

# VIVIANA BEAUTY ROOM

## Transferencia de Conocimiento sobre Wireframe

### MVP Simplificado

Flutter · Firebase · Gemini API

## 1. Alcance del MVP

Este documento describe la versión simplificada de la aplicación, centrada en resolver el flujo de agendamiento y el catálogo de servicios, que son los dos requerimientos explícitos de Viviana. Todo lo demás es fase posterior.

*Criterio de inclusión: si Viviana no lo pidió y no es necesario para que el flujo principal funcione, no está en este MVP.*

| ✓ Incluido en este MVP   | ✗ Fuera de este MVP (fase posterior)   |
|--|--|
| <ul style="list-style-type: none"><li>• Agenda con vista diaria y semanal</li><li>• Bloqueo manual de horarios</li><li>• Nueva cita integrada en la agenda</li><li>• Catálogo con foto, descripción y precio</li><li>• Botón de agendar directo desde el catálogo</li><li>• Directorio de clientas (nombre, contacto, notas básicas)</li><li>• Historial simple dentro del detalle de cita</li></ul> | <ul style="list-style-type: none"><li>• Dashboard con métricas</li><li>• Registro y reportes de ingresos</li><li>• Vista propia para la cliente</li><li>• Perfil de cliente como pantalla separada</li><li>• Sistema de roles y configuración avanzada</li><li>• Módulo de pagos y comprobantes</li><li>• Chatbot (ver sección extra al final)</li></ul> |

## 2. Pantallas del MVP

El MVP Simplificado consta de 6 pantallas. La vista de cliente no existe en esta fase; el agendamiento externo se gestiona a través del flujo de Viviana.

| # | Pantalla               | Rol en el flujo   |
|---|------------------------|---|
| 1 | Login                  | Punto de entrada. Autenticación de Viviana o colaboradora |
| 2 | Agenda                 | Pantalla principal. Vista diaria y semanal de citas       |
| 3 | Nueva cita             | Modal sobre la agenda. Creación del flujo de agendamiento |
| 4 | Catálogo de servicios  | Lista visual de servicios con botón de agendar directo    |
| 5 | Detalle de cita        | Vista completa de una cita con opciones de gestión        |
| 6 | Directorio de clientes | Lista buscable de clientes con perfil e historial         |

## 2.1 Login

Punto de entrada a la aplicación. Solo usuarios registrados por Viviana pueden acceder.

| Elemento                      | Descripción   |
|-------------------------------|---|
| Logo / nombre de marca        | Viviana Beauty Room en Playfair Display, centrado en la parte superior  |
| Campo email                   | Input de correo electrónico con validación de formato   |
| Campo contraseña              | Input de contraseña con opción de mostrar/ocultar   |
| Botón de ingreso              | Botón primario. Al autenticarse redirige a la Agenda  |
| Rol detectado automáticamente | El sistema determina si el usuario es ADMIN o COLABORADORA según las credenciales. No se selecciona manualmente |
| Error de credenciales         | Mensaje de error visible bajo el formulario si las credenciales son incorrectas                                 |

## 2.2 Agenda

Pantalla principal de la app. Es el primer lugar que Viviana ve al abrir la aplicación.

| Elemento          | Descripción  |
|-------------------|--|
| Vista por defecto | Día actual con las citas del día listadas en orden cronológico |
| Cambio de vista   | Toggle entre vista diaria y semanal                            |
| Tarjeta de cita   | Nombre de cliente, servicio, hora y estado con color           |

|                    |  |
|--------------------|--|
| Bloqueo de horario | Botón para marcar un horario como no disponible sin asociarlo a una cita           |
| Acción principal   | Botón fijo para crear nueva cita   |
| Al tocar una cita  | Abre el detalle de la cita con opción de editar, cancelar o marcar como completada |

## 2.3 Nueva Cita

No es una pantalla separada. Se abre como un modal o panel deslizable desde la agenda, para no sacar a Viviana de su contexto.

