

Programacion 3

INTRODUCCIÓN CONCEPTUAL DE PROYECTOS DE SOFTWARE.
GRUPO #3



Presentación

Nombre:

Adry Mateo Ramón (2017-5331)

Bryan De Jesus Rosa Tavaréz (2021-2239)

Domingo Gabriel Alcántara Martínez (2021-0255)

John Jairo Moreta Bautista (2020-10965)

Junior Enmanuel Sánchez Román (2021-0266)

Edwin Alexander Hernández Vicioso (2020-10486)

Roberto Adrián Cuevas Méndez (2021-0266)

Materia:

Programación 3

Profesor:

Kelyn Tejada

Sección:

Viernes 6:00 pm a 10:00 pm. Grupo 3

Fecha de entrega:

Lunes, 13 de febrero de 2023

Santo Domingo Este, R.D.



Elaborar un análisis de viabilidad de un proyecto utilizando la guía 6 pasos para llevar a cabo un estudio de viabilidad

Identificación del proyecto.

El objetivo general del proyecto es “crear un sistema que permita la modernización de la gestión y funcionamiento general de un restaurante.”

Entre los objetivos específicos están:

Crear un sistema que permita el manejo de pedidos, menú digital y flexible, intercomunicación entre las diferentes áreas del restaurante, manejo de productos y pedidos y manejo de las finanzas y del sistema de reservaciones.

Hacer el sistema accesible y viable, en el sentido de que la curva de aprendizaje no debe ser alta y que debe funcionar de la mejor manera posible.

El proyecto se puede dar como finalizado cuando se consiga hacer que el sistema abarque las áreas del restaurante que se indicaron en los requerimientos, de esta forma se tendrá la garantía de haber entregado un producto de calidad y que cumpla con los objetivos propuestos. Puesto que el cliente necesita un sistema que le permita agilizar los procesos que realiza manualmente, pero no se puede añadir complejidad a la hora de desenvolverse, se debe conseguir que el sistema luzca lo más simple y familiar posible para el cliente, de modo que su tiempo de adaptación sea mínimo y pueda aprovechar al máximo la modernización de su sistema de gestión y operaciones.

En resumen, se busca que el proyecto a largo plazo ayude al cliente a ahorrar tiempo y dinero, pues hay pasos que no le serán tan complicados, como la facturación, los cálculos de la cuenta, la toma de pedidos, la manera de realizar reservaciones y de gestionarlas, y también la manera en que se le mostrarán los productos al usuario. De este modo el cliente se asegurará de tener una infraestructura robusta y que le garantice tener todas sus



operaciones centralizadas en un mismo sitio, permitiendo así un análisis de fortalezas y debilidades más certero.

Investigación del mercado.

En el mercado existen varias opciones de sistemas de gestión y administración de restaurantes, sin embargo, los requerimientos que se mencionan son muy específicos en algunos aspectos, lo que haría que fuese necesaria una refactorización o integración de diferentes proyectos, lo que puede terminar en un sistema mal optimizado, lo que se busca evitar.

Bien es cierto que la demanda de este tipo de sistemas es alta y por esta razón el mercado ya tiene circulando softwares de gran variedad relacionados a la gestión y manejo de restaurantes, sin embargo, hemos observado que la mayoría de los restaurantes con requerimientos y manera de trabajar parecido al cliente utilizan sistemas hechos a medida, debido a que la mayoría de ofertas del mercado solo cubren aspectos básicos, por lo que no cuentan con opciones que se establecen en los requerimientos.

Otro inconveniente que se puede notar es que estos sistemas suelen estar diseñados de una forma poco amigable para un cliente que es nuevo en este aspecto, ya que no toman en cuenta la separación de muchos de los componentes que si la requieren, en la mayoría de los casos la interfaz está llena de objetos, imágenes y botones, lo que puede llevar a una sobrecarga cognitiva del usuario en sus etapa de uso inicial, lo que no va en la línea dispuesta en requerimientos de hacer el sistema lo más accesible y fácil de usar posible.



Análisis técnico.

El desarrollo será enfocado en la plataforma Windows, debido a su facilidad de desarrollo y su interfaz familiar.

