**David Alejandro López Torres 17300155 Proyecto 3**

Empecemos rememorando una secuencia de recursos clave sobre el plan generalmente, el nombre de comentado sistema va a ser BuronSchool, el propósito general va a ser posicionar a BuronSchool como una de las primordiales plataformas de cursos online en el marco de programación, no obstante, va a tener la función de expandirse a distintas zonas de la ingeniería en caso de tener el triunfo que se pretende, o sea la obra de BuronSchool va a ser flexible y con una composición que posibilite una fácil escalabilidad del sistema. La iniciativa va a ser que BuronSchool no solo ofrezca servicios a usuarios personales, sino que tras haber conseguido estatus y confianza en medio de las organizaciones BuronSchool se vuelva un ejemplo para la capacitación de empleados en las organizaciones más relevantes de desarrollo de programa. Según con las propiedades del sistema debemos hacer y examinar los periodos de desarrollo para afirmar la calidad del programa acorde a los estándares utilizados en el desarrollo de aplicaciones de tipo e-learning, así como en el desarrollo de programa de manera diaria. Definamos los tipos de planes por los que tenemos la posibilidad de garantizar una buena etapa de prueba del programa:

Gracias a la naturaleza del plan el tipo de emprendimiento que se usará va a ser el visionario: Es decir usaremos esta perspectiva de BuronSchool para convertirla en comercio no obstante mantendremos nuestras propias expectativas e ideas realísticas fundamentadas en los datos. Pues la meta a conseguir es más que viable dado las situaciones que acontecen gracias a la enfermedad pandémica del 2020 de Coronavirus 19, la sociedad ha aprendido a lidiar con las limitaciones que fueron impuestas, y basados en numerosas encuestas y datos hechos por medios de elevado prestigio, la sociedad muestra una determinada tendencia a quedarse en esta clase de modalidad de trabajo en caso de que podamos retornar en un futuro a algo parecido a la vieja “normalidad”. Es por ello, por lo cual un modelo de enseñanza con base en el e-learning tiene una extensa capacidad, y una gigantesca posibilidad de triunfo.

Tenemos la posibilidad de dividir la preparación del plan en 2 piezas, la primera es la preparación de la plataforma, la segunda es la atención de las clases y los maestros que atenderán a los cursos y la ejecución de ellos. Las dos piezas tienen que estar llevadas a cabo de manera idónea para que todo el plan logre tener el triunfo que se espera que tenga. No obstante, una virtud es que las dos piezas tienen que desarrollarse en tiempos diversos (al menos la mayoría de ella), es decir ya que no tenemos la posibilidad de comenzar con la ejecución y organización de los cursos si todavía no fue desarrollada la plataforma, no obstante, podría ser que cuando se encuentren impartiendo los cursos se ocupe de alguna actualización para llevar a cabo novedosas funciones o componer bugs (errores o trabas en el código original).

En esta situación desarrollare un plan análogo a la carrear de la que egresare, desarrollo de programa, no obstante es fundamental destacar que no está relacionada con el plan que en este instante realizo para egresar, el plan de comercio que propongo es un sistema de cursos y enseñanza online basada en el inicio del e-learning, comentado sistema tiene la tarea de dar cursos y enseñanza de calidad de manera primordial en las superficies de programación abarcando varios idiomas de programación y aplicaciones interesantes que le van a ser de utilidad al alumno para el entorno profesional en el cual se desarrollaran. El nombre de comentado sistema va a ser BuronSchool, la finalidad general va a ser posicionar a BuronSchool como una de las primordiales plataformas de cursos online dentro de programación, no obstante, va a tener la función de expandirse a distintas zonas de la ingeniería en caso de tener el triunfo que se pretende, o sea la obra de BuronSchool va a ser flexible y con una composición que posibilite una simple escalabilidad del sistema. La iniciativa va a ser que BuronSchool no solo ofrezca servicios a usuarios personales, sino que tras haber conseguido estatus y confianza en medio de las organizaciones BuronSchool se vuelva un ejemplo para la capacitación de empleados en las organizaciones más relevantes de desarrollo de programa.

