****

Memoria TFG Desarrollo Aplicaciones Web

Victor Manuel Omar Allende

Tutor: Carlos Rufiángel

# Abstrac

En el presente proyecto he llevado a cabo el desarrollo de una aplicación web, para pequeños y medianos negocios del sector de la hostelería que necesitan realizar tareas como la gestión de su oferta diaria de menú y también obtener una funcionalidad esencial en el negocio como son las reservas online para sus clientes. HostApp nace como una forma fácil, sencilla y económica.

Esta aplicación nace como solución ante grandes apps más complejas, costosas y que no se adaptan al pequeño y mediano negocio de la hostelería.

HostApp tiene un potencial de crecimiento importante dado que su patrón de diseño al ser MVC modulariza cada componente con lo cual se facilita cualquier tipo de evolutivo que cubra necesitades futuras, el poder de customización de HostApp es lo que la hace realmente interesante y válida para las pequeñas y medianas empresa del sector.

In this project I have carried out the development of a web application for small and medium-sized businesses in the hospitality sector that need to perform tasks such as managing their daily menu offer and also obtain essential functionality in the business such as online reservations for your customers. HostApp was born as an easy, simple and economical way.

This application was created as a solution to large, more complex and expensive apps that do not adapt to small and medium-sized hospitality businesses.

HostApp has significant growth potential given that its design pattern, being MVC, modularizes each component, which facilitates any type of evolution that covers future needs. HostApp's customization power is what makes it really interesting and valid for companies. small and medium-sized companies in the sector.

# Índice

[Abstrac 1](#_Toc134901380)

[Índice 2](#_Toc134901381)

[Objetivos 4](#_Toc134901382)

[Introducción 5](#_Toc134901383)

[Justificación del Proyecto 6](#_Toc134901384)

[Desarrollo 7](#_Toc134901385)

[Metodología 7](#_Toc134901386)

[Arquitectura 7](#_Toc134901387)

[IDE 8](#_Toc134901388)

[Control de versiones 8](#_Toc134901389)

[Cliente control de versiones 9](#_Toc134901390)

[Estructura del proyecto 9](#_Toc134901391)

[Modelo 10](#_Toc134901392)

[Vista 11](#_Toc134901393)

[Controlador 13](#_Toc134901394)

[Configuración del proyecto: 14](#_Toc134901395)

[Modelo de datos 14](#_Toc134901396)

[Describe usuarios: 15](#_Toc134901397)

[Describe menu: 15](#_Toc134901398)

[Describe reservas: 15](#_Toc134901399)

[Diseño 0](#_Toc134901400)

[Diseño MVC 0](#_Toc134901401)

[Sesión 0](#_Toc134901402)

[Validaciones 1](#_Toc134901403)

[Encriptación 0](#_Toc134901404)

[Diseño de vistas 0](#_Toc134901405)

[Diagrama de flujo de navegación 1](#_Toc134901406)

[Flujo de navegación usuarios tipo ‘n’ 1](#_Toc134901407)

[Flujo de navegación usuarios tipo ‘a’ 2](#_Toc134901408)

[Diagrama de peticiones 2](#_Toc134901409)

[Diagrama de clases 0](#_Toc134901410)

[Diagrama entidad relación 0](#_Toc134901411)

[Tabla usuarios 0](#_Toc134901412)

[Tabla Menu 0](#_Toc134901413)

[Tabla Reservas 1](#_Toc134901414)

[Diagrama de base de datos 1](#_Toc134901415)

[Metodología 4](#_Toc134901416)

[Diagrama de Gantt 4](#_Toc134901417)

[Tecnologías 5](#_Toc134901418)

[Trabajos futuros 6](#_Toc134901419)

[Pruebas 7](#_Toc134901420)

[Conclusiones 8](#_Toc134901421)

[Glosario 8](#_Toc134901422)

[Bibliografía 9](#_Toc134901423)

[Anexos 10](#_Toc134901424)

# Objetivos

Objetivos generales

Og\_Diseñar una aplicación o solución informática dirigida al negocio de la restauración, HostApp es una solución simple, económica y mantenible para la gestión de cualquier negocio de la restauración para abarcar las necesidades o requerimientos en la gestión de reservas de clientes, así como también proporcionar una gestión personalizada para los administradores del negocio.

