**RÁPIDO&SABROSO**

**(ATAM) Architecture Tradeoff Analysis Method**

**Versión 3.0**

**Identificación de Documento**

|  |  |
| --- | --- |
| **Identificación** |  |
| **Proyecto** | Rápido & Sabroso |
| **Versión** | 2.1 |

|  |  |
| --- | --- |
| **Documento mantenido por** |  |
| **Fecha de última revisión** | 01/12/2024 |
| **Fecha de próxima revisión** | 01/08/2024 |

|  |  |
| --- | --- |
| **Documento aprobado por** | JUAN GANA |
| **Fecha de última aprobación** | 01/12/2024 |

**Historia de Revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 27/11/2021 | 1.0 | Inicio informe | Franco Alvarez |
| 28/11/2021 | 2.0 | Continuación informe | Gabriel Morales |
| 29/11/2021 | 2.1 | Revisión y Aprobación | Pablo Carvajal |
| 01/12/2021 | 3.0 | Término informe | Johann Vega |

**Tabla de Contenidos**

[**1**](#_heading=h.gjdgxs) **Introducción 3**

[1.1](#_heading=h.30j0zll) Contexto del Problema 3

[1.2](#_heading=h.1fob9te) Propósito del Documento 3

[1.3](#_heading=h.3znysh7) Definiciones, acrónimos y abreviaciones 3

[1.4](#_heading=h.2et92p0) Referencias 3

[1.5](#_heading=h.tyjcwt) Resumen ejecutivo 3

[**2**](#_heading=h.3dy6vkm) **Presentación de los objetivos de negocio 4**

[2.1](#_heading=h.1t3h5sf) Objetivos de Negocio 4

[2.2](#_heading=h.4d34og8) Funcionalidades de Alto nivel 4

[2.3](#_heading=h.2s8eyo1) Descripción de los Stakeholders del proyecto 4

[**3**](#_heading=h.17dp8vu) **Presentación de la Arquitectura 5**

[3.1](#_heading=h.3rdcrjn) Metas de la arquitectura 5

[3.2](#_heading=h.26in1rg) Patrones y Arquetipos de Diseño. 5

[3.3](#_heading=h.lnxbz9) Descripción detallada de la Arquitectura de Solución. 5

[**4**](#_heading=h.35nkun2) **Análisis de la Arquitectura propuestas 6**

[4.1](#_heading=h.1ksv4uv) Árbol de Utilidad y priorización de Escenarios 6

[4.2](#_heading=h.2jxsxqh) Objetivos de calidad y decisiones arquitectónicas 6

[4.3](#_heading=h.z337ya) Riesgo y No Riesgos 6

[*4.3.1*](#_heading=h.3j2qqm3) *Riesgos 6*

[*4.3.2*](#_heading=h.1y810tw) *No Riesgos 6*

[4.4](#_heading=h.4i7ojhp) Puntos Sensibles y TradeOff 6

[*4.4.1*](#_heading=h.2xcytpi) *Puntos Sensibles 6*

[*4.4.2*](#_heading=h.1ci93xb) *TradeOff 6*

[4.5](#_heading=h.3whwml4) Puntos únicos de fallo (SPOF – Single Point of Failure) 7

[**5**](#_heading=h.2bn6wsx) **Conclusiones y Recomendaciones del Análisis de Arquitectura 8**

[5.1](#_heading=h.qsh70q) Conclusiones 8

[5.2](#_heading=h.3as4poj) Recomendaciones 8

1. **Introducción**
   1. **Contexto del Problema**

La página web busca ofrecer una plataforma digital para facilitar el acceso a las comidas rápidas, permitiendo a los usuarios seleccionar y pagar por sus alimentos a través de una página web intuitiva. Esta plataforma debe permitir a los usuarios filtrar precios de menús de forma eficiente y realizar pagos con distintos métodos. El sistema debe ser de fácil acceso tanto en dispositivos móviles como en computadoras de escritorio, proporcionando una experiencia de usuario simple y rápida.

