



Drink coffe

Página 1 de 15

Propuesta Comercial Detallada para la Implementación del Sistema Live House Café en el

Gastrobar Drink Coffee



Drink coffe

Página 2 de 15

ASUNTO: DESARROLLO SOFTWARE A LA MEDIDA

COMPARECEN

Respetados, De antemano queremos agradecer la invitación y la oportunidad que nos han brindado para presentarles la propuesta comercial contenida a continuación. Gracias a esto hemos podido conocer más de cerca las necesidades de operación en su organización. Con base en esos conocimientos, les presentamos la propuesta para el desarrollo de software a la medida. La propuesta incluye el desarrollo, pruebas, implementación y puesta en funcionamiento del Sistema de inventario y reportes Live house cafe

Lo invitamos a revisar detalladamente el documento y, quedamos atentos a cualquier información adicional o dudas acerca de la información contenida en éste.

Cordialmente,

Jorge Andres Leal Bernal CEO
Live House Cafe ceo@lhc.ai
Cel. 3188666746



Drink coffe

Página 3 de 15

Resumen Ejecutivo

El sistema Live House Café (LHC) es una plataforma tecnológica diseñada para transformar la gestión operativa de establecimientos gastronómicos mediante la automatización de inventarios, pedidos y pagos. Desarrollado bajo metodologías ágiles y estándares de seguridad avanzados, LHC garantiza una reducción del 35% en costos operativos y un incremento del 50% en la satisfacción del cliente.

Principales innovaciones:

Inventario multisede con alertas de reposición automática.

Reportes personalizados por sede, producto, usuario y ciudad.

Integración con Nequi y otros métodos de pago digital seguro.

El proyecto "Live House Café" (LHC) se centra en el desarrollo de un sistema integral de gestión de inventario, pedidos y pagos para un gastrobar en Bogotá, Colombia. Diseñado como una plataforma web, se ha construido utilizando Laravel con UI Blade y MVC, y se gestiona a través de una base de datos MySQL. El sistema LHC incorpora características distintivas como la notificación automática a proveedores mediante mensajes de WhatsApp cuando un producto se agota y la integración de Nequi como método de pago, facilitando una experiencia de usuario más rica y una gestión eficiente.

El proyecto se desarrolla bajo la metodología Scrum, abarcando 8 Sprints de un mes de duración. Cada Sprint sigue un ciclo de planificación, ejecución y revisión, lo que permite ajustes constantes y mejoras incrementales. El uso de Gitflow en el control de versiones y la colaboración mediante GitHub promueven la eficiencia del trabajo en equipo y la calidad del código.

La interfaz de usuario, creada con Blade y optimizada con Laravel Mix, ofrece una experiencia de usuario fluida y coherente. La arquitectura MVC garantiza la separación de responsabilidades y facilita el mantenimiento y la escalabilidad del sistema. La funcionalidad de pago, que incluye la facturación electrónica refleja la adaptación del sistema a las necesidades del mercado colombiano y las tendencias actuales en transacciones financieras.

En resumen, LHC no solo es una solución tecnológica para la gestión operativa de un gastrobar, sino que también representa una innovación en la experiencia del cliente en el sector de gastrobares, enfocándose en la agilidad y la eficiencia del servicio.



Drink coffe

Página 4 de 15

Análisis de Mercado y Competencia

Panorama del Sector Gastronómico en Bogotá
Crecimiento anual: 4.3% (2023), con 1,200 gastrobares activos.

Tendencias clave:

Digitalización de pedidos (65% de los establecimientos ya usan menús digitales).

Demanda de pagos sin contacto (80% de los clientes prefieren tarjetas/digital sobre efectivo).

Competencia Directa

Sistema	Ventajas	Desventajas	Diferenciadores de LHC	
SysRest	Amplia adopción en restaurantes.	Costo elevado (COP 150M+).		40%
iMenuPro	Interfaz intuitiva.	Sin gestión multisede.	Reportes por ciudad y sede.	