Objetivos específicos.

Oe.1\_Analizar número de negocios de hostelería potenciales para adquirir la solución.

Oe.2\_Analizar necesidad de clientes que demanden una atención de reservas online.

Oe.3\_Analizar demanda de las aplicaciones similares en el mercado.

Oe.4\_Definir la solución a medida para nuestros clientes.

# Introducción

En el presente proyecto se desarrollará una aplicación como solución para los negocios del sector de la hostelería, que necesiten o demanden actualizar su sistema de reservas para sus clientes a un sistema de reservas online.

También se pretende cubrir parte del negocio en la que los mismos administradores del negocio o empresa puedan dar a conocer, publicar y administrar online sus menús del día a día de forma rápida y sencilla.

Mantener en todo momento un control exacto de las reservas que hacen sus clientes en su negocio y poder administrarlas.

**HostApp** nace para cubrir las necesidades más demandantes en la gestión del negocio, ser sencilla e intuitiva, para así evitar costes de gestión.

# Justificación del Proyecto

En la actualidad donde cada vez se implementan más soluciones informáticas en todo ámbito de negocio, he detectado que medianas o pequeñas empresas del sector de la hostelería no tienen la suficiente oferta de soluciones informáticas a medida o para tereas simples, la mayoría de estas soluciones son grandes, complejas y costosas.

Observado esto, nace la idea de HostApp, que es una solución simple y adaptable a cualquier negocio que les permite entrar en el ámbito web, permitiendo a sus clientes hacer reservas online, modificarlas o cancelarlas, conocer su oferta día a día sin necesidad de poseer complejas soluciones informáticas, también administrar, crear ofertas (menú) actualizar o eliminar los mismos, conocer el estado de reservas y administrarlas.

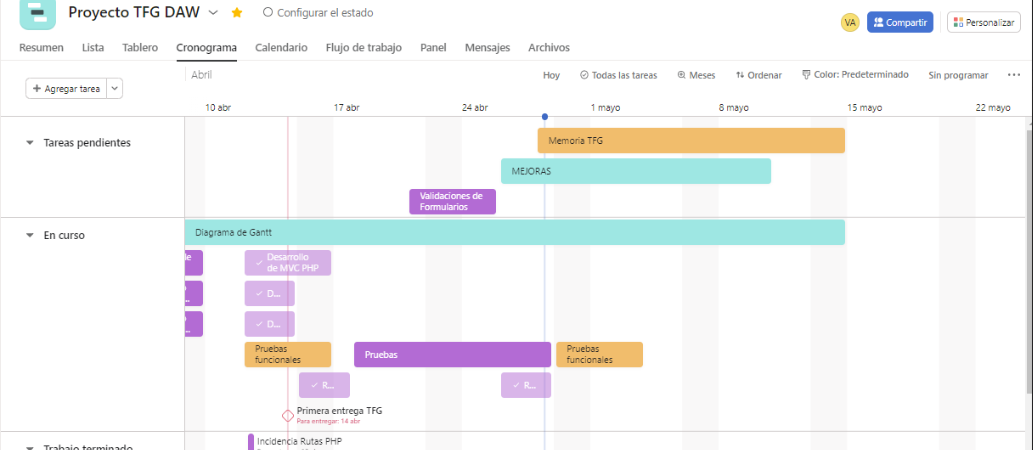
Es por estos motivos que HostApp se posiciona como una potente alternativa para pequeñas y medianas empresas que busquen una solución específica para estas necesidades.

# Desarrollo

Para el desarrollo de **HostApp** he optado por realizarlo con tecnología PHP ya que para aplicaciones web es muy potente, si bien es cierto que con el avance de nuevas tecnologías PHP tiende a perder el dominio que tiene desde el año 2004, sigue estando muy presente en el mundo web a nivel empresarial, con lo cual me parece una oportunidad única para aprender más de esta tecnología y poder desarrollar nuevos conocimientos.