* 1. **Propósito del Documento**

El propósito de este documento es detallar la arquitectura del sistema que permitirá a los usuarios de *Rápido y Sabroso* filtrar precios de comidas rápidas, seleccionar opciones de pago y realizar pedidos de forma eficiente y confiable. Este informe también analiza los distintos componentes arquitectónicos de la solución propuesta.

* 1. **Definiciones, acrónimos y abreviaciones**

|  |  |
| --- | --- |
| **ACRÓNIMO** | **DESCRIPCIÓN** |
| *ERS* | Especificación de requisitos de software. Es una descripción detallada del comportamiento esperado del sistema. |
| *DAS* | Documento de Arquitectura de Software. Documento que describe la arquitectura técnica de la solución. |
| *UI* | Interfaz de usuario. Referencia a los componentes visuales con los cuales interactúan los usuarios. |
| *API* | Interfaz de programación de aplicaciones. Permite que el sistema se comunique con servicios externos. |

* 1. **Referencias**

A continuación, se listan las referencias a otros documentos del proyecto:

* **Documento ERS**
* **Documento DAS**
  1. **Resumen ejecutivo**

El proyecto *Rápido y Sabroso* es una plataforma digital enfocada en ofrecer a los usuarios una experiencia simple y rápida para seleccionar y comprar comidas rápidas. La solución incluirá una página web que permita a los usuarios filtrar menús por precio y tipo de comida, realizar pagos online y gestionar su historial de compras. Con una interfaz intuitiva, el sistema garantizará un proceso eficiente desde el filtrado de precios hasta la confirmación del pedido. Además, ofrecerá distintas opciones de pago y soporte para una experiencia de compra sin interrupciones.

1. **Presentación de los objetivos de negocio**
   1. **Objetivos de Negocio**

Los principales objetivos de negocio que impulsan el diseño y desarrollo del Proyecto *Rápido&Sabroso*, son los siguientes:

* **Objetivo 1:**

Crear una plataforma web para que los usuarios puedan filtrar y ordenar comidas rápidas según sus preferencias de precio y tipo de comida.

* **Objetivo 2:**

Proveer diversas opciones de pago (tarjetas, billeteras electrónicas) para garantizar la facilidad de pago.

* **Objetivo 3:**

Fomentar una experiencia de compra intuitiva, reduciendo el tiempo necesario para realizar una orden.

* 1. **Funcionalidades de Alto nivel**

 **Filtrado de Precios**: Permite a los usuarios aplicar filtros para ver comidas dentro de un rango de precios determinado, mejorando la experiencia de selección.

 **Pago**: Integración de un sistema de pago online que permita a los usuarios pagar mediante tarjetas de crédito, débito o plataformas de pago.

 **Gestión de Pedidos**: Los usuarios pueden realizar pedidos, ver su historial y consultar el estado de sus pedidos en tiempo real.

* 1. **Descripción de los Stakeholders del proyecto**

 **Gabriel Morales**: Jefe de Proyecto de *Rápido y Sabroso*, encargado de coordinar y supervisar las actividades del equipo de desarrollo.

 **Johan Vega**: Desarrollador Backend, encargado de la implementación de la lógica de negocio y la base de datos.

 **Pablo Carvajal**: Desarrolladora Frontend, responsable de la creación de la interfaz de usuario.

 **Franco Alvarez**: Responsable de Marketing, encargado de la difusión de la plataforma y análisis de usuarios.

1. **Presentación de la Arquitectura**

A continuación, se describen los aspectos arquitectónicos de la solución propuesta para el Proyecto *Rápido&Sabroso*.

