una convocatoria se vasa en la autorizacion por parte del vicerrectorado, el ddecano de una facultad y

pasando a un debate por parte de docnetes participes en la convocatoria en la aprobacion de dicha convacotria o ser disuelta y cancelarse la convocatoria.

si llega a aprobarse sigue los siguientes pasos para su creacion :

#titulo de la convocatoria: que especificara para quienes y en que facultad saldra dicha convocatoria, especificando la faacultad, carrera y universidad, haciendo referencia a una resolucion.

#postulantes requisitos: la convocatoria presentara los requisitos a presentar para su habiitacion explicando y/o especificando algunos incisos que contendran infomacion de donde conseguir ciertos requisitos, o especificando algunos datos con respectos a la presentacion o la convocatoria.

#una presentacion a la postulacion: indicando donde deben presentar sus requisitos la fecha de culminacion y lanzamiento de esta convocatoria e informacion de orden o llenado de los requisito. sinedo otro caso el rechazo de algunos postulantes por falta de requisitos o incumplir en alguna de las normas que se presentaran en las convocatorias.

#realizando una eleccion por parte de docentes, decanos o personaje(s) a cargo de la convocatoria se realizaran una evaluacion y seleccion de los postulantes para su habilitacion, en otros casos selecion de ganadores de las convocatorias.

culminando estos pasos se realizara una carta por parte del decano de la carrera o facultad referente al vicerrector para realizar su revision de los requisitos, resolucion, su dictamen y sus designaciones. cumpliendo estos pasos en el vicerrectorado se realizar la confirmacion de documentacion completa, para perimitir el archivo de dicho documento para el amacenamiento de la informacion.

SISTEMA DE GESTION DE CONVOCATORIAS

en referente a lo anterior el sistema a crea tendra como manera inicial una plantilla dada por el vicerrectorado de un documento para el lanzamiento de su convocatoria este sera editable estableciendo campos para ser llenados y agregados de manera simple y directa a la plantilla del documento siendo editaspos por la decanatura de la carrera o facultad.

una manera de compartir de manera web los documentos ya editados por la carrera o decanatura para su revision y aprobacion por parte del vicerrectorado asi siendo lanzadas a los intereados en esta convocatoria.

pasando la fase de documentaicon de parte del decano de la carrera o facultad, este realizara una carta con mencion de una resolucion y dictamen de la misma, lo cual el sistema tambien enviara una platilla de dicha carta, resolucion y del dictamen, para despues el decano realizar el agregado de los requisitos en la convocatoria, y las designaciones e informe de participantes y ganadores.

este sera enviado nuevamente a revision del vicerrectorado para la verificaicon de datos, y que todo este en orden para pasar de unas opciones del documento coomo en revision, observado, concluido.

al tener la marca de concluido, el decano de la carrera o faculta realizara una impresion de dichos datos, con las firmas necesarias del decano de la facultad o carrera, firmas en los documentos de requisitos etc, para que dichos documentos fisicos sean enviados a la archivasion en fisico en el vicerrectorado asi finalmente tenner los archivos guardados en una base de datos despues de su escaneo con firmas y todo lo necesario.

eso solo para las convocatorias, de ser posible agregar una categorizacion de convocaatorias diviendolos en facultasdes/carreraas/convocatorias/docentes/y aqui dentro separarlos por las distintas categorias de docentes como ser consultor , extraordinario, invitado, ordinario, etc.

sistema de gestion de convocatorias

#creacion de una plantilla de carta de conclusion de convocatoria, incluyendo la resolucion y dictamen, con opcion de editado para agregar directamente al un documento y generar este mismo documento para su guardado en base de datos y opcion de compartir.

#estado del documento enviado con las opciones de en revision, observado, y revisado.

#almacenamiento de documentos generados o subidos

#creacion de categorizacion de convocatoria (docentes/que se subdivide en ordinarios, estraordianrios, consultores, invitados, examen de competencia, etc.)con opcion de subida de documentos para un mejor almacenamiento mas organizado.

