**Asignatura: Taller de Diseño y desarrollo de Soluciones**

**Sección:** TIHI42/D-B50-N4-P12-C1/D Talca B5

**Nombre del docente:** Jerman Espíndola Sandoval

**Nombre de los integrantes del grupo:** Alejandro Troncoso y Aline Suarez

**Martes 8 noviembre año 2022**

Guía Informe de Proyecto

Unidad 1: Arquitectura de la solución, almacenamiento y plan de pruebas

**Contenido**

[I. Introducción 3](#_Toc471831140)

[II. Objetivo 3](#_Toc471831141)

[III. Desarrollo 3](#_Toc471831142)

[IV. Conclusiones 29](#_Toc471831143)

[V. Referencias bibliográficas 29](#_Toc471831144)

1. Introducción

Principalmente, en esta fase del proyecto se lleva a cabo una breve investigación acerca de la estructura del patrón de arquitectura a usar en el proyecto de software “DinnerApp”, en este caso el modelo MVT el cual facilita la gestión con el framework django.

En segundo lugar, se mencionan los distintos tipos de servicios cloud existentes actualmente. Posteriormente, se realiza un estudio y análisis cuidadoso a cada uno, considerando costos, características, ventajas y desventajas para determinar la arquitectura a utilizar, y por qué es el más adecuado para el proyecto.

Finalmente, se desarrolla el plan de pruebas, enfocado en certificar la calidad de los requerimientos funcionales del software. El plan de pruebas es conjunto de casos de prueba, basados en los posibles escenarios que podrían ocasionar provocar fallos al usuario del sistema.

1. Objetivo

Investigar, definir, planificar, diseñar y documentar los patrones de arquitectura que serán utilizados en el desarrollo del proyecto, el proveedor del almacenamiento cloud y el plan de pruebas para la fase de testing que será ejecutada, evaluada y certificada por el equipo QA después de cada Sprint.

1. Desarrollo
2. Definir los patrones de arquitectura de la solución.

Los patrones de arquitectura son métodos base para el diseño de diferentes funcionalidades de un software. Dan una descripción de los elementos, el tipo de relación entre ellos y un conjunto de restricciones de uso. Por ejemplo, la arquitectura en capas es un estilo de llamamiento-y-regreso, cuando define uno un estilo general para interaccionar. Cuando esto es descrito estricta y comúnmente disponible, es un patrón. (Wikipedia, 2020)

En el proyecto “DinnerApp” el patrón de arquitectura a utilizar es el MVT (Modelo-Vista-Template), ya que el framework Django respeta dicho patrón. Originalmente el patron MVT surge del MVC, es django quien lo redefine. (Profe, 2022)

El patrón MVT se usa para implementar interfaces web y aplicaciones, se compone de lo siguiente: Modelo: maneja lo relacionado con la información, incluye como acceder a esta, validación, relación entre los datos y su comportamiento. Vista: es un enlace entre el modelo y el template, decide qué información será mostrada y por cual template. Template: decide cómo será mostrada la información.

MTV de Django funciona de la siguiente manera: Al momento de hacer clic en un enlace o escribir una dirección a lo primero que se accede es al mapa de URLs en este archivo cada ruta está asociado con una view, si se necesita algún dato se solicitara este a modelo el cual a su vez generara la consulta a la base de datos, cuando los datos han sido traídos estos son enviados al template que contiene la lógica de presentación para estos. Luego de "pintar" la página esta se envía al navegar que hizo la solicitud. (espifreelancer, 2019)

La siguiente imagen representa gráficamente la explicación anterior:

Diagrama

Descripción generada automáticamente

1. Definir almacenamiento de datos en cloud.

El almacenamiento en la nube es un modelo que almacena datos en Internet a través de un proveedor de informática en la nube que administra y opera el almacenamiento en la nube como un servicio. Los datos son alojados en uno o más servidores de manera virtual.

Las compañías de alojamiento operan enormes centros de procesamiento de datos, y los usuarios que requieren los servicios deben comprar, alquilar o contratar la capacidad de almacenamiento necesario. A los servicios de almacenamiento en nube, se puede acceder por diferentes medios, como un servicio web (web service), interfaz de programación de aplicaciones (API), interfaz de usuario (interfaz web).