Como tal hemos tratado sobre la filosofía y tipo de mentalidad que buscamos en el plan no obstante una sección fundamental es el financiamiento de esta puesto que no importa con que pasión se hagan las cosas sin un óptimo proyecto de financiamiento, sencillamente no es viable sustentar el plan, y únicamente se lograran deudas. Pese a que se sospecha que el alcance del plan tendrá que ser enorme posicionándose como un ejemplo para organizaciones de elevado grado, la organización tendrá que iniciar siendo una microempresa que antes que nada va a ser producido por tecnólogos que no muestran un extensa vivencia gremial no obstante si técnica y esta va a ser suficiente para llevar a cabo la primera parte del plan sin ningún gasto creado además del trabajo creado por los desarrolladores y el precio del internet/electricidad consumida. Ya que la dificultad técnica del desarrollo y mantenimiento de tal plataforma no es complicada comparativamente a otro tipo de servicios los accesorios delegados del mantenimiento y actualización del programa no tendrá que ser enorme, sin embargo tendrá que ser experimentado y hábil en el campo del desarrollo web. Para la segunda parte del plan se contará con un equipo de profesionales psicólogos en temas de enseñanza, licenciados en enseñanza, ingenieros profesionales en el sector que se busca enseñar, profesionales en control de calidad, profesionales en desempeño de marketing en línea y capacitadores para enseñar a los ingenieros la utilización de la herramienta.

**Herramientas de Trabajo:**

Postman: Es un instrumento que primordialmente nos posibilita producir pedidos sobre APIs de una manera bastante simple y poder, tal cual, probar las APIs. Todo con base en una expansión de Google Chrome. El cliente de Postman podría ser un inventor que se encuentre comprobando el desempeño de una API para desarrollar sobre ella o un operador el cual se encuentre llevando a cabo labores de monitorización sobre un API.

JMeter: es un instrumento que permite la administración integral de los procesos de pruebas de rendimiento, sin embargo, no es la exclusiva pues poseemos otras como: Micro Focus LoadRunner, IBM RPT, SilkPerformer, WebLoad, NeoLoad, OpenSTA, otras.

Selenium: es un ámbito de pruebas que se usa para verificar si el programa que se está desarrollando funciona de forma correcta. Este instrumento posibilita: filmar, editar y depurar casos de pruebas que tienen la posibilidad de automatizar.

LoadRunner: Es un framework de Microfocus para hacer pruebas de rendimiento. Es un framework que está compuesto de algunas herramientas. Es multiprotocolo puesto que nos posibilita filmar y llevar a cabo scripts en diversos protocolos, cómo podría ser HTTP, SAP, etc. Es multilenguaje de manera nativa. No es Open Source, sino que es un framework dueño. Es gratuito hasta 50 usuarios concurrentes, y desde allí es necesario bastante más de una licencia.

NeoLoad: Es un programa de prueba de carga y rendimiento para aplicaciones web y móviles que simula la actividad del cliente y monitoriza el desempeño del servidor. Permite el despliegue de tus aplicaciones originarias de web, intranet o móvil con total confianza, independientemente de las tecnologías usadas, inclusive las más actuales como GWT, Silverlight, Flex y Ajax Push.

Luego de esto debemos contabilizar y conceptualizar los peligros que hay es decir

No se cuenta con los testers suficientes para hacer las pruebas en la era predeterminado (Medio): La gran proporción de ecuaciones que debería solucionar la aplicación necesita de la más grande proporción de pruebas probables

❖ Los equipamientos de pruebas está formado mayormente por testers con escasa vivencia (Medio): No es preciso demasiada vivencia para probar al sistema: solo necesita tomar una foto a un escrito y equiparar lo cual regresa el sistema con lo deseado

El ambiente de pruebas no está configurado correctamente (Medio): Hay 3 formas de interactuar con el sistema para hacer pruebas, en donde todos ellos son bastante flexibles, por lo cual todavía en condiciones adversas se espera hallar una ruta para hacer las pruebas

**Análisis de Benchmarking**

|  |  |
| --- | --- |
| BuronSchool | Khan Academy |
| Explicación de temas básicos y avanzados en el ámbito de programación | Explicación de temas básicos en el ámbito de programación |
| Formato en línea y fuera de línea | Formato solo en línea |
| Conexión alámbrica e inalámbrica para el transporte de metadatos | Conexión inalámbrica |
| Explicación y desarrollo en varios tipos de programación | Explicación superficial de algoritmia |
| Enfocado únicamente a la programación. | extensión en explicación de varios tópicos de carácter intelectual |
| Profesores profesionales específicos | Profesores de varios indoles |
| Videos de varios temas |  |

Fortalezas: La aplicación tiene inconvenientes de estabilidad (Medio): La aplicación en sí no guarda valores bastante relevantes de los usuarios (solo su cuenta y contraseña, que se usa para el historial de consultas de ecuaciones). No obstante, la probabilidad de dañar código y con ello el desempeño puede ser arriesgado.

Oportunidades: El presupuesto para la ejecución de las pruebas es menor al que debe ser (Medio): El plan está sentado en las bases del desarrollo Open Source, aun cuando varias implementaciones de APIs necesitan un diminuto aporte. Así, una salida de la partida presupuestal podría ser complicada, sin embargo resolver por los accesorios.

