

Documentación de Actualización:

Módulo de Notificaciones

1. Introducción al Módulo de Notificaciones

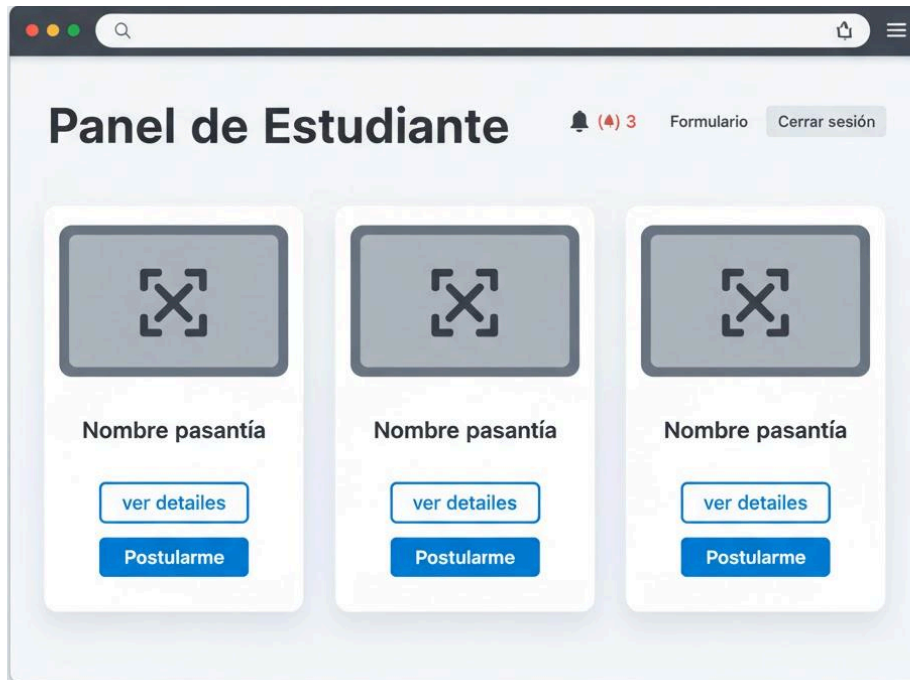
Se introduce un nuevo módulo de "Notificaciones" para gestionar alertas activas y pasivas para todos los actores (Estudiante, Empresa, SAU). Este módulo es responsable de:

1. **Alertas Basadas en Eventos:** Notificar acciones clave (ej. postulación aceptada, oferta aprobada).
2. **Alertas Basadas en Fechas:** Enviar recordatorios programados (ej. fin de pasantía).
3. **Gestión de Preferencias:** Permitir al usuario elegir *qué* notificaciones recibir y *por qué canal* (App, Email, SMS).
4. **Auditoría y Estados:** Registrar el estado de cada notificación (leída/no leída, enviada/fallida).

2. Prototipo de la Nueva Funcionalidad

2.1. Modificación: Paneles Principales (Estudiante, Empresa, Admin)

Se añade un ícono de "Campana" (Notificaciones) en la cabecera de los paneles principales (Estudiante, Empresa y Admin.). Un indicador numérico muestra las notificaciones no leídas.



2.2. Nueva Pantalla: Centro de Notificaciones (In-App)

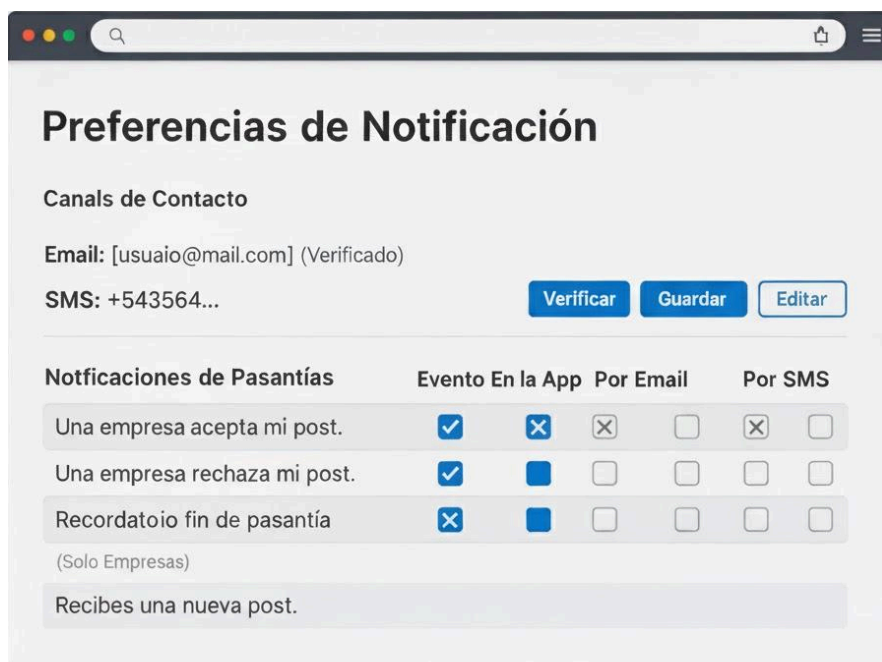
Al hacer clic en la campana (🔔), se despliega un panel o se accede a una página con el listado de notificaciones.



