

Documentación

Byteados

1er cuatrimestre 2024

Contents

1	Referencias	2
1.1	Enlaces externos:	2
2	Integrantes	2
3	Keywords	3
4	Introducción	3
5	Pruebas	4
6	Solución propuesta	4
6.1	Endpoints	5
7	Plan de actividades	7
7.1	Creación de frontend	8
7.1.1	calendario	9
7.2	Crear base de datos	9
7.3	Realización de requests	9
7.3.1	GET	10
7.3.2	POST	10

7.3.3	PATCH	10
7.3.4	DELETE	10
7.4	Creación de backend (API)	11
7.5	Cargado de imágenes en el drive	11
7.6	Backoffice	11
7.7	Hosteo de la base de datos y la página en pythonanywhere.	12
8	Hipotesis	12
9	Abstract	12

1 Referencias

1.1 Enlaces externos:

- [GIT cheatsheet](#)
- [GitHub SSH setup](#)
- [Flask docs](#)
- [Hosteo en pythonanywhere](#)
- [LaTeX](#)

2 Integrantes

Listado de integrantes, padrón y su mail de contacto.

Byteados

Jose Piñeiro Sanchez	106382	jpineiro@fi.uba.ar
Tomas Ordorica	111636	tordorica@fi.uba.ar
Ivan Colombo	111671	icolombo@fi.uba.ar
Maxi Fittipaldi	111676	mafittipaldi@fi.uba.ar
Joaquín Basile	111687	jbasile@fi.uba.ar
Lucas de la Peña	111733	ldelapena@fi.uba.ar
Lucas Mancini	111826	lumancini@fi.uba.ar
Melanie Belen Garcia Lapegna	111848	mgarcial@fi.uba.ar
Mirko Uriel Sáenz Valiente	111960	msaenz@fi.uba.ar
Mateo Parrado	111973	mparrado@fi.uba.ar

3 Keywords

Flask - API - Web - Servidor - Requests - Http - Posada - Habitaciones - Administrador - Crear - Leer - Actualizar - Eliminar

Flask - API - Web - Server - Requests - Http - Inn - Rooms - Admin - Create - Read - Update - Delete

4 Introducción

La "Posada Byteados" es una página de un lugar ficticio localizada Los Terrones 950 esquina, Batalla de Caseros, X5184 Capilla del Monte, Córdoba. Es un sitio web dedicado a una posada creado por el grupo "Byteados", en el que se muestran las habitaciones con sus características que se ofrecen para el cliente junto a un carrusel de imágenes donde se pueden ver dichas habitaciones.

Luego se puede observar una lista de espacios que se encuentran en común que ofrece la misma posada para poder disfrutar como una pileta, una sala de SPA y un arcade. También se puede observar que hay actividades adicionales que ofrece la misma posada como pueden ser un Tour en la cumbrecita o uno privado a Colonia

Caroya. Estas secciones en el funcionamiento de la pagina esto resulta ser estático, es decir, no se puede interactuar con este listado.

Por último, la página proporciona una sección donde se puede hacer una reserva si desea quedarse por un tiempo en alguna de las habitaciones que se encuentren disponibles junto con una breve descripción de cada una.

5 Pruebas

Se realizaron varias pruebas para analizar errores para aumentar la precisión en la búsqueda del objetivo de la página, en este caso hubieron varios cambios en la base de datos, donde se vio afectado el uso de las habitaciones y las reservas, se logró concretar cambios en la vista para que sea más amigable con el cliente al reservar o apreciar lo que ofrece el servicio de la página, y se implementaron cambios en la base de datos.

Se cuestionó bastante y hubo muchos cambios en la base de datos ya que pasó por cambios necesarios tanto para la tabla de habitaciones, como las reservas y la misma implementación de la tabla de imagenes. Las habitaciones tuvieron una serie de pruebas donde en un principio no tuvo las suficientes variables, o la implementación de imágenes para que sea posible era necesario hacer una tabla en la base de datos para que no solo puedan ser estáticas, sino buscar un dinamismo a la hora de agregar, eliminar u obtener una imagen.

Otra parte la cual se estudió mucho fue el mismo Front y las vistas, ya que llegaron a haber cambios para que la experiencia de usuario sea mas sencilla e interactiva. Tales cambios se ven vistos en la distribución de la pagina, el diseño estático, las funciones de la reserva y el mismo calendario.

