**PORTAFOLIO**

**COMPETENCIAS**

* *Diseñar software y modelos de datos de acuerdo con los requerimientos de las partes interesadas y los estándares definidos.*
* *Desarrollar prototipos de software utilizando tecnologías de manera innovadora de acuerdo con los objetivos del negocio.  
  Levantar requerimientos del negocio con las partes interesadas, de acuerdo con estándares definidos.*
* *Generar información del negocio, a través del análisis de datos y procesos, de acuerdo con objetivos y requerimientos de la organización.*
* *Gestionar proyectos informáticos de pequeña envergadura de acuerdo con las definiciones, requerimientos de la organización y buenas prácticas de la industria.*
* *Construir soluciones de software y componentes de acuerdo con los requerimientos de las partes interesadas, estándares definidos y pruebas de aseguramiento de la calidad.*
* *Administrar software de sistemas de la organización, monitoreando la seguridad y operación, evaluando e informando problemas detectados, de acuerdo con requerimientos.*
* *Gestionar la seguridad de la organización y sus sistemas, de acuerdo con estándares establecidos.*

**EVIDENCIA 1**

|  |  |
| --- | --- |
| **ALUMNO:** | *Rober Alonzo Astorga Berna* |
| **FECHA:** | *04/01/2021* |
| **ACTIVIDAD:** | *Control de los ingresos, egresos y vueltos de la caja chica.* |

**Descripción del requerimiento**

*La problemática es que se requiere un control de la caja chica ya que no se tiene, este control debe tomar en cuenta el ingreso, egreso y el vuelto de dinero a la caja chica.*

**Entregable**

*El entregable esperado es tener un apartado dentro del sistema que permitiera hacer realizar el registro o ingreso de dinero y también el egreso de este con una glosa. También debe quedar un registro de las operaciones realizadas y en caso de que se produjese un reingreso de dinero como un vuelto, este también debe quedar registrado.*

**Fechas**

*La fecha especificada comprendía desde el 05 - 12 de enero 2021 aproximadamente.*

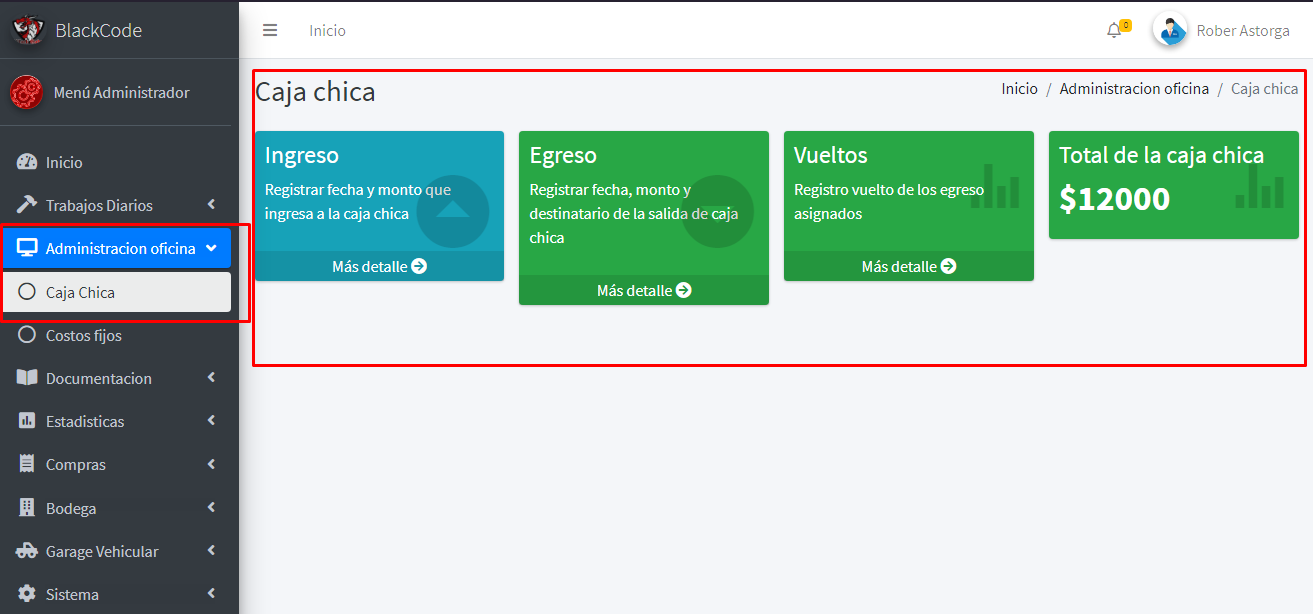
**Descripción de las acciones realizadas**

*La actividad desarrollada contempla las siguientes tecnologías web:*

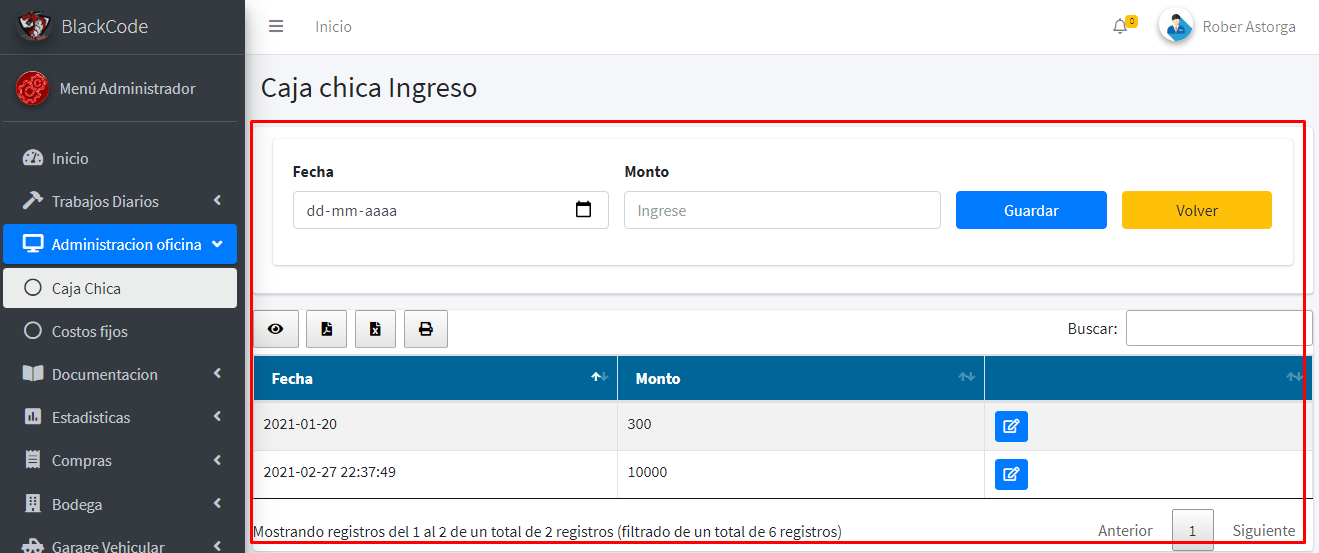
* *Lenguaje de programación: PHP y JavaScript.*
* *Framework codeigniter MVC.*
* *Diseño se utilizó la plantilla de adminlte3 que usa el framework de diseño Bootstrap 3.*
* *Para el registro de los datos se utiliza el motor base de datos MYSQL.*

