**AUTONOMOUS LEARNING ABC**

**DIANA CAROLINA GÓMEZ GONZALEZ**

**ANDRES FELIPE TORRES MORENO**

**MIGUEL ANGEL MARIN PERILLA**

**JOHAN SEBASTIAN CIFUENTES GIRALDO**

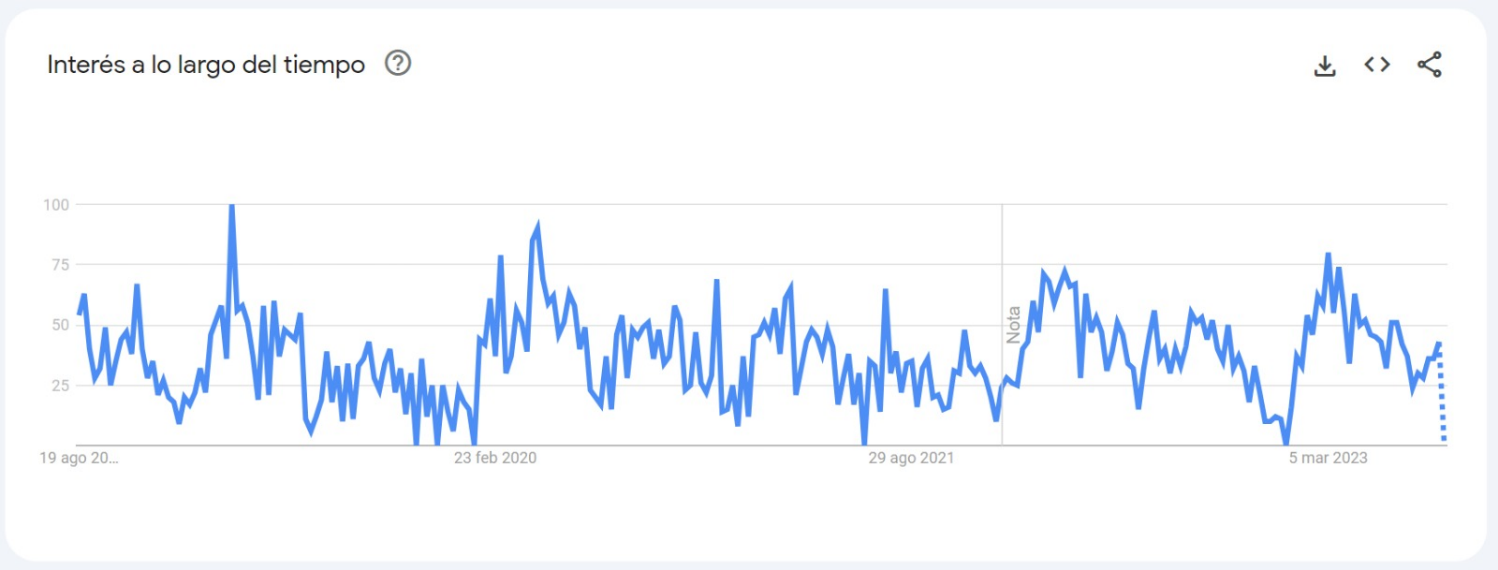
**UNIDAD CENTRAL DEL VALLE**

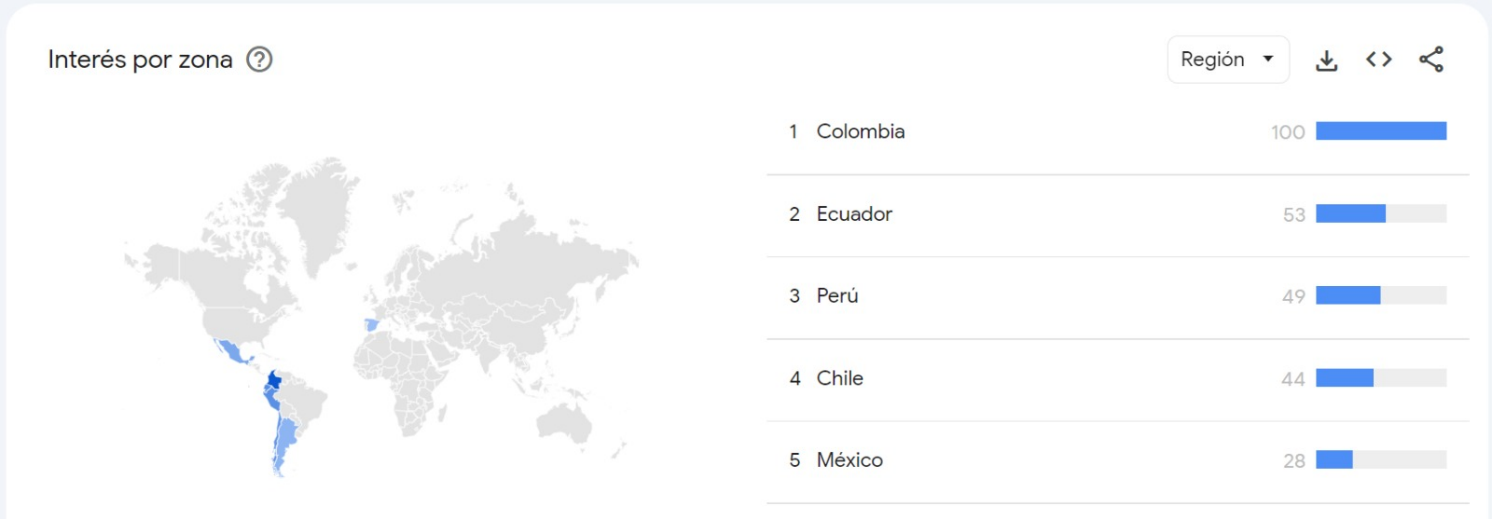
**ING DE SOFTWARE II**

**DOCENTE ROYER ESTRADA**

**TALLER 1**

**AÑO 2023**





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CRITERIO** | **DESCRIPCIÓN** | **TEMA** | **PAÍS** | **CALIFICACIÓN (1 / 5)** |
| Presencia en internet | Mayor presencia en motores de búsqueda en internet | SOFTWARE PARA NIÑOS | COLOMBIA  ECUADOR  PERÚ  CHILE  MÉXICO | 5  2.5  2.2  2  1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **CRITERIO** | **DESCRIPCIÓN** | **TEMA** | **DOCUMENTACIÓN** |
| Documentación | Volumen de libros o papers en inglés o español  referentes a la metodología | SCRUM  XP  TDD | Scrum Methodology, Sakshi Sachdeva  <https://www.academia.edu/download/46357609/11_ijecs.pdf>  The Positive Affect of the XP Methodology, Sharifah Lailee Syed-Abdullah, John Karn, Mike Holcombe, Tony Cowling & Marian Gheorge  <https://link.springer.com/chapter/10.1007/11499053_31>  Test-driven development as a defect-reduction practice, L. Williams, E.M. Maximilien, M. Vouk  <https://ieeexplore.ieee.org/abstract/document/1251029> |

|  |  |  |  |
| --- | --- | --- | --- |
| **CRITERIO** | **DESCRIPCIÓN** | **METODOLOGÍA** | **EMPRESAS** |
| Certificación | Aquella metodología en la  que se puede certificar y  por ende, garantizar el  cumplimiento de sus  prácticas. | Agile Project Management (APM) | * Dynatrace * New Relic * AppDynamics * Splunk * BMC Software * CA Technologies (ahora parte de Broadcom) * Cisco AppDynamics (anteriormente AppDynamics) * Micro Focus * Riverbed Technology |