Se utilizará C# .NET como lenguaje de desarrollo, con Entity Framework, y los frameworks Guna UI y Bunifu UI para la interfaz gráfica. Para la gestión y manejo de las bases de datos, se utilizará una base de datos relacional, esta será Microsoft SQL Server.

Las herramientas para utilizar serán Visual Studio 2022, para el desarrollo, SQL Server Management Studio, para la gestión de la base de datos, y Git y Github, para el trabajo colaborativo.

Estimación de costos.

Para la adquisición de 2 computadoras Lenovo ThinkCentre M710Q, (procesador intel core i5 de 7ma generación, 8GB de RAM, con 256 GB de almacenamiento en estado sólido) se dispondrá de \$220 dólares (\$110 cada una).

Cada una deberá contar con sus periféricos (mouse, monitor y teclado), los cuales harán un total de \$275.54 (\$92.36 mouse y teclado, \$46.18 cada uno y \$183.18 los monitores, \$91.59 cada uno).

Para mostrar los menús se precisará de una smart TV de 50" con un costo de \$374.92 (incluyendo el soporte y un Amazon Fire TV).

Para la capacitación del personal se destinará un presupuesto de \$1,000 que se implementarán en charlas y talleres didácticos donde se instruya al cliente y su personal sobre cómo sacar el mayor provecho al sistema.

El coste de desarrollo será de \$80,000.

Y por último se añadirá una tarifa de soporte, con un costo de \$30,000 por 2 años. Por lo que los gastos aproximados totales ascenderían a los \$111,869.54.



Análisis financiero.

La implementación del proyecto no presentará frutos de manera instantánea, ya que se debe tomar en cuenta el periodo de transición, pero se estima que cerca de 2 meses de ya implementado y desplegado el sistema se empezarán a ver los siguientes beneficios:

- Aumento de la eficiencia en el servicio.
- Mayor organización y estructuración de los procesos.
- Mejor manejo del inventario.
- Uso óptimo de los recursos.

Estos beneficios permitirán que el cliente se sienta más a gusto, decida visitar de manera más frecuente el restaurante, lo que subirá los ingresos cerca de un 45%. También se ahorrarán recursos, gracias a que se conocerá y tendrá mejor manejo de los alimentos, lo que reduce el riesgo de que se dañen o de que se gaste en un alimento que el cliente no consume, por lo que esto reduciría cerca del 30% de los gastos, lo que se transforma en beneficio.

Estamos hablando de un aumento del 75%, de la inversión luego de 2 meses, lo que representa un valor de \$195,771.69.

El retorno de la inversión sería: $(83,902.15 / 111,869.54) * 100$

Lo que equivale a un ROI de 0.75.

Lo que hace que el proyecto sea viable y representa un beneficio considerable para la empresa.



Identificación de riesgos.

Los principales desafíos que pueden presentarse durante el desarrollo del proyecto son:

Atrasos en los tiempos de entrega.

Debido a problemas inesperados y no previstos entre los integrantes del equipo de desarrollo pueden surgir inconvenientes personales que se perfilan como amenazas para las fechas de entrega y el flujo de desarrollo.

Problemas de adquisición de los equipos y herramientas necesarias para el despliegue.

Debido a escasez de stock y alza de los precios, es posible que llegue a ser complicado encontrar los equipos listados al precio sugerido.

Problemas de adopción.

Existe la posibilidad de que al cliente no se le termine de hacer del todo familiar la adopción de un nuevo sistema de gestión, lo que puede resultar en un proceso confuso o deficiente mediante el cual no le saque todo el provecho esperado al sistema.

Cambio en los requisitos del usuario.

Puede que durante el transcurso del desarrollo el cliente vaya viendo las funcionalidades y aspectos del sistema, lo que puede llevarlo a querer modificar cosas, que pueden resultar en retrasos y cortes en el flujo y planificación.



Toma de decisiones.

En base a los aspectos estudiados, el proyecto luce como una opción viable, justificando su implementación con beneficios y mejoras significativas y sustanciales. También abre la puerta a la posibilidad de nuevas posibilidades e implementaciones para que el restaurante inicie una transición donde aproveche todas las herramientas y facilidades que la tecnología brinda.

