

UNIVERSIDAD GERARDO BARRIOS



FACULTAD DE CIENCIA Y TECNOLOGÍA
INGENIERÍA EN SISTEMAS Y REDES INFORMÁTICAS

ASIGNATURA:

PROGRAMACION COMPUTACIONAL III

DOCENTE:

WILLIAN ALEXIS MONTES GIRÓN

ACTIVIDAD:

AVANCE #2

INTEGRANTES:

CARLOS EDUARDO GARCIA CASTILLO (SMSS045523)

CARLOS ERNESTO VÁSQUEZ STECH (SMSS023423)

DIEGO ANTHONIO CABRERA GUEVARA (SMSS020623)

HERSON EXEQUIEL VILLATORO AMAYA (SMSS025023)

STEVEN ARIEL GONZÁLEZ DÍAZ (SMSS029423)

FECHA:

DOMINGO 17 DE NOVIEMBRE DEL 2024

Establecimiento de prioridades

Prioridades altas

Control de la información de la base de datos: Sabemos que debe existir una base de datos adecuadamente nutrida de la información de las reservas. Por eso es que pensamos que es esencial que el sistema se suspenda hasta no contar con todos los datos requeridos y que la información sea sólida y que sea suficiente para que así tengamos todo lo que necesitamos.

Envío de correo de confirmación: Este sistema que creamos, debe enviar automáticamente al correo electrónico del usuario una confirmación lógica de que se ha hecho la reserva de manera correcta, ahí se manda toda la información importante acerca de la reservación que se hizo y los datos esenciales para que el usuario confirme que si hizo correctamente la reserva.

Prioridades medias

Generación y envío de códigos QR: Es deseable que el sistema genere códigos QR únicos para cada reserva y los envíe al cliente, ya que esto facilitaría el acceso a la información de la reserva, este código QR puede venir dentro del correo electrónico que se manda al usuario.

Notificación de fechas ocupadas: También pensamos en que el sistema debe de ser capaz de informar al cliente cuando intente realizar una reserva en una fecha ya que ya esté ocupada, mejorando así la transparencia y evitando errores a la hora de hacer una reserva y así llevar un mejor control.

Prioridades bajas

Diseño visual atractivo de la página web: Si bien es deseable que la interfaz sea bastante agradable a la vista, consideramos que es una funcionalidad de menor prioridad frente a las necesidades funcionales del sistema tales como el control de la información y la funcionalidad de saber que fechas están ocupadas o no.

Resolución del plan de trabajo detallado en el segundo avance: Desarrollar las funcionalidades pendientes del proyecto que estén alineadas con las descritas en el primer avance y que estén marcadas como de alta prioridad para poder implementarlas al proyecto.

1- La página web

Tenemos ahora una página web, la cual tendrá el rol de ser el medio en el cual los clientes podrán realizar sus reservaciones.

Esta página web contará con diferentes opciones las cuales el usuario tendrá que elegir en base a sus deseos a la hora de realizar la reservación. También nos hemos asegurado de que sea una página web que no sea solo funcional y efectivo, sino que sea fácil de usar y agradable a la vista, ya que si esta no fuera del agrado de los clientes, afectaría en la cantidad de personas que quisieran hacer una reservación con nosotros.



The image shows a web browser window displaying the BerlFood reservation form. The header is orange with the logo 'BerlFood' and the tagline 'Un lugar donde el café y la comodidad se encuentran'. Below the header is a dark brown navigation bar with links: Inicio, Menú, Reservación, and Contacto. The main content area has a light orange background and is titled 'Formulario de Reservación'. It contains several input fields: 'Nombre del Cliente:', 'Correo Electrónico:', 'Número de Teléfono:', 'Número de Personas:', 'Fecha de la Reserva:' (with a date picker showing 'dd/mm/aaaa'), and 'Hora de la Reserva:' (with a time picker showing '--:--').