Generalmente hay tres modelos de servicio en la nube para comparar:

* **Software como servicio (SaaS):**
  + Ejemplos comunes: *Google Workspace, Dropbox, Cisco, WebEx, Concur, GoToMeeting.*
* **Plataforma como servicio (PaaS):**
  + Ejemplos comunes: *AWS Elastic Beanstalk, Windows Azure, Heroku, Force.com, Google App Engine, Apache Stratos, OpenShift.*
* **Infraestructura como servicio (IaaS):**
  + Ejemplos comunes: *DigitalOcean, Linode, Rackspace, Amazon Web Service (AWS), Cisco Metapod, Microsoft Azure, Google Compute Engine (GCE).*

El enfoque principal de este informe está en las Plataformas como Servicio (PaaS), el cual proporciona una plataforma para la creación de software. Esta plataforma se entrega a través de la web, lo que brinda a los desarrolladores libertad de concentración en el software sin tener que preocuparse por sistemas operativos, actualizaciones de software, almacenamiento o infraestructura. Está basado en la tecnología de virtualización, por lo que los recursos se pueden escalar hacia arriba o hacia abajo, proporciona una variedad de servicios para ayudar con el desarrollo, prueba e implementación de aplicaciones, es accesible a numerosos usuarios e integra servicios web y bases de datos.

PaaS permite a las empresas diseñar y crear aplicaciones integradas en PaaS con componentes de software especiales, aplicaciones denominadas middleware, son escalables y de alta disponibilidad. Actualmente dentro de las plataformas más conocidas y utilizadas, se posicionan:

* Microsoft Azure: Servicio de computación en la nube creado por Microsoft para construir, probar, desplegar y administrar aplicaciones y servicios mediante el uso de sus centros de datos.
* Google App Engine: Servicio de hosting que permite ejecutar aplicaciones sobre la infraestructura de Google.
* IBM Cloud: Conjunto de servicios de computación en la nube para empresas que ofrece la empresa de tecnología de la información IBM

Entre los 3, el elegido para el proyecto “DinnerApp” es Microsoft Azure debido a la cantidad de productos y servicios en la nube que ofrece a un precio accesible. Ofrece máxima seguridad, es simple, y ofrece una buena experiencia al usuario.

1. Diseñar el plan de pruebas.

El plan de pruebas tiene como objetivo orientar el esfuerzo de pruebas, identificando y detallando las pruebas más importantes, para que el equipo de QA pueda enfocarse en su ejecución y pueda responder de forma adecuada a los cambios que tiene el proyecto.

El objetivo de la norma ISO 29119 es crear un estándar definitivo sobre pruebas software, que recoja y estandarice el vocabulario, los procesos, técnicas de documentación, etc., del ciclo de vida de las pruebas. (Garzas, 2012)

El objetivo del plan de pruebas es certificar la calidad de los requisitos funcionales del software a desarrollar. El cual se hace a través de la siguiente estructura:

|  |  |
| --- | --- |
| Funcionalidad: | Registrarse en la aplicación |
| Identificador: | CP\_add1 |
| Nombre: | Agregar usuario no registrado |
| Descripción: | Se realizará desde cualquier usuario (cliente o dueño de restaurante) autenticado previamente. Se ejecutarán registros con diferentes longitudes en los campos solicitados. |
| Entradas: | Nombre  Correo  Clave  Confirmación de clave |
| Salida esperada: | Que la aplicación ingrese al nuevo usuario en sus registros, muestre un mensaje en pantalla que diga “Usuario ingresado exitosamente. Bienvenid@.” Y luego de unos segundos, el sistema debe redirigir al perfil del usuario. |

|  |  |
| --- | --- |
| Funcionalidad: | Registrarse en la aplicación |
| Identificador: | CP\_add2 |
| Nombre: | Agregar clave con cantidad de caracteres inválidos |
| Descripción: | Se realizará desde cualquier usuario previamente registrado. Se ingresa clave con 7 (inferior al mínimo) caracteres y/o 13 (superior al máximo). |
| Entradas: | Nombre  Correo  Clave  Confirmación de clave |
| Salida esperada: | El sistema debe mostrar un error en pantalla indicando cuales son los requisitos de la clave (ej.: “Cantidad de caracteres invalida, debe estar entre 8 y 12”) |