|  |  |  |  |
| --- | --- | --- | --- |
| **CRITERIO** | **DESCRIPCIÓN** | **TEMA** | **Instituciones** |
| Training | Aquella metodología para  la que existe instituciones  que ofrecen programas de  entrenamientos referentes a la metodología. | Scrum  XP  TDD | * Scrum Alliance: Scrum Alliance es una de las organizaciones líderes en la enseñanza y certificación de Scrum. Ofrecen certificaciones como Certified ScrumMaster (CSM), Certified Scrum Product Owner (CSPO) y Certified Scrum Developer (CSD), entre otras. * Scrum.org: Scrum.org es otra organización de renombre que proporciona formación y certificación en Scrum. Ofrecen certificaciones Professional Scrum Master (PSM), Professional Scrum Product Owner (PSPO) y Professional Scrum Developer (PSD), entre otras. * Agile Alliance: Agile Alliance es una organización que promueve las prácticas ágiles en general, incluyendo Extreme Programming. Ofrecen eventos, conferencias y recursos de aprendizaje que pueden incluir contenido sobre XP. * Agile Training Providers: Muchas organizaciones y consultoras que ofrecen formación en metodologías ágiles también incluyen cursos y talleres específicos sobre XP. Algunas de estas organizaciones incluyen Agile42, AgileForAll y SolutionsIQ, entre otros. * Institutos de Formación Ágil: Instituciones que se especializan en formación ágil a menudo ofrecen programas de entrenamiento en Extreme Programming. Investiga instituciones reconocidas que tengan una buena reputación en la comunidad ágil. * Cursos en línea: Plataformas educativas en línea como Udemy, Coursera, edX y LinkedIn Learning ofrecen una variedad de cursos relacionados con TDD. Puedes buscar cursos específicos sobre TDD y desarrollo ágil en estas plataformas. * Institutos de Formación Ágil: Muchas organizaciones y consultoras ágiles ofrecen cursos y talleres sobre TDD y otras prácticas ágiles. Algunas de estas organizaciones incluyen Agile Alliance, Agile42, y AgileForAll, entre otros. * Desarrolladores y Consultores Independientes: Algunos desarrolladores y consultores independientes ofrecen entrenamiento personalizado en TDD. Pueden brindarte orientación y capacitación adaptada a tus necesidades específicas |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CRITERIO** | **DESCRIPCIÓN** | **METODOLOGÍA** | **ENTIDAD** | **TEMA** |
| Comunidades | Hay formada una  comunidad de  profesionales alrededor de  la metodología o está  asociada a Agile Alliance | Agile Project Management (APM) | Universidad de Cornell  Alaska Airlines | Utilizó APM para solucionar problemas en una plataforma de software crítica que se caía varias veces a la semana debido a transacciones complejas.  Desplegó APM para ayudar a optimizar sus complejos sistemas basados en la nube y mantener la satisfacción del cliente. |

|  |  |  |  |
| --- | --- | --- | --- |
| **CRITERIO** | **DESCRIPCIÓN** | **TEMA** |  |
| Presencia Empresarial | Empresas de desarrollo de  software usan la  metodología | Scrum  XP  TDD |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **CRITERIO** | **DESCRIPCIÓN** | **TEMA** |  |
| Proyecto Software | Proyectos de software en  donde es evidente el uso  de la metodología. | Scrum  XP  TDD |  |

**METODOLOGÍAS**

**SCRUM**

1. Ventajas:

* El cliente está satisfecho ya que recibe lo que necesita y esperaba
* El coste en términos de proceso y Management es mínimo, llevando a un

resultado más rápido y barato.

* Ayuda a la empresa a ahorrar tiempo y dinero
* Permite realizar proyectos en los que la documentación de los requerimientos

de negocios no están muy claros como para ser desarrolladas

* Se desarrolla rápidamente y testea. Cualquier error puede ser fácilmente

rectificado.

* Los problemas se identifican por adelantado en las reuniones diarias y por lo tanto se pueden resolver rápidamente
* Hay visibilidad clara del desarrollo del proceso
* Iterativo en su naturaleza, requiere continuo feedback del usuario
* Fácil de manejar los cambios debido a los sprints tan cortos y el feedback constante
* Poco control que insiste en la información frecuente del proceso en el trabajo

mediante reuniones regulares.

* Las reuniones diarias hacen posible medir la productividad individual. Esto

lleva a la mejor en la productividad de cada uno de los miembros del equipo.