* 1. **Metas de la arquitectura**

El diseño de la arquitectura tiene como objetivo garantizar que la plataforma sea fácil de usar, escalable y segura. Las metas incluyen:

* **Escalabilidad**: Poder manejar un número creciente de usuarios sin que se degrade el rendimiento.
* **Seguridad**: Implementar mecanismos de pago seguros y proteger la información del usuario.
* **Usabilidad**: Crear una interfaz intuitiva que facilite la navegación.
  1. **Patrones y Arquetipos de Diseño.**

**Arquitectura en Capas**: La arquitectura está organizada en capas horizontales para separar las distintas responsabilidades del sistema, mejorando la mantenibilidad y escalabilidad.

**Patrón de Diseño de Estrategia**: El sistema implementará una estrategia flexible para la selección de filtros y métodos de pago, permitiendo modificaciones rápidas sin afectar el sistema global.

* 1. **Descripción detallada de la Arquitectura de Solución.**

El sistema estará compuesto por los siguientes módulos principales:

* **Módulo de Login y Registro**: Los usuarios podrán crear una cuenta o iniciar sesión para gestionar sus compras y acceder al historial de pedidos.
* **Módulo de Filtrado de Precios**: Permite a los usuarios seleccionar opciones según el precio y tipo de comida.
* **Módulo de Gestión de Pagos**: Integración con pasarelas de pago para la realización de pagos seguros.
* **Módulo de Historial y Gestión de Pedidos**: Los usuarios pueden ver su historial de pedidos anteriores y realizar nuevos pedidos.

**3.4 implementación del sistema.**

El administrador del modelo deberá hacer la búsqueda y compra del dominio preferido que se utilizará para la implantación de la página web. También se deberá buscar un hosting adecuado que contenga el suficiente almacenamiento y que soporte toda la carga de datos que irá generando la página web ya sea con la compra de productos, registro de los clientes y contratación de servicios que ofrece la página web. Adicionalmente será el encargado de suministrar el equipo computacional correspondiente a los involucrados en el soporte del sistema.

**Herramientas a Utilizar:**

* **Framework Django**: Para el desarrollo backend de la plataforma web.
* **MySQL**: Para gestionar la base de datos de usuarios, menús y pedidos.
* **Stripe / PayPal**: Para la integración de pagos online.

**3.5 Capacitación**

Se realizará capacitación a los administradores y personal de soporte en el uso del sistema, incluyendo cómo gestionar la base de datos de productos, procesar pagos y manejar incidencias.

**3.6 Programa de actividades**

| **Revisor** | **Estado** | **Notas** |
| --- | --- | --- |
| **comité proyecto** | Aprobada | Revisión de los entregables del sprint actual. |
| **Comisión del cliente** | Aprobada | Aprobación de los requerimientos y validación de los mockups. |
| **Administrador de Base de Datos** | Aprobada | |  | | --- | | Revisión y configuración de la base de datos. |  |  | | --- | |  | |
| **Scrum Master** | Aprobada | |  | | --- | | Revisión del avance en la implementación del backend. |  |  | | --- | |  | |
| **Desarrollador Frontend** | Aprobada | |  | | --- | | Implementación de las pantallas y pruebas de usabilidad. |  |  | | --- | |  | |
| **Desarrollador Backend** | Aprobada | |  | | --- | | Integración de la lógica de negocio para filtrar precios. |  |  | | --- | |  | |

| **Revisor** | **Estado** | **Notas** |
| --- | --- | --- |
| **Product Owner** | Aprobada | |  | | --- | | Validación continua del producto con los stakeholders. |  |  | | --- | |  | |
| |  | | --- | | **Control de Calidad** |  |  | | --- | |  | | En curso | |  | | --- | | Pruebas de control de calidad en las funcionalidades del carrito de compras. |  |  | | --- | |  | |
| **Gestión de Proyecto** | Aprobada | |  | | --- | | Supervisión del avance general del proyecto. |  |  | | --- | |  | |
| **Supervisor de Proyecto** | Aprobada | |  | | --- | | Supervisión de la implementación y seguimiento de tareas. |  |  | | --- | |  | |
| **Personal de Soporte** | En curso | Capacitación y soporte para el equipo de backend y frontend. |

**3.7 Análisis de riesgo**

Algunos de los riesgos que podrían surgir durante la implementación de Rápido y Sabroso incluyen:

* **Riesgos de Implementación**:
  + Problemas con la integración de pagos.
  + Caídas en el servidor por alta demanda.
  + Falta de seguridad en el manejo de datos personales.
* **Riesgos No Críticos**:
  + Problemas menores con la interfaz de usuario en dispositivos móviles.