Debilidades: Unos de los inversionistas retiran su apoyo al plan, por lo que los accesorios de prueba sufren recortes de presupuesto (Alto): Si por alguna razón las tecnologías que se aplican de forma Open Source cambian abruptamente a cobrar una cuota por servicio el plan se ve en apuros ya que el plan en sí se fundamenta en la relación de cada una de estas herramientas: si una se bloquea, la aplicación no puede funcionar.

**Puntos de licenciamiento**

|  |  |  |
| --- | --- | --- |
| Tipo de Licencia | Pros | Contras |
| Software Libre | Otros usuarios pueden aprender de  nuestras implementaciones, así  como aportar al proyecto. | Tener libre control sobre la información o código  fuente puede darle usos que van en contra de lo que  desearíamos como sus autores. |
| Copyleft con  Protección | Otros usuarios pueden aprender de  nuestras implementaciones, así como aportar al proyecto. | En caso de querer comercializar el producto u obtener  alguna clase de remuneración, no podremos atribuirlo  al código (bits secretos). |
| como aportar al proyecto. | como aportar al proyecto. | como aportar al proyecto. |
| FreeDSB | El crédito se debe garantizar para  los desarrolladores del código y sus  derivados. | La comercialización es libre luego no sería posible  cobrar comisiones por trabajo derivados del código  original |
| MPL | El crédito se debe garantizar para  los desarrolladores del código y sus  derivados.  Las modificaciones se presentan a  los creadores del código original  La comercialización de tercero solo  puede realizarse a través de  archivos binarios.  La evolución o dirección del código  está más controlada por los  creadores. | Se perderían las ganancias o licenciamientos de los  archivos binarios. |
| X.Org | Podemos mezclar las mejores  características del código abierto y  código cerrado. | Puede llegar a ser muy abierta en el sentido de  “personalización”, de forma que terminar la licencia  será más complicado que solo adoptar una existente. |
| Freeware | El autor mantiene control sobre el  código original, no permite que otros  modifiquen o distribuyan el código.  En si permitirá solo el aprendizaje de  las técnicas y métodos utilizados en  el programa. | No permite la evolución del programa mediante  aportaciones externas pues no se permiten las  modificaciones. |
| Donationware | El programa podría continuar su  evolución puramente diseñada por el  equipo original del código gracias a  las aportaciones voluntarias de los  usuarios. | Puede que los donativos no sean suficientes para  mantener el software, así mismo habría que lidiar con  modificaciones ilegales. |
| Shareware | En versión demo y trial podría  motivar a los usuarios a querer  pagar una versión completa. | Las ganancias deberán depender únicamente de las  licencias compradas del software. |
| Addware | El adware podría proveer  financiamiento extra para el  desarrollo y mantenimiento del  proyecto. | La cantidad de anuncios debe ser precisa para  mantener el balance en la aplicación, pero al mismo  tiempo poder darle mantenimiento a la misma. |
| Criplleware | Mostrar un poco de las capacidades  del software puede motivar a los  usuarios a querer utilizar las  capacidades completas. | Puede ser que los usuarios se den con la versión  gratuita de forma que las ganancias no serán  suficientes para mantener el proyecto. |
| Propietario | Permite la personalización de las  limitaciones de modificación y  redistribución del proyecto.  Permite tener un mayor control  sobre el proyecto, cualquier  distribución o modificación  extraordinaria cobrara una comisión  el propietario original. | La debido al pago por redistribución puede que las  modificaciones no estén obligadas a compartirse al  proyecto original, lo que puede llevar a muchas  versiones del mismo producto envés de uno muy  evolucionado. |

Gracias a la modalidad web que tiene el plan, y ya que necesita de servicios externos (que cobran) para seguir estando, funcionando necesitamos obtener alguna clase remuneración (al menos para conservar el servicio en funcionamiento). Tomando esto presente en el entorno de este plan la mejor manera de obtener remuneración es mediante anuncios en el servicio web.

Sin embargo, el enfoque que buscamos con este plan es incentivar la utilización de las matemáticas en los estudiantes y proveerles una forma fácil de solucionar ecuaciones diferenciales (muy frecuentes en tópicos de ciencia o ingeniería). Por esto de determinada forma manejar la alternativa del código abierto con el fin de enseñar u ofrecer un caso muestra a continuar a otros alumnos de manera que no empiecen a partir de 0, es lo cual estamos intentando encontrar. No obstante, preferimos continuar teniendo el control de la aplicación, de manera que buscamos un estilo de trabajo parecido a LINUX en el cual la sociedad labora sobre un mismo producto en lugar de producir derivados de este producto (que tienen la posibilidad de concluir asciendo auto competencia). Tomando en cuenta esta filosofía de trabajo la elección de utilizar anuncios como forma de financiamiento cobra fuerza puesto que no posee caso cobrar por el programa si es abierto. Sin embargo, la elección de cobrar por servicios estilo add-on no parece ser una elección por ahora puesto que la aplicación está enfocada a un objetivo bastante especial no obstante esta clase de filosofía posibilita que, si en un futuro la aplicación se lleva a cabo a una escala más grande, se logre llevar a cabo esta técnica de remuneración.