Elabore un análisis FODA a este mismo proyecto

A continuación, se presenta un análisis FODA (Fortalezas, Oportunidades, Debilidades y Amenazas) de nuestro proyecto, un sistema que permita la modernización de la gestión y funcionamiento general de un restaurante:

Fortalezas:

- Aumento de la eficiencia en la gestión de los procesos del restaurante.
- Mejora en la toma de decisiones gracias a la obtención de datos y estadísticas precisas y actualizadas.
- Reducción de los errores y demoras en la atención al cliente.
- Mejora en la coordinación entre las diferentes áreas del restaurante.
- Aumento de la satisfacción del cliente gracias a la rapidez y eficiencia en el servicio.



Oportunidades:

- Aprovechar la tendencia a la digitalización en la industria gastronómica.
- Ofrecer una experiencia más personalizada y satisfactoria a los clientes.
- Mejorar la imagen del restaurante como un lugar moderno y tecnológicamente avanzado.
- Posibilidad de expandir el negocio a través de la integración con aplicaciones de delivery y reservaciones en línea.

Debilidades:

- Costo de implementación y adquisición del sistema.
- Necesidad de capacitación del personal en el uso del sistema.
- Dependencia del sistema y posible interrupción en su funcionamiento.
- Dificultad en la integración con sistemas y tecnologías existentes.

Amenazas:

- Competencia de otros restaurantes que ya han implementado soluciones similares.



- Cambios en las preferencias y expectativas de los clientes en relación con la tecnología.
- Desconfianza y resistencia al cambio por parte del personal y clientes.
- Riesgo de seguridad y privacidad de los datos.

Estas son los resultados que obtuve realizando el análisis FODA, a continuación, daré más detalles sobre cada punto de los anteriores:

Detalle de las fortalezas del proyecto:

1- Aumento de la eficiencia en la gestión de los procesos del restaurante: Al automatizar y optimizar los procesos, se reduce el tiempo y esfuerzo necesario para llevar a cabo tareas como la toma de pedidos, la gestión de inventarios, la generación de facturas, entre otros.

2- Mejora en la toma de decisiones gracias a la obtención de datos y estadísticas precisas y actualizadas: El sistema permitirá recopilar y analizar datos relevantes sobre el rendimiento del restaurante, lo que permitirá tomar decisiones basadas en información objetiva y actualizada.



3- Reducción de los errores y demoras en la atención al cliente: Al automatizar y optimizar los procesos, se reduce la posibilidad de errores humanos y se acelera el tiempo de atención al cliente, lo que conduce a una experiencia más satisfactoria para el cliente.

4- Mejora en la coordinación entre las diferentes áreas del restaurante: El sistema permitirá una comunicación más eficiente y coordinación entre las diferentes áreas del restaurante, lo que resulta en una operación más fluida y sin interrupciones.

5- Aumento de la satisfacción del cliente gracias a la rapidez y eficiencia en el servicio: Al optimizar los procesos y mejorar la atención al cliente, se aumenta la satisfacción del cliente y, por lo tanto, su lealtad hacia el restaurante.

Detalle de las oportunidades del proyecto:

1- Aprovechar la tendencia a la digitalización en la industria gastronómica: En la actualidad, existe una creciente demanda por parte de los clientes por una experiencia más tecnológica y digital en los restaurantes, por lo que



implementar un sistema de gestión puede ser una oportunidad para diferenciarse y ofrecer un valor agregado.

2- Ofrecer una experiencia más personalizada y satisfactoria a los clientes: Al recopilar y analizar datos sobre los clientes, se puede ofrecer una experiencia más personalizada y satisfactoria, lo que resulta en un aumento de la lealtad y satisfacción de los clientes.

3- Mejorar la imagen del restaurante como un lugar moderno y tecnológicamente avanzado: La implementación de un sistema de gestión puede mejorar la imagen del restaurante como un lugar moderno y tecnológicamente avanzado, lo que puede atraer a nuevos clientes y aumentar la reputación del restaurante.

4- Posibilidad de expandir el negocio a través de la integración con aplicaciones de delivery y reservaciones en línea: Al integrar el sistema de gestión con aplicaciones de delivery y reservaciones en línea, se puede expandir el alcance del negocio y atraer a nuevos clientes.