(Leyenda: (●) No Leída, () Leída)

2.3. Nueva Pantalla: Preferencias de Notificación

Se accede desde el "Formulario" o "Perfil" del usuario.



3. Procesos de Negocio y Casos de Uso (Agregados/Modificados)

Se definen nuevos casos de uso y se modifican los existentes para integrar el Módulo de Notificaciones.

3.1. Caso de Uso (Nuevo): Gestionar Preferencias de Notificación

- **ID:** CU-006
- **Descripción:** Permite a un usuario (Estudiante, Empresa, Admin) configurar qué notificaciones desea recibir y por qué canales.
- **Actores Principales:** Estudiante, Empresa, Administrador (SAU).
- **Precondiciones:** El usuario ha iniciado sesión.
- **Flujo Principal:**
 1. El usuario accede a su "Perfil" / "Formulario" y selecciona "Preferencias de Notificación".
 2. El sistema muestra la pantalla "Preferencias de Notificación" (Prototipo 2.3), cargando sus preferencias actuales.
 3. El usuario marca o desmarca las casillas (checkboxes) según sus preferencias.
 4. El usuario presiona "Guardar Cambios".
 5. El sistema almacena las preferencias en la base de datos asociadas al id_usuario.
- **Post-condiciones (Éxito):** Las preferencias del usuario han sido actualizadas.
- **Flujos Alternativos:**
 - **3.a. Usuario quiere añadir/verificar SMS:**
 1. (Paso 2.1) El usuario ingresa su número de celular y presiona "Verificar".
 2. (Paso 2.2) El sistema envía un código de verificación al celular.
 3. (Paso 2.3) El usuario ingresa el código y el sistema lo valida.
 4. (Paso 2.4) El canal SMS queda marcado como "Verificado".
 - **3.a.1. Falla envío de SMS (Gateway):** El sistema muestra un error: "No se pudo enviar el código de verificación. Intente más tarde."
 - **3.a.2. Código de verificación incorrecto:** El sistema muestra un error: "Código incorrecto."
 - **5.a. Falla al guardar en BD:** El sistema muestra un error: "No se pudieron guardar las preferencias. Intente nuevamente." (El sistema debe hacer *rollback* de la transacción).

3.2. Proceso de Negocio (Nuevo): Envío de Alertas Programadas (Fin de Pasantía)

- **ID:** PB-001
- **Descripción:** El sistema envía recordatorios automáticamente antes de la finalización de una pasantía activa.

- **Actores Principales:** Sistema.
- **Trigger:** Proceso programado (Cron Job) que se ejecuta una vez al día (ej. 08:00 AM).
- **Flujo Principal:**
 1. El sistema busca todas las pasantías en estado "Activa" cuya fechaFin esté a N días (ej. 7 días y 1 día antes).
 2. Por cada pasantía encontrada, el sistema genera un evento FIN_PASANTIA_PROXIMO.
 3. El Módulo de Notificaciones recibe el evento (que incluye id_estudiante e id_empresa).
 4. El módulo consulta las preferencias del Estudiante y la Empresa involucrados.
 5. El módulo envía la notificación (ej. "Recordatorio: Tu pasantía en [Empresa] finaliza el [Fecha]") por los canales configurados por cada usuario (App, Email, SMS).
 6. El sistema registra el envío en la auditoría (LogEnvioNotificacion) con estado "Enviada".
- **Flujos Alternativos:**
 - **1.a. No se encuentran pasantías:** El proceso finaliza sin acción.
 - **5.a. Falla un canal de envío (ej. SMS Gateway caído):**
 1. El módulo intenta enviar por SMS.
 2. El servicio externo (Gateway) devuelve un error.
 3. El sistema registra el envío en la auditoría con estado "Fallida" y el motivo del error.
 4. El sistema continúa enviando por los otros canales configurados (App, Email).
 - **5.b. Falla creación de notificación In-App (Falla BD):**
 1. El sistema intenta escribir la notificación en la tabla de la BD.
 2. La BD devuelve un error.
 3. El sistema registra el envío en la auditoría como "Fallida". (Este es un error crítico que debe ser monitoreado).