*también se utilizaron algunas librerías extras como la librería de DataTable, la cual sirve para mostrar una lista con un buscador integrado.*

* *De acuerdo con análisis previos, se ubicó esta nueva opción en el menú de administrador dentro del sistema, como se representa en la siguiente imagen:*



* *Se dividió en 3 opciones que permiten el registro de ingreso, egreso y el vuelto de dinero. La última opción representa el balance total de la caja chica.*
* *Para el ingreso de dinero se tiene la siguiente vista:*



* *Donde se permite el registro de ingreso indicando una fecha y el monto a ingresar luego dando clic al botón guardar se envía la información ingresada a un archivo JavaScript, el cual después mediante una petición Ajax envía esta información a un controlador que se encarga de validar esta información y enviarla al modelo para realizar el registro en la base de datos. Una vez registrado el monto, se actualiza la página web y queda el registro reflejado en la tabla de más abajo, donde tiene la opción de editar el registro si así el usuario lo requiriese. El proceso de registro se refleja de la siguiente forma.*

Archivo JavaScript

Datos desde la vista (fecha y monto)

*notificación al*

*Al usuario*

Generar notificación de registro

Controlador

(validación de datos)

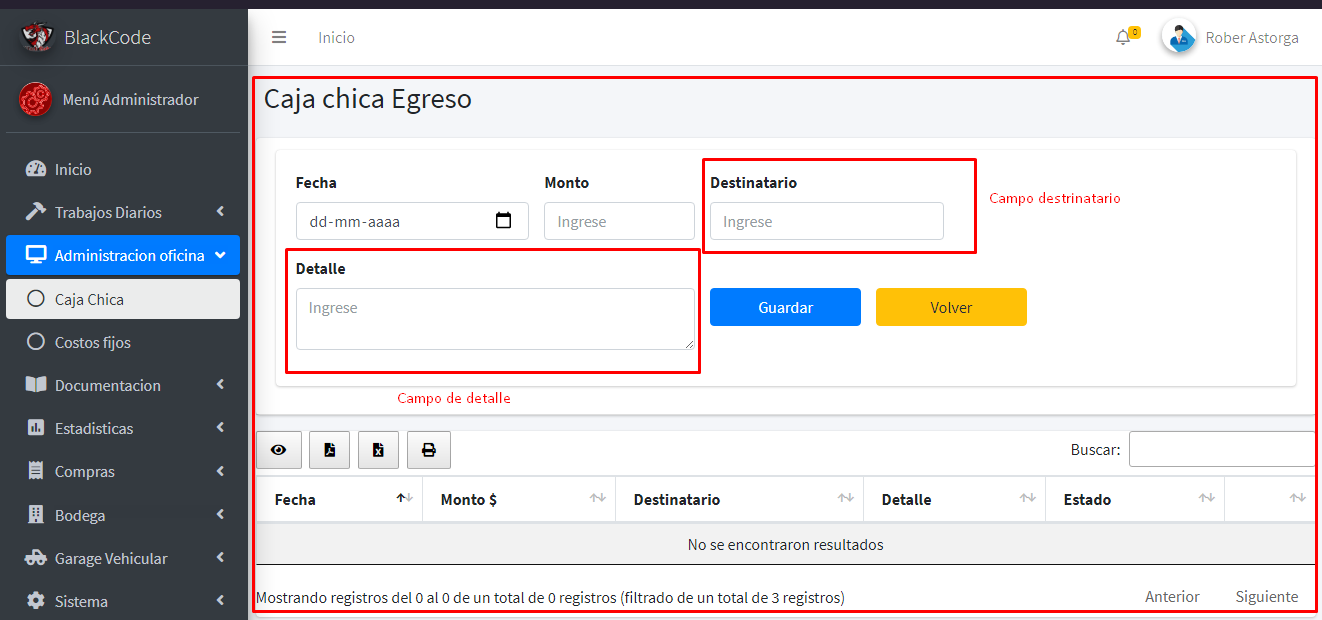
*validación correcta*

Modelo

(Registro base datos)

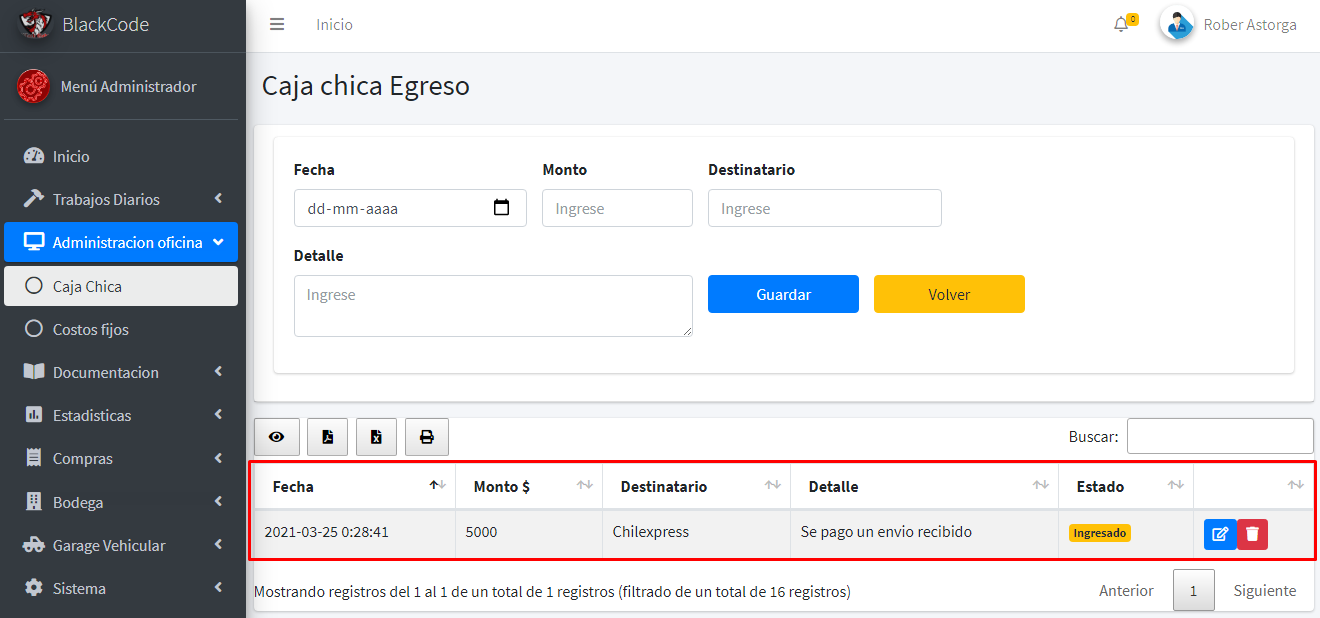
*Registro correcto*

* *Para el proceso de registro de egreso es el mismo que el ingreso, solo que el egreso debe descontar al total que hay en la caja chica para que así se produzca el balance. La vista realizada para el egreso de dinero se realizó de la siguiente imagen:*

**

*También en el egreso de dinero se agregaron 2 campos extras, los cuales son el destinatario y el detalle ya que el requerimiento era que se necesitaba saber para quien va dirigido el dinero que se sacara y el motivo del porque se sacó tal dinero. Luego de la misma forma el ingreso, el registro queda grabado en una lista, la cual también tiene un buscador integrado.*

