

Ciclo 01 2023



UNIVERSIDAD DON BOSCO

Asignatura: Desarrollo de Aplic. Web con Soft. Interpret. en el Servidor

Evaluación de proyecto Catedra

Caso de estudio.

El **Banco de Agricultura Salvadoreño SA de CV**, ha solicitado a usted y su grupo de desarrolladores la creación de un sistema web que le permita organizar de mejor manera las actividades de las áreas internas de la empresa. Para lo cual se le detallará a continuación los diferentes roles de usuario y las acciones permitidas a realizar en el sistema:

Cliente o prestamista: Persona la cual deberá registrarse en el sistema y validar su usuario vía correo electrónico. Una vez ha validado su usuario, podrá crear un máximo de 3 cuentas bancarias las cuales tendrán un identificador numérico generado de forma automática y aleatoria (aconsejable usar función random php), También podrá ver la lista de movimientos realizados en sus cuentas bancarias, tanto depósitos como retiros y transferencias de dinero a otros usuarios del mismo banco.

Dependiente del banco: Tipo de usuario especial; dueño de cualquier comercio (farmacias, gasolineras, tiendas) el cual podrá dirigirse al banco y solicitar la creación de un usuario, para poder ingresar al sitio y brindar servicios especiales sin necesidad de ir al banco:

- Ingreso y retiro de efectivo a una cuenta de un usuario del banco, para lo cual por medio de un formulario que solicite el número de DUI, podrá ver todas las cuentas asignadas al cliente y poner abonarle y retirarle el efectivo.

Esto mediante la creación de un servicio web, **especificación al final del caso de estudio.**

Cajero: Empleado del banco el cual podrá realizar.

- Ingresar nuevos clientes o prestamistas al banco (función específica para usuarios sin acceso a internet o busquen una asesoría al respecto).
- Agregar dependientes del banco.
- Ingresar o retirar dinero a una cuenta de algún prestamista (deberá validar que el DUI y la cuenta bancaria estén relacionadas, de lo contrario deberá dar error en la transacción).
- Apertura un préstamo a un cliente el cual se debe validar de la siguiente manera:
 - Para empleados con salarios menores a un sueldo de \$365 el préstamo máximo que se puede hacer es de \$10,000 con un interés del 3%.
 - Para empleados con salarios menores a un sueldo de \$600 el préstamo máximo que se puede hacer es de \$25,000 con un interés del 3%.
 - Para empleados con salarios menores a un sueldo de \$900 el préstamo máximo que se puede hacer es de \$35,000 con un interés del 4%.
 - Para empleados con salarios mayores a un sueldo de \$1,000 el préstamo máximo que se puede hacer es de \$50,000 con un interés del 5%.

NOTA:

- Para cada préstamo deberá apertura un caso en el cual su estado puede ser: en espera, aprobado o rechazado
- La cuota mensual a pagar no debe superar el 30% del salario del cliente.
- Con base a lo anterior debe calcular la cantidad de años en la que se pagara el total del préstamo.

Gerente de sucursal: Persona encargada de las actividades administrativas de la sucursal del banco, el cual podrá:

- Ingresar nuevos empleados: Cajeros del banco, personal de limpieza, secretarias o recepcionistas y personal de mesa (asesores financieros), a la sucursal, para lo cual deberá de asignarles una acción de personal que será aceptada en siguiente rol y con lo cual su estado quedara **en espera de aprobación**.
- Dar de baja a un empleado, esto sin eliminarlo de la base de datos cambiando su estado de activo a inactivo.
- Ver lista de casos de préstamos abiertos por los cajeros y aprobar o rechazar un credito.

Gerente general del banco: Persona encargada de aceptar o rechazar las acciones de personal generadas por el gerente de sucursal. También podrá crear el registro de una nueva sucursal del banco y asignar a ella un nuevo gerente de sucursal (esto en caso se desee expandir el banco). También podrá ver todos los movimientos de las cuentas creadas en las sucursales y las cuentas independientes de sucursal.

Web services con laravel.

Finalmente deberá crear web services con laravel, el cual pueda ser consumido por un dependiente del banco en php (sitio oficial), esta sección deberá cumplir con los siguientes puntos:

- Deberá recibir ya sea por post o por get el número de DUI de la persona, con lo cual se deberá retornar al cliente un JSON con la lista de cuentas relacionadas a este número.
- El módulo del dependiente del banco deberá consumir el JSON enviado a la página principal y pintar en una vista de forma creativa la lista de cuentas asignadas.
- Finalmente, en un formulario solicitar el monto a ingresar o retirar a una cuenta en específico, y usando Eloquent con Laravel abonar o retirar a dicha cuenta el monto y añadir el registro del movimiento realizado.