# Metodología

Como metodología y al ser un desarrollador no aplica o no es eficiente la demanda para hacer alguna metodología ágil como ***Scrum*** o ***Kanvan*** etc. Sin embargo, Utilice un sistema de tareas a realizar, tareas en curso, tareas finalizadas, fechas de entrega y poder realizar hitos, utilice ***Asana*** en su periodo de pruebas únicamente ya que Asana requiere licencia:



## Arquitectura

La arquitectura de la aplicación **HostApp** es la siguiente:

|  |  |  |
| --- | --- | --- |
| Servidor | Base de datos | Tecnología |
| Apache Tomcat XAMMP | MySql MariaDB | PHP 7.4.27 |

**El patrón de arquitectura** del software es **Modelo-Vista-Controlador o MVC** que es el estándar que permite una separación de los datos o lógica del negocio de las interfaces, dando así una independencia de funcionamiento y mayor facilidad en su mantenimiento.

El módulo encargado de la comunicación entre vistas y modelo es el módulo de controlador.

Diagrama

Descripción generada automáticamente

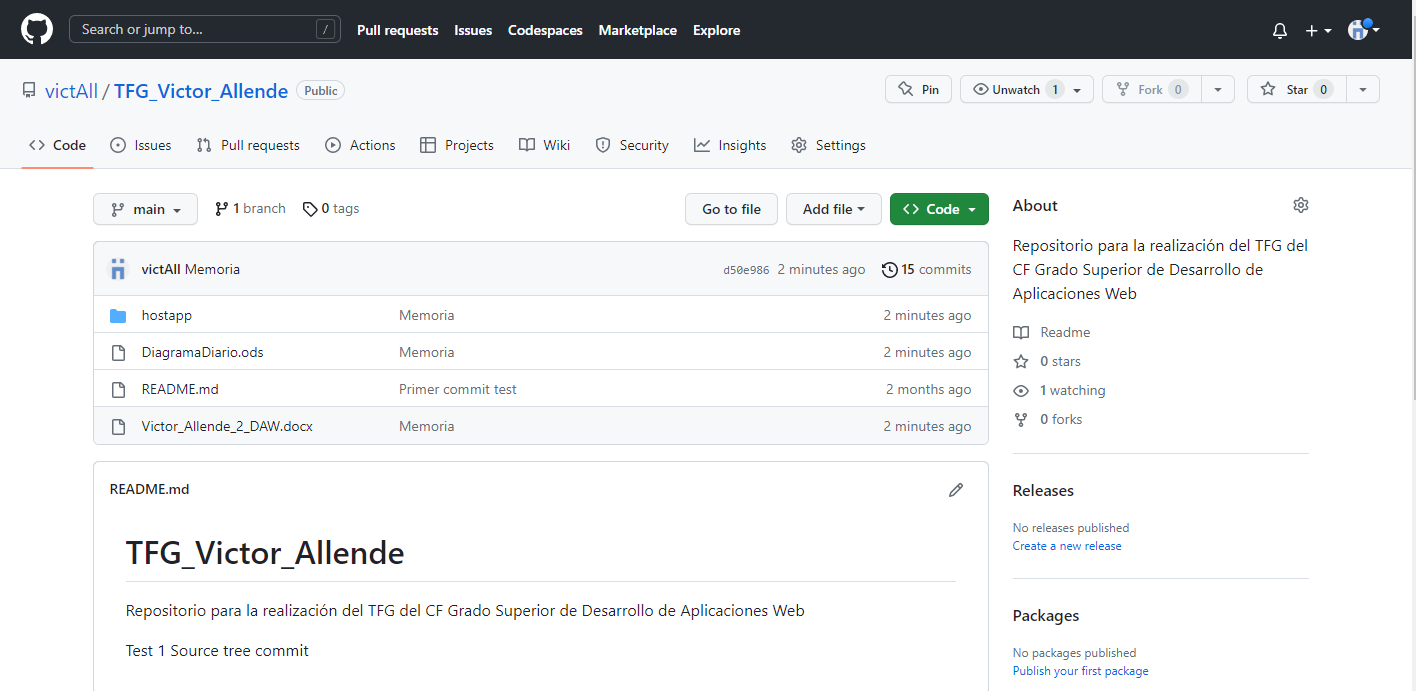
Fuente: <https://hocphp.net/tong-quan/tin-tuc/mo-hinh-mvc-trong-php/>