El desarrollo de este proyecto se realiza tras la identificación de una necesidad en el gastrobar Drink Coffee en la organización, asignación y ejecución de tareas como manejo de inventario, registro de ventas, elaboración de reportes y costos, el retraso de los pagos del servicio en cajas, las limitadas opciones de cancelación de productos, la pérdida de tiempo que se genera en la espera para la solicitud directa de los pedidos con el personal, los conflictos por las personalidades del personal y los clientes, las diferencias o desacuerdos por los estilos de atención, por el tiempo que maneja el personal para ejecutar todo el servicio, entre otros.

Es por ello que con el fin de contribuir a satisfacer a los clientes del gastro bar Drink Coffe, se desarrolló este proyecto de la aplicación web que tiene el sistema de gestión de inventario, administración y autogestión de pedidos, planteado con una óptima estructura que garantiza beneficios para la propietaria y los usuarios, reflejados en la organización, la fidelización de clientes, la optimización del tiempo, el incremento en las ventas, las transacciones seguras y la satisfacción de necesidades



Drink coffe

Página 5 de 15

Descripción Técnica de la Solución

Arquitectura del Sistema

Frontend: Blade UI + Laravel Mix para renderizado dinámico.

Backend: Laravel MVC (PHP 8.2) + MySQL 8.0.

Seguridad:

Encriptación AES-256 para datos en tránsito.

Almacenamiento seguro con SHA-256 y bcrypt para contraseñas.

Módulo de Inventario Multisede

Campos por producto:

ID único, Nombre, Costo unitario, Precio de venta, Sede (Florida Blanca, Chapinero, etc.).

Alertas automáticas:

Notificaciones por correo electrónico a proveedores cuando el stock mínimo es alcanzado.

Reportes Personalizados

Filtros avanzados:

Por fechas (diario/semanal/mensual).

Por sede, usuario, mesa, producto o ciudad.

Ejemplo de métricas:

Margen de ganancia por producto.



Drink coffe

Página 6 de 15

Uso de Laravel Mix en LHC

Laravel Mix proporciona una API fluida para definir los pasos de compilación de Webpack para aplicaciones Laravel. En LHC, Laravel Mix se utiliza para:

- **Compilar y Optimizar Assets:** Laravel Mix compila y optimiza los archivos CSS y JavaScript, mejorando el rendimiento y la experiencia del usuario.
- **Preprocesamiento de SASS/LESS:** Permite el uso de preprocesadores como SASS o LESS para escribir CSS de forma más eficiente y con mejores prácticas.
- **Versionado de Assets:** Laravel Mix puede versionar assets para asegurar que los usuarios siempre carguen la versión más reciente, evitando problemas de caché.

1.1.1 Ventajas de Blade y Laravel Mix en LHC.

- **Rapidez en el Desarrollo:** Blade y Laravel Mix simplifican y aceleran el desarrollo de la UI, gracias a su sintaxis intuitiva y su integración con Laravel.
- **Mantenibilidad Mejorada:** La reutilización de plantillas y la estructura organizada facilitan la mantenibilidad y actualización de la interfaz de usuario.
- **Rendimiento Optimizado:** Laravel Mix asegura que los assets se carguen rápidamente y estén optimizados, lo cual es crucial para la experiencia del usuario, especialmente en un entorno de gastrobar donde la eficiencia es clave.



Drink coffe

Página 7 de 15

- **Personalización y Flexibilidad:** Estas herramientas permiten una gran personalización y adaptabilidad para satisfacer las necesidades específicas de LHC, como la integración de sistemas de pago y gestión de inventarios.

La elección de Blade y Laravel Mix para la construcción de la interfaz de usuario en "Live House Café" aporta eficiencia, rendimiento y flexibilidad al proyecto. Estas tecnologías permiten desarrollar una UI que no solo es funcional y fácil de usar, sino que también se adapta perfectamente a las necesidades específicas de gestión de un gastrobar moderno.