Detalle de las debilidades del proyecto:

1- Costo de implementación y mantenimiento: El desarrollo y implementación de un sistema de gestión puede requerir una inversión significativa en términos de tiempo y dinero, además del costo de mantenimiento y actualización del sistema.

2- Curva de aprendizaje para los empleados: Los empleados pueden requerir un período de adaptación para aprender a utilizar el nuevo sistema, lo que puede resultar en interrupciones en la operación del restaurante durante el período de transición.

3- Dependencia tecnológica: Al depender del sistema de gestión, el restaurante puede verse afectado por posibles fallos o interrupciones en el sistema, lo que puede resultar en una mala experiencia para los clientes y una interrupción en la operación del restaurante.

4- Posible resistencia al cambio por parte de los empleados y clientes: Algunos empleados y clientes pueden resistirse al cambio y tener dificultades para



adaptarse a un nuevo sistema, lo que puede resultar en una reducción en la productividad y satisfacción.

5- Riesgo de seguridad de los datos: Al almacenar y procesar informaciones sensibles, como datos de los clientes y financieros, existe un riesgo de seguridad que debe ser considerado y mitigado para proteger la privacidad y seguridad de los datos.

Detalles de las amenazas del proyecto:

1- Competencia de otros restaurantes que hayan implementado sistemas de gestión: Al haber otros restaurantes que ya hayan implementado sistemas de gestión, el restaurante puede verse amenazado por una competencia más fuerte y una posible pérdida de clientes.

2- Cambios en las preferencias de los clientes: Las preferencias de los clientes pueden cambiar con el tiempo, y un sistema de gestión que fue implementado para satisfacer sus necesidades puede perder su relevancia o ser reemplazado por soluciones más actuales.



3- Regulaciones gubernamentales: Las regulaciones gubernamentales pueden cambiar con el tiempo, y un sistema de gestión que fue implementado de acuerdo con las regulaciones actuales puede requerir cambios para cumplir con las nuevas regulaciones.

4- Avances tecnológicos: Los avances tecnológicos pueden resultar en soluciones más avanzadas y eficientes, lo que puede reemplazar el sistema de gestión actual y requerir una inversión adicional en un sistema más actualizado.

5- Desastres naturales o incidentes de seguridad: Desastres naturales o incidentes de seguridad pueden interrumpir el funcionamiento del restaurante y afectar la operación del sistema de gestión. Por lo tanto, es importante tener un plan de contingencia en caso de un evento no deseado.

En conclusión, un sistema de gestión para un restaurante puede ofrecer muchas fortalezas, como una mayor eficiencia en la operación, una mejora en la satisfacción del cliente, y una mejora en la toma de decisiones. Sin embargo, también existen desafíos, como un costo de implementación y



mantenimiento alto, la curva de aprendizaje para los empleados, la dependencia tecnológica y la posible resistencia al cambio. Además, hay amenazas externas, como la competencia, cambios en las preferencias de los clientes, regulaciones gubernamentales y avances tecnológicos. Por lo tanto, es importante evaluar cuidadosamente todos los aspectos antes de tomar la decisión de implementar un sistema de gestión para un restaurante y considerar medidas para mitigar los riesgos y maximizar las fortalezas.

Elabore una matriz con los posibles riesgos del proyecto

La matriz de identificación de riesgos es una herramienta muy útil para identificar y evaluar los riesgos que pueden surgir durante el desarrollo del proyecto. La matriz permite que el equipo del proyecto visualice los riesgos potenciales de una manera clara y manejable y tome medidas para mitigarlos o eliminarlos.

La matriz creada para el proyecto de modernización de la gestión de restaurantes incluyó los cinco riesgos identificados. Cada riesgo incluye información sobre su nivel de riesgo, fuente de riesgo, descripción del riesgo y plan de contingencia.

El nivel de riesgo se refiere a la probabilidad de que ocurra un riesgo y el impacto que tendrá en el proyecto una vez que ocurra. La matriz divide los



riesgos en tres categorías: alto, medio y bajo. Los riesgos de alto nivel se refieren a riesgos con una alta probabilidad de ocurrencia y un impacto significativo en el proyecto, mientras que los riesgos de bajo nivel se refieren a riesgos con una baja probabilidad de ocurrencia y un impacto limitado en el proyecto.

