

DOCUMENTO DE LEVANTAMIENTO DE REQUERIMIENTOS

Simulador de Turno Bancario

1. Objetivo del Sistema

Desarrollar un simulador de turnos bancarios que permita gestionar filas de atención por tipo de trámite (Caja, Servicio al Cliente y Créditos), asignando turnos consecutivos y mostrando la posición del usuario en la fila.

El sistema será desarrollado en el framework **Laravel** y gestionará la información mediante sesiones del servidor, sin utilizar base de datos.

2. Requerimientos Funcionales

2.1 RF1 – Registro del Cliente

El sistema debe permitir al usuario ingresar:

- Nombre completo.
- Tipo de trámite (Caja, Servicio al Cliente, Créditos).

El sistema debe validar que:

- El nombre no esté vacío.
- El tipo de trámite seleccionado sea válido.
- Los datos ingresados cumplan con el formato esperado.

2.2 RF2 – Generación Automática de Turno

El sistema debe generar un número de turno consecutivo por tipo de trámite:

- Caja → C001, C002, C003...
- Servicio → S001, S002, S003...
- Créditos → CR001, CR002, CR003...

Condiciones:

- El turno debe ser único dentro de su categoría.
- El turno debe asignarse en orden de llegada.
- El consecutivo debe formatearse con tres dígitos.

2.3 RF3 – Gestión de Filas

El sistema debe:

- Mantener una fila independiente por cada tipo de trámite.
- Utilizar estructuras de datos tipo arreglo (array).
- Implementar el modelo FIFO (First In, First Out).
- Calcular la cantidad de personas que se encuentran antes del usuario en la fila.

2.4 RF4 – Visualización del Turno

Una vez generado el turno, el sistema debe mostrar:

- Nombre del cliente.
- Tipo de trámite seleccionado.
- Número de turno asignado.
- Cantidad de personas en espera antes del turno asignado.

2.5 RF5 – Simulación de Atención

El sistema debe permitir:

- Atender el siguiente turno de una fila específica.
- Eliminar el primer elemento de la cola correspondiente.
- Actualizar automáticamente la cantidad de personas en espera.

3. Requerimientos No Funcionales

3.1 RNF1 – Arquitectura

El sistema debe seguir el patrón MVC (Modelo-Vista-Controlador) del framework Laravel:

- Rutas definidas en web.php.
- Controladores para la lógica del negocio.
- Vistas desarrolladas con Blade.
- Separación clara entre lógica y presentación.

3.2 RNF2 – Persistencia

- No se utilizará base de datos.
- La información será almacenada temporalmente mediante sesiones del servidor.

- Los datos se perderán al cerrar sesión o reiniciar el servidor.

3.3 RNF3 – Usabilidad

- La interfaz debe ser clara y minimalista.
- El proceso de generación de turno debe ser simple.
- El flujo del sistema debe seguir el siguiente orden:
 1. Ingresar datos.
 2. Generar turno.
 3. Visualizar posición en la fila.

3.4 RNF4 – Rendimiento

- El sistema debe generar el turno en menos de 2 segundos.
- Las operaciones de atención deben ejecutarse de manera inmediata.

5. Reglas de Negocio

1. Cada tipo de trámite tiene su propia fila independiente.
2. Los turnos se asignan estrictamente por orden de llegada.
3. No se pueden saltar turnos.
4. El número de turno es incremental y consecutivo dentro de su categoría.
5. No se permite registrar un turno sin nombre.
6. El sistema puede ser reiniciado manualmente.

6. Requerimientos Técnicos

- Framework: Laravel
- Lenguaje: PHP
- Servidor local: php artisan serve
- Gestión de colas: Arrays en sesión
- Modelo de atención: FIFO (First In, First Out)
- Entorno de desarrollo: Repositorio versionado en GitHub