**3.8 Método de implementación**

## Fase de Desarrollo: En esta fase se construirá la plataforma web, configurando la base de datos, los módulos de filtrado de precios, pago y gestión de pedidos.

## Fase de Pruebas: Se realizarán pruebas unitarias y de integración para garantizar que el sistema funcione correctamente.

## Fase de Puesta en Marcha: Se lanzará la plataforma al público general, monitoreando el rendimiento y realizando ajustes según sea necesario.

## Plan de implementación

## *Para que el plan de implantación se desarrolle de la mejor manera es primordial tener un plan de contingencia que permita la vuelta atrás en caso de que la implementación falle en alguna de sus etapas, en esta parte del documento se deben describir las actividades que se llevarán a cabo en caso de tener la necesidad de volver atrás el proceso de implementación.*

**1. Fallo en el Sistema de Pagos**

* **Riesgo**: Los usuarios no podrán realizar pagos correctamente debido a un fallo en la integración con el sistema de pagos.
* **Probabilidad**: Moderada (existen múltiples proveedores de pago y algunos podrían presentar fallos técnicos).
* **Impacto**: Alto (sin pagos, no se podrá completar el proceso de compra, afectando directamente la funcionalidad de la aplicación).
* **Estrategia**: Implementar un sistema de backup que permita procesar pagos de forma manual en caso de fallos del sistema de pagos principal. Además, contar con una integración secundaria con otro proveedor de pagos como opción de respaldo.
* **Responsable**: Administrador del sistema y Equipo de desarrollo Backend.
* **2. Desactualización de Precios en el Menú**
* **Riesgo**: Los precios de los productos mostrados en la plataforma no se actualizan correctamente desde los restaurantes asociados.
* **Probabilidad**: Alta (es posible que los restaurantes no actualicen los precios de manera constante).
* **Impacto**: Medio (puede generar confusión en los usuarios y reclamaciones si los precios no coinciden con los reales).
* **Estrategia**: Implementar un sistema de notificación automática para que los restaurantes actualicen los precios y productos. En caso de que no se actualicen, el sistema mostrará un mensaje de advertencia.
* **Responsable**: Administrador de la base de datos y Equipo de Integración de restaurantes.
* **3. Problemas de Conexión a Internet**
* **Riesgo**: Los usuarios no pueden acceder a la plataforma debido a problemas de conexión a internet o red inestable.
* **Probabilidad**: Moderada (dependiendo de la ubicación geográfica de los usuarios, puede haber fluctuaciones en la conectividad).
* **Impacto**: Medio (puede afectar la experiencia del usuario y generar pérdida de transacciones, pero no bloquea completamente el uso de la aplicación).
* **Estrategia**: Desarrollar una funcionalidad de "caché" que permita a los usuarios navegar y realizar transacciones limitadas sin conexión. Informar al usuario sobre la conectividad mediante mensajes en la app.
* **Responsable**: Equipo de desarrollo Frontend y Equipo de soporte técnico.
* **4. Fallo en la Base de Datos**
* **Riesgo**: La base de datos no está disponible o presenta fallos de rendimiento, afectando el acceso a los menús, el historial de pedidos y los pagos.
* **Probabilidad**: Baja (siempre y cuando la base de datos esté correctamente optimizada y respaldada).
* **Impacto**: Alto (afectaría gravemente a la operativa del sistema, impidiendo el acceso a la información y la realización de transacciones).
* **Estrategia**: Implementar un sistema de respaldos automáticos y pruebas de recuperación ante desastres para restaurar la base de datos en caso de fallo. Además, contar con un servidor de base de datos redundante para minimizar el tiempo de inactividad.
* **Responsable**: Administrador de bases de datos y Equipo de infraestructura de TI.