| Campo          | Detalle   |
|----------------|---|
| Cliente        | Buscar en directorio existente o crear nueva con nombre y teléfono                  |
| Servicio       | Selector desde el catálogo. Al elegir, bloquea automáticamente la duración estimada |
| Fecha y hora   | Selector de calendario que solo muestra horarios disponibles                        |
| Monto anticipo | Se sugiere automáticamente (\$50 o \$100) según el servicio. Editable.              |
| Notas          | Campo libre para alergias, preferencias o indicaciones especiales                   |
| Estado inicial | PENDIENTE hasta que Viviana registre el anticipo como recibido                      |

## 2.4 Catálogo de Servicios

Lista visual de todos los servicios disponibles. Es el equivalente digital del catálogo que hoy Viviana envía por mensaje.

| Elemento            | Descripción   |
|---------------------|---|
| Tarjeta de servicio | Foto, nombre, precio y duración estimada  |
| Detalle de servicio | Al tocar: descripción completa, indicaciones previas y fotos de referencia              |
| Combos              | Sección separada con los paquetes disponibles (maquillaje + peinado / cejas + pestañas) |
| Gestión             | Viviana puede editar precio, descripción y fotos desde esta pantalla                    |

|                   |   |
|-------------------|---|
| Servicios futuros | Campo de estado: Activo / Próximamente — para mostrar microblading y uñas sin que se puedan agendar aún |
|-------------------|---|

## 2.5 Detalle de Cita

Se abre al tocar cualquier cita en la agenda. Centraliza todas las acciones de gestión sobre una cita específica.

| Elemento                    | Descripción  |
|-----------------------------|--|
| Encabezado                  | Nombre de la cliente, servicio, fecha y hora de la cita  |
| Estado actual               | Badge de color con el estado actual: PENDIENTE, CONFIRMADA, etc.   |
| Alerta de alergia           | Si la cliente tiene alergias registradas, se muestra un banner de advertencia visible en la parte superior del detalle |
| Monto anticipo              | Monto sugerido y si ya fue recibido o está pendiente. Botón para marcarlo como recibido                                |
| Notas de la cita            | Campo de notas específicas de esta cita: alergias, indicaciones, preferencias  |
| Acción: Confirmar           | Disponible solo si el anticipo está registrado. Cambia estado a CONFIRMADA   |
| Acción: Completar           | Marca la cita como COMPLETADA. Solo disponible en estado CONFIRMADA  |
| Acción: Reprogramar         | Abre selector de nueva fecha y hora con validación de disponibilidad   |
| Acción: Cancelar cita       | Botón destructivo. Abre modal de confirmación antes de ejecutar la cancelación   |
| Acceso al perfil de cliente | Enlace al perfil completo de la cliente desde el detalle de la cita  |

## 2.6 Directorio de Clientas

Lista de clientes registradas. Reemplaza la memoria de Viviana con un registro consultable.

| Elemento          | Descripción   |
|-------------------|---|
| Lista principal   | Nombre, inicial o foto, teléfono y fecha de última visita   |
| Búsqueda          | Por nombre o número de teléfono   |
| Perfil de cliente | Se abre al tocar. Muestra: datos de contacto, notas (alergias, preferencias) e historial de citas |
| Historial         | Lista simple de citas anteriores: fecha, servicio y estado  |

|       |  |
|-------|--|
| Notas | Campo libre editable que Viviana puede actualizar después de cada visita |
|-------|--|

### 3. Flujo Principal de Agendamiento

Este es el flujo que debe reflejarse en el wireframe con todas sus transiciones entre pantallas.

| Paso | Acción                         | Resultado   |
|------|--------------------------------|---|
| 1    | Viviana abre la app            | Ve la agenda del día actual                             |
| 2    | Toca el botón de nueva cita    | Se abre el modal de nueva cita                          |
| 3    | Selecciona o crea la cliente   | Se asocia el perfil a la cita                           |
| 4    | Elige el servicio              | La duración se bloquea automáticamente en la agenda     |
| 5    | Selecciona fecha y hora        | Solo muestra horarios disponibles                       |
| 6    | El sistema sugiere el anticipo | Viviana confirma o ajusta el monto                      |
| 7    | Agrega notas si es necesario   | Alergias, preferencias o indicaciones                   |
| 8    | Guarda la cita                 | Queda en estado PENDIENTE en la agenda                  |
| 9    | Cliente envía el anticipo      | Viviana lo marca como recibido en el detalle de la cita |
| 10   | Estado cambia a CONFIRMADA     | La cita queda activa en la agenda con color verde       |
| 11   | Se presta el servicio          | Viviana marca la cita como COMPLETADA                   |

