SnowBall Finance

Sotomayor Vargas Angelo, [asotomayor70969@ufide.ac.cr](mailto:asotomayor70969@ufide.ac.cr)

Guilder W. Milliner, gmilliner70418@ufide.ac.cr

Glennmark Milliner Whitaker, gmilliner70419@ufide.ac.cr

*Ingeniería en Sistemas de Computación, Universidad Fidélitas, San José, Costa Rica*

***Abstract*—The following project seeks to create a transactional website that can be implemented in real life for SMEs or any other similar entity by applying the knowledge and skills acquired through the course. All the tools and methodologies learned will be used, such as Java + SpringBoot, HTML, CSS, databases and the MVC model, among others, to create a solution that meets the needs of the proposal.**

**Keywords: WebSite.**

1. INTRODUCCIÓN.

El siguiente proyecto busca crear un sitio web transaccional que pueda ser implementado en la vida real por empresas PYMES o cualquier otra entidad similar mediante la aplicación de los conocimientos y habilidades adquiridas a través del curso. Se utilizarán todas las herramientas y metodologías aprendidas como Java + SpringBoot, HTML, CSS, bases de datos y modelo MVC entre otros que permitan crear una solución que cumpla la necesidad de la propuesta.

1. PROPUESTA DE PROYECTO

Según estudios realizados por los sectores bancarios públicos y privados, en los últimos 5 años se refleja un alarmante nivel de endeudamiento en la población costarricense de hasta un 101%, principalmente asociados a tarjetas de crédito. La población más afectada corresponde a funcionarios públicos, pequeñas empresas, jóvenes y mujeres.

A partir de la problemática planteada, se propone desarrollar un sitio web que permita generar acción a corto y mediano plazo mediante la orientación y capacitación de los usuarios que ayuden a tomar las mejores decisiones de consumo.

La temática principal del sitio se basará en el “método de la bola de nieve” el cual, es una estrategia propuesta por Dave Ramsey, autor de la “Transformación Total de su Dinero”, para eliminar deudas económicas personales.

En pocas palabras, el método trata de liquidar lo más rápido posible el préstamo o la deuda más pequeña que se tenga activa mientras se van aportando pagos mínimos al resto de deudas. Una vez que haya pagado esa deuda, podrá destinar el dinero extra que usaba para ese pago para la siguiente deuda más pequeña. El método está diseñado para salir de deudas de una forma más rápida que lo que se tardaría continuando con el pago de los créditos normalmente.

También los usuarios podrán suscribirse a programas de educación financiera de paga en los cuales tendrán acceso (dependiendo del plan comprado) a material, herramientas, planes de acción y contacto directo con profesionales en el área.

1. HISTORIAS DE USUARIO

A continuación, se presenta un primer acercamiento de los requerimientos del sitio web en cuestión, mediante historias de usuario:

1. *Us-01*

Cómo usuario necesito poder visualizar la información de forma clara y precisa.

Condiciones de Satisfacción:

* El sitio web deberá contener la información siempre organizada por secciones o categorías.

1. *Us-02*

Cómo usuario necesito poder navegar a través de la aplicación por medio de un menú de opciones o enlaces.

Condiciones de Satisfacción:

* El sistema deberá contar con un menú que sea visible en todo momento independientemente la página donde se encuentre el usuario.
* El menú deberá estar diseñado de tal forma que organice la información por categorías y en concordancia con la temática del sitio.
* El menú se presentará siempre en la cabecera de la página.

1. *Us-03*

Como usuario necesito visualizar los planes de asesoría financiera disponibles.

Condiciones de Satisfacción:

* La información será desplegada en cuadrículas para facilitar la visualización al cliente o usuario.
* Cada cuadrícula mostrará un detalle del plan seleccionado, con los beneficios que incluye y su precio, además de un botón para comprar.

1. *Us-04*

Como usuario necesito confirmar la compra del producto seleccionado.

Condiciones de Satisfacción:

* Se mostrará un formulario donde se cargarán todos los datos del usuario en sesión, además solicitará el método de pago y la dirección.
* El sitio deberá mostrar una ventana de confirmación de compra.
* Una vez realizada la compra, el sitio le mostrará un mensaje de transacción exitosa y generará el documento de la factura.

1. *Us-05*

Como usuario necesito poder hacer uso de las calculadoras de deudas que el sitio facilita.

