República Bolivariana de Venezuela

Ministerio del Poder Popular para la Educación Superior

Universidad Valle del Momboy

Facultad de Ingeniería

Escuela de Ingeniería de Computación

Carvajal, Edo. Trujillo

**Modelo CANVAS**

**Y Diseño de la aplicación**

Integrantes:

José Ferrer C.I.: 25.173.145

José Urbina C.I.: 18.458.980

Ángelo Tarascio C.I.: 22.624.401

Miguel Colmenares C.I.: 24.612.665

Desarrollo de Aplicaciones Móviles

Representando a: Vertex Caster Studios

Prof.: Jhonmer Bencomo

**1.- Modelo CANVAS:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Fecha:** 22-2-17 | **Propuesta:** Question-man | **Empresa:** Vertex Caster Studios. | | | **Creadores:** José Ferrer, José Urbina, Ángelo Tarascio, Edgar Tarascio, Miguel Colmenares | **Iteración:** 1 |
| **8.- Socios Clave:**  - Usuarios quienes serán los que crearán el contenido de la plataforma, y lo moderarán.  - Librerías tanto físicas como digitales, las cuales ofrecerán las obras que tengan en existencia para su publicitación. | **6.- Actividades clave:**  - Mantenimiento y desarrollo de la plataforma web.  - Publicidad.  - Administración.  - Revisión de cuestionarios.  - Corrección de cuestionarios.  - Gestión de usuarios. | | | **1.- Propuesta de valor:**  Desarrollar una aplicación web adaptable que le permita a una persona crear sus propios cuestionarios acerca de diferentes temas para luego repasarlos o bien hacer intentos de resolución de dichos cuestionarios mediante selección simple entre 4 opciones, el usuario puede ver, resolver, calificar y notificar los cuestionarios que otros usuarios han publicado. | **4.- Relación con los clientes:**  - Un usuario puede crear cuestionarios con varias preguntas acerca de un tema en particular, el usuario guarda el cuestionario, y puede verlo en cualquier dispositivo ya sea en modo de contenido para recurso de estudio, o en modo prueba para evaluar sus conocimientos.  - La empresa le ofrece al usuario la posibilidad de publicar su cuestionario a otros usuarios, y de promocionarlo en las búsquedas si así el usuario lo desea. | **2.- Segmento de Mercado:**  -Estudiantes de educación media y superior.  -Profesores de educación media y superior. |
| **7.- Recursos Clave:**  - Personal: Programador Web, Diseñador Web, Publicista, Docente especializado en Metodología del aprendizaje, Contador, Moderador, Gerente.  - Infraestructura: Plataforma Web, un local físico.  - Tecnología: Servidor, Dominio Web, Bases de datos, Telefonía.  - Capital: Computadores, Teléfonos, Equipo de Oficina y Bibliografía en Digital. | | | **3.- Canales:**  - Online, a través de la aplicación web adaptable a cualquier dispositivo que soporte el motor Webkit. |
| **9.- Estructura de costos:**  - Sueldos y salarios.  - Adquisición y mantenimiento de equipos tecnológicos.  - Servicios públicos e impuestos.  - Hosting de la plataforma.  - Publicidad en línea, y folletos para estudiantes.  Modelo de negocio orientado al valor | | | **5.- Fuente de Ingresos:**  - Promoción de cuestionarios: Los usuarios interesados podrán promocionar aquellos cuestionarios que consideren están muy completos ante otros usuarios que busquen cuestionarios similares.  - Publicidad de editoriales: Las tiendas que venden libros electrónicos y de texto podrán publicar obras que sean acordes a los cuestionarios que un usuario busque.  Se aceptarán pagos vía transferencias y depósitos tanto para las promociones como para la publicidad, que serán verificados antes de ser activados. | | | |