*En esta lista también, otro punto a destacar es que tiene un estado, este estado representa el estado del dinero egresado, es decir si el dinero que se saco tuvo vuelto o no. La siguiente imagen representa un registro donde se sacó $5000 de la caja chica.*

**

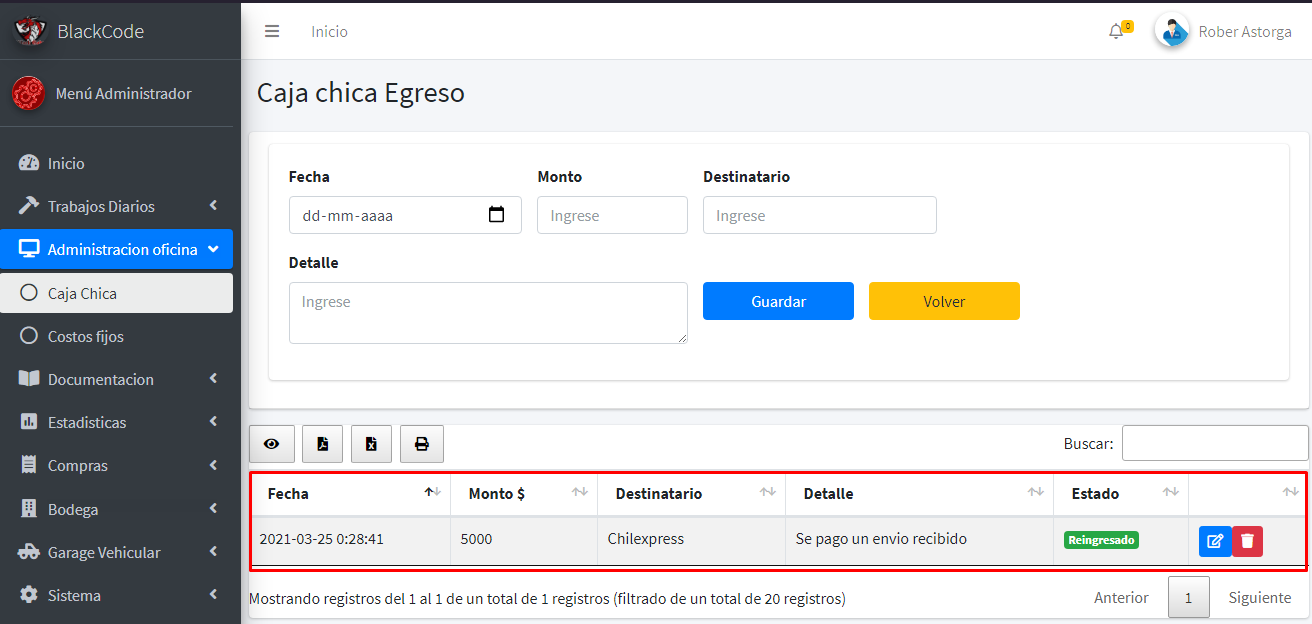
* *Entonces el estado esta como ingresado, quiere decir que no ha habido un vuelto.*

*La siguiente imagen representa que se ha registrado un vuelto.*

**

*Gracias a este estado se sabe si el dinero que se sacó tuvo un reingreso mediante un vuelto de dinero.*

*De esta manera si se requiere realizar un egreso de dinero nuevamente, se puede saber que, en un regreso anterior, se registró un vuelto. Este requerimiento también fue se suma importancia para tener más control del dinero que se saca y el vuelto que es devuelto a la caja chica.*

**

**Conclusiones**

*La actividad fue llevada con éxito, un punto que fue complicado implementar fueron las tablas donde quedan los registros, es decir la librería de DataTable. Esta librería era desconocida para mí, por lo que tuve que investigar un poco para aprender a usarla, pero al final es muy útil ya que contiene un buscador propio y además un estilo bien particular en cuanto a las fechas propuestas estuvo dentro del rango y por último es posible mejorar la aplicación quizás mediante un comprobante del dinero egresado, donde la aplicación web permita la subida de documentos y sea posible subir la boleta o el documento que respalda lo retirado de la caja chica.*

**EVIDENCIA 2**

|  |  |
| --- | --- |
| **ALUMNO:** | *Rober Alonzo Astorga Berna* |
| **FECHA:** | *04-01-2021* |
| **ACTIVIDAD:** | *Control de documentos subidos al sistema, documentos permanentes y actualizables.* |

**Descripción del requerimiento**

Se requiere poder tener un registro de la documentación subida, esta documentación puede ser de 2 tipos

* Permanente: Solo subir documentos con una fecha indicada
* Actualizable: Subir documentos, pero con una fecha límite que indique o notifique al usuario un máximo de 3 días antes de la fecha límite, el vencimiento del documento

**Entregable**

El entregable solicitado es tener un apartado en el sistema que permita subir documentos con una fecha, pero dividirlo en 2 secciones, permanente y actualizable. La documentación actualizable se debe poder registrar una fecha límite para que el sistema notifique en un máximo de 3 días antes de la fecha limite indicada antes, el vencimiento del documento para que el usuario pueda actualizar el documento.

**Fechas**

La fecha para el desarrollo de la actividad abarca desde el 13 – 18 de enero aproximadamente.

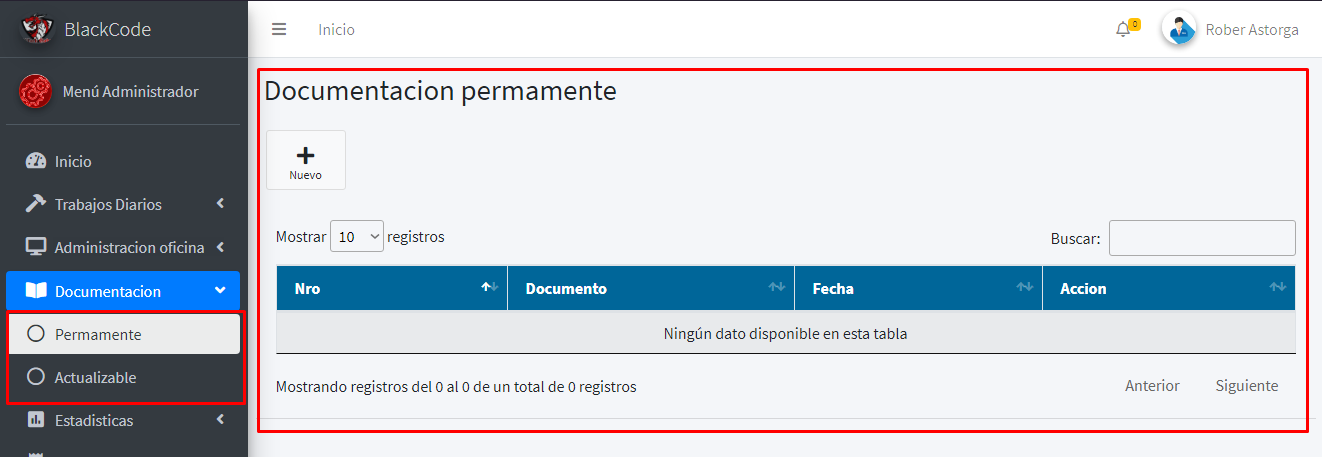
**Descripción de las acciones realizadas**