## IDE

Como ide de desarrollo he seleccionado **Visual Studio Code** en su Version 1.74.2, por su potencial como ide de desarrollo de aplicaciones web y ser de libre distribución, no se necesita licencia.

## Control de versiones

Como sistema de gestión de versiones he utilizado GitHub, ya que poseo una cuenta en GitHub hace unos años y es totalmente gratuita, aunque los repositorios son públicos.



<https://github.com/victAll/TFG_Victor_Allende.git>

## Cliente control de versiones

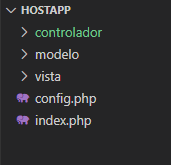
Como cliente GIT he utilizado SourceTree en su versión 3.4.12

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## Estructura del proyecto

La estructura del proyecto se divide en 3 carpetas principales *controlador* contiene la capa controladora, *modelo* contiene la clase encargada de la interacción con la base de datos y *vista* que contine las vistas o interfaces de usuario y formularios, el fichero *index.php* es el fichero de inicio del proyecto y también el encargado de distribuir las peticiones que se envían de los formularios desde las diferentes interfaces, ya sean de tipo *GET* o *POST* y el fichero *config.php* que es el encargado de la configuración del proyecto.



### Modelo

Esta carpeta contiene el principal fichero que permite la interacción de la aplicación con el modelo de datos: *modelo.php* También incluyo un fichero *conexión.php* que es el encargado de abrir conexión con la base de datos, el fichero *sesión.php* encargado de gestionar las sesiones y por último una carpeta *scriptDB* con el fichero *scriptdb.sql* que contiene un script para generar la base de datos desde cero y un juego de datos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Fichero** | **Clase** | **Descripción** |
| modelo.php | class Modelo | Funciones y métodos para interactuar con el modelo de datos. |
| conexión.php |  | Funciones para gestionar conexiones a la base de datos |
| session.php |  | Gestiona las sesiones de los usuarios. |

### Vista

Contiene las vistas y formularios de la aplicación además de los archivos CSS de la aplicación y recursos:

**Carpeta vista**: Vistas de la aplicación, vistas genéricas, de administración y usuarios comunes.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

|  |  |
| --- | --- |
| **Fichero** | **Descripción** |
| actualizar\_menu.php | Vista de administrador de la aplicación |
| actualizar\_reserva.php | Vista de usuario de la aplicación |
| bienvenida\_admin.php | Vista de administrador de la aplicación |
| bienvenida.php | Vista de todos los usuarios de la aplicación |
| crear\_menu.php | Vista de administrador de la aplicación |
| crear\_reserva.php | Vista de usuario de la aplicación |
| despedida.php | Vista de todos los usuarios de la aplicación |
| login.php | Vista de todos los usuarios de la aplicación |
| logout.php | Vista de todos los usuarios de la aplicación |
| mostrar\_menu.php | Vista de administrador de la aplicación |
| mostrar\_reservas\_admin.php | Vista de administrador de la aplicación |
| mostrar\_reservas.php | Vista de usuario de la aplicación |
| registro.php | Vista de todos los usuarios de la aplicación |

**Carpeta CSS**: Estilos de la `aplicación.



|  |  |
| --- | --- |
| **Fichero** | **Descripción** |
| style.css | Estilos de la aplicación |

**Carpeta layouts**: Vistas genéricas header y footer de la aplicación.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

|  |  |
| --- | --- |
| **Fichero** | **Descripción** |
| footer.php | Vista de footer común a toda la aplicación |
| header.php | Vista cabecera común a toda la aplicación |