BerlFood
Un lugar donde el café y la comodidad se encuentran

Inicio Menú Reservación Contacto

Formulario de Reservación

Nombre del Cliente:

Correo Electrónico:

Número de Teléfono:

Número de Personas:

Fecha de la Reserva:

Hora de la Reserva:

Correo Electrónico:

Número de Teléfono:

Número de Personas:

Fecha de la Reserva:

Hora de la Reserva:

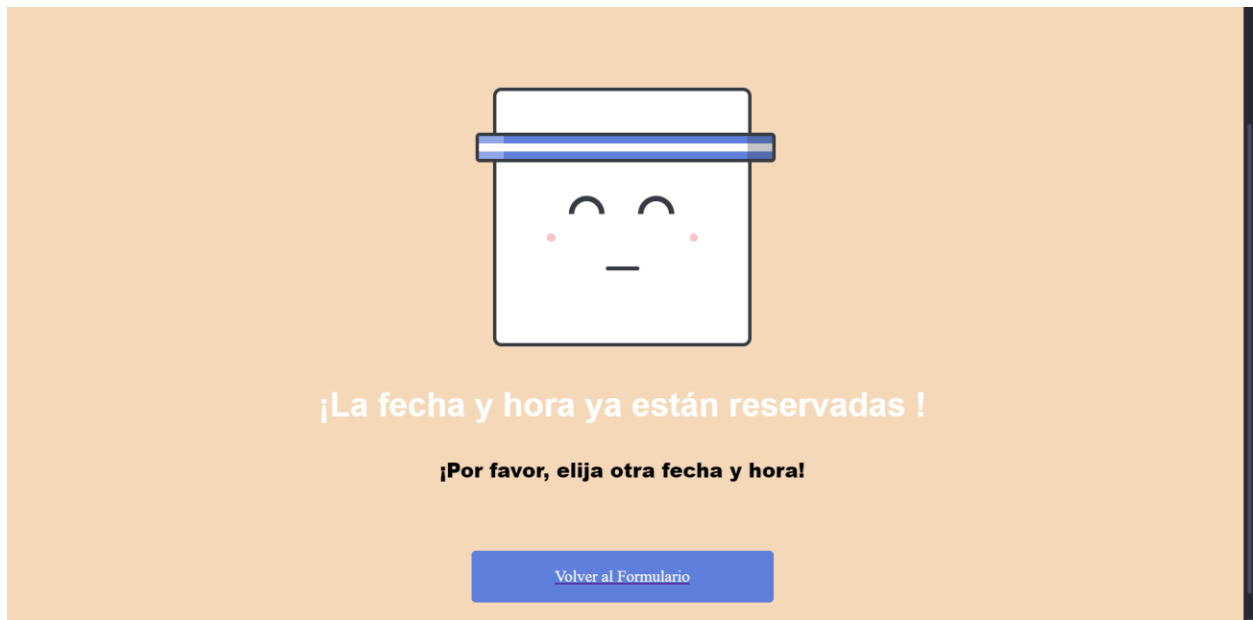
Tipo de Reserva:

Notas:

Acceso:

[Enviar Reservación](#)

También tenemos otras funciones, como, por ejemplo, si un cliente intenta reservar una hora que ya está ocupada, la página redirigirá al usuario a otra página, la cual informa al usuario que esa fecha ya está reservada.

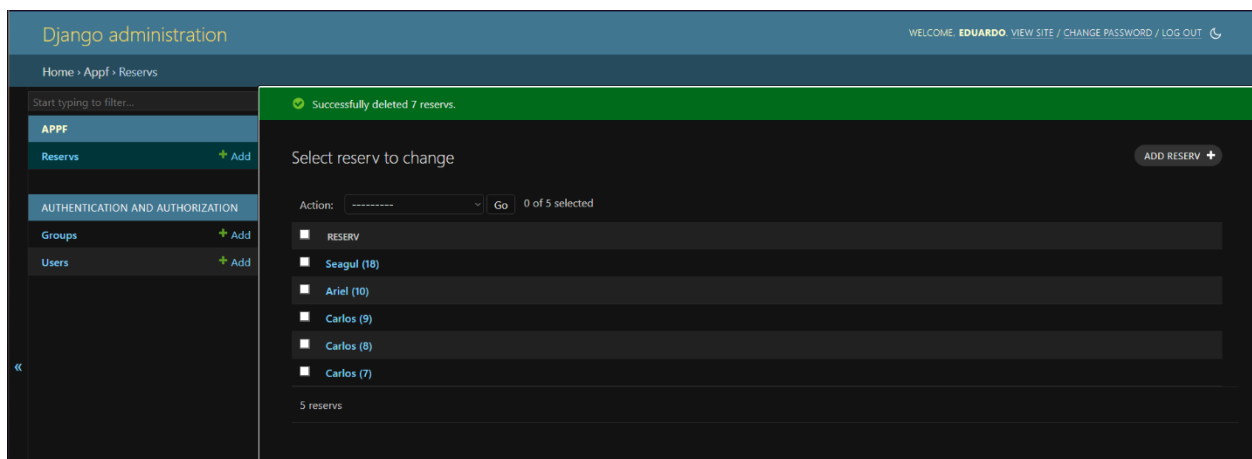


2- La base de datos:

Teniendo página web, también hemos realizado exitosamente una pequeña base de datos, la cual, guarda los datos ingresados por el cliente a la hora de realizar su reservar.