*La actividad desarrollada contempla las siguientes tecnologías web:*

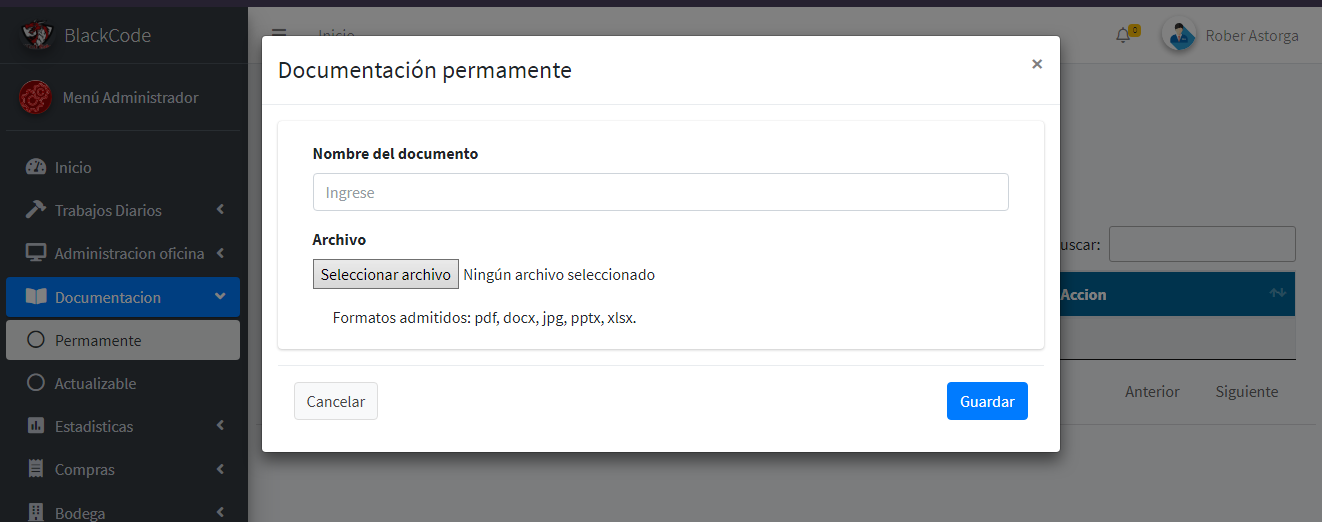
* *Lenguaje de programación: PHP y JavaScript.*
* *Framework codeigniter MVC.*
* *En cuanto a diseño se utiliza la plantilla de adminlte3 que usa el framework de diseño Bootstrap 3.*
* *Para el registro de los datos se utiliza el motor base de datos MYSQL.*

*La actividad comienza analizando el requerimiento solicitado y este es tener un apartado para poder subir documentación permanente y actualizable, esta documentación está asociada a los proyectos con que la empresa se involucra, por lo tanto, son de suma importancia. Un punto clave del requerimiento es que, si se sube un documento del tipo actualizable, el usuario debe indicar una fecha límite, entonces llegada a esa fecha limite con un máximo de 3 días de anterioridad, el sistema debe notificar al usuario que el documento está por vencer o expirar, para que así el usuario deba actualizarlo.*

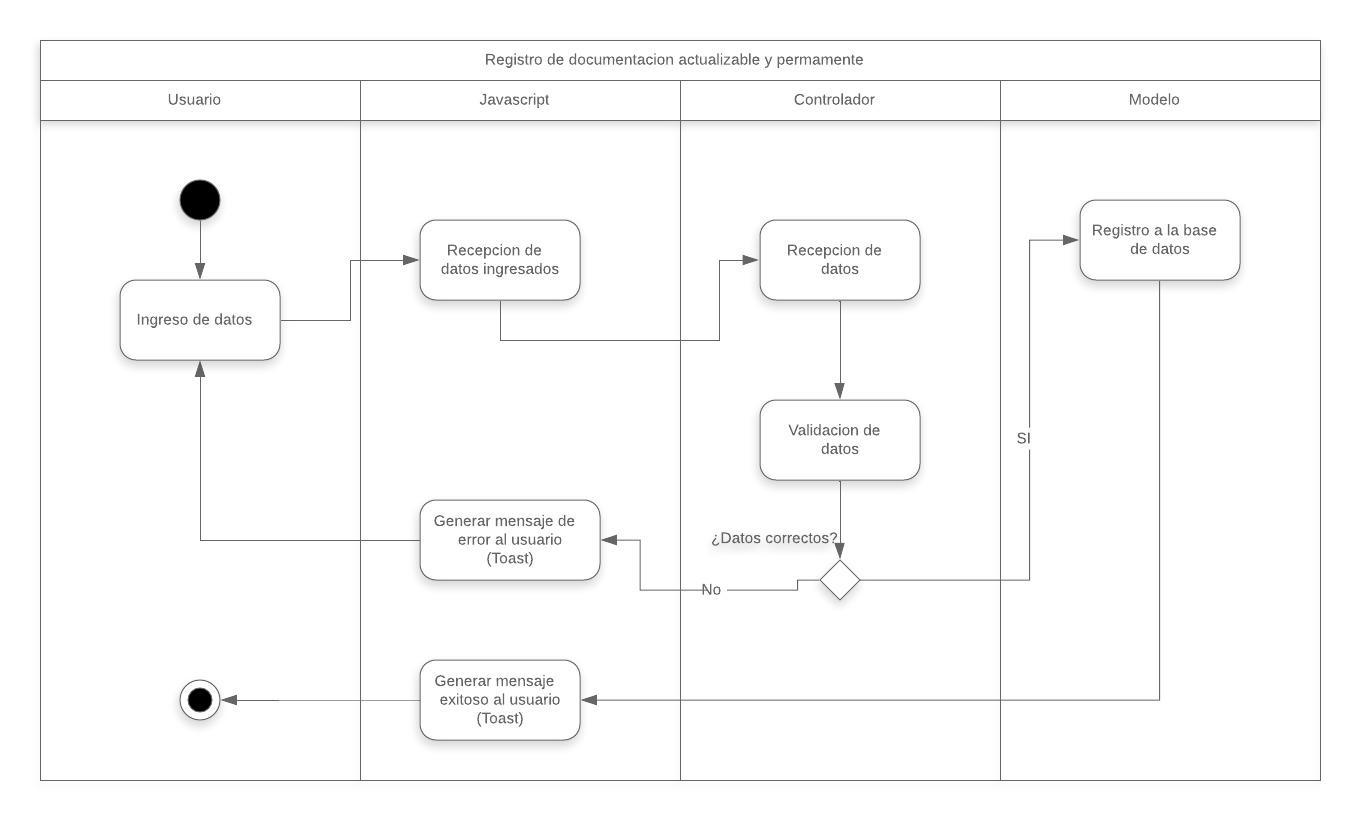
*Una vez teniendo el requerimiento claro, se procedió a diseñar la vista o interfaz donde ira la opción para subir los documentos y esta se encuentra en el menú de administrador, como se muestra en la siguiente imagen:*