Condiciones de Satisfacción:

* Las calculadoras siempre estarán disponibles para cualquier usuario independientemente si es un usuario registrado o no ya que es un servicio gratuito del sitio.
* El usuario deberá ingresar la información tal como se solicita, llenando todos los espacios del formulario para finalmente hacer click en el botón “Calcular”.
* La calculadora luego de unos segundos mostrará los resultados y de ser el caso recomendará al usuario algunos de los planes de asistencia financiera que dispone el sitio.

1. *Us-06*

Como usuario necesito registrar los datos personales para poder generar mis credenciales y poder acceder al sitio web.

Condiciones de Satisfacción:

* Los datos personales que se necesitan a la hora del registro son los siguientes: nombre, apellido, correo electrónico, teléfono y contraseña.
* El usuario no podrá registrarse si no se ingresan todos los datos personales.

1. *Us-07*

Como usuario necesito poder ingresar al sitio haciendo log in con mis credenciales.

Condiciones de Satisfacción:

* El usuario debe de poder ingresar su correo electrónico y contraseña para poder hacer log in en el sitio.
* Si no se ingresan alguno o ninguno de los datos al sitio (correo electrónico y contraseña) el mismo le indicara al usuario que dichos campos son obligatorios.
* El sitio deberá brindar la opción al usuario de poder iniciar con cuentas de Google o Facebook.

1. *Us-08*

Como usuario necesito poder cerrar la sesión del sitio web de forma segura.

Condiciones de Satisfacción:

* El sitio debe habilitar un botón que le permita al usuario finalizar la sesión con el servidor.

1. *Us-09*

Cómo administrador necesito que sitio se desarrolle bajo el modelo de arquitectura de software MVC para cumplir con las buenas prácticas de programación e implementación.

Condiciones de Satisfacción:

* Como lo indica el modelo, se deben separar los datos, principalmente lo que es la lógica de negocio de su representación, e implementar el módulo encargado de gestionar los eventos y las comunicaciones.
* El usuario podrá modificar una solicitud existente, deberá seleccionar el nuevo periodo y guardar el cambio. Una vez finalizado el proceso, el sistema le confirmará que los datos nuevos han sido guardados exitosamente mediante una ventana emergente.

1. *Us-10*

Como administrador requiero que el sitio web tenga un diseño atractivo y de percepción amigable para el usuario, que facilite su navegación y que el nombre y logo de la empresa siempre sea visible e identificable.

Condiciones de Satisfacción:

* El sitio contará deberá usar fuentes de letra acorde a la sección del sitio, donde dependiendo del contenido el tamaño de la fuente puede variar para resaltar información importante.
* El uso de colores se utilizará de forma sutil y debe seguir un patrón de paleta seleccionada ya previamente según el diseño deseado.

1. *Us-11*

Cómo administrador requiero que el sitio cuente con una página principal o inicial.

Condiciones de Satisfacción:

* La página debe ser alineada con la imagen corporativa.
* Debe tener un diseño atractivo, que llame la atención.
* Debe brindarle al usuario datos muy precisos sobre el propósito de la empresa y los servicios que brinda.

1. *Us-12*

Cómo administrador requiero que el sitio cuente con una página de “Acerca de nosotros” o “About”.

Condiciones de Satisfacción:

* La página mostrará información de contacto, como correo electrónico, redes sociales y ubicación física en mapa.
* También brindará al usuario una opción para llenar un formulario de sugerencia.

1. *Us-13*

Cómo administrador requiero que el sitio cuente con una sección de blog o comentarios para que los usuarios puedan compartir experiencias.

Condiciones de Satisfacción:

* Solamente podrán comentar los usuarios con cuenta.
* Para cada comentario se registrará el nombre del usuario y un campo de texto amplio.
* Un usuario podrá responder a otro generando una cadena de comentarios.

1. *Us-14*

Como administrador se necesita garantizar que los usuarios elijan una contraseña robusta con el fin de resguardar sus datos de una forma segura.

Condiciones de Satisfacción:

* En caso de que no se ingrese ningún valor de contraseña se debe indicar que el campo es requerido.
* La contraseña debe de tener un mínimo de 8 caracteres y un máximo de 20
* La contraseña debe de tener al menos un carácter especial (#?!@$%^&\*-)
* La contraseña al menos debe de tener un número y una letra mayúscula y minúscula.
* El usuario deberá de digitar la contraseña dos veces en campos diferentes para garantizar que el usuario en verdad eligió bien la contraseña.
* Se deberán de mostrar mensajes de advertencia o errores que le hagan saber al usuario los parámetros que debe seguir la contraseña.