1. **Análisis de la Arquitectura propuestas**

Esta sección describe los escenarios funcionales y no funcionales que han sido evaluados y priorizados durante el análisis de la arquitectura del proyecto *Rapido&Sabroso*.

1. **Casos de Uso y Escenarios Funcionales Prioritarios:**
2. **Registro de Usuario**: Permite a los usuarios crear una cuenta en la aplicación para poder hacer pedidos.
   * **Escenario**: El usuario ingresa sus datos personales, configura sus preferencias de pago y recibe un mensaje de confirmación.
   * **Atributos de Calidad**: Usabilidad, seguridad.
3. **Selección de Comida**: Los usuarios pueden navegar a través del menú de comida rápida, seleccionando productos para añadir al carrito.
   * **Escenario**: El usuario puede ver el menú actualizado en tiempo real, con descripciones, imágenes y precios.
   * **Atributos de Calidad**: Disponibilidad, rendimiento.
4. **Carrito de Compras y Checkout**: El usuario revisa su carrito y realiza el proceso de pago.
   * **Escenario**: El sistema verifica los productos seleccionados y calcula el precio total, con un proceso de pago seguro.
   * **Atributos de Calidad**: Fiabilidad, seguridad.
5. **Seguimiento del Pedido**: Los usuarios pueden ver el estado de su pedido en tiempo real.
   * **Escenario**: El sistema muestra el estado de preparación y envío del pedido, con estimaciones de tiempo de entrega.
   * **Atributos de Calidad**: Rendimiento, usabilidad.
6. **Sistema de Notificaciones**: Los usuarios reciben actualizaciones sobre el estado de su pedido, promociones y eventos.
   * **Escenario**: El sistema envía notificaciones en tiempo real sobre los cambios en el estado del pedido.
   * **Atributos de Calidad**: Fiabilidad, rendimiento.

**4.5 Árbol de Utilidad y Priorización de Escenarios**

En este análisis, se presentan los objetivos de calidad asociados a cada escenario, clasificándolos según su prioridad y el riesgo que conlleva cada uno.

**Árbol de Utilidad y Priorización**

| **Escenario** | **Objetivos de Calidad** | **Importancia** | **Riesgo** | **Clasificación Final** |
| --- | --- | --- | --- | --- |
| **Registro de Usuario** | Usabilidad, Seguridad | Alta | Bajo | Alta |
| **Selección de Comida** | Disponibilidad, Rendimiento | Alta | Moderado | Alta |
| **Carrito de Compras y Checkout** | Fiabilidad, Seguridad | Alta | Alto | Alta |
| **Seguimiento del Pedido** | Rendimiento, Usabilidad | Media | Bajo | Media |
| **Sistema de Notificaciones** | Fiabilidad, Rendimiento | Media | Moderado | Media |

* 1. **Riesgo y No Riesgos**

**Riesgos:**

* **Validación de pagos y/o recargas:** Existen posibles riesgos al integrar sistemas de pago, como pagos rechazados o fallas en el procesamiento, que podrían generar una mala experiencia para el usuario.
* **Interacción con el servicio de restaurantes:** Pueden existir problemas de comunicación entre la plataforma y los restaurantes que actualizan sus menús y precios, lo que puede llevar a mostrar precios desactualizados.
* **Tiempo de respuesta de la base de datos:** Si la base de datos no es optimizada correctamente, puede haber tiempos de respuesta lentos, afectando la experiencia del usuario.

