



**QUERCUS**  
**Project Design**

Juan Daniel Jossa Soliz  
Kevin David Rodriguez Riveros  
Michel Mauricio Castañeda Braga

Ingeniería de Software II G1  
Liliana Marcela Olarte Mesa

Universidad Nacional de Colombia  
Bogota DC  
2025

## TARJETAS CRC

### Usuario

|   |   |
|---|---|
| <b>Responsabilidades</b> <ul style="list-style-type: none"><li>• Registrar, autenticar y gestionar su perfil.</li><li>• Inscribirse en eventos y consultar historial.</li><li>• Recibir notificaciones y recordatorios.</li></ul> | <b>Colaboradores</b><br>Rol, Evento, Registro, Notificación |
|---|---|

### Rol

|  |  |
|--|--|
| <b>Responsabilidades</b> <ul style="list-style-type: none"><li>• Definir permisos de acceso y funciones del sistema.</li><li>• Asociar a los usuarios según su tipo.</li></ul> | <b>Colaboradores</b><br>Usuario, ControlAcceso |
|--|--|

### Evento

|   |   |
|---|---|
| <b>Responsabilidades</b> <ul style="list-style-type: none"><li>• Almacenar información del evento (título, fecha, lugar, tipo, estado).</li><li>• Publicarse para que los usuarios puedan inscribirse.</li><li>• Asociar inscripciones, pagos y asistencia.</li></ul> | <b>Colaboradores</b><br>Usuario, Registro, Pago, Notificación |
|---|---|

### Registro

|   |   |
|---|---|
| <b>Responsabilidades</b> <ul style="list-style-type: none"><li>• Vincular usuarios con eventos.</li><li>• Validar disponibilidad de cupos.</li><li>• Generar comprobantes y validar asistencia.</li></ul> | <b>Colaboradores</b><br>Usuario, Evento, Pago |
|---|---|

### Pago

|  |  |
|--|--|
| <b>Responsabilidades</b> <ul style="list-style-type: none"><li>• Registrar las transacciones realizadas.</li><li>• Confirmar o rechazar pagos según respuesta de la pasarela.</li><li>• Asociarse a las inscripciones.</li></ul> | <b>Colaboradores</b><br>Registro, Evento |
|--|--|

### ControlAcceso

|  |                                      |
|--|--------------------------------------|
| <b>Responsabilidades</b> <ul style="list-style-type: none"><li>• Validar autenticación y permisos.</li><li>• Restringir funciones según el rol.</li><li>• Gestionar inicio y cierre de sesión.</li></ul> | <b>Colaboradores</b><br>Usuario, Rol |
|--|--------------------------------------|

### Notificación

|  |   |
|--|---|
| <b>Responsabilidades</b> <ul style="list-style-type: none"><li>• Enviar avisos sobre eventos o cambios relevantes.</li><li>• Recordar fechas y confirmar</li></ul> | <b>Colaboradores</b><br>Usuario, Evento |
|--|---|

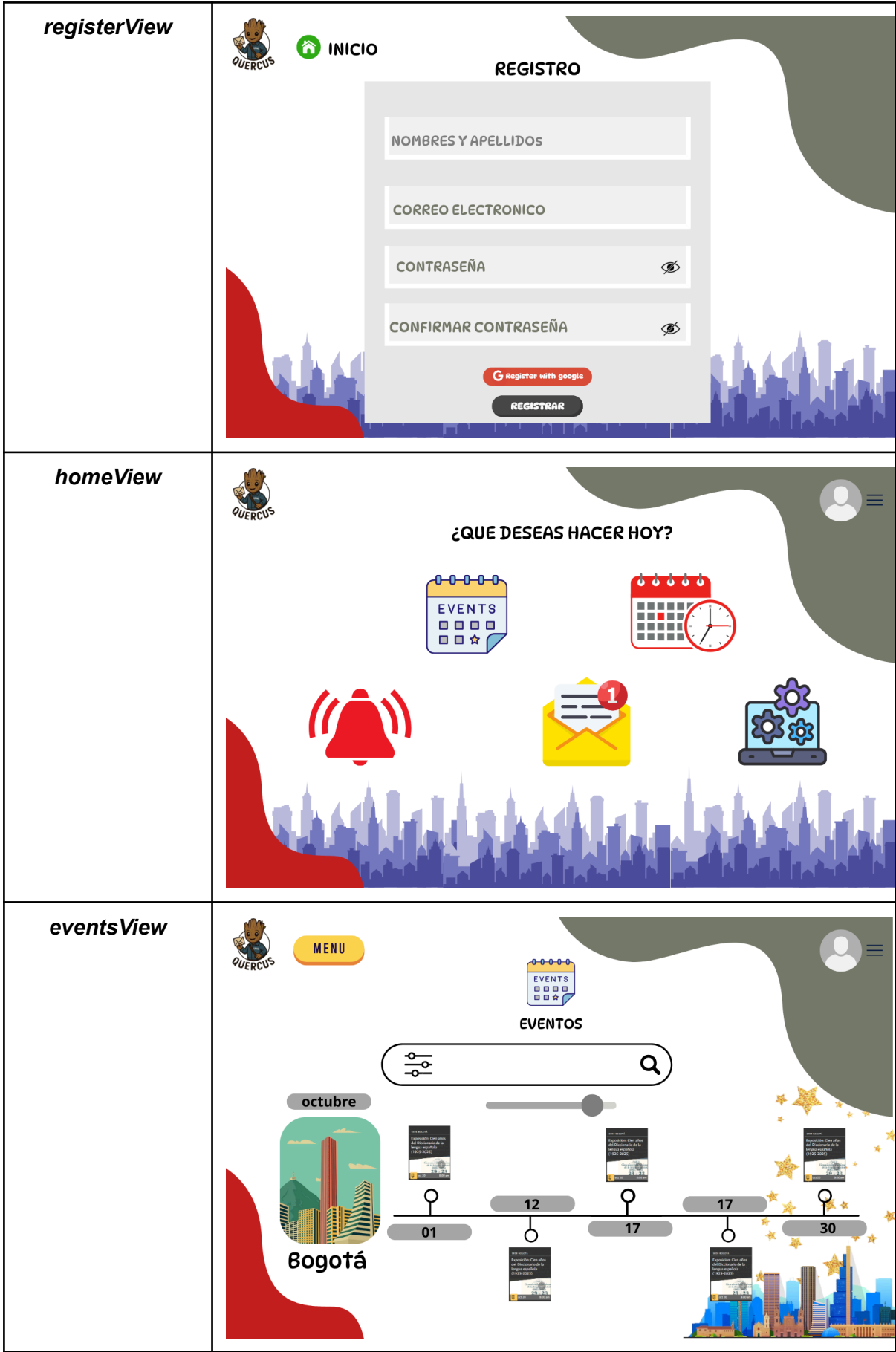
|  |  |
|--|--|
| inscripciones. <ul style="list-style-type: none"> <li>Alertar sobre modificaciones o cancelaciones.</li> </ul> |  |
|--|--|