6 Solución propuesta

Las reservas de habitaciones pueden ser un problema, dado que hay que tener en cuenta las fechas disponibles y la coordinación entre el cliente y la posada. Por eso,

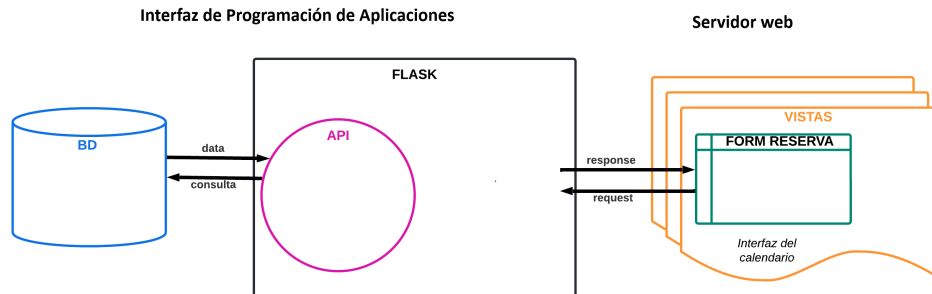


Figure 1: Estructura del proyecto

lo que nosotros proponemos es una aplicación web que resulte atractiva, llamativa y que provea a los clientes una forma sencilla y cómoda de reservar, sin pasar por consultas por mail o Whatsapp. En lugar de eso, el que esté interesado en poder estanciarse en la posada puede encontrar en el calendario las fechas en las que desea hospedarse y la página realiza la consulta. El cliente luego confirma la reserva y en el caso que se quiera revertir la reserva lo podrá hacer introduciendo sus datos y un código que se le es otorgado para lograr mas seguridad tanto para el cliente como para la misma reserva de la página.

La estructura del proyecto consta de una API (Application Program Interface) y de un servidor web, ambas son aplicaciones separadas que se complementan para el buen funcionamiento de la página. En la figura 1 se puede observar como la parte del servidor web contiene toda el frente de la página, capaz de generar interactividad y comunicación con el cliente,

6.1 Endpoints

Endpoint: /cabanias

metodo: GET

recibe: no se reciben parámetros

Endpoint: /cabanias/calendario/<string:cabania-id>

metodo: GET

recibe: no se reciben parámetros

Endpoint: /cabanias

metodo: POST

recibe: {"secreto":..., "id":..., "nombre":...,
"descripcion":..., "cap-max":..., "precio-noche":...}

Endpoint: /cabanias/<string:id>

metodo: PATCH

recibe: {"secreto":..., "nombre":..., "descripcion":..., "cap-max":..., "precio-noche":...}

metodo: DELETE

recibe: id (por URI)

Endpoint: /reservas

metodo: GET

recibe:

{"codigo-reserva": codigo-reserva, "nombre-cliente": nombre-cliente,
"cliente-id": cliente-id, "email": email,
"telefono": telefono, "fecha-ingreso": fecha-ingreso,
"fecha-egreso": fecha-egreso, "precio-total": precio-total}

Endpoint: /reserva

metodo: GET

recibe: {"cliente-id": cliente-id|pasaporte, "nombre-cliente": nombre-cliente}

Endpoint: /crear-reserva

metodo: POST

recibe: {"cabania-id": id, "fecha-ingreso": "aaaa-mm-dd",
"fecha-egreso": "aaaa-mm-dd", "nombre-cliente": "nombre-cliente",
"cliente-id": cliente-id, "telefono": telefono,
"email": "example@example.com"}

Endpoint: /reserva/<int:id>

metodo: DELETE

recibe: {"email": example@mail.com}

Byteados

Si "email" es proporcionado, la reserva del cliente con el código de reserva 'id', será eliminada.

Endpoint: /reserva/<int:id>

metodo: PATCH

recibe: {"secreto" : passw}

luego se puede modificar lo que se quiera de la reserva

Endpoint: /imagenes

metodo: GET

recibe: {"secreto" : passw, "cabania-id" : id *opcional*}

Endpoint: /crear-imagen

metodo: POST

recibe:

{"secreto" : passw, "cabania-id" : id, *opcional*

"link" : url-img, "descripcion" : portada}

Endpoint: /imagen

metodo: DELETE

recibe: {"secreto" : passw, "link" : url}

[API del proyecto](#)

7 Plan de actividades

La creación de software se enfocó en la división de tareas desde el principio y la creación de una estructura que sustentaría todo el proyecto. Primero y principalmente creamos una ilustración que explicaba gráficamente la forma en que sería el proyecto. Con el tiempo, esta ilustración se modificó en función de las tareas que surgían y las habilidades de cada miembro del equipo o su desempeño con respecto a las propuestas.