Una fuente de riesgo es la causa u origen del riesgo. Por ejemplo, la incompatibilidad de hardware es una fuente de riesgo que puede afectar el correcto funcionamiento del sistema.

La descripción del riesgo describe los riesgos y sus consecuencias en detalle. Por ejemplo, una descripción de riesgo de incompatibilidad de hardware indica que es posible que el sistema no funcione correctamente si el hardware es incompatible.

Un plan de contingencia es un plan que se puede implementar para reducir o gestionar los riesgos. Por ejemplo, en los casos en que exista riesgo de incompatibilidad de hardware, el plan de contingencia puede incluir una evaluación preliminar del hardware y, si es necesario, actualizaciones.



En conclusión, la matriz de identificación de riesgos es una importante herramienta de gestión de proyectos porque puede identificar, evaluar y mitigar los riesgos de manera efectiva. Además, permite que los equipos de proyecto trabajen juntos para lograr los objetivos del proyecto y reducir el riesgo. Por lo tanto, es importante evaluar constantemente los riesgos y desarrollar planes de contingencia adecuados para mitigarlos o gestionarlos.

Descripción de la tabla de la matriz con los posibles riesgos del Proyecto.

Incompatibilidad de hardware y software: este riesgo se considera de bajo impacto porque el equipo de desarrollo se asegurará de que el software funcione con diferentes tipos de hardware y que los requisitos técnicos sean claros y fáciles de cumplir. Para evitar este riesgo, se recomienda realizar pruebas exhaustivas en diferentes sistemas operativos y hardware antes de lanzar el producto.

Cambios en los requisitos: pueden ocurrir cambios en los requisitos y objetivos específicos durante el proceso de desarrollo. Si esto sucede, es importante tener un plan de contingencia.



Retrasos en el desarrollo: Este riesgo se considera de impacto medio, ya que diversos factores como problemas técnicos, falta de recursos o cambios en los requisitos pueden causar retrasos en el desarrollo. Para evitar este riesgo, es importante crear un plan de proyecto y cronogramas claros, definir responsabilidades claras y mantener una comunicación constante entre todas las partes involucradas.

Problemas de Entity Framework ORM: este riesgo se considera de bajo impacto porque el equipo de desarrollo tiene experiencia trabajando con el marco y es muy consciente de sus fortalezas y debilidades. Para evitar este riesgo, se recomienda proporcionar una buena documentación y una planificación cuidadosa para garantizar que el marco se utilice de la manera más eficaz.

Incompatibilidad con otros sistemas: Este riesgo se considera de impacto medio debido a que el sistema debe ser compatible con otros sistemas utilizados por el restaurante, como los sistemas de facturación y reserva. Se recomienda realizar pruebas exhaustivas y una buena comunicación con los equipos responsables de estos otros sistemas para evitar este riesgo.



Consideraciones de seguridad: este riesgo se considera un impacto moderado porque el sistema necesita proteger la información confidencial, como la información financiera y la información del cliente. Para evitar este riesgo, se recomienda utilizar métodos de seguridad sólidos y realizar pruebas exhaustivas para garantizar que la información esté protegida.

Problemas del modelo de arquitectura: este riesgo se considera de bajo impacto porque el equipo de desarrollo tiene experiencia en el uso de un modelo de arquitectura en capas y conoce sus ventajas y desventajas. Se recomienda una buena documentación y una planificación cuidadosa para evitar este riesgo.



Tabla de una matriz con los posibles riesgos del Proyecto

Nivel de Riesgo	Fuente de Riesgo	Descripción del Riesgo	Plan de Contingencia
Alto	Incompatibilidad del hardware.	El sistema puede no funcionar correctamente debido a la incompatibilidad del hardware del restaurante	Evaluación previa del hardware y actualización si es necesario
Medio	Falta de recursos financieros	El presupuesto puede no ser suficiente para cubrir todos los costos del proyecto	Revisión y ajuste del presupuesto y búsqueda de financiamiento adicional
Bajo	Inconsistencia de los datos	Los datos pueden ser inconsistentes o incorrectos debido a una mala integración del sistema	Implementación de un sistema de verificación de datos y un proceso de corrección de errores
Medio	Cambios en los requisitos del usuario	Los requisitos del usuario pueden cambiar a lo largo del desarrollo del proyecto	Comunicación continua con el cliente y ajuste del plan de proyecto en consecuencia