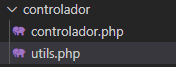
**Carpeta res**: recursos de la aplicación, iconos, banner utilizados por la aplicación.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

### Controlador

En esta carpeta almaceno el fichero Controller de la aplicación y un fichero de utilidades o de validaciones:



|  |  |  |
| --- | --- | --- |
| **Fichero** | **Clase** | **Descripción** |
| controlador.php | class ModeloController | Métodos y funciones que permiten la comunicación entre la vista y el modelo de datos |
| utils.php |  | Métodos y funciones de validaciones |

### Configuración del proyecto:

En la raíz del proyecto se encuentran los ficheros de configuración de HostApp, estos ficheros son *index.php.* y *config.php.*

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente con confianza media

|  |  |
| --- | --- |
| **Fichero** | **Descripción** |
| index.php | Recibe las peticiones *Get* o *Post* desde las vistas o formularios y llama a controlador |
| config.php | Configuración de variables globales a toda la apliicación |

# Modelo de datos

El modelo de datos consiste en 3 tablas, las que se encargan de almacenar *usuarios*, *menú* y *reservas.* A continuación, mostraré un *“describe table”* de cada tabla y las detallo en profundidad cada una en el diagrama de base de datos más adelante.

## Describe usuarios:

Texto

Descripción generada automáticamente

## Describe menu:

Imagen que contiene texto, pizza, tabla, grande

Descripción generada automáticamente

## Describe reservas:

Imagen que contiene texto, marcador, pizza, tabla

Descripción generada automáticamente

# Diseño

El objetivo de este apartado es presentar y dar a conocer el diseño de HostApp, en su totalidad funcional, Mostraré un esquema de las peticiones realizadas desde HostApp, cada una con su ***method*** sea *get* o *post* diagrama de peticiones*,* también mostraré los diagramas de Diseño MVC, Sesión, Diseño de vistas. Diseño de estilos, Diagrama de clases, Diagrama de entidad relación, Diagrama de base de datos, diagrama de flujo de navegación.

## Diseño MVC

El patrón de diseño escogido para HostApp es el más utilizad para animaciones web. Por su alta abstracción de capas, separación de capa de lógica , vistas y modelo, a continuación mostraré el diagrama de MVC utilizado por HostApp.

Diagrama

Descripción generada automáticamente

## Sesión

La Sesión PHP se inicia al hacer login ***sesión\_start()*** y finaliza ***sesión\_destroy()*** al cerrar sesión por parte del usuario logeado:

Diagrama

Descripción generada automáticamente

## Validaciones

Para las validaciones de los datos inyectados a ***HostApp***, he utilizado un fichero ***utils.php*** que es el encargado de realizar estas funciones. Las funciones para las validaciones son las siguientes:



**(\*1)**

* / = Delimitador de la expresión regular, define el inicio y el fin.
* [] = Definición de una clase, el carácter puede ser cualquiera de los valores que se encuentran definidos dentro de la clase.
* a-z = Rango de letras desde la a hasta la z, sería lo mismo que escribir abcdefg...
* A-Z = Rango de letras desde la A hasta la Z, lo mismo que antes, pero en mayúsculas
* \s = El equivalente a un carácter de tipo espacio
* ñáéíóúÁÉÍÓÚ = caracteres que no están incluidos en a-z y A-Z, para que la expresión coincida cuando estén estos caracteres.
* ^ = Define que la expresión debe aplicar desde el comienzo de la cadena
* ·          $ = Define que la expresión debe aplicar hasta el final de la cadena
* ·          {4,} requiere de una cadena de al menos 4 caracteres

**(\*2”)**

* validar longitud && validar que la 9º posición es '-' ej: 53380908-R
* nos quedamos con los números del dni
* nos quedamos con la letra del dni
* fórmula para encontrar la letra
* resto de los números del dni / 23 tiene que ser el numero de la letra final del dni

## Encriptación

La encriptación utilizada por HostApp para el tratamiento y seguridad en las password es el ***MD5*** por su sencillez y eficacia, el producto PHP define una función md5($var) preparada para su consumo.