La metodología que se utilizó fue mediante una comunicación activa para aclarar la planificación de los objetivos y analizar en cada encuentro la realización de dichas

actividades, donde se acordó que uno de los objetivos principales para llevar a cabo era el diseño que iba a presentar la pagina, para luego poder desarrollar la parte funcional y programar un entorno de desarrollo utilizando tecnologías para el back-end. Después de haber cumplido con la importante tarea de haber trabajado en el Backend y en el Frontend quedaba poner a prueba y asegurar que tuviera un funcionamiento óptimo para la experiencia del usuario, ya que la pagina debía cumplir con las funciones recientemente hechas y, por otro lado, la interfaz del usuario debía ser intuitiva y fácil de usar para así poder corregir y ajustar los errores que se presentaban.

Las herramientas que tuvimos a nuestro favor y para llevar a cabo el plan de actividades fue el gestor de proyectos [Trello](#), donde pudimos listar las tareas entre los integrantes y así tener un seguimiento de cada una. El desarrollo de la pagina lo logramos hacer gracias a la implementación de Git y [GitHub](#), y otros programas como Visual Studio Code, MySQLite, Docker, entre otros.

Los integrantes del proyecto contaron con una división de roles y responsabilidades que tuvieron que cumplir como Desarrolladores de Backend y Frontend, pero mas allá del rol principalmente impuesto se cada uno logró contribuir en cualquiera de las áreas, para ello realizábamos encuentros para charlar las cuestiones que se nos dificultaba, trabajar como equipo y documentar cualquier cambio que hiciéramos por medio de un grupo de Discord.

7.1 Creación de frontend

El desarrollo del Frontend fue una de las primeras tareas a realizar, ya que antes de comenzar la codificación se presentó un diseño y planificación de la estructura con la implementación de **HTML**, para la base de la página web, **CSS**, para definir el aspecto visual y aplicar estilos, y **JavaScript**, para agregar una interactividad y dinamismo.

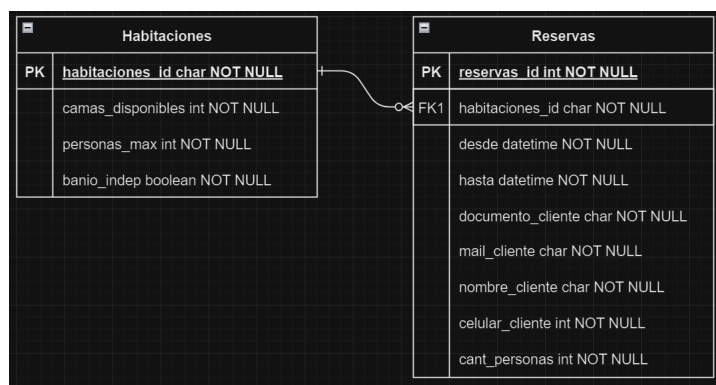


Figure 2: Idea principal de la Base de Datos

7.1.1 calendario

La implementación de un calendario es importante a la hora de realizar las reservas, por que es la conexión entre el cliente y la funcionalidad, por lo que se tuvo que implementar desarrollo en el Front para la vista, como en el Back para la funcionalidad.

7.2 Crear base de datos

La base de datos es una parte crucial del trabajo ya que almacena toda la información de la pagina, ya sean las reservas, habitaciones e imagenes. Por lo que se implemento distintos diseños para realizar dicha base de datos como se pueden observar en las figuras 2 y 3

7.3 Realizacion de requests

El realizar los requests era necesario para todas las tablas de la base de datos del proyecto, ya sea para obtener los datos de las cabañas, como realizar la reserva o agregar alguna imagen de la cabaña, es por eso que se llevaron a cabo diferentes tareas donde se vieron implementadas cada una de estas funciones del CRUD (Create, Read, Update, Delete) para las diferentes tablas

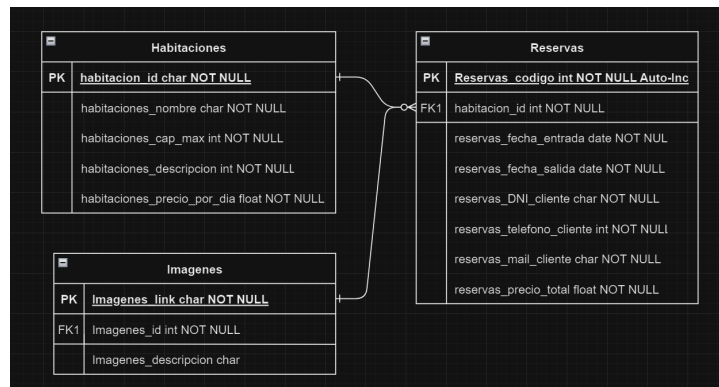


Figure 3: Idea final de la Base de Datos

7.3.1 GET

Los requests del tipo GET se utilizan para obtener datos del servidor, es por eso que se utilizó para saber que cabañas hay con disponibilidad, las reservas que hizo el mismo cliente o mismo para ver las imagenes implementadas.

