Model-View-Controller (MVC) architecture played a pivotal role in development. SQLAlchemy facilitated interaction with the MySQL database, while Jinja2 dynamically rendered views.

The Frontend team initially modeled using Figma and implemented key functionalities, while Backend managed application logic and server requests.

Comprehensive testing, both manual and automated, ensured the application's seamless operation. Manual tests involved site navigation and feature usage to identify and address issues. Daily meetings and tools like GitHub and Trello facilitated team organization, task coordination, and project progress tracking.

The outcome was an efficient web page enabling users to register, log in, and manage reservations effortlessly. User experience optimization focused on minimizing clicks required for critical actions. The methodology and work structure employed fostered effective collaboration, delivering a high-quality final product meeting user expectations and needs.

11 Anexos

Figura 1: Vista de login.

Figura 2: Detalles habitación.

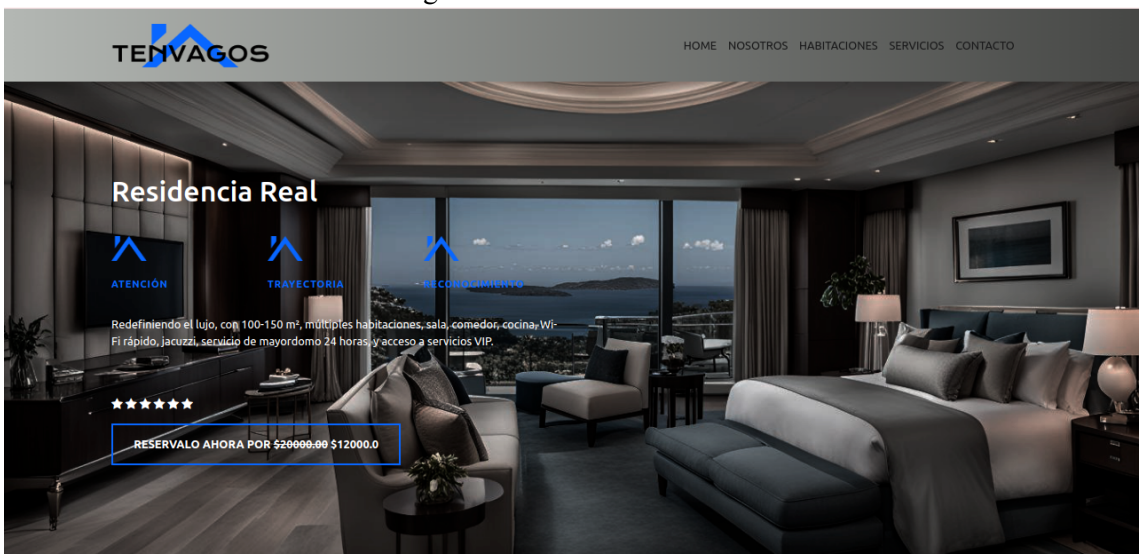

Figura 3: Vista reserva de habitaciones.

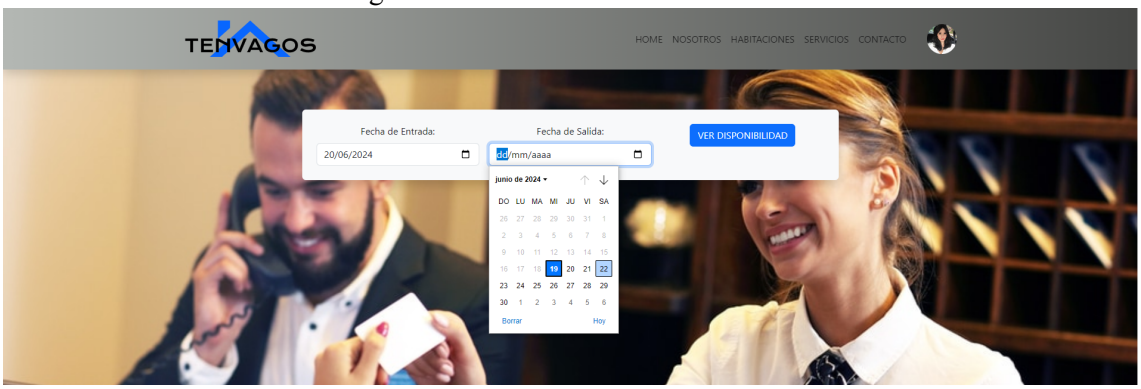

Figura 4: Vista gestión de reservas.