1. *Us-15*

Cómo administrador necesito poder gestionar, visualizar, agregar o eliminar los usuarios registrados en el sitio web.

Condiciones de Satisfacción:

* El sitio deberá habilitar un módulo “admin” para el administrador.
* La información que se mostrará de los usuarios será el nombre, apellido, teléfono y correo electrónico en formato de lista.
* Se podrá modificar cualquier dato por cada usuario.
* El administrador podrá eliminar usuarios, para lo cual la aplicación deberá solicitar una confirmación por medio de una ventana emergente para continuar, en caso de que el administrador acepte, el usuario se eliminará definitivamente.
* El administrador podrá agregar nuevos usuarios si lo desea, deberá realizar el mismo procedimiento ya descrito en las historias de usuario Us-06 y cumplir con los requisitos de contraseña mencionados en Us-13.

1. HERRAMIENTAS A UTILIZAR

Se utilizarán las siguientes herramientas para la implementación del proyecto:

* **Github:** Al ser un proyecto en grupo, se utilizará Github como herramienta colaborativa en la nube para alojar el código de la aplicación mediante un repositorio público el cual puede ser accesado mediante el siguiente enlace: <https://github.com/Guilderm/SC-403_SnowBallalculator_CapstoneProject.git>
* **Netbeans:** Para la programación del proyecto se usará el IDE utilizado en clase, Netbeans, el cuál es un entorno de desarrollo gratuito orientado principalmente al desarrollo de aplicaciones Java que cuenta con variedad de módulos para su extensión.
* **MySQL Workbench:** Se utilizará Workbench como herramienta para el diseño, administración, gestión y mantenimiento de la base de datos MySQL para la integración con el proyecto.
* **Heroku:** Plataforma PaaS (plataforma como servicio) que brinda servicios de computación en la nube basada en contenedores que soporta diversos lenguajes de programación. Heroku nos permite desplegar nuestra aplicación en la web gracias a la directa integración con el repositorio en Github, facilitando que los cambios realizados en el código en Git se vean reflejados inmediatamente en el proyecto final. Puede ver el despliegue de la aplicación en el siguiente enlace: <https://snowballalculator.herokuapp.com/>
* **HTML5:** Lenguaje de etiquetas de marcado de hipertexto utilizado para la estructuración y presentación de contenido en las páginas web.
* **CCS3:** Lenguaje que se utiliza en conjunto con HTML, se utiliza para definir y crear la presentación de un documento ya estructurado, es decir, permite generar un diseño visual de las páginas web e interfaces del usuario.
* **Java Spring Boot:** Módulo del Framework Spring que simplifica drásticamente el desarrollo de aplicaciones. Cuenta con muchas características como la inyección de dependencias y el uso de módulos completos como lo son Spring MV, Spring Security entre otros.
* **Boostrap:** Es un framework de CSS o bien una librería o conjunto de herramientas de código abierto para el diseño del front-end para sitios y aplicaciones web que se adaptan a cualquier dispositivo.

1. MODELO MVC

El Modelo Vista Controlador (MVC) es un patrón de diseño de arquitectura de software que separa los datos de una aplicación (la lógica del negocio) de la interfaz del usuario (representación), y el módulo de control (gestión de eventos y comunicaciones) en 3 componentes diferentes.

El modelo busca promover las buenas prácticas de programación, para llevar el código de manera formal así como una estructura ordenada y entendible.

MVC se usa principalmente en sistemas donde se requiere el uso de interfaces de usuario, orientadas al consumidor o usuario donde el objetivo es crear un portal web amigable.

A continuación, se detalla la función que constituyen los 3 componentes del modelo:

1. Modelo

El Modelo contiene una representación de los datos que maneja el sistema, su lógica de negocio. Se encarga de acceder a la capa de almacenamiento de datos.

1. Vista

La Vista, o interfaz de usuario, compone la información que se envía al cliente y los mecanismos interacción con éste. Representa los datos del modelo al usuario. La vista sabe cómo acceder a los datos del modelo, pero no sabe qué significa esta información o qué puede hacer el usuario para manipularla.

1. Controlador

El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

El flujo que sigue la interacción del usuario con el código es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario).
4. El modelo interactúa con la base de datos.
5. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista.
6. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Diagrama

Descripción generada automáticamente

Fig. 1 Modelo MVC

1. PROCESO DE CREACIÓN DEL SITIO
2. CONCLUSIÓN
3. REFERENCIAS