## Reporte

|   |  |
|---|--|
| <b>Responsabilidades</b> <ul style="list-style-type: none"> <li>Generar reportes de participación, pagos y asistencia.</li> <li>Mostrar datos estadísticos por evento o periodo.</li> </ul> | <b>Colaboradores</b><br>Evento, Registro, Pago |
|---|--|

## MOCKUPS

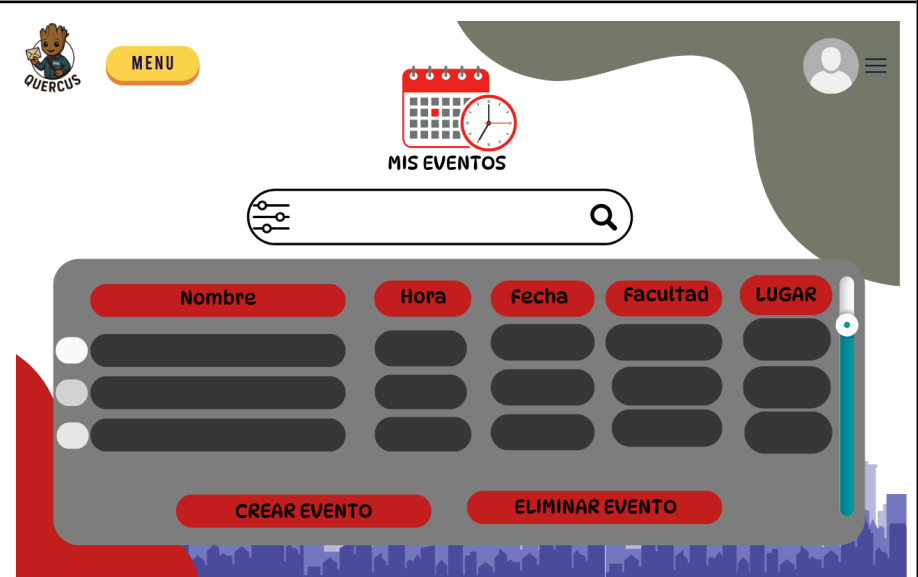
| Código             | Mockup   |
|--------------------|--|
| <i>welcomeView</i> |   |
| <i>loginView</i>   |  |



detailsView



managementView



createEventView



## **MODELO DE PROCESOS DE NEGOCIO (BPMN)**

El siguiente modelo BPMN representa el ciclo de vida completo del proceso de participación en eventos dentro de la plataforma Quercus, desde la creación del evento por parte del organizador hasta la validación final de asistencia.

El enfoque principal se centra en las interacciones entre los organizadores y los estudiantes, considerando al sistema como un actor autónomo que gestiona las validaciones, notificaciones y registros.

Este flujo refleja cómo la aplicación permite gestionar eventos académicos o estudiantiles de manera eficiente, asegurando la trazabilidad de cada paso —desde la publicación hasta la asistencia— e incorporando decisiones automáticas que garantizan la integridad de la información.