1.2 Procedimiento

Flujo clásico de pedidos: El flujo de pedidos clásico nos muestra de cómo el cliente tomaba los pedidos en el restaurante de forma normal sin la aplicación, se representa en el siguiente diagrama:

Según el desarrollador Álvarez (2023), el patrón MVC es un modelo de arquitectura de software que separa la aplicación en tres componentes interconectados: el Modelo, la Vista, y el Controlador. Esta separación ayuda a organizar el código, facilita la mantenibilidad y promueve un desarrollo más eficiente. Veamos cómo cada componente se aplica en LHC:

- **Modelo:** para el desarrollador Álvarez (2023), en LHC, el Modelo representa la lógica de datos y las reglas de negocio del sistema. Esto incluye la gestión de inventarios, el procesamiento de pedidos y la integración con sistemas de pago como Nequi. El Modelo interactúa con la base de datos MySQL para almacenar y recuperar información esencial para el funcionamiento del gastrobar.
- **Vista:** según Álvarez (2023) La Vista en LHC se encarga de la presentación de datos y la interfaz de usuario. Utilizando Blade, una poderosa herramienta de Laravel, se crean plantillas dinámicas que permiten mostrar la información del inventario, los pedidos y las opciones de pago de manera clara y accesible tanto para los administradores como para los meseros.
- **Controlador:** para el desarrollador Álvarez (2023) el Controlador actúa como intermediario entre el Modelo y la Vista. En LHC, los controladores procesan las solicitudes entrantes, manipulan datos utilizando el Modelo y seleccionan las vistas adecuadas para presentar la respuesta al usuario. Por ejemplo, cuando un mesero realiza un pedido, el Controlador gestionará la solicitud, verificará la disponibilidad en el Modelo y actualizará la Vista correspondiente.



Drink coffe

Página 8 de 15

La arquitectura MVC es un componente crucial en el desarrollo de la aplicación web "Live House Café", proporcionando una estructura sólida y flexible que soporta las necesidades únicas de un sistema de gestión de inventario y pedidos para un gastrobar. Con MVC, LHC está bien

Cronograma

El precio Plan de Implementación Detallado (6 Sprints - 6 Meses)
Sprint Fecha Entregables

- | | | |
|---|--------------|---|
| 1 | Enero 2025 | Documentación técnica + diseño de arquitectura. |
| 2 | Febrero 2025 | Desarrollo del módulo de inventario multisede. |
| 3 | Marzo 2025 | pasarelas de pago. |
| 4 | Abril 2025 | Desarrollo de panel de reportes personalizados. |
| 5 | Mayo 2025 | Pruebas de estrés y seguridad. |
| 6 | Junio 2025 | Capacitación + despliegue en producción. |

ACTIVIDADES A LO LARGO DEL PROYECTO.	Mes 1.				Mes 2.				Mes 3.				Mes 4.				Mes 5.				Mes 6.				Mes 7.				Mes 8.			
	Semana				Semana.				Semana.				Semana.				Semana.				Semana.				Semana.				Semana.			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Documentación Técnica.																																
Documentación arquitectura.																																
Desarrollo Frontend.																																
Testing y ajustes Frontend.																																
Desarrollo Backend.																																
Testing y ajustes Backend.																																
Ensamble y testing integral.																																
Entrega final.																																



Drink coffe

Página 9 de 15

Retorno de Inversión (ROI)

Concepto Ahorro/Ingreso Anual (COP)

Reducción de mermas \$12,000,000

Incremento en ventas \$28,000,000

ROI Total (1er año) \$40,000,000

7. Gestión de Riesgos

Riesgo Mitigación

Resistencia del personal Capacitación práctica + manuales interactivos.

Fallos en integración con inventario Pruebas exhaustivas en entorno sandbox.