**

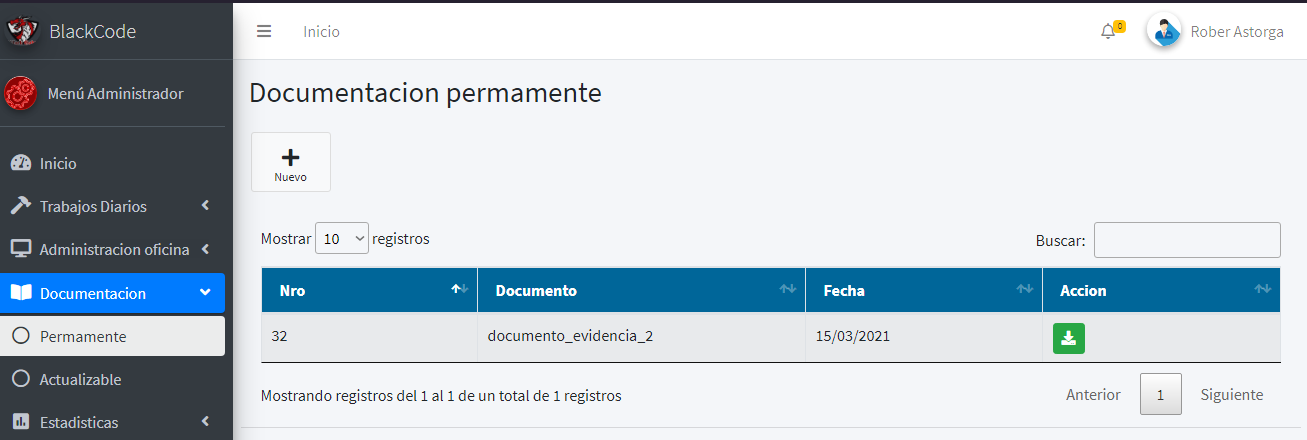
*Para la subida de documentos en cuanto a documentos permanentes, el usuario debe realizar un clic en el botón de “Nuevo” para poder registrar un nuevo documento, al hacer clic, este lo captura la vista y se encarga de abrir un modal para mostrar lo siguiente:*

**

*En este modal el usuario debe ingresar el nombre del documento, este es obligatorio ya que permite identificar el documento subido y también debe subir el documento como tal, haciendo clic en seleccionar archivo, una vez completados estos campos al presionar el botón de guardar, son capturados por un archivo de JavaScript que luego envía los datos a un controlador para que sean validados y si todo va bien, los datos son enviados hacia el modelo para poder ser registrados en la base de datos, en la base de datos se registra solo el nombre del documento y el archivo es subido a la raíz del sistema. Este proceso se puede ver reflejado de la siguiente forma.*

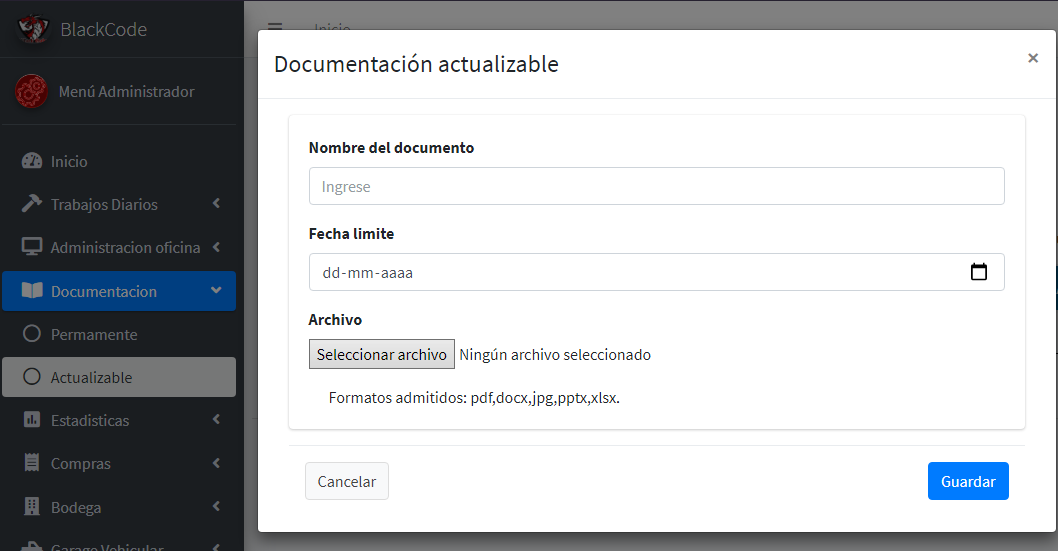


*Una vez subido el documento con el nombre, el registro quedo reflejado en la siguiente imagen:*

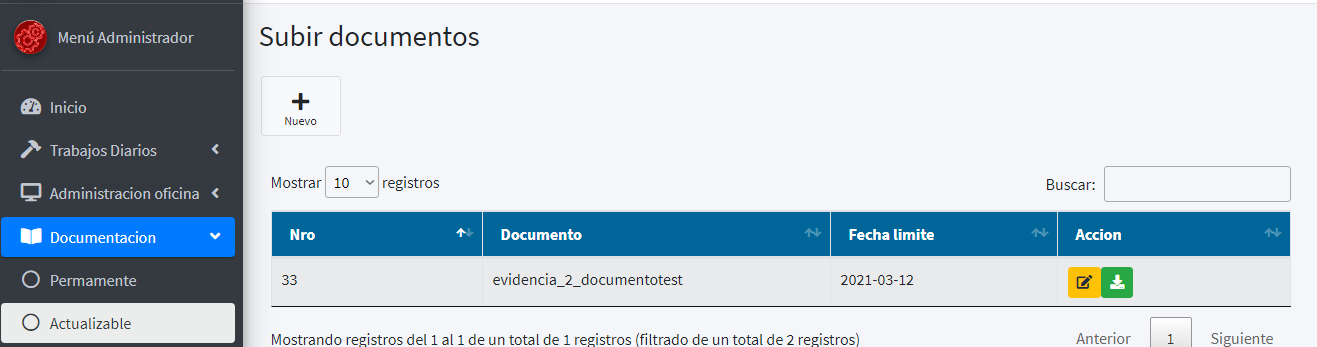
**

*Una vez subido el documento, el usuario puede clicar el botón de descargar y se comenzaría a descargar dicho documento. Para realizar esto primero se captura el clic dado por el usuario en el botón de descargar mediante el archivo de JavaScript y luego se realiza la petición al controlador y entonces se realiza la consulta a la base de datos y si el documento se encuentra, se procede la descarga del documento.*

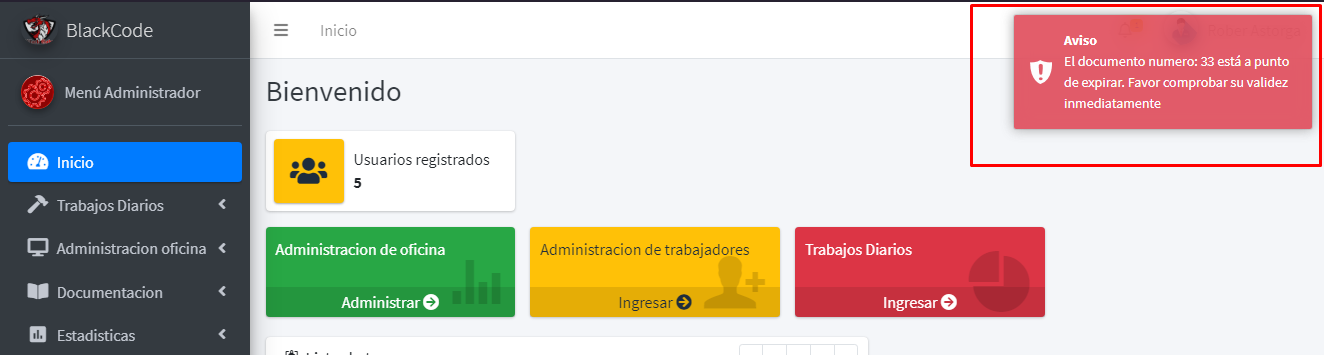
*Para la documentación actualizable el proceso es el mismo, la diferencia de esta parte es que la documentación actualizable lleva un campo adicional, el cual es la fecha límite.*

