**Manual de Implementación: Bandejas de Correo Colaborativas**

**1. Introducción**

Este documento describe la implementación de un sistema de bandejas de correo colaborativas, incluyendo gestión de estados, KPIs, adjuntos y vinculación con otros procesos departamentales.

**2. Arquitectura General**

**2.1 Tecnologías Utilizadas**

* **Backend:** Django (Python)
* **Base de Datos:** PostgreSQL
* **Frontend:** Django Templates + Bootstrap
* **Correo:** Integración con Gmail API / IMAP/SMTP
* **Almacenamiento de Adjuntos:** AWS S3 o almacenamiento local
* **Seguridad:** Escaneo de archivos, control de accesos y registros de auditoría

**3. Funcionalidades Clave**

**3.1 Gestión de Estados de Correos**

Los correos pueden pasar por distintos estados:

* **Pendiente**: Recibido y aún no atendido.
* **En proceso**: Vinculado a otro departamento para su resolución.
* **Respondido**: Se ha enviado una respuesta al remitente.
* **Escalado**: Transferido a otra bandeja o usuario.
* **Cerrado**: No requiere más acciones.

Cada cambio de estado se registra en el historial con usuario, fecha y comentarios. Además, si un correo cambia de estado y está vinculado a una tarea externa, se enviará una actualización tanto al cliente como al remitente notificando el progreso y la reasignación cuando aplique.

**3.2 Gestión de KPIs**

Los KPIs registrados incluyen:

* **Tiempo promedio de respuesta**
* **Tiempo de resolución**
* **Correos atendidos por usuario**
* **Distribución de carga de trabajo**
* **Correos sin respuesta después de X horas**
* **Picos de actividad**

Estos KPIs pueden visualizarse en un dashboard con gráficos y reportes.

**3.3 Manejo de Adjuntos**

* Los adjuntos se almacenan y se asocian con cada correo.
* Se permite la previsualización sin descarga.
* Se registran todas las acciones sobre adjuntos en el historial.
* Si un correo está en "En proceso", los adjuntos son accesibles en el área vinculada.

**3.4 Transferencia Interna de Correos**

* Se permite mover correos entre bandejas manteniendo historial y notificaciones.
* Un correo puede ser vinculado con tareas en otros departamentos.
* Cada transferencia de correo genera una notificación y una actualización automática al cliente y al remitente.

**3.5 Control de Respuestas**

* Un usuario bloquea temporalmente un correo mientras lo responde.
* Historial de respuestas con fecha, remitente y destinatario.
* Uso de respuestas predefinidas con opción de revisión antes de enviar.

**4. Implementación Técnica**

**4.1 Base de Datos**

**4.1.1 Tablas Principales**

* **Correos**: ID, remitente, destinatario, asunto, contenido, estado, fecha de recepción, última actualización, ID de usuario asignado.
* **Adjuntos**: ID, correo\_id, nombre\_archivo, tipo, tamaño, fecha\_subida, usuario\_subió.
* **Historial\_Correos**: ID, correo\_id, usuario\_id, cambio\_estado, fecha, comentario.
* **Respuestas**: ID, correo\_id, usuario\_id, fecha\_envío, contenido.
* **Tareas\_Vinculadas**: ID, correo\_id, departamento\_id, estado, última\_actualización.

**4.2 Flujos de Trabajo**

1. 📩 **Correo recibido → Estado: Pendiente**
2. 🔄 **Asignación interna → Puede pasar a "En proceso" si requiere acción de otro departamento**
3. 📂 **Si es "En proceso", se genera una tarea vinculada y se notifica al cliente y al remitente sobre el cambio de estado y la reasignación**
4. 🔗 **El departamento actualiza la tarea con comentarios y cambios de estado, enviando actualizaciones al cliente y al remitente**
5. ✅ **Cuando el proceso finaliza, el correo se responde y se marca como "Respondido" o "Cerrado". Se envía una notificación final al cliente y al remitente.**

**4.3 API Endpoints**

**Correos**

* GET /correos/pendientes/ → Lista de correos pendientes.
* POST /correos/asignar/ → Asignar un correo a un usuario.
* PATCH /correos/{id}/estado/ → Cambiar el estado de un correo y notificar al cliente y remitente.

**Adjuntos**

* GET /correos/{id}/adjuntos/ → Listar adjuntos.
* POST /correos/{id}/adjuntos/ → Subir adjuntos.
* GET /adjuntos/{id}/descargar/ → Descargar adjunto.

**Historial y Respuestas**

* GET /correos/{id}/historial/ → Ver historial de cambios.
* POST /correos/{id}/responder/ → Enviar una respuesta.

**KPIs y Reportes**

* GET /kpis/ → Obtener métricas generales.
* GET /reportes/actividad/ → Reporte de actividad por usuario y departamento.

**5. Seguridad y Control de Accesos**

* **Roles y permisos**: Control de acceso según usuario y departamento.
* **Cifrado de datos**: Protección de información sensible.
* **Registros de auditoría**: Historial detallado de acciones en el sistema.

**6. Despliegue y Monitoreo**

* **Docker** para contenedorización del sistema.
* **CI/CD** con GitHub Actions para despliegues automatizados.
* **Logs y monitoreo** con herramientas como Grafana o ELK Stack.

**7. Conclusión**

Este sistema permite gestionar correos de forma colaborativa, asegurando trazabilidad, control y eficiencia en la comunicación entre departamentos, con actualizaciones automáticas al cliente y al remitente en cada etapa del proceso.