#

* Crear primeramente una forma de categorización de las convocatorias es decir como crear una carpeta que diga (facultad y carrera): cienciaspuras/ matematicas, esta se subdividad en las convocatorias como docente extraordinario, docente invitado, docente consultor,etc (ciencias puras/matematicas/convocatoria\_docente\_consultor )
* las convocatorias se clasificaran mediante: tipo de convocatoria (docente: invitado, consultor, extraordinario,etc) , categoría (cfacultad y carrera), Numero de convocatoria, titulo de la convocatoria (examen de competencia para docente consultor),fecha publicación, fecha final.
* Estado de la convocatoria, como subido, revisado, observado
* Lo siguiente será dichas carpetas serán editadas, guardades por un usuario que tendrá su propio id, nombre, rol (secretaria, docente; etc), contraseña, que también tendrá la opción de mover las convocatorias entre carpetas, ya sea por seguridad o errores , etc
* Lo usuarios tendrán la opción de subir a las convocatorias documentos, los documentos tendra una plantilla subida por el vicerrectorado con la idea de tner un menú donde se pueda editar la resolución, titulo de convocatoria, dictamen, numero de carta, y el contenido necesario
* Vistas de las convocatorias, incluyendo un redireccion para ver el documento de su interios como un documento cualquiera que se ve en el navegador.

#### ############################################################################# BASE DE DATOS: Código de base de datos

**-- Crear la tabla de usuarios con restricción CHECK en Rol**

CREATE TABLE usuarios (

id VARCHAR(20) PRIMARY KEY,

Nombres VARCHAR(50) NOT NULL,

Apellido\_paterno VARCHAR(50),

Apellido\_materno VARCHAR(50),

Rol VARCHAR(20) CHECK (Rol IN ('admin', 'usuario', 'secretaria', 'decanatura', 'vicerrectorado')),

Contraseña VARCHAR(100) NOT NULL,

Celular VARCHAR(15)

);

**-- Crear la tabla de facultades**

CREATE TABLE facultad (

id\_facultad SERIAL PRIMARY KEY,

Nombre\_facultad VARCHAR(100) UNIQUE NOT NULL

);

**-- Crear la tabla de carrera**

CREATE TABLE carrera (

id\_carrera SERIAL PRIMARY KEY,

Nombre\_carrera VARCHAR(100) UNIQUE NOT NULL,

Cod\_facultad INT,

FOREIGN KEY (Cod\_facultad) REFERENCES facultad(id\_facultad)

);

**-- Crear la tabla de tipo de convocatoria**

CREATE TABLE tipo\_convocatoria (

id\_tipoconvocatoria SERIAL PRIMARY KEY,

Nombre\_convocatoria VARCHAR(100) UNIQUE NOT NULL,

);

**-- Crear la tabla de documentos**

CREATE TABLE documentos (

id SERIAL PRIMARY KEY,

documento\_path VARCHAR(255),

resolucion\_path VARCHAR(255),

dictamen\_path VARCHAR(255),

carta\_path VARCHAR(255),

id\_convocatoria INT,

FOREIGN KEY (id\_convocatoria) REFERENCES convocatorias(id\_convocatoria)

);

**-- Crear la tabla de convocatorias**

CREATE TABLE convocatorias (

id\_convocatoria SERIAL PRIMARY KEY,

cod\_convocatoria VARCHAR(100) NOT NULL,

Nombre VARCHAR(100),

Fecha\_inicio DATE NOT NULL DEFAULT CURRENT\_DATE,

Fecha\_fin DATE,

id\_usuario VARCHAR(20);

id\_tipoconvocatoria INT,

id\_carrera INT,

id\_facultad INT,

estado VARCHAR(20) CHECK (estado IN ('Para Revisión', 'En Revisión', 'Observado', 'Revisado')) DEFAULT 'Para Revisión,

prioridad VARCHAR(30);

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id),

FOREIGN KEY (id\_tipoconvocatoria) REFERENCES tipo\_convocatoria(id\_tipoconvocatoria),

FOREIGN KEY (id\_carrera) REFERENCES carrera(id\_carrera),

FOREIGN KEY (id\_facultad) REFERENCES facultad(id\_facultad)

);