Bajo	Retraso en la entrega	El proyecto puede retrasarse debido a problemas técnicos o humanos	Implementación de un sistema de seguimiento del progreso y una planificación detallada del tiempo
------	-----------------------	--	---

Hay varios riesgos asociados con el proyecto que propusimos. A continuación, se describen algunos de los riesgos más comunes:

Inconsistencia de datos: en un sistema de gestión de restaurantes, la coherencia y la integridad de los datos son extremadamente importantes. Si los sistemas no logran garantizar la precisión y consistencia de los datos, pueden ocurrir errores en el procesamiento de pedidos y la gestión financiera, lo que puede afectar negativamente la satisfacción del cliente y la rentabilidad.

Rendimiento del sistema: es importante asegurarse de que el sistema funcione al máximo rendimiento y sea capaz de manejar la cantidad esperada de transacciones. Si el sistema no es lo suficientemente rápido o eficiente, puede haber retrasos en el procesamiento de pedidos y la comunicación entre las diferentes áreas del restaurante.



Integración con otros sistemas: El sistema debe estar integrado con otros sistemas existentes como sistemas de facturación y sistemas de reservas. Si estos sistemas no están debidamente integrados, el procesamiento de pedidos y la gestión financiera pueden fallar.

Complejidad del sistema: los sistemas pueden ser complejos y requieren una curva de aprendizaje importante para los usuarios. Si la curva de aprendizaje es demasiado alta, es posible que los usuarios no adopten el sistema y que no se alcancen los objetivos de modernización de la gestión.

Cambios en los requisitos comerciales: los requisitos comerciales cambian con el tiempo. La flexibilidad del sistema y la capacidad de adaptarse a los cambios son importantes.

Problemas de calidad del software: el uso de diferentes tecnologías y sistemas puede dificultar la resolución de problemas y garantizar la calidad del software. Por lo tanto, es importante asegurarse de que los desarrolladores tengan experiencia con estas tecnologías y puedan solucionar problemas de manera efectiva.



Para evitar estos riesgos, es importante implementar un programa sólido de calidad del software que incluya pruebas exhaustivas, garantice que los desarrolladores tengan experiencia con la tecnología que se utiliza y tenga planes de contingencia en caso de que cambien los requisitos. También es importante trabajar con el equipo de seguridad de la información para garantizar que la información esté protegida y garantizar la disponibilidad del sistema a través de las pruebas de los usuarios y la incorporación de los comentarios de los usuarios.

¿Qué pasa si todo sale bien y se cumple todo lo que se tiene planeado, pues fácil, si el proyecto se ejecuta exitosamente, se lograrán los siguientes resultados:

- ❖ **Mayor eficiencia en la gestión de restaurantes:** un sistema integrado para gestionar diferentes aspectos de un restaurante proporcionará una gestión más eficiente y eficaz, lo que dará como resultado un mejor control y una mayor toma de decisiones. preciso.
- ❖ **Comunicación cruzada mejorada:** al usar un sistema de comunicación entre los miembros de la empresa, se logra una mejor coordinación y cooperación, lo que resulta en un mejor desempeño y servicio al cliente.
- ❖ **Flexibilidad del menú:** con un menú digital y flexible, los clientes tendrán una experiencia más agradable y podrán elegir entre una variedad de opciones, lo que a su vez conducirá a una mayor satisfacción y una mejor retención de clientes.



- ❖ **Mejora financiera:** Mediante un sistema de facturación y gestión financiera se logra un mejor control de ingresos y gastos, lo que permite una mejor planificación y toma de decisiones financieras.
- ❖ **Reservas mejoradas:** con un sistema de reservas, se puede lograr una mejor coordinación y planificación, lo que resulta en un mejor servicio al cliente y una mejor utilización de los recursos.

Si este proyecto sale bien y cumple con los requisitos, modernizará las operaciones del restaurante y las operaciones generales, lo que conducirá a un mejor servicio al cliente, una mejor coordinación y cooperación entre los miembros de la empresa, y una mejor planificación financiera y toma de decisiones.

