Índice

[HELP – DESK 2](#_Toc96898850)

[Inicio de Sesión (LOGIN) 2](#_Toc96898851)

[Vistas Usuario 4](#_Toc96898852)

[Crear un Reporte 8](#_Toc96898853)

[Mis Servicios Pendientes 12](#_Toc96898854)

[Mis Servicios Terminados (PDF Final) 15](#_Toc96898855)

[Firmar Reporte 18](#_Toc96898856)

[Vistas Administrador 20](#_Toc96898857)

[Usuarios 20](#_Toc96898858)

[Editar Datos del Usuario 23](#_Toc96898859)

[Servicios Pendientes 27](#_Toc96898860)

[Cambiar Estado de Reporte 29](#_Toc96898861)

[Terminar Reporte 30](#_Toc96898862)

[Filtro por Fecha 32](#_Toc96898863)

[Reporte Firmado y Recogido 33](#_Toc96898864)

[Recoger Reporte 34](#_Toc96898865)

# HELP – DESK

Aplicación en la cual se atenderán peticiones de los usuarios a problemas técnicos en el centro de cómputo. Dicho sistema ayudará a agilizar el proceso de atención y respuesta a dichas peticiones, dando una visualización de que etapa lleva la petición, desde pendiente, en proceso y finalizado, una vez llegando a esta etapa, también se llevará acabo las firmas correspondientes al documento y se verá si el documento ya ha sido recogido por el personal encargado.

# Inicio de Sesión (LOGIN)

Texto

Descripción generada automáticamente

Ilustración 1 Abstracción de los datos para el inicio de sesión

Texto

Descripción generada automáticamente

Ilustración 2 Función para el inicio de sesión

Texto

Descripción generada automáticamente

Ilustración 3 Obteniendo los datos y validado el login de usuario

Para el funcionamiento del inicio de sesión se usaron varios archivos los cuales tienen distintos propósitos, para iniciar el archivo loginUsuarios.php obtiene los datos necesarios para entrar iniciar sesión en el sistema, en este caso los elementos únicamente son el nombre de usuario y la contraseña, una vez que los obtiene se mandan como parámetros a otro archivo (Usuarios.php) que contiene una función para validarlos con los datos registrados en la base de datos y se envía una respuesta de 1 en caso de que se haya encontrado a un usuario con esas credenciales y un 0 en caso de que no se encuentre nada, esta respuesta es usada en un archivo JS (JavaScript) el cual dependiendo de la respuesta envía al usuario al inicio del sistema o en caso de haber escrito mal las credenciales o simplemente que no se encuentre un usuario, manda un mensaje de error. Es importante mencionar que hay funciones de encriptación, pero esas funciones serán explicadas más adelante.

# Vistas Usuario

Texto

Descripción generada automáticamente

Ilustración 4 Código header parte 1



Ilustración 5 Código header parte 2

Texto

Descripción generada automáticamente

Ilustración 6 Código footer

Implementamos el uso de dos archivos denominados como header y footer (encabezado y pie de página) para hacer que el resto de la programación fuese más limpia y únicamente se mostraran los elementos necesarios de una página y no estar repitiendo tantas veces el mismo código en cada archivo, observando la parte del código del archivo header, se puede apreciar que recurrimos al uso de sesiones para determinar lo que cada usuario puede ver dependiendo de los roles que tenemos asignados en la base de datos, en este caso el número 1 indica al sistema que el usuario es un cliente mientras que el número 2 indica que es un administrador.

Texto

Descripción generada automáticamente

Ilustración 7 Código de servicios pendientes parte 1

Se puede apreciar en las imágenes del código que el archivo PHP contiene únicamente los elementos que necesita para funcionar, lo que se indica con las primeras líneas de código es que el archivo incluirá lo que está presente en el archivo header.php para que de esta forma la parte del encabezado del sistema (la cual corresponde al menú superior) pueda ser visualizada una vez se ingrese a la parte de servicios pendientes del usuario y con las consultas de las sesiones se puede saber quién es el que está ingresando al sistema y de esta forma obtener los datos correspondientes, en cuanto al resto del código es lo que se necesita para crear un elemento que contenga en este caso una tabla que será utilizada para mostrar los reportes o servicios generados y almacenados en la base de datos.