**-- Crear la tabla de materia**

CREATE TABLE materia (

id\_materia SERIAL PRIMARY KEY,

codigomateria VARCHAR(100),

nombre VARCHAR(100) NOT NULL,

id\_carrera INT,  
 horas\_teoria INT,

horas\_practica INT,

horas\_laboratorio INT,

total\_horas INT;

FOREIGN KEY (id\_carrera) REFERENCES carrera(id\_carrera)

);

**\*\*\*\*\*\*\*triggers de tabla materias\*\*\*\*\*\*\***

**CREATE OR REPLACE FUNCTION actualizar\_total\_horas()**

RETURNS TRIGGER AS $$

BEGIN

NEW.total\_horas := NEW.horas\_teoria + NEW.horas\_practica + NEW.horas\_laboratorio;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

**#########otro trigger#########**

**CREATE TRIGGER trigger\_actualizar\_total\_horas**

BEFORE INSERT OR UPDATE ON materia

FOR EACH ROW

EXECUTE FUNCTION actualizar\_total\_horas();

CREATE TRIGGER trigger\_actualizar\_total\_horas

BEFORE INSERT OR UPDATE ON materia

FOR EACH ROW

EXECUTE FUNCTION actualizar\_total\_horas();

**-- Crear la tabla de convocatoria\_materia**

CREATE TABLE convocatoria\_materia (

id SERIAL PRIMARY KEY,

id\_convocatoria INT,

id\_materia INT,

TOTAL\_horas INT,

tiempo\_trabajo VARCHAR(30),

perfil\_profesional VARCHAR(100) NULL,

FOREIGN KEY (id\_convocatoria) REFERENCES convocatorias(id\_convocatoria),

FOREIGN KEY (id\_materia) REFERENCES materia(id\_materia)

);

**###########TRigers para convocatoria\_materia##################**

CREATE OR REPLACE FUNCTION sumar\_horas\_materia()

RETURNS TRIGGER AS $$

BEGIN

-- Sumar las horas de las materias y actualizar el campo TOTAL\_horas en convocatoria\_materia

UPDATE convocatoria\_materia

SET TOTAL\_horas = (

SELECT SUM(m.horas)

FROM convocatoria\_materia cm

JOIN materia m ON cm.id\_materia = m.id\_materia

WHERE cm.id\_convocatoria = NEW.id\_convocatoria

)

WHERE id\_convocatoria = NEW.id\_convocatoria;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

**#############otro tigger de convocatoria\_materia###########**

CREATE TRIGGER trigger\_sumar\_horas

AFTER INSERT OR UPDATE ON convocatoria\_materia

FOR EACH ROW

EXECUTE FUNCTION sumar\_horas\_materia();

**-- Crear la tabla de honorarios**

CREATE TABLE honorarios (

id\_honorario SERIAL PRIMARY KEY,

id\_convocatoria INT NULL,

id\_tipoconvocatoria INT NULL,

pago\_mensual VARCHAR(100) NULL,

resolucion VARCHAR(30),

dictamen VARCHAR(30),

FOREIGN KEY (id\_convocatoria) REFERENCES convocatorias(id\_convocatoria),

FOREIGN KEY (id\_tipoconvocatoria) REFERENCES tipo\_convocatoria(id\_tipoconvocatoria)

);  
  
**-- Crear la tabla de Generacion de PDF**

CREATE TABLE GeneracionPDF (

id\_generacion SERIAL PRIMARY KEY,

id\_convocatoria INT,

resolucion\_path VARCHAR(255),

dictamen\_path VARCHAR(255),

fecha\_generacion DATE,

FOREIGN KEY (id\_convocatoria) REFERENCES Convocatorias(id\_convocatoria)

);

**#########################triggers##########################**

**-- Creación de trigger para actualizar TOTAL\_horas en convocatoria\_materia**

CREATE OR REPLACE FUNCTION update\_total\_horas()

RETURNS TRIGGER AS $$

BEGIN

NEW.TOTAL\_horas := (

SELECT COALESCE(horas\_teoria, 0) + COALESCE(horas\_practica, 0) + COALESCE(horas\_laboratorio, 0)