* Para los desarrolladores las ventajas pueden ser motivación y satisfacción

de realizar el trabajo de una manera eficiente.

* Es fácil entregar un producto de calidad en el tiempo estipulado
* Puede trabajar con cualquier tecnología o lenguaje de programación
* Hace el proceso del desarrollo de software más centrado y manejable

1. Desventajas:

* Si no existe una fecha definitiva de finalización del proyecto es posible que se siga solicitando, y añadiendo, nueva funcionalidad.
* Si una tarea no está bien definida, los costes de tiempo y dinero estimados

del proyecto no serán demasiado exactos. En ese caso, la tarea se puede extender sobre varios sprints.

* Si los miembros del equipo no están centrados y convencidos, el proyecto nunca se completará o incluso fallará.
* Está bien para proyectos pequeños, de rápido movimiento ya que trabaja bien solo con equipos pequeños.
* Esta metodología necesita solo miembros de equipo experimentados. Si el equipo consiste en gente que son junior, el proyecto no puede ser completado a tiempo.
* Además de los recursos sin suficiente experiencia , la falta de dirección firme puede llevar a los proyectos a no completarse o incluso fallar.
* La metodología Scrum funciona bien cuando el scrum master confía en el equipo que lleva. Si se practican controles muy estrictos sobre los miembros

del equipo, puede ser extremadamente frustrante para ellos, llevando a la

desmoralización y el fallo del proyecto.

* Si algunos de los miembros del equipo se marchan durante el desarrollo puede

tener un efecto negativo enorme en el desarrollo del proyecto.

* El control de la calidad del proyecto es difícil de implementar y cuantificar a menos que el equipo de test puedan llevar a cabo testeo de regresión después de cada sprint.

**Test Driven Development (TDD)**

1. Ventajas:

* Mayor calidad
* Diseño enfocado en las necesidades
* Mayor simplicidad en el diseño
* El diseño se va adaptando al entendimiento del problema
* Mayor productividad
* Menos tiempo invertido en debugging de errores

1. Desventajas:

* Interfaz de usuario
* Base de datos
* Errores no identificados
* Perder la visión general (Ver el árbol en lugar del bosque)
* Pronunciada curva de aprendizaje

**Extreme Programming (XP)**

1. Ventajas:

* Da lugar a una programación sumamente organizada.
* Ocasiona eficiencias en el proceso de planificación y pruebas.
* Cuenta con una tasa de errores muy pequeña.
* Propicia la satisfacción del programador.
* Fomenta la comunicación entre los clientes y los desarrolladores.
* Facilita los cambios.
* Permite ahorrar mucho tiempo y dinero.
* Puede ser aplicada a cualquier lenguaje de programación.
* El cliente tiene el control sobre las prioridades.
* Se hacen pruebas continuas durante el proyecto.
* La XP es mejor utilizada en la implementación de nuevas tecnologías.

1. Desventajas:

* Es recomendable emplearla solo en proyectos a corto plazo.
* En caso de fallar, las comisiones son muy altas.
* Requiere de un rígido ajuste a los principios de XP.
* Puede no siempre ser más fácil que el desarrollo tradicional

**Comparativa**

* Una diferencia en el sistema de seguimiento entre Scrum y programación extrema es que para el proceso de seguimiento y entrega trabajan con tiempos diferentes. En el caso de Scrum las iteraciones (pequeños bloques) que se denominan Sprints se entregan entre 2 semanas a un mes. En cambio, la metodología xp se maneja con tiempos relativamente más cortos que van desde 1 a 2 semanas.
* En el estado de cambios los equipos de Scrum y Xp no permiten cambios en los sprints después de que el Product Owner lo apruebe. No obstante, el mismo proceso crea algo llamado Sprint Backlog o lista de tareas del sprint, en donde se pueden hacer modificaciones solo en el caso de que sea necesario para mejorar el desarrollo de las tareas.
* En general, Scrum y TDD son dos métodos diferentes de desarrollo de software. Scrum se enfoca en la entrega de productos de software funcionales y de alta calidad en un corto período de tiempo, mientras que TDD se enfoca en crear un código que sea testable, robusto y fácil de mantener. Ambos métodos tienen sus propias ventajas y desventajas, y el método adecuado para un proyecto en particular dependerá de las necesidades específicas del proyecto.