3.3. Modificación a Casos de Uso Existentes

- **Caso de Uso: Aprobar Oportunidad de Pasantía (Modificación de Flujo de SAU)**
 - *Flujo existente (basado en 8.5 [cite: 135-136]):* SAU aprueba la oferta.
 - *Paso 7 (Modificado):* El sistema actualiza el estado de la oferta a "Aprobada".
 - *Paso 8 (Nuevo):* El sistema dispara el evento OFERTA_APROBADA (dirigido a la Empresa).
 - *Paso 9 (Nuevo):* El Módulo de Notificaciones procesa el evento y envía la alerta por los canales configurados por la empresa.
 - *Paso 10 (Nuevo):* El sistema registra el envío en la auditoría.
 - **Flujos Alternativos:**
 - **9.a. Falla el Módulo de Notificaciones:** La aprobación de la pasantía (Paso 7) **no debe revertirse**. El error de notificación (ej. email no enviado) debe registrarse en la auditoría (LogEnvioNotificacion) como

"Fallida", pero el flujo de aprobación de la oferta es exitoso.

- **Caso de Uso: Aceptar Postulación de Estudiante (Modificación - Flujo de Empresa)**
 - *Contexto:* Este flujo no estaba explícitamente detallado como Caso de Uso, pero es fundamental.
 - *Paso 1:* Empresa selecciona "Aceptar" en la postulación de un estudiante.
 - *Paso 2:* El sistema actualiza el estado de la postulación a "Aceptada".
 - *Paso 3 (Nuevo):* El sistema dispara dos eventos:
 1. POSTULACION_ACEPTADA (dirigido al Estudiante).
 2. CONFIRMACION_ACEPTACION (dirigido a la Empresa, como confirmación).
 - *Paso 4 (Nuevo):* El Módulo de Notificaciones procesa ambos eventos y envía las alertas por los canales configurados para cada usuario.
 - *Paso 5 (Nuevo):* El sistema registra los envíos en la auditoría.
 - **Flujos Alternativos:**
 - **4.a. Falla notificación al Estudiante (ej. Email inválido):** El Módulo registra el envío al Estudiante como "Fallida". El sistema continúa y procesa la notificación a la Empresa (Paso 4.2). La postulación sigue "Aceptada".
 - **4.b. Falla notificación a la Empresa:** El Módulo registra el envío a la Empresa como "Fallida". El flujo principal (aceptación de la postulación) no se revierte.

4. Endpoints de la API (Agregados y Modificados)

Para dar soporte a la nueva funcionalidad, se añaden los siguientes endpoints a la API (Sección 8 del Proyecto_.docx [cite: 110-140]). Todos requieren autenticación (Token JWT).

4.1. Endpoints Nuevos

- **GET /api/notificaciones**
 - **Descripción:** Obtiene el listado de notificaciones (para el centro In-App) del usuario autenticado.
 - **Respuesta (Éxito):** 200 OK

```
[
  {
    "id": "n-123",
    "mensaje": "¡Felicitaciones! Tu postulación para 'Analista Jr.' ha sido aceptada.",
    "estado": "no_leida",
    "fecha": "2025-11-04T17:00:00Z",
    "tipo": "POSTULACION_ACEPTADA"
  },
  {
```

```

    "id": "n-122",
    "mensaje": "Recordatorio: Tu pasantía en Arcor finaliza en 7 días.",
    "estado": "leída",
    "fecha": "2025-11-03T08:00:00Z",
    "tipo": "FIN_PASANTIA_PROXIMO"
  }
]

```

- **POST /api/notificaciones/{id}/leída**

- **Descripción:** Marca una notificación específica como "leída".
- **Respuesta (Éxito):** 204 No Content

- **GET /api/preferencias**

- **Descripción:** Obtiene las preferencias de notificación del usuario actual.
- **Respuesta (Éxito):** 200 OK


```

{
  "celular": "+543564...",
  "celular_verificado": true,
  "preferencias": [
    { "evento": "POSTULACION_ACEPTADA", "app": true, "email": true, "sms": false },
    { "evento": "FIN_PASANTIA_PROXIMO", "app": true, "email": true, "sms": true }
  ]
}

```

- **PUT /api/preferencias**

- **Descripción:** Actualiza el listado de preferencias del usuario.
- **Body (Request):**

```

{
  "celular": "+543564...", // Opcional si se actualiza
  "preferencias": [
    { "evento": "POSTULACION_ACEPTADA", "app": true, "email": true, "sms": false },
    { "evento": "FIN_PASANTIA_PROXIMO", "app": false, "email": true, "sms": false }
  ]
}

```

- **Respuesta (Éxito):** 200 OK (devuelve el objeto actualizado).