**No Riesgos:**

* **Diseño de la aplicación:** No se anticipan problemas significativos con el diseño, ya que se ha diseñado teniendo en cuenta la simplicidad y accesibilidad.
* **Historial de pedidos:** Se garantiza que el historial de pedidos y transacciones del usuario se almacenará correctamente y se podrá acceder de forma automática.
  1. **Puntos Sensibles y TradeOff**

**Puntos Sensibles:**

* **Actualizaciones del sistema**: La actualización del sistema puede generar problemas si no se testean adecuadamente las nuevas funcionalidades, especialmente cuando se realizan cambios a nivel de base de datos o de pagos.
* **Interacción entre el frontend y backend**: La eficiencia de las solicitudes entre el frontend (interfaz de usuario) y el backend (servidor) es un aspecto crítico para garantizar que el tiempo de carga y la navegación sean óptimos.
* **Base de datos**: La correcta implementación y optimización de la base de datos es vital para el rendimiento del sistema, especialmente en lo que respecta a la consulta de precios, disponibilidad de productos y gestión de usuarios.

Trade-offs:

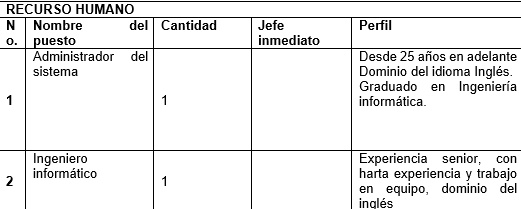
* **Base de datos relacional vs. no relacional**: Optamos por una base de datos relacional (MySQL) debido a que se adapta mejor a la estructura de datos de los productos y usuarios, aunque esto puede implicar una mayor complejidad en su escalabilidad futura.
* **Aplicación web y móvil**: Decidimos priorizar la versión web inicialmente, ya que la mayoría de los usuarios accederán al servicio desde computadoras, y la versión móvil se desarrollará en una segunda fase. Sin embargo, la versión web también debe ser totalmente adaptable a móviles para no perder usuarios.
  1. **Puntos únicos de fallo (SPOF – Single Point of Failure)**

**Interacción con proveedores de comida**: Si uno de los proveedores de comida (restaurantes) experimenta un problema técnico o no actualiza sus precios de manera correcta, los usuarios podrían ver información errónea.

**Base de datos**: La base de datos es un componente clave del sistema, y si falla, puede afectar la disponibilidad de precios o el proceso de compra de los usuarios.

**Redes de pago**: Si hay fallas en la integración con sistemas de pago (por ejemplo, tarjetas de crédito, PayPal), los usuarios no podrán completar sus compras, lo que afectará directamente la operación.

**5**. **Necesidades de recursos para implementar**



## Recurso tecnológico

**Dependencias**: Servidores para hosting de la aplicación, base de datos y servidor de pruebas.

**Herramientas de desarrollo**: Visual Studio Code, Django, MySQL, herramientas de análisis de rendimiento.

**Equipos**: Computadoras de desarrollo, dispositivos móviles para pruebas, equipos para capacitación.

**Conexión de red**: Conexión estable a internet para realizar pruebas de carga y funcionamiento en entornos reales.

**Equipo multimedia**: Proyector para capacitaciones, impresoras para generar manuales y apuntes.

**Conclusiones y Recomendaciones del Análisis de Arquitectura**

**Conclusiones**

En conclusión, en base al análisis realizado por nuestro equipo sobre la arquitectura del software concluimos que la evaluación realizada en este informe ATAM permite adecuar la arquitectura con respecto a los atributos de calidad especificados para el sistema, donde sin duda hay posibilidad de mejora, como identificar mejor y más puntos sensibles y realizar un mejor análisis de riesgo del software.

**Recomendaciones**

Como equipo, creemos que nuestras recomendaciones son, un estudio previo de los puntos a tratar para tener una mejor idea de lo que hay que añadir y mejorar, descripción de la información más amplia y detallada, mejor visualización de los diagramas, no tener miedo de guiarse en base a ejemplos de internet e ir trabajando cada día para así nunca quedar atrás y mejorar todos los días.