**Metodología para el proyecto de software para niño:**

Test Driven Development (**TDD**) es la más viable para nuestro proyecto ya que requiere de un desarrollo que inicialmente se adapta al problema es de mediana complejidad , de alta calidad , fácil de refactorizar donde permite una eficiencia del producto para nuestros clientes.

**ESTIMACIÓN DE RIESGOS**

**LISTA DE SUPUESTOS(Condiciones iniciales)**

1. El equipo estará conformado por 4 integrantes.
2. El proyecto será un Desarrollo web.
3. Plazo estimado de entrega 6 meses.
4. Disponibilidad de fondos para licencias en la aplicación.
5. Poca disponibilidad de personas para realizar pruebas de calidad de la aplicación.
6. El personal de la empresa asistirá a todas las reuniones para modificación o evaluación del producto.
7. Voluntad política para permitir convenios entre empresas y el producto
8. El software contará con un apartado de sugerencias para nuevas dinámicas o actividades.
9. El software cuente con permisos de controles parentales.
10. Se trabajará con método de tecnología ágil.

**LISTA DE RIESGOS (Solución a supuestos)**

1. Teniendo en cuenta diferentes aspectos puede surgir la necesidad de un nuevo integrante para el desarrollo de nuestra aplicación.
2. Puede existir un fallo de energía o del servidor que no va a permitir la funcionalidad de la aplicación de forma correcta.
3. Es posible que alguno de los integrantes no cumpla con los sprint requeridos en el tiempo estimado.
4. No disponer con el suficiente capital que se ha tenido en cuenta para adquirir las licencias de la aplicación.
5. No contar con suficiente personal que tenga el conocimiento previo para realizar las pruebas de calidad de nuestra aplicación.
6. No contar con la disponibilidad de todos los integrantes por motivo o fuerzas externas que no se puedan controlar.
7. Las empresas no vean necesario hacer uso de nuestra aplicación ya que no cubre sus necesidades.
8. El software no cuente con restricciones para su uso en el apartado de sugerencias.
9. Falta de supervisión del adulto o tutor responsable en el momento que el menor haga uso de la aplicación.
10. Falta de experiencia al utilizar la metodología.

**DEFINICIÓN DE ESCALA DE EVALUACIÓN DE RIESGOS**

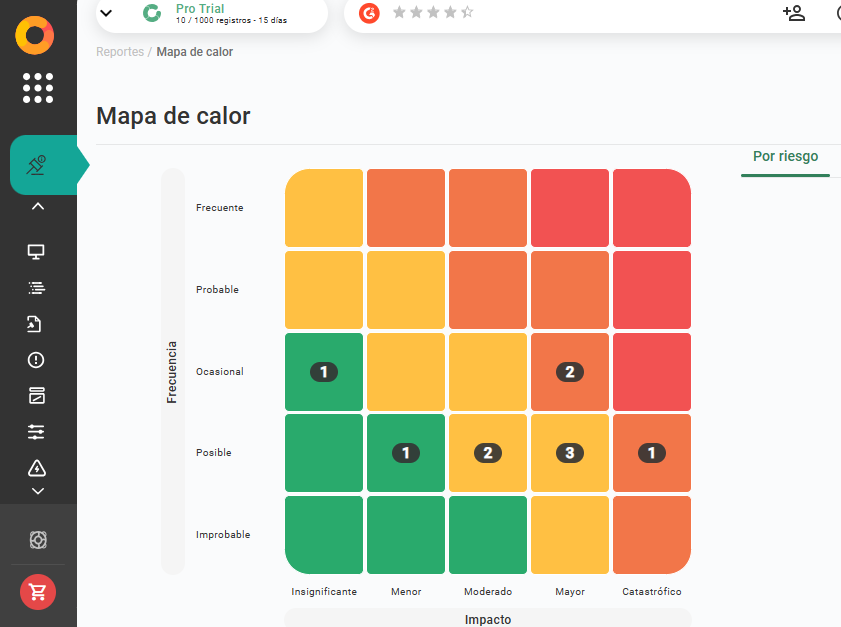
|  |  |
| --- | --- |
| Probabilidad | |
| Muy baja | 0,1 |
| Baja | 0,3 |
| Moderada | 0,5 |
| Alta | 0,7 |
| Muy Alta | 0,9 |