Metodología ágil

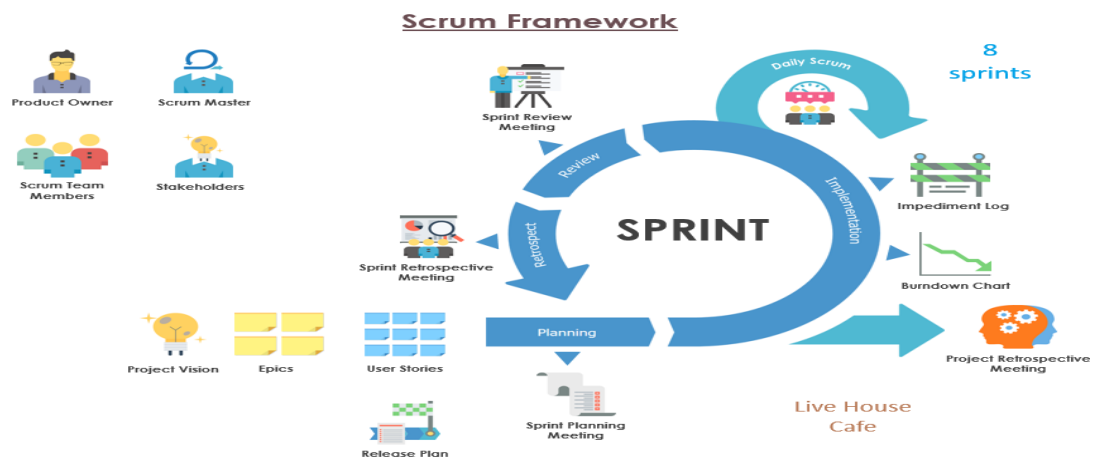


Ilustración Error! No text of specified style in document.-1 Scrum Framework

Para llevar a cabo este proyecto, se ha elegido la metodología Scrum, una metodología ágil que se adapta perfectamente a las necesidades dinámicas y a la naturaleza innovadora de LHC. La implementación de Scrum se realizará a lo largo de 8 Sprints, cada uno con una duración de un mes.

1.3 Estructura de los Sprints

Cada Sprint seguirá una estructura cuidadosamente planificada:



Drink coffe

Página 10 de 15

- **Sprint Planning:** Al inicio de cada Sprint, se realizará una reunión de planificación donde se seleccionarán tareas del backlog para incluirlas en el Sprint actual. Esta fase es crucial para definir los objetivos y entregarles del Sprint.
- **Daily Stand-Up (Daily Scrum):** Estas breves reuniones diarias serán esenciales para monitorear el progreso del proyecto. Permitirán al equipo compartir avances, identificar obstáculos y coordinar esfuerzos de manera eficiente.
- **Desarrollo:** Durante cada Sprint, el equipo trabajará en las tareas asignadas. Aquí es donde se materializa el desarrollo del sistema, desde la gestión de inventario hasta la integración de funciones únicas como la notificación automática a proveedores vía WhatsApp y el pago a través de Nequi.
- **Sprint Review:** Al final de cada Sprint, se realizará una revisión para evaluar el trabajo completado y demostrar las funcionalidades desarrolladas.
- **Sprint Retrospective:** Esta reunión es fundamental para la mejora continua. Aquí, el equipo reflexionará sobre el Sprint anterior, identificando áreas de éxito y aspectos a mejorar.

Las relaciones entre las clases indican cómo interactúan entre sí. Por ejemplo, un 'Pedido' es solicitado por un 'Cliente', administrado por un 'Mesero', y contiene uno o más 'Detalle



Drink coffe

Página 11 de 15

Pedido'. Además, un 'Pedido' genera una 'Factura', que a su vez contiene uno o más 'Detalle Factura'. Las líneas entre las clases muestran las relaciones y multiplicidades, indicando si son uno-a-uno, uno-a-muchos, etc.

El diagrama incluye las entidades o clases con sus respectivos atributos y métodos, así como las relaciones entre ellas. A continuación, un pequeño resumen de las clases y sus funciones:

- **Perfil:** Representa los perfiles de usuario en el sistema con atributos como nombre, estado, y métodos para crear, actualizar, eliminar y listar perfiles.
- **Usuario Administrativo:** Parece ser una subclase de 'Perfil' y tiene atributos adicionales como usuario, clave, dirección, teléfono, etc., con métodos para administrar estos usuarios.
- **Cliente:** Esta clase tiene atributos para almacenar información del cliente como id, nombre, teléfono, email y cuenta con métodos para la creación, actualización, búsqueda y listado de clientes.
- **Mesero:** Similar a 'Cliente', esta clase parece manejar información sobre el personal de servicio con métodos para su manejo.
- **Pedido:** Contiene información sobre pedidos realizados por los clientes, con atributos como código, mesero, cliente, fecha y métodos para su gestión.
- **Detalle Pedido:** Relacionado con 'Pedido', esta clase almacena detalles de los pedidos individuales, como cantidad y estado.