|  |  |
| --- | --- |
| Funcionalidad: | Registrarse en la aplicación |
| Identificador: | CP\_add3 |
| Nombre: | Agregar nombre con espacios |
| Descripción: | Se realizará desde cualquier usuario previamente registrado. El nombre no debe contener espacios. |
| Entradas: | Nombre  Correo  Clave  Confirmación de clave |
| Salida esperada: | El sistema debe mostrar un error en pantalla indicando cuales son los requisitos del nombre de usuario. (ej.: “El nombre no debe contener espacios y debe incluir al menos una mayúscula y una minúscula”). |

|  |  |
| --- | --- |
| Funcionalidad: | Inscribir restaurante |
| Identificador: | CP\_add4 |
| Nombre: | Agregar restaurante ya inscrito. |
| Descripción: | Debe hacerlo un usuario dueño del restaurante, previamente registrado. Se intenta agregar un restaurante con datos de uno que ya existe en la base de datos del sistema. |
| Entradas: | Nombre del local  Teléfono  Dirección  Hora de apertura  Hora de cierre  Días disponibles  Agregar menú (opcionalmente) |
| Salida esperada: | El sistema debe mostrar un error indicando que dato primario fue encontrado en los registros de la base de datos. (ej.: Nombre o patente del restaurante). |

|  |  |
| --- | --- |
| Funcionalidad: | Inscribir restaurante |
| Identificador: | CP\_add5 |
| Nombre: | Agregar caracteres de texto en campo teléfono |
| Descripción: | Función ejecutada por el dueño del restaurante, previamente registrado. Intenta agregar caracteres de tipo cadena de texto en un campo que debe ser de tipo numérico. |
| Entradas: | Nombre del local  Teléfono  Dirección  Hora de apertura  Hora de cierre  Días disponibles  Agregar menú (opcionalmente) |
| Salida esperada: | El sistema debe mostrar un error en pantalla indicando que el número telefónico debe ser de tipo numérico. |

|  |  |
| --- | --- |
| Funcionalidad: | Realizar reservación |
| Identificador: | CP\_add6 |
| Nombre: | Agregar campo fecha requerida |
| Descripción: | Función disponible solo para el usuario cliente, previamente registrado. El cual, intenta reservar una mesa, el campo fecha es requerido para averiguar que mesas estarán disponibles tal día. |
| Entradas: | Fecha  Hora  Seleccionar mesa |
| Salida esperada: | El sistema debe redirigir a la página de selección de mesa. |

|  |  |
| --- | --- |
| Funcionalidad: | Publicar ofertas del negocio |
| Identificador: | CP\_add7 |
| Nombre: | Agregar oferta |
| Descripción: | Función disponible solo para usuario dueño de restaurante previamente registrado e inscrito en la aplicación. El usuario desea agregar oferta |
| Entradas: | Formulario con detalles de la oferta:  Precio normal  Precio  Horario de validez |
| Salida esperada: | El sistema debe validar si los campos obligatorios del formulario están completos para agregar oferta en el listado y mostrarlo en pantalla. De lo contrario, muestra error en pantalla indicando requisitos del formulario. |

|  |  |
| --- | --- |
| Funcionalidad: | Inscribir restaurante |
| Identificador: | CP\_Leave1 |
| Nombre: | Dejar campo dirección requerido vacío |
| Descripción: | Esta función es ejecutada por un dueño de restaurante, previamente registrado. Intenta inscribir restaurante dejando el campo requerido en blanco. |
| Entradas: | Nombre del local  Teléfono  Dirección  Hora de apertura  Hora de cierre  Días disponibles  Agregar menú (opcionalmente) |
| Salida esperada: | El sistema muestra error en pantalla, indicando que dicho campo es obligatorio. |

|  |  |
| --- | --- |
| Funcionalidad: | Realizar reservación |
| Identificador: | CP\_Leave2 |
| Nombre: | Dejar campo fecha requerido vacío |
| Descripción: | Función disponible para el usuario cliente, previamente registrado. El campo es requerido para hacer la reservación, se deja en blanco para ver si el sistema permite realizar la reservación. |
| Entradas: | Fecha  Hora  Seleccionar mesa |
| Salida esperada: | El sistema debe indicar en pantalla que el campo es obligatorio. |

|  |  |
| --- | --- |
| Funcionalidad: | Realizar reservación |
| Identificador: | CP\_Leave3 |
| Nombre: | Dejar campo horas requerido vacío |
| Descripción: | Función disponible solo para el usuario cliente, previamente registrado. Se requiere indicar hora para hacer una reservación, el cliente intenta dejarlo en blanco para ver si el sistema le permite continuar de igual manera. |
| Entradas: | Fecha  Hora  Seleccionar mesa |
| Salida esperada: | El sistema debe indicar un error en pantalla especificando que no puede quedar en blanco ese campo. |