## Diseño de vistas

En el siguiente diagrama se puede **identificar** a **alto nivel** el diseño de las vistas, identificándose por el tipo de usuario ‘n’ o ‘a’, las vistas comunes y las específicas de cada funcionalidad, más adelante, en el diagrama de flujo de navegación identificaré particularmente cada flujo para cada tipo de usuario.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

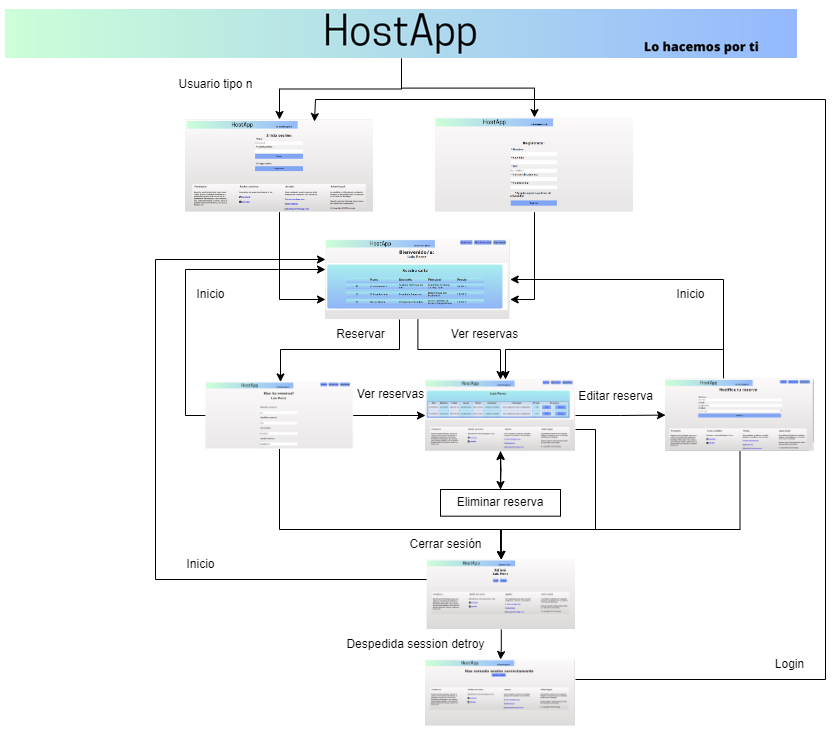
## Diagrama de flujo de navegación

En el siguiente diagrama se muestra el flujo de navegación de **HostApp** en este flujo se definen 2 posibles flujos:

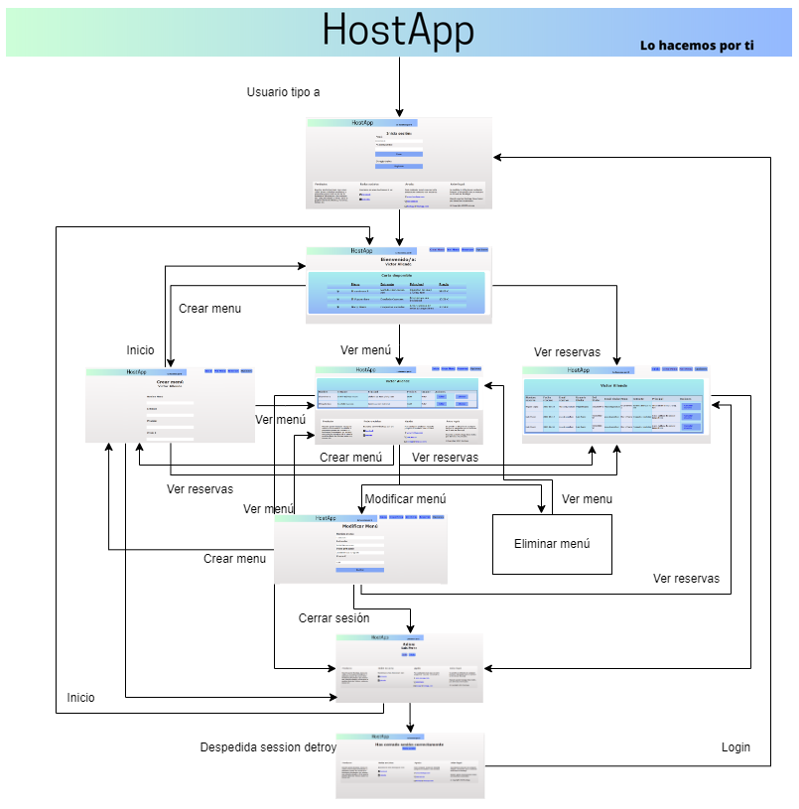
1. Usuario normal o cliente
2. Administrador del sistema