#### 3.1 Flujo de Cancelación

| Paso | Acción                              | Resultado   |
|------|-------------------------------------|---|
| 1    | Viviana abre el detalle de una cita | Ve las opciones disponibles según el estado                       |
| 2    | Selecciona Cancelar cita            | El sistema muestra un modal de confirmación                       |
| 3    | Confirma la cancelación             | Estado cambia a CANCELADA. El anticipo se registra como retenido. |
| 4    | El horario se libera                | El espacio vuelve a estar disponible en la agenda                 |

---

## 4. Reglas de Negocio

Estas reglas deben implementarse en el backend (Firebase Functions). No son validaciones de interfaz, son restricciones que el sistema debe respetar independientemente de lo que muestre Flutter.

| Regla                               | Implementación   |
|-------------------------------------|--|
| Bloqueo de horario automático       | Al crear una cita, se bloquea la duración del servicio + 30 min de buffer                              |
| Anticipo obligatorio para confirmar | Una cita no puede pasar de PENDIENTE a CONFIRMADA sin anticipo registrado                              |
| Cancelación sin reembolso           | Al cancelar, el anticipo queda como ingreso registrado en el perfil de la cliente                      |
| Límite diario de citas              | Máximo 5 citas por día. Configurable por Viviana   |
| Reprogramación                      | Solo válida si la nueva fecha está disponible y el aviso es con 3 días de anticipación mínimo          |
| Alerta de alergia                   | Si la cliente tiene alergias registradas, se muestra una alerta visible al abrir el detalle de la cita |

---

## 5. Stack Técnico

Este stack debe estar configurado completamente antes de escribir la primera pantalla en Flutter. La app depende de que el backend exista primero.

### 5.1 Capa de Interfaz — Flutter

| Elemento          | Decisión  |
|-------------------|---|
| Framework         | Flutter — multiplataforma iOS, Android y Web desde un solo código |
| Gestión de estado | Riverpod — suficiente para el tamaño del proyecto, escalable      |
| Navegación        | Go Router — navegación declarativa compatible con deep links      |
| Diseño            | Paleta de marca: rosa #D4748F, negro #1A1A1A, blanco #FAF8F8      |

|             |   |
|-------------|---|
| Tipografía  | Poppins para cuerpo. Playfair Display para nombre de la marca |
| Iconografía | Lucide Icons — trazo fino, coherente con el estilo visual     |

## 5.2 Capa de Backend — Firebase

| Servicio                 | Uso en el proyecto  |
|--------------------------|---|
| Firestore                | Base de datos principal. Colecciones: citas, clientes, servicios, horarios bloqueados |
| Firebase Auth            | Autenticación de Viviana y su hermana como usuaria secundaria                         |
| Firebase Functions       | Lógica de negocio: validaciones, reglas de cancelación, cálculo de disponibilidad     |
| Firebase Cloud Messaging | Notificaciones push para recordatorios de cita  |
| Firebase Storage         | Almacenamiento de fotos del catálogo de servicios                                     |
| Plan requerido           | Blaze (pago por uso). El plan gratuito (Spark) no permite llamadas a APIs externas    |

*El plan Blaze requiere tarjeta de crédito registrada en Firebase pero el costo real con el volumen actual de Viviana (máximo 5 citas diarias) es prácticamente cero.*

## 5.3 Modelo de Datos en Firestore

Las colecciones deben definirse antes de escribir cualquier pantalla en Flutter. Si cambian después, impactan todo el proyecto.

| Colección          | Campos principales   |
|--------------------|--|
| citas              | id, clientald, serviciold, fechaHora, duracion, estado, montoAnticipo, anticípоОPagado, notas, creadaEn  |
| clientas           | id, nombre, telefono, alergias, preferencias, notas, creadaEn  |
| servicios          | id, nombre, descripcion, precio, duracionMin, fotoUrl, activo, proximamente, esCombo, serviciosIncluidos |
| horariosBlockeados | id, fechaHora, duracionMin, motivo   |
| usuarios           | id, nombre, rol (ADMIN   COLABORADORA), email  |