|  |  |
| --- | --- |
| Funcionalidad: | Realizar reservación |
| Identificador: | CP\_Leave4 |
| Nombre: | Dejar campo mesa requerido vacío. |
| Descripción: | Función disponible solo para el usuario cliente, previamente registrado. Se requiere elegir mesa para hacer una reservación, el cliente intenta continuar con el campo vacío. |
| Entradas: | Fecha  Hora  Seleccionar mesa |
| Salida esperada: | El sistema muestra error en pantalla, indicando que se requiere elegir mesa a reservar. |

|  |  |
| --- | --- |
| Funcionalidad: | Inscribir restaurante |
| Identificador: | CP\_Leave5 |
| Nombre: | Dejar campo nombre vacío |
| Descripción: | Función disponible para usuario dueño de restaurante, previamente registrado. El campo es requerido para poder continuar, es imposible inscribir el restaurante sin ingresar nombre. |
| Entradas: | Nombre del local  Teléfono  Dirección  Hora de apertura  Hora de cierre  Días disponibles  Agregar menú (opcionalmente) |
| Salida esperada: | El sistema muestra error indicando que el campo es obligatorio. |

|  |  |
| --- | --- |
| Funcionalidad: | Ver horarios de reservaciones |
| Identificador: | CP\_Select1 |
| Nombre: | Seleccionar zoom al gráfico |
| Descripción: | Función disponible solo para dueños de restaurantes, previamente registrado y con su restaurante inscrito en la aplicación. Función que acerca o aleja el gráfico según el usuario lo manipule. |
| Entradas: | No tiene |
| Salida esperada: | Si el usuario indica acercar, el grafico y su contenido debe verse más grande y legible. De lo contrario, se va haciendo pequeño el contenido y no se entiende. |

|  |  |
| --- | --- |
| Funcionalidad: | Agregar restaurante a favoritos |
| Identificador: | CP\_Select2 |
| Nombre: | Seleccionar corazón |
| Descripción: | Función disponible para usuario cliente, previamente registrado. El cliente hace clic al corazón, cuando el corazón esta en negro significa que el restaurante fue agregado a favoritos. De lo contrario, el restaurante esta fuera de la lista de favoritos. |
| Entradas: | No tiene |
| Salida esperada: | El sistema debe mostrar el cambio de estado del corazón (en negro o no), según sea el estado, se debe mostrar un mensaje “Restaurante agregado a favoritos” o “Restaurante eliminado de favoritos”. |

|  |  |
| --- | --- |
| Funcionalidad: | Ver categorías de restaurantes |
| Identificador: | CP\_Select3 |
| Nombre: | Seleccionar ver categorías |
| Descripción: | Función disponible para cualquier usuario, no necesariamente debió registrarse previamente. El usuario puede ver, buscar, seleccionar categorías de los restaurantes inscritos. |
| Entradas: | No tiene |
| Salida esperada: | El sistema debe mostrar una lista con las categorías de los restaurantes inscritos. |

|  |  |
| --- | --- |
| Funcionalidad: | Ver restaurantes cerca |
| Identificador: | CP\_select4 |
| Nombre: | Seleccionar zoom al mapa |
| Descripción: | Función disponible para usuario cliente, no necesariamente registrado previamente. El sistema toma la ubicación del dispositivo. Función que acerca o aleja el mapa según el usuario lo manipule. |
| Entradas: | Se requiere ubicación del dispositivo. |
| Salida esperada: | Si el usuario indica acercar, el grafico y su contenido debe verse más grande y legible. De lo contrario, se va haciendo pequeño el contenido y no se entiende. |

|  |  |
| --- | --- |
| Funcionalidad: | Ver restaurantes cerca |
| Identificador: | CP\_select5 |
| Nombre: | Seleccionar restaurante |
| Descripción: | Función disponible para cualquier usuario, no necesariamente debe estar registrado previamente. El usuario intenta acceder a un restaurante a través del mapa. |
| Entradas: | Selección del usuario a cualquier restaurante en pantalla. |
| Salida esperada: | El sistema debe redirigir a una pantalla donde se muestren los detalles del restaurante seleccionado. |