Elabore una estimación gastos de su proyecto

La estimación de gastos para un proyecto de software para un restaurante depende de muchos factores, como el tamaño y la complejidad del proyecto, la tecnología utilizada, el número de desarrolladores y otros recursos necesarios, y la duración del proyecto. Hay varios factores que deben tenerse en cuenta al estimar los gastos de un proyecto de software de un restaurante, algunos de los cuales incluyen:



Complejidad del proyecto: La complejidad del proyecto es un factor clave en la estimación de los costos. Proyectos más complejos requieren más tiempo, recursos y esfuerzo para desarrollarse, lo que a su vez aumenta el costo total.

Equipo de desarrollo: El tamaño y la experiencia del equipo de desarrollo también pueden afectar el costo total. Un equipo más grande o con más experiencia puede ser más eficiente y tener un impacto positivo en la estimación de los costos.

Tiempo de desarrollo: El tiempo de desarrollo es un factor crítico en la estimación de los costos. Un proyecto de corto plazo puede requerir menos recursos y tener un costo total más bajo que un proyecto de largo plazo.

Ubicación geográfica: La ubicación geográfica puede afectar el costo total debido a diferencias en los precios locales de mano de obra, materiales y servicios.



Calidad y requisitos: La calidad y los requisitos deseados también pueden afectar el costo total. Un proyecto con requisitos más altos o un mayor enfoque en la calidad requerirá más tiempo y recursos, lo que aumentará el costo total.

Desarrollo ágil vs. Desarrollo tradicional: El enfoque adoptado para el desarrollo del software, ya sea ágil o tradicional, también puede afectar el costo total. El enfoque ágil puede ser más flexible y adaptable a cambios, pero también puede requerir más tiempo y recursos.

Costos indirectos: Los costos indirectos, como viajes, alojamiento y transporte, también deben tenerse en cuenta al estimar los costos totales.

Estamos utilizando una estimación basada en análisis de costo-beneficio, en la cual se evalúan los costos estimados de los diferentes componentes del proyecto y se estima un rango de costo probable para cada uno. Este tipo de estimación es una técnica común en la planificación de proyectos de software, y es una buena manera de obtener una visión general de los costos aproximados asociados con un proyecto. Sin embargo, es importante tener en cuenta que las estimaciones de costos siempre tienen cierta incertidumbre, por lo que es importante realizar un seguimiento cuidadoso



de los costos reales y ajustar la estimación según sea necesario a medida que avanza el proyecto.

Estimación por juicio de expertos

La estimación de costos que hemos proporcionado se parece más a una estimación por juicio de expertos. La estimación por juicio de expertos implica que un grupo de expertos en el tema evalúan los costos probablemente asociados con un proyecto de software, basándose en su experiencia previa y su conocimiento del proyecto y la industria en general. La estimación que proporcioné es una combinación de este enfoque y una estimación basada en análisis de costo-beneficio, en la cual se estiman los costos de los componentes individuales del proyecto y se realiza una estimación aproximada del costo total.

En una estimación por juicio de expertos, un grupo de expertos en el tema evalúan los costos probablemente asociados con un proyecto de software. Estos expertos pueden incluir gerentes de proyectos, ingenieros de software, arquitectos de sistemas, entre otros. Estos expertos utilizan su experiencia previa y su conocimiento del proyecto y la industria en general para realizar una evaluación cuidadosa de los costos asociados con el proyecto.

La estimación que proporcionamos es una combinación de este enfoque y una estimación basada en análisis de costo-beneficio. En esta estimación, se



estiman los costos de los componentes individuales del proyecto, como el desarrollo de software, la adquisición de hardware y software, la contratación de personal, entre otros. Estas estimaciones se realizan utilizando una combinación de métodos, incluyendo la investigación de precios y la evaluación de la complejidad y el alcance del trabajo requerido para cada componente.

Una vez que se han estimado los costos de los componentes individuales, se realiza una estimación aproximada del costo total para el proyecto. Esta estimación puede ser revisada y ajustada por los expertos, teniendo en cuenta factores como la incertidumbre, la contingencia y otros factores que pueden afectar el costo final.