FROM materia

WHERE id\_materia = NEW.id\_materia

);

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

**-- Integración del trigger**

CREATE TRIGGER before\_insert\_update\_convocatoria\_materia

BEFORE INSERT OR UPDATE ON convocatoria\_materia

FOR EACH ROW

EXECUTE FUNCTION update\_total\_horas();

**-- Crear un trigger para generar el código automáticamente**

CREATE OR REPLACE FUNCTION generar\_cod\_convocatoria()

RETURNS TRIGGER AS $$

BEGIN

NEW.cod\_convocatoria := LPAD(nextval('convocatoria\_codigo\_seq')::text, 3, '0');

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

**-- Asignar el trigger a la tabla de convocatorias**

CREATE TRIGGER trigger\_cod\_convocatoria

BEFORE INSERT ON convocatorias

FOR EACH ROW

WHEN (NEW.cod\_convocatoria IS NULL)

EXECUTE FUNCTION generar\_cod\_convocatoria();

### 2. Interfaz de Usuario (UI)

En la interfaz de usuario (UI), puedes implementar menús desplegables (dropdowns) que muestren los nombres en lugar de los códigos. Para lograr esto, deberás realizar consultas a la base de datos para obtener los nombres correspondientes a los códigos y mostrarlos en los componentes de la UI.

**Sprint 1: Configuración Inicial y Categorización**

**Objetivo:**

Configurar el entorno de desarrollo y comenzar con la estructura básica de categorización.

**Tareas:**

1. **Configurar el entorno de desarrollo**:
   * **Repositorios de código**: Crea un repositorio en GitHub o GitLab para tu proyecto.
   * **Herramientas de colaboración**: Configura herramientas como Slack para la comunicación del equipo y Trello o Jira para la gestión de tareas.
2. **Crear la estructura básica de la aplicación**:
   * **Backend**: Inicializa un proyecto con Node.js y Express.
   * **Frontend**: Inicializa un proyecto con React.
   * **Base de datos**: Diseña la estructura inicial utilizando postgres (Gestion\_de\_convocatorias)
3. **Diseñar la base de datos inicial**:
   * Define las tablas (colecciones) y campos necesarios para las categorías (facultad y carrera).
4. **Implementar la funcionalidad para crear y gestionar categorías**:
   * **Backend**: Implementa rutas API para crear, leer, actualizar y eliminar categorías.
   * **Frontend**: Crea formularios y vistas para que los usuarios puedan interactuar con las categorías.

**Salida esperada:**

* Entorno de desarrollo configurado.
* Estructura básica de la aplicación con backend y frontend.
* Funcionalidad inicial para la gestión de categorías.

**Sprint 2: Subdivisión de Convocatorias**

**Objetivo:**

Implementar la subdivisión de convocatorias dentro de las categorías.

**Tareas:**

1. **Desarrollar la funcionalidad para crear y gestionar tipos de convocatoria**:
   * **Backend**: Implementa rutas API para manejar los tipos de convocatoria (docente extraordinario, invitado, consultor, etc.).
   * **Frontend**: Crea formularios y vistas para gestionar los tipos de convocatoria.
2. **Implementar la lógica de clasificación de convocatorias dentro de categorías**:
   * Establece la lógica para que las convocatorias se organicen correctamente dentro de las categorías definidas.
3. **Realizar pruebas unitarias y de integración**:
   * Asegúrate de que la nueva funcionalidad esté probada y funcione correctamente dentro del sistema.

**Salida esperada:**

* Funcionalidad para subdividir convocatorias dentro de las categorías.
* Formularios y vistas para gestionar los tipos de convocatoria.
* Pruebas de funcionalidad.

**Sprint 3: Clasificación de Convocatorias**

**Objetivo:**

Implementar la clasificación completa de las convocatorias.

**Tareas:**

1. **Desarrollar la funcionalidad para clasificar las convocatorias**:
   * Clasificación basada en tipo, categoría, número de convocatoria, título, fechas.
2. **Implementar la lógica para gestionar el estado de las convocatorias**:
   * Estados como subido, revisado, observado.
3. **Crear vistas y formularios para la creación y edición de convocatorias**:
   * Interfaces de usuario que permitan crear y editar las convocatorias.
4. **Realizar pruebas unitarias y de integración**:
   * Asegurar la correcta funcionalidad de la clasificación de convocatorias.