|  |  |
| --- | --- |
| Funcionalidad: | Ver restaurantes mejor valorados |
| Identificador: | CP\_select6 |
| Nombre: | Seleccionar detalles |
| Descripción: | Función disponible para cualquier usuario, no necesariamente debió registrarse previamente. El usuario intenta ver los detalles del restaurante. |
| Entradas: | Selección según criterio del usuario. |
| Salida esperada: | El sistema debe redirigir a pantalla donde se muestren los detalles del restaurante elegido por el usuario. |

|  |  |
| --- | --- |
| Funcionalidad: | Ver restaurantes mejor valorados |
| Identificador: | CP\_select7 |
| Nombre: | Seleccionar precio |
| Descripción: | Función disponible para cualquier usuario no necesariamente debe estar registrado. El usuario intenta acceder a los precios según menús disponibles. |
| Entradas: | Selección según criterio del usuario. |
| Salida esperada: | El sistema debe redirigir a otra pantalla con los detalles de cada uno de los precios en el restaurante seleccionado por el usuario. |

|  |  |
| --- | --- |
| Funcionalidad: | Publicar ofertas del negocio |
| Identificador: | CP\_select8 |
| Nombre: | Seleccionar eliminar |
| Descripción: | Función disponible solo para usuario dueño de restaurante previamente registrado e inscrito. El usuario puede eliminar la oferta publicada desde la lista. |
| Entradas: | Selección según criterio del usuario. |
| Salida esperada: | El sistema debe mostrar una interfaz para confirmar que desea eliminar la publicación. Según la selección del usuario, el sistema decide si sacar la oferta o si continua en la lista. |

|  |  |
| --- | --- |
| Funcionalidad: | Sacar restaurante del servicio |
| Identificador: | CP\_select9 |
| Nombre: | Seleccionar confirmación en “SI” |
| Descripción: | Función disponible solo para usuario dueño de restaurante previamente registrado e inscrito. El usuario confirma que desea salir de la aplicación, ingresando un código que le llega al correo. |
| Entradas: | Código |
| Salida esperada: | Si el usuario confirma, la pagina debe redirigir al perfil de usuario, indicando en un mensaje “gracias por su tiempo con nosotros”. |

|  |  |
| --- | --- |
| Funcionalidad: | Modificar reservación |
| Identificador: | CP\_select10 |
| Nombre: | Seleccionar fecha en tiempo pasado |
| Descripción: | Función disponible para usuario cliente previamente registrado. El usuario ingresa fecha errónea. |
| Entradas: | Fecha |
| Salida esperada: | El sistema debe mostrar un error en pantalla indicando que la fecha indicada es inferior a la actual, por lo que es imposible realizar la modificación de reserva. |

|  |  |
| --- | --- |
| Funcionalidad: | Modificar reservación |
| Identificador: | CP\_select11 |
| Nombre: | Seleccionar hora no disponible |
| Descripción: | Función disponible para usuario cliente previamente registrado. El usuario intenta ingresar hora no disponible |
| Entradas: | Fecha  Hora |
| Salida esperada: | El sistema muestra error en pantalla e indica los requisitos para ingresar una hora. |

|  |  |
| --- | --- |
| Funcionalidad: | Cancelar reservación |
| Identificador: | CP\_select12 |
| Nombre: | Seleccionar enlace restaurante |
| Descripción: | Función disponible para usuario cliente previamente registrado. El usuario intenta acceder al perfil del restaurante. |
| Entradas: | No tiene |
| Salida esperada: | El sistema debe redirigir al perfil del restaurante con los detalles principales. |

|  |  |
| --- | --- |
| Funcionalidad: | Cancelar reservación |
| Identificador: | CP\_select13 |
| Nombre: | Seleccionar a “X” en campo “cancelar” de la tabla |
| Descripción: | Función disponible solo para el usuario cliente. El usuario desea cancelar una reservación, entonces debe hacer clic en la “X” que aparece al nivel del nombre del restaurante en la tabla. |
| Entradas: | Código de confirmación |
| Salida esperada: | El sistema debe mostrar ventana emergente para ingresar el código de confirmación, una vez validado, el sistema borra la reservación de la lista. |