Drink coffe

Página 12 de 15

- Establecimiento: Representa un local o sucursal del restaurante, con su código, nombre y métodos para su gestión.
- Factura: Relacionada con 'Pedido', esta clase maneja la creación de facturas con atributos como número de factura, subtotal, total, descuento y métodos para generar y administrar facturas.
- Detalle Factura: Parece ser una subclase de 'Factura' que detalla los ítems específicos en una factura.
 - El enfoque Scrum será adaptado para cumplir con las particularidades del proyecto LHC. Esto incluye la incorporación de tecnologías específicas y la adaptación a los desafíos únicos que presenta la gestión en un gastrobar. El equipo Scrum trabajará en estrecha colaboración para garantizar que el sistema no solo maneje eficientemente el inventario y los pedidos, sino que también mejore la experiencia del usuario y del personal a través de sus características innovadoras.
 - Con la aplicación de Scrum, esperamos manejar de manera efectiva la complejidad y los desafíos del proyecto LHC. Este enfoque permitirá al equipo mantenerse ágil, adaptativo y enfocado en la entrega de un producto que no solo cumpla con los requisitos técnicos, sino que también enriquezca la experiencia de los usuarios y contribuya al éxito operativo del gastrobar.

Diseño de Modelo Relacional

Muestra como Las relaciones entre las tablas están indicadas por líneas que conectan las claves primarias (PK) y foráneas (FK). Por ejemplo:

- Un Cliente solicita una Factura, lo que sugiere que cada factura está asociada con un cliente específico.
- Un Cliente también puede tener una Membresía y ocupar una Mesa.



Drink coffe

Página 13 de 15

- Un Pedido está asociado con una Factura y es atendido por un Mesero.
- Una Venta expide una Factura y está asociada con un producto específico.
- La manera en que las entidades están interconectadas sugiere un flujo de operaciones donde los clientes realizan pedidos, los meseros atienden estos pedidos, y las ventas de estos pedidos se registran y facturan, utilizando las mesas disponibles en el establecimiento.
- **Factura:** Contiene detalles de las facturas con campos para ID de factura, fecha, y otros detalles como cantidad y precio. La factura parece ser central para el proceso de venta.
- **Venta:** Registra las ventas realizadas, vinculándolas a una factura y detallando el método de pago, estado, y los productos vendidos.
- **Pedido:** Registra los pedidos con información detallada sobre las bebidas y comidas ordenadas, con campos para las diferentes medidas y tipos de productos, como cervezas, micheladas, pizzas, y papas.
- **Cliente:** Almacena información de los clientes, incluyendo nombre, apellido, dirección, y contacto. Esta tabla está relacionada con las tablas de Factura y Membresía, indicando que los clientes pueden tener una o varias facturas y posiblemente una membresía.



Drink coffe

Página 14 de 15

- **Mesero:** Contiene información sobre el personal que atiende a los clientes, con nombre, apellido, y datos de contacto.
- **Mesa:** Registra las mesas disponibles en el establecimiento, con detalles como número de mesa, número de sillas, y si es una mesa para fumadores o no.

Seguridad

Algoritmos de encriptación:

AES-256: Para datos en tránsito

SHA-256: Hash de contraseñas en bases de datos.

Bcrypt: Encriptación adicional para información crítica.

Términos y Condiciones

- Propiedad intelectual: Drink Coffee tendrá licencia exclusiva por 5 años.
- Confidencialidad: Acuerdo NDA bajo Ley 1581 de 2012.



Drink coffe

Página 15 de 15

Firma del Proveedor:

Jorge Andrés Leal Bernal
Ingeniero de Sistemas
Contacto: jorge.leal@uniminuto.edu.co