**2.- Llamadas de la API:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Función** | **Parámetros** | **Lógica** | **Respuesta** |
| 1 | Registrar un nuevo usuario | Información completa del nuevo usuario: Correo, alias, clave, nombre, y | Se envían los datos al servidor, si el correo es único en la tabla de usuarios se guarda el nuevo usuario en el modelo de los usuarios, sino, se avisa que el correo ya está usado y que se ingrese otra dirección de correo electrónico. | JSON con el mensaje de éxito si guarda sin fallos, o un mensaje de fallo si no logra guardar los datos por existencia, u otro fallo. |
| 2 | Identificar un usuario | Correo del usuario y su clave, combinadas y cifradas. | Se envía el cifrado al servidor, y se compara con cada una de las claves almacenadas en el modelo de usuarios, si hay coincidencia, manda una señal de aceptación que deja al usuario continuar en el espacio, sino, rechaza al usuario. | JSON con la aceptación si la cadena cifrada existe en un usuario, o mensaje de fallo si no lo encuentra. |
| 3 | Modificar los datos de un usuario | Correo y clave del usuario. | Envía al servidor la dirección de correo y la clave del usuario, se valida la coincidencia del usuario con los datos almacenados y se devuelven los datos del usuario, el usuario edita la información y la envía de vuelta, y el servidor modifica según corresponda en el modelo de los usuarios. | JSON con los datos del usuario cuando se inicia la operación, y luego un JSON con el mensaje de operación exitosa u operación fallida dependiendo del resultado. |
| 4 | Eliminación de un usuario | Correo del usuario | Envía la dirección de correo electrónico del usuario al servidor, se busca el correo en la tabla y se marca el usuario del modelo como borrado. | JSON con el mensaje de éxito si marca el usuario sin fallos, o un mensaje de fallo si no logra guardar los datos. |
| 5 | Reactivación de un usuario | Correo del usuario y su clave | Envía la dirección de correo electrónico del usuario al servidor, se busca el correo en la tabla y se marca el usuario del modelo como no borrado. | JSON con el mensaje de éxito si desmarca el usuario sin fallos, o un mensaje de fallo si no logra guardar los datos. |
| 6 | Búsqueda de un cuestionario | Texto dentro del nombre del cuestionario | Envía al servidor la cadena, ahí la convierte en tokens y compara cada token que no sea de 3 caracteres o menos con los términos contenidos en la frase de búsqueda con los nombres guardados en el modelo de Cuestionarios y luego se carga en un arreglo los identificadores y los nombres de los cuestionarios coincidentes, los alias de sus creadores, y la cantidad de preguntas que tiene el cuestionario. | JSON con el arreglo de los datos de las coincidencias de formularios. |
| 7 | Seleccionar un cuestionario de la búsqueda | ID del cuestionario seleccionado. | Se envía al servidor el ID del cuestionario, y crea el objeto que contendrá la información del cuestionario, la aplicación que pida los datos creará en local el formulario para su verificación y posterior almacenamiento en el modelo. | JSON con el nombre del cuestionario, el alias de su creador, las preguntas, y sus respectivas respuestas. |
| 8 | Reportar un cuestionario | ID del cuestionario a reportar. | Se envía al servidor el ID del cuestionario, busca el cuestionario, e incrementa el contador de reportes del cuestionario para que el moderador verifique los datos si el registro obtienen muchos reportes. | JSON con el mensaje de reporte exitoso si logra hacer el reporte, o mensaje de fallo si falla en el envío del reporte. |
| 9 | Reportar al usuario creador de un cuestionario | ID del cuestionario a reportar | Se envía al servidor el ID del cuestionario, busca el cuestionario, e incrementa el contador de reportes del cuestionario para que el mantenedor verifique la situación y tome acciones. | JSON con el mensaje de reporte exitoso si logra hacer el reporte, o mensaje de fallo si falla. |
| 15 | Registrar un pago para promocionar un cuestionario | ID del cuestionario, y datos del pago. | Se envían los datos del pago y el ID del cuestionario que se va a promocionar, y se guarda en el modelo de los Pagos. | JSON con el mensaje de éxito de registro de pago si logra registrar el pago, o con un mensaje de fallo si no logra registrar el pago. |
| 16 | Listar todos los pagos hechos en la aplicación | Correo del usuario | Se envía el correo y la clave del usuario, y el servidor crea una lista de los pagos del usuario y finalmente la envía | JSON con la lista de los pagos realizados en la aplicación. |
| 10 | Crear un cuestionario | Correo del usuario, clave del usuario, nombre y descripción del cuestionario. | Envía el correo del usuario con su clave en una combinación y los datos del cuestionario, lo añade al modelo de cuestionarios | JSON con el mensaje de éxito de creación si logra crear el cuestionario, o con un mensaje de fallo si no logra guardarlo. |
| 11 | Añadir un par pregunta/respuesta | ID del cuestionario, correo del usuario y su clave, la pregunta, y la respuesta. | Envía la información indicada, valida los datos, crea el registro en los modelos de Pregunta y Respuesta respectivamente, asigna la clave de respuesta a la pregunta, y asigna tanto la pregunta como la respuesta al cuestionario donde se añadieron | JSON con el mensaje de éxito de registro de éxito si logra crear el cuestionario, o con un mensaje de fallo si no logra guardar el par. |
| 12 | Modificar un par pregunta/respuesta | ID del cuestionario, ID de la pregunta, correo del usuario y su clave. | Envía los datos del usuario cifrados, valida si el cuestionario en cuestión pertenece al usuario, y luego busca la pregunta para mandar los datos del par al usuario que los pidió. | JSON con el ID de la pregunta su contenido, el ID de la respuesta y el contenido de esta, para su posterior actualización. |
| 13 | Eliminar un par pregunta/respuesta | ID del cuestionario, ID de la pregunta, correo del usuario y su clave. | Envía los datos al servidor, se valida si el cuestionario a editar es del usuario, y si lo es, se elimina la pregunta con el ID indicado. | JSON con el mensaje de éxito de registro de éxito si logra quitar la pregunta del cuestionario, o con un mensaje de fallo si no lo logra borrar. |
| 14 | Mostrar un cuestionario para su lectura desde la lista de búsqueda | ID del cuestionario | Envía los datos al servidor, se busca el cuestionario, y se envía un objeto con los datos del cuestionario, las preguntas, y sus respectivas respuestas. | JSON con los datos del cuestionario que se solicitó. |