|  |  |
| --- | --- |
| Funcionalidad: | Reservar menú |
| Identificador: | CP\_select14 |
| Nombre: | Seleccionar menú disponible |
| Descripción: | Función disponible para usuario cliente. El usuario intenta ver una lista con los menús disponibles en el restaurante elegido. |
| Entradas: | Elección del restaurante |
| Salida esperada: | El sistema debe mostrar lista de menús disponibles en el restaurante, el usuario selecciona el menú que desea y el sistema debe mostrar un mensaje que indique que el menú fue seleccionado. |

|  |  |
| --- | --- |
| Funcionalidad: | Habilitar/deshabilitar mesa |
| Identificador: | CP\_select15 |
| Nombre: | Seleccionar mesa para habilitar/deshabilitar |
| Descripción: | Función disponible para usuario dueño de restaurante. Al hacer clic se cambia el estado de la mesa. (Ej.: Esta en rojo (inhabilitada) al hacer clic automáticamente cambia a verde (habilitada)) |
| Entradas: | No tiene |
| Salida esperada: | El sistema debe mostrar el cambio de color de la mesa e indicar el cambio de estado a través de un mensaje en pantalla. |

|  |  |
| --- | --- |
| Funcionalidad: | Elegir restaurante |
| Identificador: | CP\_enter1 |
| Nombre: | Ingresar nombre Restaurante no inscrito |
| Descripción: | Función disponible para cualquier usuario, no necesariamente debe estar previamente registrado en la aplicación. El usuario busca/filtra restaurantes por nombre para acceder directamente a sus detalles. |
| Entradas: | Nombre |
| Salida esperada: | El sistema indica con un mensaje en pantalla que el restaurante no aparece en inscrito en la aplicación. De lo contrario, se redirige a interfaz con los detalles del restaurante. |

|  |  |
| --- | --- |
| Funcionalidad: | Elegir restaurante |
| Identificador: | CP\_enter2 |
| Nombre: | Ingresar a menú disponible |
| Descripción: | Función disponible para cualquier usuario, no necesariamente debe estar previamente registrado en la aplicación. |
| Entradas: | No tiene |
| Salida esperada: |  |

|  |  |
| --- | --- |
| Funcionalidad: | Elegir restaurante |
| Identificador: | CP\_enter3 |
| Nombre: | Ingresar a menú no disponible |
| Descripción: |  |
| Entradas: | No tiene |
| Salida esperada: |  |

|  |  |
| --- | --- |
| Funcionalidad: | Ver horarios de reservaciones |
| Identificador: | CP\_enter4 |
| Nombre: | Ingresar código de reservación |
| Descripción: | Función disponible solo para dueños de restaurante previamente inscrito. El usuario ejecuta en la barra de búsqueda ingresando un código de reservación existente. |
| Entradas: | Código |
| Salida esperada: | Si el sistema encuentra un código que coincida con el buscado, el sistema muestra detalles de la reservación en pantalla. |

|  |  |
| --- | --- |
| Funcionalidad: | Ver horarios de reservaciones |
| Identificador: | CP\_enter5 |
| Nombre: | Ingresar fecha reservación |
| Descripción: | Función disponible solo para dueños de restaurante previamente inscrito. El usuario ejecuta en la barra de búsqueda una filtración de datos ingresando una fecha de reservación existente. |
| Entradas: | Fecha |
| Salida esperada: | El sistema busca en los registros de la base de datos, si coincide, muestra todos los resultados coincidentes en pantalla y la opción disponible para acceder a detalles de reservación. |

|  |  |
| --- | --- |
| Funcionalidad: | Ver restaurantes mejor valorados |
| Identificador: | CP\_enter6 |
| Nombre: | Ingresar a menú |
| Descripción: | Función disponible para cualquier usuario no necesariamente debió estar registrado previamente. El usuario desea visualizar menús disponibles a través de la interfaz de restaurantes mejor valorados. |
| Entradas: | Elección de restaurante según criterio del usuario |
| Salida esperada: | El sistema debe mostrar detalles de menús disponibles en el restaurante elegido por el usuario. |

|  |  |
| --- | --- |
| Funcionalidad: | Ver restaurantes mejor valorados |
| Identificador: | CP\_enter7 |
| Nombre: | Ingresar a ubicación |
| Descripción: | Función disponible para cualquier usuario no necesariamente debió estar registrado previamente. El usuario desea conocer donde se ubica el restaurante. |
| Entradas: | No tiene |
| Salida esperada: | El sistema debe mostrar la dirección del restaurante. |