**Salida esperada:**

* Funcionalidad para clasificar convocatorias.
* Vistas y formularios para gestión de convocatorias.
* Pruebas de funcionalidad.

**Sprint 4: Gestión de Usuarios**

**Objetivo:**

Implementar la gestión de usuarios y permisos.

**Tareas:**

1. **Desarrollar la funcionalidad para gestionar usuarios**:
   * Campos como ID, nombre, rol, contraseña.
2. **Implementar la lógica de autenticación y autorización**:
   * Uso de JWT o cualquier otro método de autenticación.
3. **Crear vistas y formularios para la gestión de usuarios**:
   * Interfaces de usuario para manejar la gestión de usuarios.
4. **Realizar pruebas de seguridad y validación de usuarios**:
   * Asegurar que la autenticación y autorización funcionen correctamente y sean seguras.

**Salida esperada:**

* Gestión de usuarios implementada.
* Vistas y formularios para la gestión de usuarios.
* Pruebas de seguridad y funcionalidad.

**Sprint 5: Documentos y Plantillas**

**Objetivo:**

Implementar la gestión de documentos y plantillas.

**Tareas:**

1. **Desarrollar la funcionalidad para subir y gestionar documentos**:
   * Implementar rutas API para subir y manejar documentos.
2. **Implementar plantillas proporcionadas por el vicerrectorado**:
   * Plantillas para resolución, título de convocatoria, dictamen, número de carta, contenido.
3. **Crear vistas y formularios para la edición y gestión de documentos**:
   * Interfaces de usuario para editar y gestionar documentos.
4. **Realizar pruebas unitarias y de integración**:
   * Asegurar la correcta funcionalidad de la gestión de documentos.

**Salida esperada:**

* Funcionalidad para subir y gestionar documentos.
* Vistas y formularios para la gestión de documentos.
* Pruebas de funcionalidad.

**Sprint 6: Vistas de Convocatorias**

**Objetivo:**

Implementar las vistas de convocatorias y redirección para visualizar documentos.

**Tareas:**

1. **Desarrollar las vistas de convocatorias**:
   * Permitir a los usuarios ver las convocatorias.
2. **Implementar la funcionalidad para redireccionar y visualizar documentos en el navegador**:
   * Integrar un visor de documentos en la aplicación.
3. **Crear vistas y formularios para la visualización de convocatorias y documentos**:
   * Interfaces de usuario para ver convocatorias y documentos.
4. **Realizar pruebas de usabilidad**:
   * Asegurar que la funcionalidad cumpla con los requisitos y sea fácil de usar.

**Salida esperada:**

* Vistas de convocatorias implementadas.
* Funcionalidad para visualizar documentos.
* Pruebas de usabilidad.

**Sprint 7: Mantenimiento y Optimización**

**Objetivo:**

Realizar tareas de mantenimiento, optimización y mejoras basadas en feedback.

**Tareas:**

1. **Revisar y optimizar el código y la base de datos**:
   * Mejoras en la eficiencia y rendimiento del sistema.
2. **Implementar mejoras basadas en el feedback de los usuarios**:
   * Ajustes y mejoras según las sugerencias y problemas reportados por los usuarios.
3. **Realizar pruebas adicionales y solucionar cualquier problema encontrado**:
   * Asegurar la estabilidad y funcionalidad del sistema.
4. **Documentar el sistema y preparar el lanzamiento**:
   * Crear documentación para el usuario final y el equipo de desarrollo.

**Salida esperada:**

* Sistema optimizado y mejorado.
* Documentación completa.
* Sistema listo para el lanzamiento.

**Resumen del Plan de Trabajo**

Cada sprint debe ser de duración razonable (por ejemplo, dos semanas), y al final de cada sprint, se debe realizar una revisión para asegurarse de que los objetivos se han cumplido. Este enfoque asegura una implementación gradual y estructurada del sistema de gestión de convocatorias.

Base de datos,