Texto

Descripción generada automáticamente

Ilustración 8 Código servicios pendientes parte 2

La segunda imagen del código incluye la importación de otros tres archivos PHP que son necesarios para su correcto funcionamiento, como lo es el caso del footer.php, como elemento adicional se creó un loader, este loader es una forma visual de indicarle al usuario o administrador que se están cargando los datos del sistema, una vez cargados completamente el loader desaparece y muestra la página correspondiente con los datos completamente cargados, por ultimo pero no menos importante, se está haciendo una validación desde el inicio del archivo PHP hasta el final. Esta validación lo que hace es comprobar que el usuario exista y mantenga iniciada la sesión, esto se puede comprobar al ingresar como un usuario y en una ventana nueva se ingrese como un administrador, la validación tomará como prioridad la última sesión iniciada, en este caso la de administrador y si se recarga la página el sistema lo redireccionará al inicio de sesión del sistema.

## Crear un Reporte

Texto

Descripción generada automáticamente

Ilustración 9 Código modal crear reporte parte 1

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración 10 Código modal crear reporte parte 2

Para la generación de servicios o reportes, se implementó el uso de un elemento de Bootstrap llamado modal, este elemento genera una ventana emergente que nos puede ayudar para el llenado de formularios o mensajes de alerta, más sin embargo es un elemento que no nos ayuda por sí solo por lo que se hace el uso de otros tres archivos, dos archivos PHP y un archivo JavaScript para el registro de los datos.

Texto

Descripción generada automáticamente

Ilustración 11 Código crearReporte.php

Texto

Descripción generada automáticamente

Ilustración 12 Función crearReporte (Reportes.php)

Texto

Descripción generada automáticamente

Ilustración 13 Funciones que proporcionan el id del usuario que inició sesión

Texto

Descripción generada automáticamente

Ilustración 14 Función crearReporte (Reportes.js)

El concepto del funcionamiento es básicamente el mismo, mediante el uso del archivo crearReporte.php (Ilustración 11) se obtienen los datos que fueron ingresados en el modal para la creación del reporte y haciendo el uso de la función generarID le asignamos el siguiente ID disponible en la base de datos, posteriormente estos datos son enviados al archivo de Reportes.php (Ilustración 12) en forma de un arreglo de datos para hacer su uso en la función crearReporte.php del mismo archivo, en esta parte se hace una consulta SQL que lo que hace es insertar los datos obtenidos del modal dentro de la base de datos y posteriormente enviar una respuesta de 1 en caso de que la consulta se haya realizado o 0 en caso contrario, esta respuesta es enviada por último al archivo reportes.js (Ilustración 14) para hacer la validación del proceso y mostrarla al usuario.

## Mis Servicios Pendientes

Texto

Descripción generada automáticamente

Ilustración 15 Código para la generación de la tabla de reportes pendientes parte 1

Texto

Descripción generada automáticamente

Ilustración 16 Código para la generación de la tabla de reportes pendientes parte 2

Texto

Descripción generada automáticamente

Ilustración 17 Código para la generación de la tabla de reportes pendientes parte 3

Texto

Descripción generada automáticamente

Ilustración 18 Código para la generación de la tabla de reportes pendientes parte 4

Texto

Descripción generada automáticamente

Ilustración 19 Carga de las tablas

La generación de las tablas en el sistema es demasiado sencillo, cada tabla hace una consulta SQL la cuál obtiene los datos de los registros en la base de datos, en el caso de los usuarios, se obtienen primero los datos del usuario que inició sesión y en la consulta SQL se comparan para que se muestren únicamente los datos correspondientes al usuario que está activo, es decir si quien inició sesión tiene el ID de usuario 1 la tabla únicamente mostrará los reportes o servicios que haya solicitado ese usuario. Por último, se denomina una función JavaScript que se encarga de cargar la tabla cada vez que se necesite y al momento de haber creado un reporte la tabla se recarga para mostrar los datos registrados en la tabla.

Estas funciones son utilizadas en todas las veces que se genera una tabla, únicamente cambiando la consulta SQL.

Texto

Descripción generada automáticamente

Ilustración 20 Función para recargar la tabla

## Mis Servicios Terminados (PDF Final)

Texto

Descripción generada automáticamente

Ilustración 21 Generación de PDF del servicio terminado (formato 004) parte 1

Texto

Descripción generada automáticamente

Ilustración 22 Generación de PDF del servicio terminado (formato 004) parte 2

Texto

Descripción generada automáticamente

Ilustración 23 Generación de PDF del servicio terminado (formato 004) parte 3

Texto

Descripción generada automáticamente

Ilustración 24 Generación de PDF del servicio terminado (formato 004) parte 4

Es importante mencionar que la generación de este documento toma los campos de la tabla de reportes terminados y obtiene los datos correspondientes al reporte o servicio que se está terminando, de igual forma la extensión con la que se puede guardar este archivo es en PDF, de forma que los archivos generados no tienen un gran tamaño de almacenamiento.