|  |  |
| --- | --- |
| Impacto | |
| Muy bajo | 0,05 |
| Bajo | 0,1 |
| Moderado | 0,2 |
| Alto | 0,4 |
| Muy Alto | 0,8 |

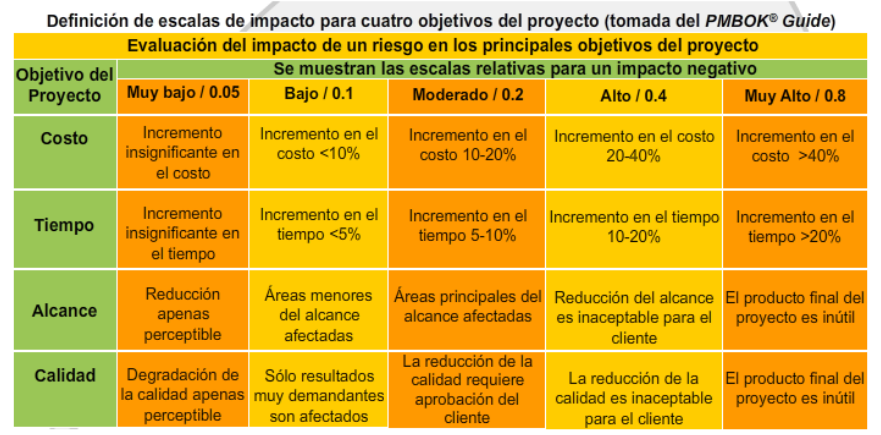
|  |  |
| --- | --- |
| Riesgo | |
| Bajo | <0,05 |
| Moderado | 0,05-0,15 |
| Alto | >0,15 |

**MATRIZ P/I**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Probabilidad | Riesgo | | | | |
| 0,9 | 0,05 | 0,09 | 0,18 | 0,36 | 0,72 |
| 0,7 | 0,04 | 0,07 | 0,14 | 0,28 | 0,56 |
| 0,5 | 0,03 | 0,05 | 0,1 | 0,2 | 0,4 |
| 0,3 | 0,02 | 0,03 | 0,06 | 0,12 | 0,24 |
| 0,1 | 0,01 | 0,01 | 0,02 | 0,04 | 0,08 |
| Impacto | 0,05 | 0,1 | 0,2 | 0,4 | 0,8 |



**EVALUACIÓN DE IMPACTO**

****

**PROBABILIDAD**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Probabilidad | Riesgo | | | | |
| 0,05 | 0,09 | 0,18 | 0,36 | 0,72 | 0,05 |
| 0,04 | 0,07 | 0,14 | 0,28 | 0,56 | 0,04 |
| 0,03 | 0,05 | 0,1 | 0,2 | 0,4 | 0,03 |
| 0,02 | 0,03 | 0,06 | 0,12 | 0,24 | 0,02 |
| 0,01 | 0,01 | 0,02 | 0,04 | 0,08 | 0,01 |