Fases del proyecto.

Fase I (Primer avance de proyecto 10%).

Deberá entregar un documento con la siguiente estructura:

- Portada
- Índice
- Introducción
- Cuerpo del trabajo
 - o Mockups (Vistas por rol del proyecto) (Usar un software profesional de maquetado, NO PAINT).
 - o Diagrama relacional de base de datos.
 - o Enlace al repositorio de GitHub (una rama por integrante).
 - o Enlace al Trello del proyecto, con las actividades programadas para todo el ciclo y sus asignaciones.
- Conclusión.

Fase II (Segundo avance de proyecto 10%).

Entrega del 30% del proyecto.

Criterio	Porcentaje
Entrega de por lo menos un módulo completo del sistema.	30%
Correcta conexión a la base de datos	10%
Uso de un estilo agradable para la vista	10%
Formulario de ingreso completo	10%
Formulario de registro completo	10%
Correcta implementación de validaciones tanto en el servidor como en el cliente	20%
Entrega puntual	10%

Fase III (Defensa final de proyecto 20%).

Actividad a evaluar	Criterio a evaluar	Cumplió		Puntaje
		%	Nota	
Proyecto web	Creación de sitio web bien diseñado, llamativo, ordenado, con uso equilibrado de colores. (2%)	2%		

	<p>Creación de sistema de navegación intuitivo mediante menús. Debe incorporar un menú principal y uno secundario (cuando aplique) con las opciones de menú extraídas desde una matriz o arreglo, incluyendo las subopciones que pudieran tenerse. (3%)</p>	3%		
	<p>Selección de fuentes incrustadas o de Google para mejorar la tipografía establecida por defecto en los navegadores. (2%)</p>	2%		
	<p>Visualización correcta en todos los navegadores y todos los dispositivos de diferentes dimensiones (computadoras de escritorio, laptops, tablets y smartphones). Es decir, creado bajo directrices de diseño web adaptable a dispositivo. (3%)</p>	3%		

	<p>Las páginas del sitio web deben tener diseño uniforme y a parte de la página de inicio deben existir páginas para cada una de las opciones de primer nivel del menú principal y secundario. Esto puede diferir en el caso de los sitios que trabajen con Landing Pages en donde se requerirá que todas las secciones estén bien acabadas. (5%)</p>	5%		
	<p>Correcta implementación del cliente con Laravel, se envía la data de forma correcta y se recibe el JSON, se consume para su posterior pintado. Se usa Eloquent para hacer el update al monto y guardar el registro del movimiento</p>	20%		

	<p>Reutilización de código haciendo uso de funciones bien diseñadas, lo cual quiere decir que en su mayoría las funciones deben recibir uno o más argumentos (por valor, por referencia y/o con valores por defecto), devolver algún tipo de valor (escalares y asociativas), implementación de funciones con lista variable de argumentos o funciones variables.</p> <p>(15%)</p>	15%		
	<p>Desarrollo de formularios de captura de información desde la parte administrativa del sitio web que deberá manipular en el script PHP que reciba los datos haciendo uso de las matrices superglobales de PHP (\$_SERVER['REQUEST_METHOD'] para validar envío de formulario y \$_POST o \$_GET para obtener valores con las validaciones de rigor en el cliente y en el servidor antes de asignarlos y utilizarlos), dependiendo del método de envío. (5%)</p>	5%		

	<p>Mostrar los datos obtenidos a través de las matrices superglobales \$_POST o \$_GET en una página web con los datos capturados de una forma ordenada y adecuada para su correcta visualización. En caso de existir errores indicarlos de explícitamente en cada dato del formulario y mostrar un enlace que permita volver a ingresar la información. (10%)</p>	10%		
	<p>Organización: Disposición en distintas carpetas de los diferentes tipos de recursos utilizados para construir el sitio web, como imágenes, vídeos, hojas de estilo, scripts js, fuentes, etc. (5%)</p>	5%		

	<p>Estructura del contenido de las páginas: (10%) - Todas las páginas se deben verse ordenadas y cumpliendo con la estructura de HTML5 y sus estándares, - Las hojas de estilo están bien organizadas, de modo que se agrupan las reglas en base a su función y a los elementos del documento que afectan. - El código PHP utiliza estructuras de control y se ajusta a buenas prácticas de programación.</p>	10%		
	<p>Funcionalidad: (20%) - Los programas y secuencias de comando PHP que se muestran durante la defensa realizan un proceso útil y que produce los resultados esperados de acuerdo con los requerimientos del tema que se ha desarrollado. - Se muestran las páginas generadas dinámicamente sin errores de funcionamiento y sin mensajes de error (fatal errors), de advertencia (warnings) o de notificación (notice).</p>	20%		
	Promedio	100%		10