Explicando el código presentado en las ilustraciones, el inicio de la generación del documento PDF es con una validación en la base de datos para saber si se encuentra el reporte o solicitud terminado, en caso de ser así, pasaría a ejecutar una consulta SQL la cual pide todos los datos relacionados con el ID de reporte que se quiere imprimir y posteriormente se guardan esos datos en un arreglo el cual es recorrido dependiendo que depende de los campos registrados. Al ser complicado modificar el documento para indicar que tipo de mantenimiento se realizó, se hizo una validación la cual modifica el documento y tacha la opción que corresponda al tipo de mantenimiento, posteriormente a eso los demás campos son llenados llamando los datos uno por uno en los diferentes campos (Ilustración 23).

## Firmar Reporte

Texto

Descripción generada automáticamente

Ilustración 25 Código modaFirmarReporte parte 1

Texto

Descripción generada automáticamente

Ilustración 26 Código modalFirmarReporte parte 2

Texto

Descripción generada automáticamente

Ilustración 27 Código firma.php

Texto

Descripción generada automáticamente

Ilustración 28 Función FirmarReporte

La firma del reporte contiene el mismo concepto de otras funciones del sistema, obtener los datos de un modal para poder ser utilizados en las funciones que realizan las operaciones correspondientes al cambio en la base de datos, en este caso no se usó un archivo de JavaScript nuevo para validar la realización de las funciones ya que se generaba un problema al tener más de un modal con las mismas funcionalidades y se terminó remplazando el uso de Ajax por Fetch de JavaScript, este cambio no afecta a como se realizan las funciones, únicamente afecta a el como se obtienen los datos. Para terminar, simplemente recalcar que el uso del archivo firma.php (Ilustración 27) es para obtener los datos del modal y mandarlos a la función FirmarReporte (Ilustración 28) que se encuentra en el archivo de Reportes.php.

# Vistas Administrador

## Usuarios

Texto

Descripción generada automáticamente

Ilustración 29 Consulta SQL para obtener los datos de los usuarios

Texto

Descripción generada automáticamente

Ilustración 30 Código modalAgregar

Texto

Descripción generada automáticamente

Ilustración 31 Obtención de los datos del modal

Texto

Descripción generada automáticamente

Ilustración 32 Función agregar persona

Texto

Descripción generada automáticamente

Ilustración 33 Función para agregar un usuario

Texto

Descripción generada automáticamente

Ilustración 34 Código para validar que el proceso se realizó correctamente

El proceso para agregar un nuevo usuario es similar al de crear un reporte, obteniendo los datos de un modal, usando un archivo PHP para enviarlos a la función correspondiente y por último validar que todo se haya hecho correctamente, sin embargo, la diferencia entra en que aquí hay dos funciones que ejecutarse para funcionar las cuales son agregar persona y agregar usuario (Ilustraciones 32 y 33).

## Editar Datos del Usuario

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración 35 Código modalEditar

Texto

Descripción generada automáticamente

Ilustración 36 Código para obtener los datos del modalEditar

Texto

Descripción generada automáticamente

Ilustración 37 Función para obtener los datos del usuario a editar y mostrarlos en el modal parte 1

Texto

Descripción generada automáticamente

Ilustración 38 Función para obtener los datos del usuario a editar y mostrarlos en el modal parte 2

Texto

Descripción generada automáticamente

Ilustración 39 Funciones para editar los datos de la persona a partir del id de usuario

Texto

Descripción generada automáticamente

Ilustración 40 Función para editar los datos del usuario

Texto

Descripción generada automáticamente

Ilustración 41 Funciones para encriptar y desencriptar las contraseñas

Editar los datos de un usuario conlleva un proceso algo más tardado o detallado ya que en realidad se pueden estar editando los datos de dos tablas a la vez, el proceso es el mismo, obtener los datos de un archivo y pasarlos a otro para ser utilizados en las funciones correspondientes, sin embargo aquí entran varias funciones en juego, como lo es el caso de obtenerIdPersona y editarPersona (Ilustración 39), la primer función obtiene el ID de la tabla persona que corresponde al usuario que se quiere editar, por ejemplo, puede haber un usuario con ID usuario 5 al que le corresponde el ID persona 3, esto con el fin de editar datos como el nombre de esa persona o el correo electrónico, después de esto tenemos la función editarPersona, esta función se encarga de actualizar los datos de la base de datos mediante una consulta SQL y para ello depende de la primer función ya que se necesita saber a quien se le esta haciendo el cambio para poder realizar los cambios en el registro pertinente.