|  |  |
| --- | --- |
| Funcionalidad: | Publicar ofertas del negocio |
| Identificador: | CP\_enter8 |
| Nombre: | Ingresar a detalles |
| Descripción: | Función disponible solo para usuario dueño de restaurante previamente registrado e inscrito en la aplicación. El usuario intenta ver los detalles de las ofertas ingresadas. |
| Entradas: | Código de validación de usuario |
| Salida esperada: | El sistema debe mostrar ventana emergente que permita ingresar un código para validar usuario. El sistema lo valida y si se acepta, se redirecciona a una pagina donde se muestran detalles de la oferta. |

|  |  |
| --- | --- |
| Funcionalidad: | Sacar restaurante del servicio |
| Identificador: | CP\_enter9 |
| Nombre: | Ingresar motivo sin seleccionar opción |
| Descripción: | Función disponible para usuario dueño de restaurante previamente registrado e inscrito en la aplicación. El usuario debe seleccionar un motivo obligatoriamente. Para confirmar que deja la aplicación, debe agregar código de validación. |
| Entradas: | Motivo  Código de validación |
| Salida esperada: | El sistema debe mostrar ventana emergente que permita ingresar un código para validar usuario. El sistema lo valida y si se acepta, se redirecciona al perfil del usuario. |

|  |  |
| --- | --- |
| Funcionalidad: | Modificar reservación |
| Identificador: | CP\_enter10 |
| Nombre: | Ingresar al mapa |
| Descripción: | Función disponible para usuario cliente. Se intenta acceder al mapa que incluye la ubicación del restaurante. |
| Entradas: | Selección de restaurante por el usuario. |
| Salida esperada: | El sistema debe mostrar en el mapa el nombre del restaurante, indicar las calles y numero entregados por el dueño del cliente al inscribirse en la aplicación como restaurante. |

|  |  |
| --- | --- |
| Funcionalidad: | Modificar reservación |
| Identificador: | CP\_enter11 |
| Nombre: | Ingresar a “ver mesa” para seleccionar otra. |
| Descripción: | Función disponible para el usuario cliente. Se intenta seleccionar una mesa. La mesa debe estar en color verde (habilitada) |
| Entradas: | Mesa para elegir |
| Salida esperada: | El sistema muestra las mesas del restaurante, cada una tiene un color (rojo o verde), lo que indica que esta deshabilitada o habilitada, respectivamente. Si el usuario selecciona una mesa verde, el sistema le indica que seleccionó esa mesa en la modificación de reserva. De lo contrario, se muestra un mensaje de error indicando que la mesa que fue seleccionada estaba inhabilitada, ya que estaba reservada por otro usuario cliente. |

|  |  |
| --- | --- |
| Funcionalidad: | Cancelar reservación |
| Identificador: | CP\_enter12 |
| Nombre: | Ingresar a modificaciones |
| Descripción: | Función disponible para usuario cliente previamente registrado. El usuario desea modificar un dato de una reservación ya realizada. Debe ingresar código de reservación para verificar que existe. |
| Entradas: | Código de reservación |
| Salida esperada: | El sistema muestra ventana emergente para ingresar el código de reservación, una vez ingresado, el sistema valida la coincidencia del código, si acepta, el sistema redirige a la interfaz de “realizar reservación” donde se accede a los datos por modificar. Una vez realizada la modificación, el sistema muestra un mensaje como notificación y envía un correo avisando sobre la modificación y adjunta documento con los detalles. |

1. Conclusiones

* Presentar una síntesis, donde se expongan ideas principales y algunas ideas personales en torno al tema. También puede incorporar ideas fuerza y/o aportes a partir del trabajo desarrollado.
* También es posible incorporar reflexiones, incluso dejar propuestas de profundización que no fueron posibles de abordar en este informe o trabajo.

1. Referencias bibliográficas

espifreelancer. (20 de Octubre de 2019). *espifreelancer.com*. Obtenido de https://espifreelancer.com/mtv-django.html

Garzas, J. (1 de febrero de 2012). *javiergarzas.com*. Obtenido de www.javiergarzas.com

Profe, H. (20 de Octubre de 2022). *docs.hektorprofe*. Obtenido de https://docs.hektorprofe.net/django/web-personal/patron-mvt-modelo-vista-template/

Wikipedia. (2020). *wikipedia.org*. Obtenido de https://es.wikipedia.org/wiki/Django\_(framework)