7.3.2 POST

Los requests del tipo POST se utilizan para enviar datos al servidor, su implementación fue necesaria y esencial para realizar las mismas reservas y así poder guardarlas en la base de datos.

7.3.3 PATCH

El PATCH se utiliza para poder actualizar dichos datos del servidor, y fue necesario en el caso que se quiera modificar algún dato o el mismo día que se haya realizado la reserva.

7.3.4 DELETE

Los requests del tipo DELETE son utilizados para eliminar algún dato del servidor, para ello es de importancia su uso en el caso que se quiera cancelar una reserva a nombre de un cliente.

7.4 Creación de backend (API)

La Interfaz de Programación de Aplicaciones es esencial para el Backend ya que es el gestor de lógica del servidor, sus operación junto a la base de datos y su comunicación con el Front. Su creación era necesaria para la conexión con la Base de Datos, las rutas, el propio servidor y los mismos Endpoints de la API en el cual estaban presentes todas las operaciones del CRUD.

7.5 Cargado de imágenes en el drive

Las imágenes son necesarias para poder generar una mejor experiencia con el usuario, ya que se muestran algunas de las cabañas en cuestión o mismo las instalaciones, actividades o servicio. Es por eso que se implementó el uso de [Google Drive](#) para cargar las imágenes, donde están todas subidas en una carpeta de llamada "Imágenes". Su implementación es necesaria por la buena gestión de archivos.

7.6 Backoffice

El Backoffice es una tarea que se precisaba para que el administrador sea capaz de modificar la pagina en el caso que sugiera algún cambio, ya sea en las imágenes, como en las reservas o las mismas cabañas. El administrador debe ser capaz de poder gestionar la información del mismo cliente en el caso que se precise para las reservas realizadas, o también debe tener la posibilidad de administrar la información de las mismas cabañas de la pagina, ya sea para implementar o quitar información. El administrador debe tener la posibilidad de modificar y gestionar la pagina de la posada con operaciones del CRUD de una forma sencilla.

7.7 Hosteo de la base de datos y la página en pythonanywhere.

Para exhibir la pagina de la posada se utilizó Pythonanywhere, el cual nos permite tener una conexion entre los datos almacenados y la misma pagina.

8 Hipotesis

Durante el desarrollo del proyecto se pudo meditar distintas hipótesis en relación al impacto de la pagina.

La mejora de la experiencia del usuario es necesaria en todo momento, ya sea para lograr que el cliente logre cumplir su objetivo en el menor tiempo posible y de la mejor manera con las herramientas adecuadas, es por eso que implementamos distintas tareas a realizar en las reservas para que sean mas seguras y sencillas, los botones que hacen fácil la movilización dentro de la pagina, la barra de navegación la cual te dirige a realizar la tarea que busques, entre otras cuestiones.

El buen uso del Backoffice ayudará con la gestión operativa de la página y brindara sencillez al administrador para manejar correctamente la página web ante cualquier imprevisto que ocurra, ya sea desde un problema con la reserva, el mal uso de una imagen o una modificación en las cabañas.

La seguridad y administración a la hora de reservar un tiempo en la posada se debe implementar en todo momento para reducir los incidentes con el cliente, es por eso que se implemento un sistema capaz de recibir un identificador de la reserva que solo lo tenga el cliente en el caso que quiera cancelar su reserva.

9 Abstract

Desde un principio decidimos elegir este desafío por que nos pareció adecuado para utilizar lo que aprendimos sobre el tema y pensamos que podía ser buena practica, ya que también para crear la página era necesario la investigación en profundidad

Byteados

de los temas enseñados.

El proyecto de la página web para "Posada Byteados" busca no solo establecer una presencia digital atractiva, sino también mejorar la experiencia del cliente mediante un sistema de reservas intuitivo y brindar toda la información necesaria. Se organizó con cuidado y atención a los detalles para garantizar todo el funcionamiento de la página sea estable.

Tanto en el Frontend como en el Backend se resolvieron y ajustaron numerosos problemas durante el trabajo, que incluyeron problemas visuales o problemas que el cliente no notaba, pero pudimos solucionarlos y estar satisfechos con lo que pudimos hacer.

English:

From the beginning we decided to choose this challenge because it seemed appropriate to use what we learned about the subject and we thought it could be good practice, since also to create the page it was necessary to research in depth the topics taught.

The website project for "Posada Byteados" seeks not only to establish an attractive digital presence, but also to improve the customer experience through an intuitive booking system and provide all the necessary information. It was organized with care and attention to detail to ensure the site's stable operation.

On both the frontend and backend numerous issues were resolved and adjusted during the work, which included visual issues or issues that the client didn't notice, but we were able to solve them and be satisfied with what we were able to do.