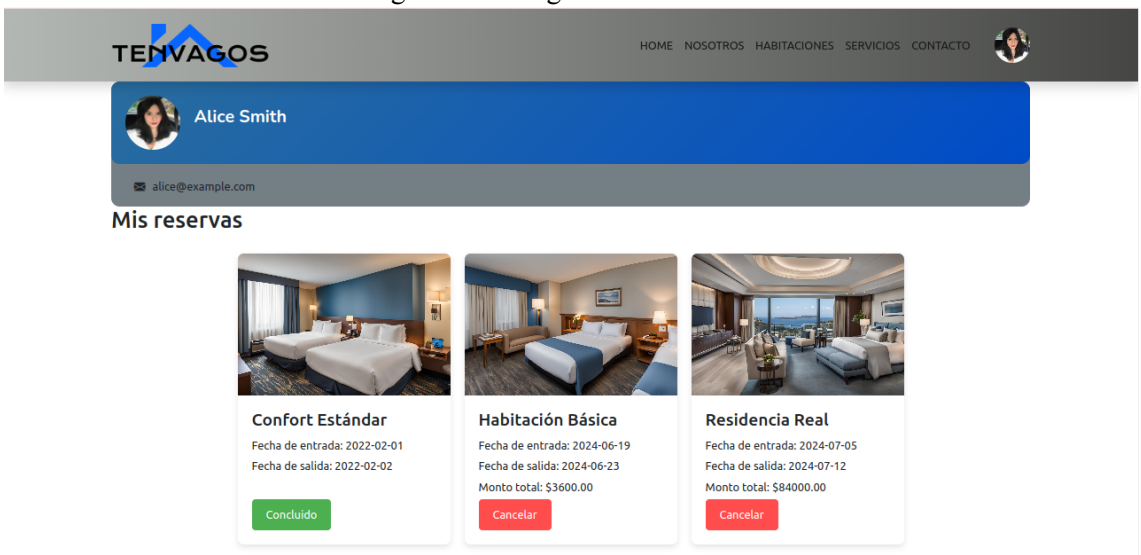

Figura 5:

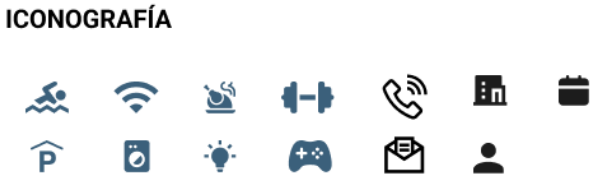

Figura 6:

Figura 7: Maquetado.

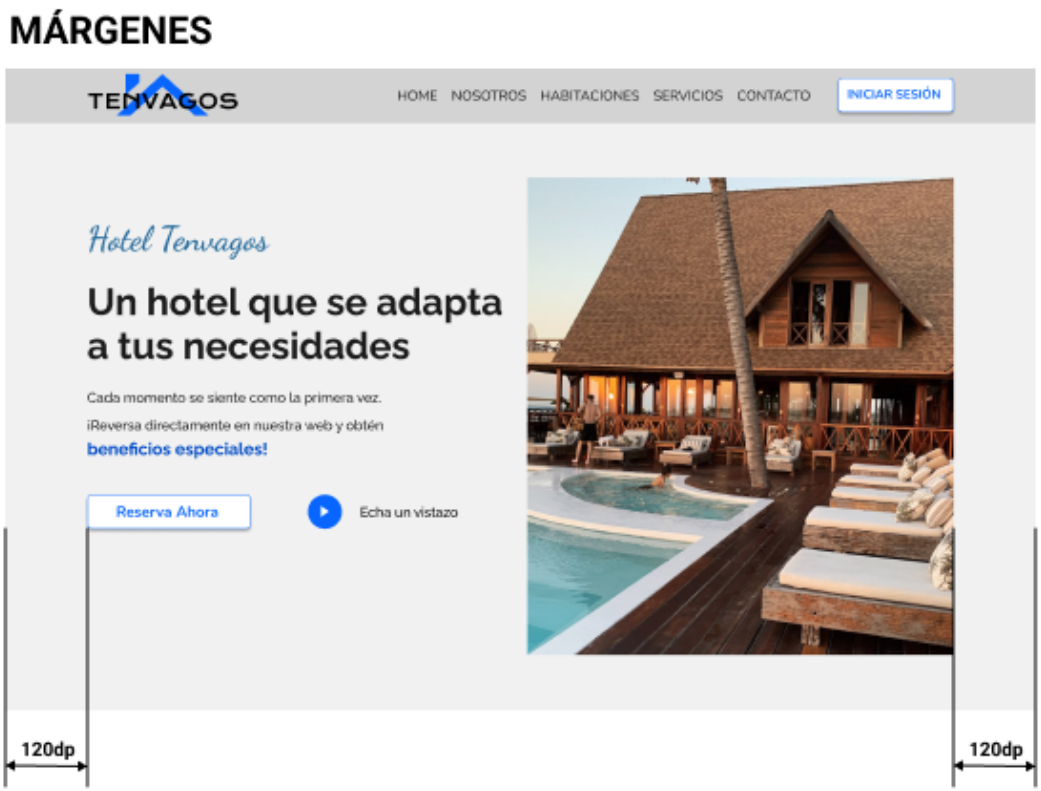

Figura 8: Mockup figma.

Figura 9: Figma dispositivo móvil.