## 5.4 Orden de Implementación Recomendado

El mismo equipo hace diseño, Flutter y backend. Este orden evita bloqueos entre tareas.

| Fase | Qué se hace                              | Por qué primero   |
|------|--|---|
| 1    | Modelo de datos en Firestore             | Todo lo demás depende de esto. Si cambia después, rompe Flutter y Functions |
| 2    | Firebase Auth + reglas de seguridad      | Sin esto, cualquiera puede leer y escribir la base de datos                 |
| 3    | Firebase Functions con lógica de negocio | El backend debe existir antes de que Flutter lo consuma                     |
| 4    | Pantalla de Agenda                       | Es el núcleo de la app y el requerimiento principal                         |
| 5    | Catálogo de servicios                    | Segundo requerimiento explícito de Viviana                                  |
| 6    | Directorio de clientes                   | Complementa el flujo de nueva cita  |
| 7    | Pruebas, ajustes y entrega               | Validación con Viviana antes de cerrar el proyecto                          |

## EXTRA — Chatbot de Agendamiento (Si hay tiempo)

*Esta sección es adicional al MVP. Solo debe implementarse si el proyecto principal está completo y el tiempo lo permite. No es un requerimiento de Viviana, es una mejora futura.*

### ¿Qué hace?

La cliente escribe en lenguaje natural dentro de la app (por ejemplo: 'quiero una cita este viernes a las 4'). El chatbot interpreta el mensaje, verifica disponibilidad en Firestore, notifica a Viviana con la solicitud y el número de la cliente, y le envía una confirmación automática de regreso.

### Qué se agrega al stack

| Componente         | Detalle   |
|--------------------|---|
| Gemini API         | Procesamiento de lenguaje natural. Plan gratuito: 15 req/min, 1M tokens/día — suficiente para el volumen de Viviana |
| Firebase Functions | Ya existe. Se agrega la función que recibe el mensaje, llama a Gemini y procesa la respuesta                        |
| Rate limiting      | Obligatorio desde el inicio para evitar que un abuso agote el plan gratuito de Gemini                               |

|                             |  |
|-----------------------------|--|
| Pantalla de chat en Flutter | Interfaz simple de conversación. Se agrega como una quinta pantalla en la navegación |
| Notificación a Viviana      | Ya configurado con FCM. Se agrega el trigger desde la función del chatbot            |
| Registro de la cliente      | La cliente debe identificarse con su número de teléfono antes de usar el chatbot     |

## Flujo del chatbot

| Paso | Actor  | Acción   |
|------|--|--|
| 1    | Cliente                                      | Escribe su solicitud en lenguaje natural                             |
| 2    | Firebase Function                            | Recibe el mensaje y lo envía a Gemini con el prompt de instrucciones |
| 3    | Gemini API                                   | Interpreta la intención, extrae fecha, hora y servicio               |
| 4    | Firebase Function                            | Consulta Firestore para verificar si el horario está disponible      |
| 5    | Firebase Function (si hay disponibilidad)    | Crea la cita en estado PENDIENTE y notifica a Viviana vía FCM        |
| 6    | Viviana                                      | Recibe la notificación con el nombre, servicio y hora solicitada     |
| 7    | Viviana                                      | Acepta desde la notificación o llama a la cliente para reprogramar   |
| 8    | Firebase Function                            | Actualiza el estado de la cita y envía confirmación a la cliente     |
| 9    | Firebase Function (si no hay disponibilidad) | El chatbot responde con los horarios disponibles más cercanos        |

## Lo que el chatbot no reemplaza

- Viviana sigue siendo quien aprueba o rechaza cada cita. El chatbot propone, nunca decide.
- El anticipo sigue siendo manual. El chatbot informa cómo enviarlo pero no lo procesa.
- Si el chatbot no entiende el mensaje, debe notificar a Viviana directamente para que ella atienda el caso. Nunca dejar a la cliente sin respuesta.