Esta, como mencionamos anteriormente, guarda el nombre del cliente, su correo electrónico, su número de teléfono, el número de personas que asistirán, la fecha y hora de la reserva, el tipo de reservación, alguna nota de parte de los clientes y el tipo de acceso que estos tendrán. Estos datos se almacenarán en esta base de datos, ya que estos son los datos que pedimos a la gente que nos brinden en la página web para realizar su reservación. Con la información guardada funciona la confirmación de las horas que ya hayan sido ocupadas.

Estos datos pasan de la página web a la base de datos, teniendo así nosotros un mejor registro y control sobre toda la información de los clientes.



3- Aplicación con el código.

Ahora que los clientes ingresan su información en nuestra página web, se les envía un correo electrónico con dicha información de forma detallada, para que puedan verla y corroborarla.

También, a parte de la informa en forma únicamente escrita, hemos agregado una confirmación en forma de QR, así, los clientes tienen más seguridad de que su reserva se ha realizado sin errores.

La página web en general también está hecha con código, pero acá nos encargamos de unirla con nuestra base de datos para que las cosas anteriores pudieran funcionar. La redirección a la página de fechas ya ocupadas también es parte de este código.

```
FinalProject > fproject > appF > views.py > enviar_correo
22 def contacto(request):
23     nombre = request.POST.get('nombre')
24     correo = request.POST.get('correo')
25     telefono = request.POST.get('telefono')
26     numero_personas = request.POST.get('numero_personas')
27     fecha = request.POST.get('fecha')
28     hora = request.POST.get('hora')
29     tipo_reserva = request.POST.get('tipo_reserva')
30     notas = request.POST.get('notas')
31     acceso = request.POST.get('acceso')
32
33     reservasExis = Reserv.objects.filter(fecha=fecha, hora=hora)
34     if reservasExis.exists():
35         return render(request, 'error.html')
36
37     # Llama a la función para enviar el correo
38     enviar_correo(nombre, correo, telefono, numero_personas, fecha, hora, tipo_reserva, notas, acceso)
39
40     #llama la base
41     Reserva(nombre, correo, telefono, numero_personas, fecha, hora, tipo_reserva, notas, acceso)
42
43     return HttpResponseRedirect("Reservación enviada con éxito")
44
```

Objetivos faltantes y plan de desarrollo

-Enlazar los botones de la barra de navegación (NavBar)

Para ello, una de las consideraciones a implementar es la navegación funcional entre todas las páginas a través de la barra de navegación. Se refiere a la creación de los enlaces en el navbar que se encuentra en la página principal mencionando, a través de los cuales se podrá acceder a las secciones Acerca de Nosotros y Promociones y Contacto, y de estas últimas con un enlace de retorno a la página y otras secciones principales específicas.

Plan de desarrollo: Se necesitará un framework de la web de Python, tal vez como Flask o Django (Nada más para mencionar alguna) para manejar las rutas de la página. Y digamos que si utilizamos esos nos tendremos que asegurar de que las vistas de sus páginas estén conectadas a su NavBar por medio de plantillas HTML de juego altering y utilizar cualquier sistema de plantillas proporcionadas por Django o Flask para no repetir el código.

-Crear una página “Acerca de Nosotros”

La página presentará información sobre el propósito del sistema, el equipo detrás del proyecto (Nosotros) y los valores que representamos. Esta sección es clave para generar confianza al usuario y también humanizar el sistema.

Plan de desarrollo: Se desarrollará una plantilla HTML específica para esta página, utilizando **CSS** para el diseño visual. La información será almacenada en **SQLite** como documentos JSON, lo que permitirá una fácil actualización del contenido desde el backend. Flask o Django se encargará de recuperar esta información y mostrarla dinámicamente en la página.

-Crear una página de promociones

Esta página va a mostrar una lista dinámica de las promociones disponibles, con imágenes, descripciones y enlaces que sean relevantes para así incentivar la participación de los usuarios.

Plan de desarrollo: Las promociones se almacenarán como documentos en **SQLite**. La página se desarrollará utilizando Flask o Django para extraer los datos y mostrarlos dinámicamente. Y si utilizamos las bibliotecas como **Jinja2** (Flask) o **Django Templates** las vamos a usar para diseñar una interfaz atractiva y que responda de manera correcta.