Las acciones de ambos casos o ambos tipos de usuario se verán más en detalle en el diagrama de casos de uso, más adelante.

### Flujo de navegación usuarios tipo ‘n’



### Flujo de navegación usuarios tipo ‘a’



## Diagrama de peticiones

Esquema de llamadas WEB (Get y Post) y funcionalidades de ***HostApp***, se describe cada llamada desde cada vista o formulario, sean get o post así como su tratamiento por parde de la capa controler y su transacción con la capa de modelo de negocio.





## Diagrama de clases

Imagen que contiene Texto

Descripción generada automáticamente

## Diagrama entidad relación

A continuación, presento el diagrama de entidad relación del modelo de datos utilizado por ***HostApp***, en este diagrama se presentan las 3 tablas con las que ***HostApp*** ofrece su funcionalidad.

### Tabla usuarios

Almacena los usuarios registrados en ***HostApp*** tanto, ***administradores*** tipo ‘a’ de ***HostApp*** dados de alta por el administrador de la base de datos y usuarios tipo ‘n’ o clientes que se registran a través de la WEB.

### Tabla Menu

En esta taba se almacena los menús creados por los administradores, son visibles desde la UI para los usuarios tipo ‘n’ y los tipos ‘a’, en el caso de los administradores tipo ‘a’ pueden ver todos los menús creados por todos los administradores, pero solo se permiten acciones sobre los menús propios creados por el administrador logeado en la sesión, esto da una capa de seguridad a ***HostApp*** no permitiendo el borrado intencional o accidental de los menús del negocio. Los usuarios tipo ‘n’ tienen visibilidad sobre todos los menús para la selección en la fase de reserva.

### Tabla Reservas

En esta tabla se almacenan las reservas realizadas por los usuarios tipo ‘n’ a través de ***HostApp***, cada usuario tipo ‘n’ tiene a su disposición sus reservas en la UI, pudiendo actuar sobre las mismas, editar o eliminar.

Los usuarios tipo ‘a’ o administradores tiene visibilidad sobre todas las reservas, no se permite su edición, pero si su eliminación o cancelación.

Diagrama

Descripción generada automáticamente

## Diagrama de base de datos

El diagrama de base de datos presentado a continuación muestra las 3 tablas detalladas anteriormente.

Tabla **usuario**, almacena los usuarios registrados en ***HostApp***, la tabla **menu** que almacena los menús creados por los administradores de ***HostApp*** tiene como FK un usuario administrador, que es el creador y dueño de ese menú, solo se permiten acciones al administrador creador del menú, tiene visibilidad sobre todos los menús de ***HostApp,*** pero no el control del total sobre los menús en su totalidad,

La tabla **reservas** almacena las reservas realizadas por los clientes registrados en ***HostApp.*** Se permite la edición y eliminación a los clientes realizadores de la reserva, a los administradores se les da permiso de delete sobre las mismas.

Diagrama

Descripción generada automáticamente

# Metodología

## Diagrama de Gantt

# Tecnologías

# Trabajos futuros

# Pruebas

# Conclusiones

Resultados obtenidos

# Glosario

# Bibliografía

Fuentes de información

# Anexos

Tablas, glosarios, referencias