**

*Entonces al subir el documento, se ve reflejado de la siguiente manera en el datatable:*

**

*Para este ejemplo se utilizó una fecha con 3 días de anterioridad a la fecha de subida del documento para que notifique que este documento necesita ser actualizado, entonces cuando el usuario inicie sesión nuevamente al sistema, se le notificara mediante un mensaje Toast generado mediante un archivo JavaScript que el documento numero 33 está por vencer.*



*Esta notificación saldrá siempre y cuando el documento no haya sido actualizado, este punto era clave en el requerimiento. Así el usuario o administrador podrá ingresar al sistema y verá reflejado si existe un documento que se debe actualizar.*

**Conclusiones**

La actividad fue llevaba a cabo con éxito, el elemento más complejo a realizar fue la notificación al usuario en caso de que un documento estuviese pronto a expirar, pero finalmente se logró realizarlo. El resultado fue el esperado, el usuario sube su documentación con fecha y queda el registro en el sistema, también puede buscar un documento indicando el nombre en la lista donde quedan todos los registros.

**EVIDENCIA 3**

|  |  |
| --- | --- |
| **ALUMNO:** | *Rober Alonzo Astorga Berna* |
| **FECHA:** | *04/01/2021* |
| **ACTIVIDAD:** | *Registro planilla de trabajos diarios por parte de trabajadores en terreno* |

**Descripción del requerimiento**

La problemática es que, se requiere digitalizar el registro del trabajo que realizan los trabajadores diariamente para que estos sean visualizados por la administración de manera instantánea, ya que en la actualidad este registro se realiza de forma manual realizando el registro en una hoja y escribiendo la información con lápiz y luego terminada la jornada laboral recién es vista la información por administración. En ocasiones esto no es así ya sea por inconveniente o por diferentes razones, lo que provoca que se acumule la información y dificulte la corroboración por administración.

**Entregable**

El entregable solicitado era que se necesitaba que un trabajador en terreno mediante su dispositivo móvil ingrese al sistema y pueda realizar el registro de la planilla del trabajo diario, esta planilla contiene datos de servicio como de gastos. El punto clave solicitado es que el sistema debe ser responsive al dispositivo utilizado, es decir que se debe adaptar a las dimensiones del dispositivo y debe mostrar claramente los componentes para que se puedan utilizar de manera correcta.

**Fechas**

*La actividad para desarrollar abarca desde el 19 – 26 de enero aproximadamente.*

**Descripción de las acciones realizadas**

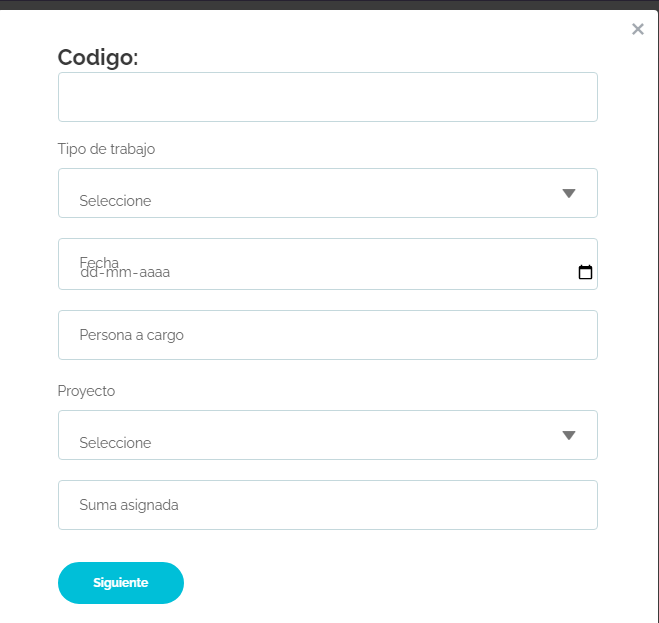
*La actividad desarrollada contempla las siguientes tecnologías web:*

* *Lenguaje de programación: PHP y JavaScript.*
* *Framework codeigniter MVC.*
* *Diseño se utilizó la plantilla de adminlte3 que usa el framework de diseño Bootstrap 3.*
* *Para el registro de los datos se utiliza el motor base de datos MYSQL.*

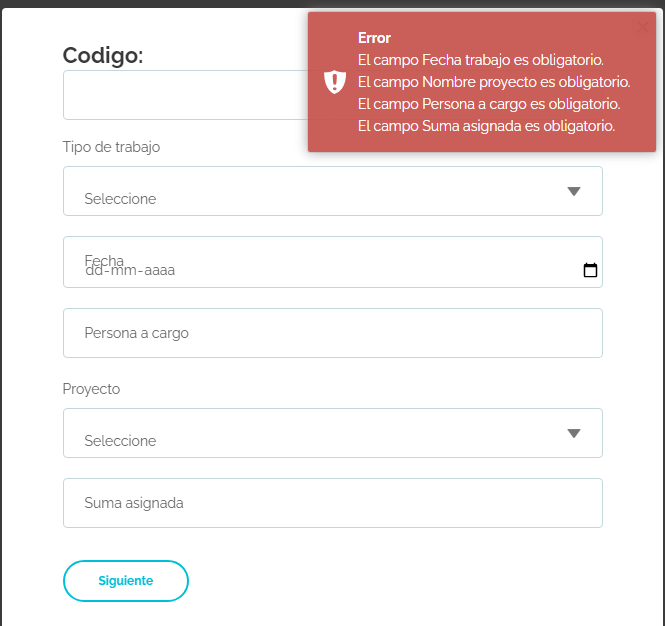
La actividad comienza analizando el requerimiento solicitado y este es tener un apartado para realizar el registro de una planilla en el sistema, esta planilla contiene los siguientes datos:

* Monto asignado
* Proyecto
* Fecha
* Personal a cargo
* Personal que asiste
* Tipo de trabajo
* Gasto de viáticos
* Gasto de materiales
* Gastos extras
* Subida de documentos
* Subida de imágenes
* Detalle de trabajos

Para realizar esta actividad, se implementó una plantilla que se adapta a cualquier dispositivo utilizado y además tiene los componentes necesarios para cada dato que se requiere, solo bastaba con editar la plantilla y adaptarla al requerimiento. Esta plantilla comienza pidiendo los siguientes datos:

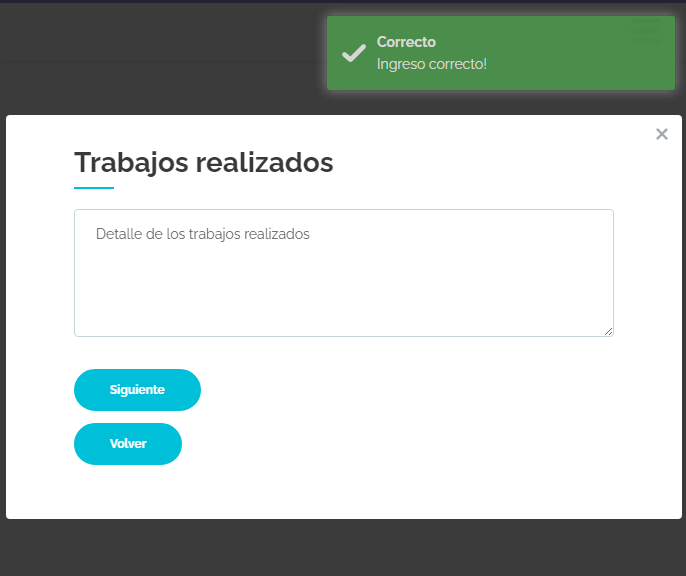


Estos datos son ingresados por el usuario y mediante un archivo JavaScript son enviados al controlador para poder ser validados, en caso de que ocurra algún error, se le notifica mediante una notificación Toast al usuario, indicando el error como se muestra en la siguiente imagen:



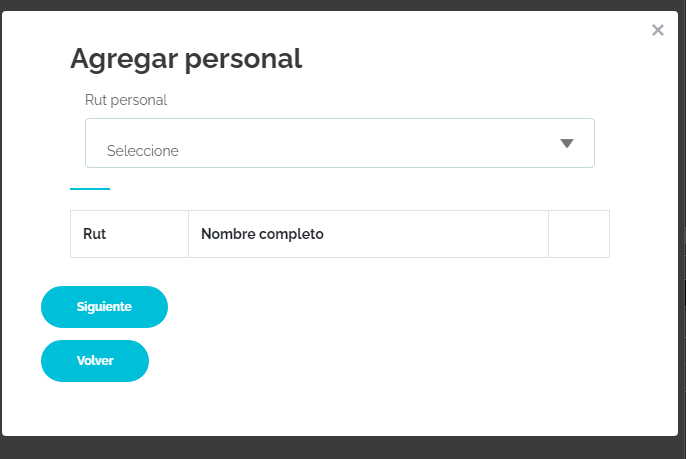
De esta manera el sistema se asegura que el ingreso sea el correcto y no se produzcan errores futuros por el mal ingreso de los datos.

En caso de que el ingreso sea el correcto, sucede lo siguiente:

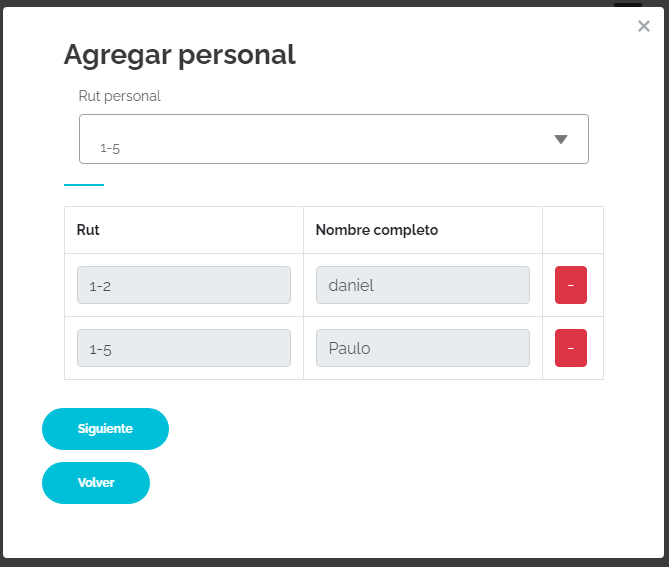


Se cambia a la siguiente vista, es decir que se abre otro modal con los siguientes datos a ingresa y se notifica mediante un Toast al usuario que el ingreso anterior fue correcto.

Entonces, se procede a ingresar el detalle y pasa al siguiente modal.

En este nuevo modal se debe registrar el personal que asistió al trabajo, mediante un select que el usuario debe seleccionar el rut ya previamente ingresado en el sistema y debe seleccionar el personal.

Una vez que el usuario selecciono el rut en el select, se captura el evento mediante el archivo JavaScript y se carga una tabla donde aparece el rut y el nombre de personal seleccionado. Esto queda de la siguiente manera:

también da la opción de poder eliminar un registro, haciendo clic en el botón rojo, se realiza la petición al archivo JavaScript y este se encarga de eliminar este elemento de la vista. El lenguaje de JavaScript en esta actividad es de suma importancia.

Luego el usuario debe clicar en siguiente y el sistema genera la vista con todos los datos pedidos en el requerimiento inicial.

* Se adjunta anexo llamado “video\_1\_(3)” donde muestra la vista generada por el sistema.

En esta vista general, el trabajador puede ingresar los siguientes datos:

* Asistencia personal
* Gasto de viáticos
* Gasto de materiales
* Materiales utilizados de bodega
* Gasto de combustible
* Gastos varios
* Subir fotografías de que comprueben el trabajo realizado

*En cuanto a la vista de registrar la asistencia del personal.*

* *Se adjunta anexo llamado “video\_2\_(3)” donde muestra la vista de la asistencia del personal.*

*En a esta vista de asistencia, al registrar una asistencia el sistema realiza el cálculo de horas extras, de acuerdo con el horario de entrada y salida de cada personal, también se agrega un campo adicional que es por qué cambio la hora, es decir un detalle. Este último punto también era importante en el requerimiento de la actividad. Una vez ingresado la asistencia se realiza el registro en la base de datos.*

*Todos estos datos son capturados por un archivo JavaScript y son enviados al controlador para su posterior validación y envió al modelo para su registro en la base de datos*

*Una vez que los datos son ingresados, la plantilla también permite la modificación de estos modificando el botón que antes era de “ingresar” ahora es “modificar”.*

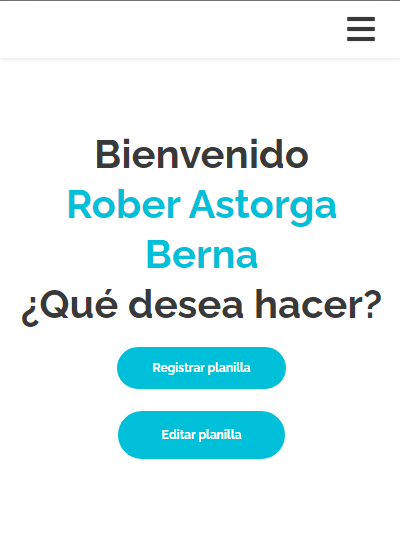
* *Se adjunta anexo llamado “video\_3\_(3)” donde muestra la planilla con los datos ingresados y que permite la modificación de estos, cabe destacar que no es necesario que se ingresen todos los datos para que se puedan modificar, a medida que se ingresen datos el sistema permite la modificación de este.*

*Por último, una vez registrada la planilla, se muestra la siguiente confirmación con el detalle del balance:*