En seguida de esas funciones tenemos la función de editarUsuario (Ilustración 40) la cual se encarga de editar los datos del usuario en forma general y su realización depende de si se ha hecho correctamente la función de editarPersona, cabe aclarar que no importa cual dato se esté alterando, las funciones siguen una de otra independientemente de lo que se haga.

Por último pero no menos importante se hace el uso de la encriptación de contraseñas, las funciones de encriptación (Ilustración 41) se usan en el login para poder comparar la contraseña tecleada por el usuario y compararla con la encriptada en la base de datos; al agregar un nuevo usuario para que la contraseña esté encriptada de forma automática y por último en editar usuario para poder visualizar la contraseña en texto plano y no encriptada, en caso de cambiar la contraseña esta se vuelve a encriptar y actualizar en la base de datos.

## Servicios Pendientes

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Ilustración 42 Consulta SQL para mostrar los reportes pendientes o en proceso

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración 43 Detalles de un servicio/reporte

En este caso los servicios o reportes tienen una vista diferente a la que se puede apreciar en el lado del usuario, primero es importante destacar que el administrador puede imprimir la solicitud para tener un respaldo del servicio ya sea de forma física o de forma digital, otro aspecto importante es el cambio de estado, en ocasiones un servicio puede llevar más de una jornada laboral en llevarse a cabo, por lo que es importante brindar la posibilidad de marcar un servicio en proceso para indicarle al solicitante que el problema está siendo resuelto y por último está el botón de terminar un reporte o servicio.

Texto

Descripción generada automáticamente

Ilustración 44 Generación en PDF del primer formato (002)

La diferencia entre la generación del primer al segundo documento recae en la consulta SQL que obtiene los datos de la base de datos, pero en esencia es lo mismo que cuando se genera el segundo documento.

## Cambiar Estado de Reporte

Texto

Descripción generada automáticamente

Ilustración 45 Código modalCambiarEstado

Captura de pantalla de un celular

Descripción generada automáticamente

Ilustración 46 Obtención de los datos de CambiarEstado

Pantalla negra con letras blancas

Descripción generada automáticamente

Ilustración 47 Función para cambiar el estado en la base de datos

La diferencia entre los demás procesos recae en la consulta que realiza la función CambiarEstado (Ilustración 47).

## Terminar Reporte

Texto

Descripción generada automáticamente

Ilustración 48 Código modalTerminarReporte

Texto

Descripción generada automáticamente

Ilustración 49 Obtención de los datos de modalTerminarReporte

Texto

Descripción generada automáticamente

Ilustración 50 Función terminarReporte

En términos prácticos los procesos para agregar un registro en una base de datos son lo mismo, la diferencia está en la tabla a la que se mandan los datos y los datos a mandar, por lo tanto, el crear un nuevo reporte o terminarlo son iguales, sin embargo, la función de terminarReporte tiene una consulta SQL adicional para cambiar el estado del reporte a terminado.

## Filtro por Fecha

Texto

Descripción generada automáticamente

Ilustración 51 Consulta SQL si se ingresa un lapso

Texto

Descripción generada automáticamente

Ilustración 52 Consulta SQL si no se ingresa un lapso

Por último, la parte de los servicios terminados, en esta sección se implementó un filtrado de búsqueda por fechas para que los administradores puedan consultar de forma sencilla los servicios generados en un determinado tiempo que puede ser desde semanas hasta años dentro del Instituto Tecnológico de Aguascalientes y en general es bastante similar a los otros apartados que incluyen una tabla.

## Reporte Firmado y Recogido

Texto

Descripción generada automáticamente

Ilustración 53 Validaciones de firma y documento recogido

Aunque la generación de las tablas es el mismo es importante destacar estas validaciones ya que son un comprobante visual para el usuario y administrador de que cada uno ha cumplido con su responsabilidad, cuando el solicitante del servicio o reporte haya firmado el documento el botón de firmado se verá de otra forma para el administrador y a este mismo se la dará la opción de marcar el reporte como recogido en caso de haberlo recogido.

## Recoger Reporte

Texto

Descripción generada automáticamente

Ilustración 54 Código modalRecogerReporte

Texto

Descripción generada automáticamente

Ilustración 55 Obtención de los datos de modalRecogerReporte

Texto

Descripción generada automáticamente

Ilustración 56 Función RecogerReporte

Nuevamente el proceso de obtener los datos y enviarlos a una función se hace presente, en este caso la función RecogerReporte se encarga de actualizar en la base de datos si el reporte se ha recogido o no.