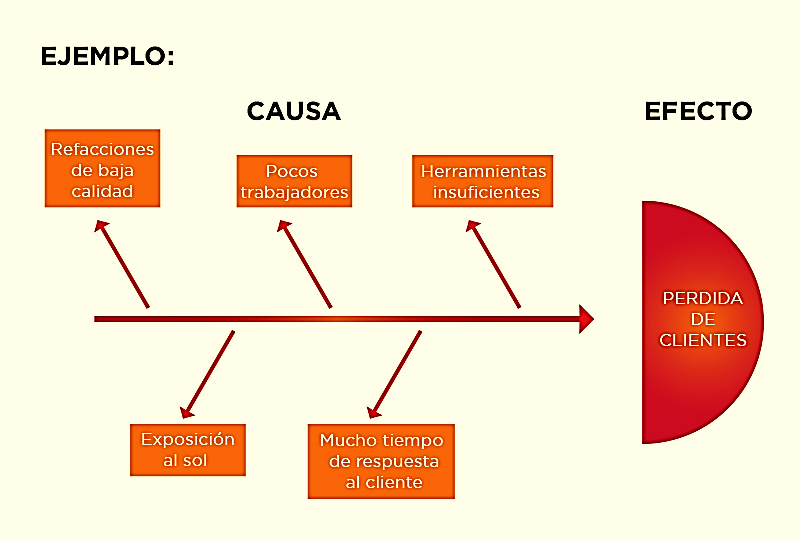
Como tal buscamos eludir que se formen programas derivados o basados en el código original gracias a la utilización de nuestras propias ideas de parte de otros programadores, más todavía, intentamos eludir que lucren con él. Por consiguiente, optaremos por políticas precisas sobre el reparto y comercialización de derivados del código original.

No va a poder ser publicado, ni reclamarse como de su autoría cualquier programa derivado del código original, no obstante, cualquier modificación va a poder reportarse a un conjunto delegado del código original y tras examinar sus alteraciones van a poder implementarse las mejoras propuestas sobre el código original y dichos usuarios recibirán crédito por la variación (al estilo de linux).

Pues por ahora la finalidad y las funciones del sistema resultan muy concretas y no bastante amplias, las políticas por “trial” o “demo” no parecen ser una alternativa conveniente para el sistema además al ser opensource los usuarios con entendimiento podría verse tentados a explicar por su cuenta el trabajo en lugar de abonar por la versión completa.

En conclusión, el mejor tipo de licenciamiento va a ser de tipo copyleft con defensa en adicionamiento implementaremos la filosofía addware para dar un tipo de financiamiento al plan. Si la evolución del plan da frutos vamos a poder llevar a cabo la comercialización de servicios sobre el programa (add-ons), que tienen la posibilidad de ser desarrollados así sea por creadores originales o por la sociedad de desarrolladores que aporta al plan.

**Proceso de Ishikawa:**



Fundamentalmente dichos son los pasos que debemos continuar respecto al programa para lograr llegar a un estado de calidad total recordemos que a calidad total es un plan de administración por medio de la cual la organización sacia las necesidades y expectativas de sus consumidores, de sus empleados, de los accionistas y de toda la sociedad generalmente, usando los recursos de que dispone: personas, materiales, tecnología, sistemas de producción entre otros. Esto involucra la Satisfacción de consumidores internos y externos. Es fundamental notar los beneficios económicos que se derivan de acometer la utilización de calidad total en este programa puesto que su triunfo se fundamenta en ser de mejor calidad comparativamente con la competencia. Una vez conocidas los beneficios resulta más simple la puesta en marcha de medidas que, aun cuando inicialmente tenga, pongan en prueba los costes, todavía más grandes, de la no calidad que existe en su organización derivados de una inadecuada implementación de los recursos. La calidad total es un instrumento necesario para la administración en las empresas y organizaciones, tomar medidas para mejorar su administración por medio de la calidad total que redunde en la optimización de su competitividad. Para nuestra conveniencia es mejor apegarnos a un modelo en específico de calidad total en nuestro caso Modelo Malcom Baldrige. Es el modelo de excelencia usado en EE. UU. y constituye el marco de alusión usado para la evaluación del premio nacional del territorio norteamericano. Este modelo estima 7 puntos de vista clave, que son los que sirven para evaluar el sistema de administración de la calidad: liderazgo, idealización estratégica, orientación al comprador y al mercado, información y estudio, orientación a los recursos humanos, administración de los proceso y resultados del comercio.