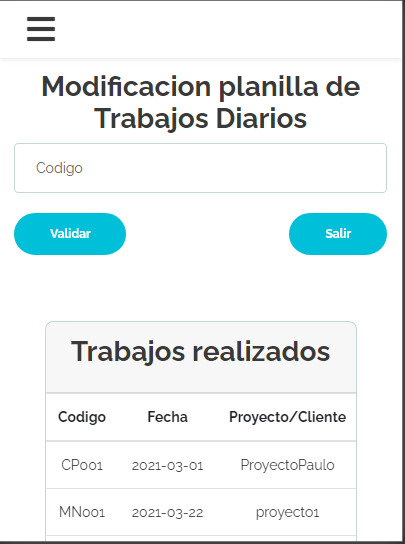


*De esta manera queda guardada la planilla de un trabajo diario, y es inmediatamente notificada al administrador para su posterior corroboración.*

*también en caso de que se requiera, el sistema también permite la modificación de una planilla, haciendo clic en editar planilla:*



*Muestra la siguiente vista, donde se debe ingresar el código de servicio y listo, ya puedes modificar la planilla.*

*En la lista de abajo se muestran los trabajos previamente registrados, estos van asociados a la cuenta ingresada en el sistema, es decir a la sesión del sistema. Otro usuario tiene otros trabajos realizados.*

*Si se ingresa el código, el usuario debe clicar en validar y el sistema genera la vista para poder modificar la planilla.*

* *Se adjunta anexo llamado “video\_4\_(3)” donde muestra la vista de la modificación de una planilla en específico.*

**Conclusiones**

*La actividad fue realizada con éxito, se logró realizar una plantilla los más interactiva para el usuario que en este caso es un trabajador en terreno. Lo más complejo fue toda la interacción que tiene esta plantilla, es decir la cantidad de datos que involucra y que estos sean correctamente formateados para su posterior visualización por administración. Las mejoras serán posibles una vez que sea utilizado por los trabajadores, ellos mismo pueden sugerir mejoras ya que son ellos los que van a utilizar esta plantilla y son los únicos capaces de sugerir algunos cambios que este módulo pueda tener.*

**EVIDENCIA 4**

|  |  |
| --- | --- |
| **ALUMNO:** | *Rober Alonzo Astorga Berna* |
| **FECHA:** | *04/05/2021* |
| **ACTIVIDAD:** | *Permitir la visualización de trabajos diarios por parte de la administración* |

**Descripción del requerimiento**

*La problemática es que, se requiere la visualización de los trabajos diarios por parte de la administración de forma inmediata ya que actualmente esta información puede durar horas o días en llegar a la administración, este requerimiento va en función a la actividad anterior, es decir que depende de los trabajos registrados por los trabajadores en terreno para que este registro sea visualizado por la administración.*

**Entregable**

*El entregable solicitado es un apartado en el sistema donde se visualice el registro de los trabajos diarios en el sistema, esto junto a todas sus variables involucradas.*

**Fechas**

*La actividad desarrollada abarca desde el 27 enero hasta el 03 de febrero.*

**Descripción de las acciones realizadas**

*La actividad desarrollada contempla las siguientes tecnologías web:*

* *Lenguaje de programación: PHP y JavaScript.*
* *Framework codeigniter MVC.*
* *En cuanto a diseño se utilizó la plantilla de adminlte3 que usa el framework de diseño Bootstrap 3.*
* *Para el registro de los datos se utiliza el motor base de datos MYSQL.*

*La actividad comienza analizando el requerimiento solicitado y este es tener un apartado para revisar los trabajos diarios realizados por los trabajadores en terreno.*

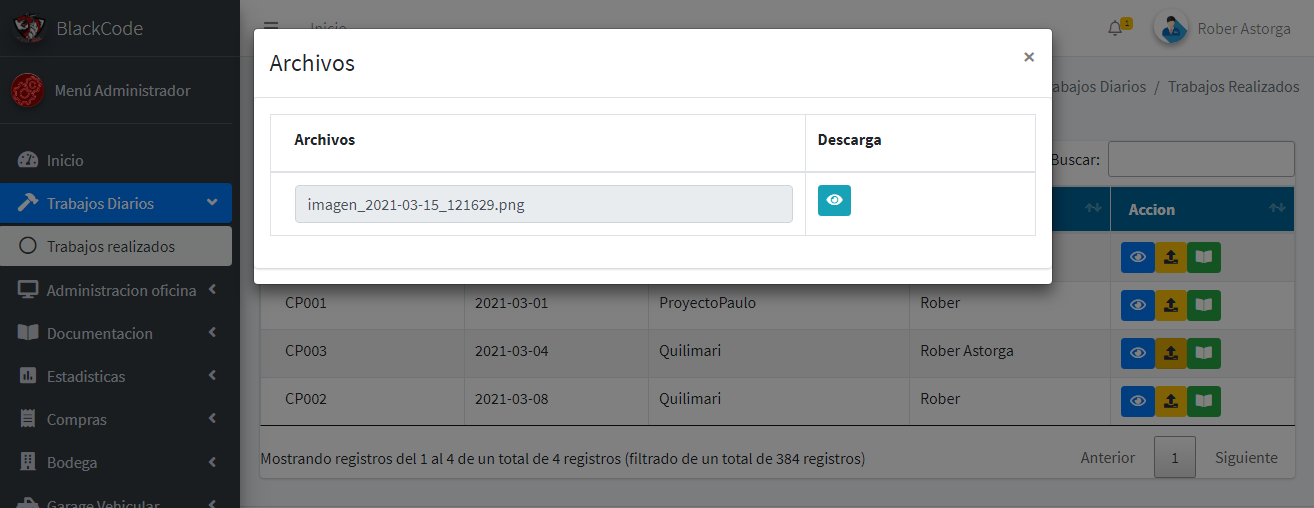
*Esto se logra mediante archivos de JavaScript y una librería llamada DataTable que permite la visualización de los trabajos como se muestra en la siguiente imagen:*



*Esta tabla contiene 3 botones, el primero es para ver el detalle del trabajo realizado, mostrando los gastos involucrados y el detalle en sí, esto se logra mediante el archivo JavaScript que captura el clic por el usuario y abre un modal con los datos que vienen desde el controlador, producto de una consulta hacia la base de datos por medio del modelo.*

* Se adjunta anexo llamado “video\_1\_(4)” donde muestra el detalle del trabajo realizado.

*El segundo botón es para ver los archivos subidos al sistema, esto igual se realiza mediante un archivo JavaScript que captura el clic realizado por el usuario y realiza una petición al controlador para luego hacer la petición al modelo sobre el nombre de los archivos y así devolver la petición al controlador y el controlador genera la respuesta al archivo JavaScript para generar la siguiente tabla:*



*El tercer botón es para ver la asistencia del personal, junto a su horario de entrada, salida, horas extras y una observación.*

* *Se adjunta anexo llamado “video\_2\_(4)” donde muestra el personal asistido.*

**Conclusiones**

*La actividad fue realizada con éxito, con esta actividad se logra digitalizar el ingreso de información por parte de trabajadores y su visualización por parte del personal administrativo, también se logra una mejor optimización de los tiempos ya que antes este proceso era manual y se tardaba horas o días incluso antes de que la información sea visualizada por la administración, de esta forma se espera que la empresa optimice sus tiempos y obtenga mejores resultados. Fue una actividad simple ya que más bien se realizaba solo consultas a la base de datos y estos son mostrados en la vista del usuario*