Es importante tener en cuenta que, aunque esta combinación de enfoques proporciona una buena estimación de los costos asociados con un proyecto de software, siempre existe cierta incertidumbre y es posible que los costos reales varíen de la estimación original. Por lo tanto, es importante realizar un seguimiento cuidadoso de los costos reales y ajustar la estimación según sea necesario a medida que avanza el proyecto.



Concepto	Descripción	Costo estimado
Desarrollo de software	Desarrollo de la aplicación de software para el restaurante	\$50,000 - \$70,000
Adquisición de hardware	Adquisición de dispositivos necesarios para el uso de la aplicación, como tabletas o terminales de punto de venta	\$15,000 - \$25,000
Adquisición de software	Adquisición de software necesario para el funcionamiento de la aplicación, como sistemas operativos o licencias de software	\$5,000 - \$10,000
Contratación de personal	Contratación de personal para implementar y mantener la aplicación, como ingenieros de software o técnicos de soporte	\$20,000 - \$30,000
Capacitación	Capacitación del personal en el uso de la aplicación y del hardware	\$2,500 - \$5,000
Costos indirectos	Costos indirectos asociados con el proyecto, como viajes, alojamiento y transporte	\$5,000 - \$10,000
Contingencia	Costos adicionales previstos para hacer frente a imprevistos o desafíos imprevistos	\$10,000 - \$20,000

Elabore un esquema de perfiles técnicos necesarios para su proyecto.

Project Manager

- Conocimiento en metodologías de gestión de proyectos de software, como Agile o Scrum.
- Experiencia en la gestión de equipos de desarrollo de software.
- Habilidad para comunicarse efectivamente con el equipo y los clientes.
- Habilidad para el liderazgo.
- Habilidad para motivar.



- Habilidades de comunicación.
- Dominio en el manejo de conflictos.
- Construcción de confianza.
- Manejo en la toma de decisiones.
- Eficiencia en la organización.
- Dominio en gestión del tiempo.

Arquitecto de software

- Conocimiento en arquitectura de software y patrones de diseño.
- Experiencia en la definición e implementación de arquitecturas de software a gran escala.
- Conocimiento de tecnologías y herramientas de desarrollo de software.

Desarrollador Front-End

- Dominio de los lenguajes HTML5 y CSS3
- Experiencia en el desarrollo de aplicaciones web.
- Conocimiento de JavaScript
- Conocimiento de optimización de sitios web para usuarios y motores de búsqueda.
- Experiencia en el desarrollo de aplicaciones multiplataforma.



- Conocimiento de herramientas de control de versiones, como Git.
- Experiencia actual en diseño responsive y compatible con dispositivos móviles

Desarrollador Back-End

- Conocimiento de uno o más lenguajes de programación, como Java, Python, C++, etc.
- Experiencia en el desarrollo de aplicaciones de software.
- Conocimientos sobre bases de datos.
- Dominio de algoritmos.
- Conocimiento de herramientas de control de versiones, como Git.

Tester de software

- Conocimiento en pruebas de software y metodologías de pruebas.
- Experiencia en la ejecución de pruebas manuales y automatizadas.
- Conocimiento de herramientas de pruebas, como Selenium.



DevOps Engineer

- Conocimiento en automatización de despliegues y gestión de infraestructuras.
- Experiencia en la configuración y gestión de servidores y sistemas.
- Conocimiento de herramientas de DevOps, como Docker y Ansible.



Bibliografía:

<https://www.iebschool.com/blog/project-manager-agile-scrum/>

<https://sq.com.mx/revista/04/el-arquitecto-software-habilidades-necesarias>

<https://dtagency.tech/10-habilidades-y-conocimientos-clave-de-un-desarrollador-front-end/>

<https://www.airavirtual.com/blog/habilidades-back-end-dev>

<https://www.computerweekly.com/es/consejo/Habilidades-del-ingeniero-DevOps-necesarias-para-el-despliegue-continuo>

<https://openwebinars.net/blog/tester-software-tareas-principales/>

<https://asana.com/es/resources/feasibility-study>

<https://softwarepara.net/restaurantes/>

<https://www.piranirisk.com/es/blog/asi-puedes-hacer-una-matriz-de-riesgos-para-tu-empresa>