**IMPACTO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Costos | Tiempo | Alcance | Calidad | Calificación Impacto |
| Necesidad de un nuevo integrante para el desarrollo de nuestra aplicación. | 0,2 | 0,2 | 0,1 | 0,05 | 0,1375 |
| Fallo de energía o del servidor | 0,4 | 0,4 | 0,4 | 0,4 | 0,4 |
| Integrantes no cumplen con tiempo estimado | 0,4 | 0,5 | 0,8 | 0,8 | 0,625 |
| Reducción de Presupuesto | 0,8 | 0,4 | 0,4 | 0,2 | 0,45 |
| Insuficiente personal | 0,05 | 0,4 | 0,2 | 0,1 | 0,1875 |
| Poca disponibilidad de integrantes | 0,05 | 0,2 | 0,1 | 0,05 | 0,1 |
| Empresas sin interés de la aplicación | 0,8 | 0,05 | 0,8 | 0,8 | 0,6125 |
| No tener restricciones de uso | 0,8 | 0,2 | 0,2 | 0,2 | 0,35 |
| Falta de supervisión de tutor responsable | 0,05 | 0,05 | 0,1 | 0,05 | 0,0625 |
| Falta de experiencia en metodología ágil | 0,4 | 0,8 | 0,4 | 0,4 | 0,5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Riesgo | Probabilidad (%) | Impacto | Calificación | Estrategia |
| Necesidad de un nuevo integrante para el desarrollo de nuestra aplicación. | 0,5 | 0,2 | 0,1 | Transferencia |
| Fallo de energía o del servidor | 0,1 | 0,4 | 0,04 | Prevención  Reducción |
| Integrantes no cumplen con tiempo estimado | 0,3 | 0,4 | 0,12 | Prevención  Reducción |
| Reducción de Presupuesto | 0,7 | 0,2 | 0,14 | Reducción |
| Insuficiente personal | 0,1 | 0,4 | 0,04 | Aceptación |
| Poca disponibilidad de integrantes | 0,5 | 0,4 | 0,2 | Reducción |
| Empresas sin interés de la aplicación | 0,5 | 0,2 | 0,1 | Transferencia |
| No tener restricciones de uso | 0,1 | 0,4 | 0,04 | Prevención -Reducción |
| Falta de supervisión de tutor responsable | 0,3 | 0,4 | 0,12 | Prevención |
| Falta de experiencia en metodología ágil | 0,7 | 0,2 | 0,14 | Prevención |

**PLANIFICACION DE LA RESPUESTA A LOS RIESGOS**

* El equipo estará conformado por 4 integrantes: Conseguir personal de medio tiempo para realizar el trabajo que haga falta.
* El proyecto será un Desarrollo web: Conseguir un servidor de respaldo que muestre la información que se necesite, mientras se le hace el mantenimiento correspondiente.
* Plazo estimado de entrega 6 meses: plan de organización que demuestre avances cada cierto tiempo
* Disponibilidad de fondos para licencias en la aplicación: buscar alternativas gratuitas y disponer un contacto con el proveedor de las licencias y explicarles la situación.
* Poca disponibilidad de personas para realizar pruebas de calidad de la aplicación: Se podría establecer colaboraciones con instituciones educativas locales para crear oportunidades de prácticas o proyectos conjuntos por esto se considera la posibilidad de capacitar a miembros del equipo.
* El personal de la empresa asistirá a todas las reuniones para modificación o evaluación del producto: Utiliza herramientas de videoconferencia y colaboración en línea para permitir la participación remota en las reuniones, Si no es posible que todos los miembros asistan en tiempo real, considera grabar las reuniones.
* Voluntad política para permitir convenios entre empresas y el producto ofrece demostraciones personalizadas de tu aplicación o períodos de prueba gratuitos para que las empresas puedan experimentar directamente cómo el producto puede beneficiarlas
* El software contará con un apartado de sugerencias para nuevas dinámicas o actividades: Diseña un apartado de sugerencias intuitivo, Implementa un formulario de sugerencias en línea que los usuarios puedan completar. Asegúrate de que sea fácil de llenar y que recoja información relevante.
* El software cuente con permisos de controles parentales: Permite que los padres o tutores creen cuentas separadas para ellos y para los niños que utilizarán el software. Esto permite un control más efectivo, Proporciona a los padres la capacidad de monitorear la actividad de sus hijos.
* Se trabajará con método de tecnología ágil: invierte en la formación y capacitación de tu equipo en la metodología ágil. Se puede contratar a un entrenador ágil o utilizar recursos en línea, cursos y talleres para enseñar al equipo los principios y prácticas ágiles.