4.2. Endpoints Modificados (Lógica Interna)

Los siguientes endpoints no cambian su *firma* (la URL y el método), pero su *lógica interna* en el backend (Sección 7.2 Proyecto_.docx) debe ser modificada para disparar los eventos de notificación.

- POST /api/postulaciones/{id}/aceptar (Endpoint hipotético de Empresa)
- POST /api/postulaciones/{id}/rechazar (Endpoint hipotético de Empresa)
- POST /api/admin/ofertas/{id}/aprobar (Endpoint hipotético de SAU)
- POST /api/admin/empresas/{id}/aprobar (Endpoint hipotético de SAU)

5. Casos de Prueba de la Nueva Funcionalidad

Se añaden los siguientes casos de prueba (basados en las plantillas TestCaseTemplate.xls).

Caso de Prueba 1

- **Test Case ID:** NOT-001
- **Descripción:** Verificar la correcta recepción de una notificación (In-App y Email) tras la aceptación de una postulación.
- **Precondiciones:**
 1. Estudiante "Estudiante_Prueba" está logueado.
 2. Estudiante_Prueba tiene preferencias: POSTULACION_ACEPTADA (App: true, Email: true, SMS: false).
 3. Empresa "Empresa_Prueba" está logueada.
 4. Estudiante_Prueba tiene una postulación "Pendiente" en Empresa_Prueba.
- **Flujo de Pasos:**
 1. Empresa_Prueba navega a sus postulaciones y "Acepta" la de Estudiante_Prueba.
 2. Estudiante_Prueba refresca su panel.
 3. Estudiante_Prueba revisa su cliente de email.
- **Resultados Esperados:**
 1. (Paso 1) La API devuelve 200 OK a la Empresa.
 2. (Paso 2) El ícono de campana (🔔) del Estudiante_Prueba muestra un indicador "1".
 3. (Paso 2) Al abrir el centro de notificaciones, aparece un mensaje "¡Felicitaciones!..." en estado "no_leída".
 4. (Paso 3) El Estudiante_Prueba recibe un email con el mismo contenido.
 5. (BD) La tabla LogEnvioNotificacion registra 2 envíos: APP (enviada) y EMAIL (enviada).

Caso de Prueba 2

- **Test Case ID:** NOT-002
- **Descripción:** Verificar que una notificación no se envía por un canal deshabilitado (SMS).
- **Precondiciones:**
 1. Mismas que NOT-001, pero las preferencias del Estudiante son: POSTULACION_ACEPTADA (App: true, Email: false, SMS: false).
- **Flujo de Pasos:**
 1. Empresa_Prueba "Acepta" la postulación de Estudiante_Prueba.

2. Estudiante_Prueba revisa su cliente de email.
3. Estudiante_Prueba revisa sus mensajes SMS.

- **Resultados Esperados:**

1. El Estudiante_Prueba recibe la notificación In-App.
2. El Estudiante_Prueba **NO** recibe ningún email.
3. El Estudiante_Prueba **NO** recibe ningún SMS.
4. (BD) La tabla LogEnvioNotificacion registra 1 envío: APP (enviada).

Caso de Prueba 3

- **Test Case ID:** NOT-003

- **Descripción:** Verificar el manejo de estado Leída / No Leída.

- **Precondiciones:**

1. Estudiante está logueado.
2. El ícono de campana (🔔) muestra "2" (dos notificaciones no leídas).

- **Flujo de Pasos:**

1. Estudiante hace clic en la campana.
2. El sistema muestra el listado (Prototipo 2.2) con 2 mensajes marcados como (●) No Leídos.
3. Estudiante hace clic en el primer mensaje.
4. Estudiante vuelve al panel principal.

- **Resultados Esperados:**

1. (Paso 3) El primer mensaje cambia su estado visual de (●) a (). (La API POST /api/notificaciones/{id}/leida fue llamada).
2. (Paso 4) El ícono de campana (🔔) ahora muestra "1".

Caso de Prueba 4

- **Test Case ID:** NOT-004

- **Descripción:** Verificar la auditoría de un envío fallido (SMS).

- **Precondiciones:**

1. Estudiante tiene preferencias: FIN_PASANTIA_PROXIMO (SMS: true).
2. El número de celular del estudiante es inválido o el servicio de SMS Gateway está caído.
3. El Proceso Programado (PB-001) se ejecuta.

- **Flujo de Pasos:**

1. El Cron Job (PB-001) se ejecuta e identifica la pasantía del estudiante.
2. El Módulo de Notificaciones intenta enviar un SMS.
3. El Gateway de SMS retorna un error.

- **Resultados Esperados:**

1. (BD) La tabla LogEnvioNotificacion registra 1 envío: CANAL (SMS), ESTADO (fallida), y un campo MENSAJE_ERROR (ej. "Número inválido" o "Gateway timeout").