Además, se incluyen condiciones alternativas y posibles errores, como formularios incompletos, falta de cupos o códigos QR inválidos, los cuales son gestionados automáticamente por el sistema para mantener la continuidad operativa del proceso.

### **Lógica del Proceso**

#### **Creación y publicación del evento**

- El organizador inicia sesión y selecciona la opción “Crear evento”.
- Diligencia el formulario con la información requerida (título, fecha, ubicación, capacidad, etc.).
- El sistema valida los datos ingresados.
  - Si el formulario es válido → el evento se publica en el catálogo general.
  - Si no → el sistema muestra un mensaje de error y solicita corrección.

#### **Búsqueda e inscripción del estudiante**

- El estudiante inicia sesión y accede al listado de eventos.
- Utiliza filtros por categoría, fecha o ubicación para encontrar uno de interés.
- Selecciona el evento y elige la opción “Inscribirse”.
- El sistema verifica disponibilidad de cupos.
  - Si hay cupos → registra la inscripción, genera un código QR único y envía una confirmación al correo electrónico.
  - Si no hay cupos → el sistema muestra “Cupos agotados.”

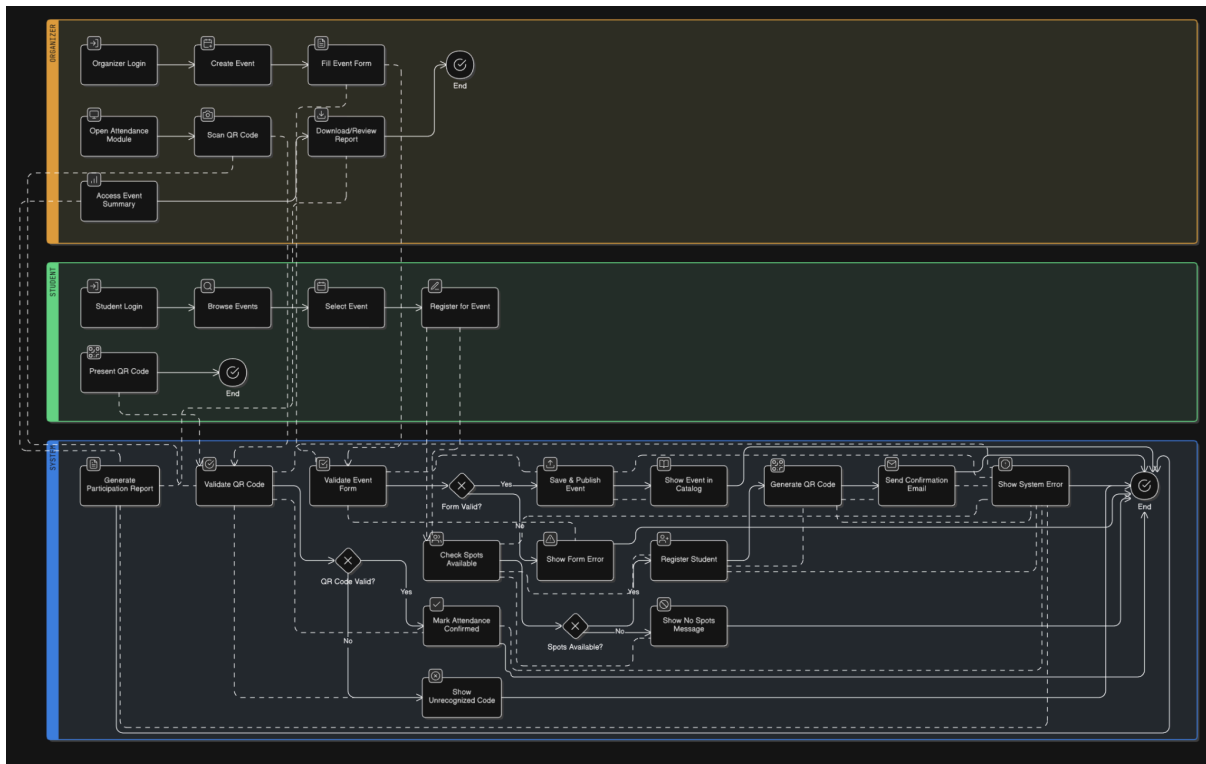
#### **Validación de asistencia durante el evento**

- En la fecha del evento, el organizador abre el módulo de asistencia.
- Cada estudiante presenta su código QR generado al inscribirse.
- El organizador escanea el código con la herramienta de verificación.
- El sistema valida la autenticidad del código.
  - Si es válido → marca la asistencia como “Confirmada.”
  - Si no → muestra el mensaje “Código no reconocido.”

#### **Generación de reportes**

- Al finalizar el evento, el organizador accede al panel de reportes.

- El sistema genera automáticamente estadísticas sobre inscritos, asistencia confirmada y estado del evento.
- El proceso concluye con la emisión del reporte consolidado del evento.



## DIAGRAMA DE ARQUITECTURA

El sistema está diseñado bajo un enfoque multicapa que separa las responsabilidades en diferentes niveles para garantizar mantenibilidad, seguridad y escalabilidad.

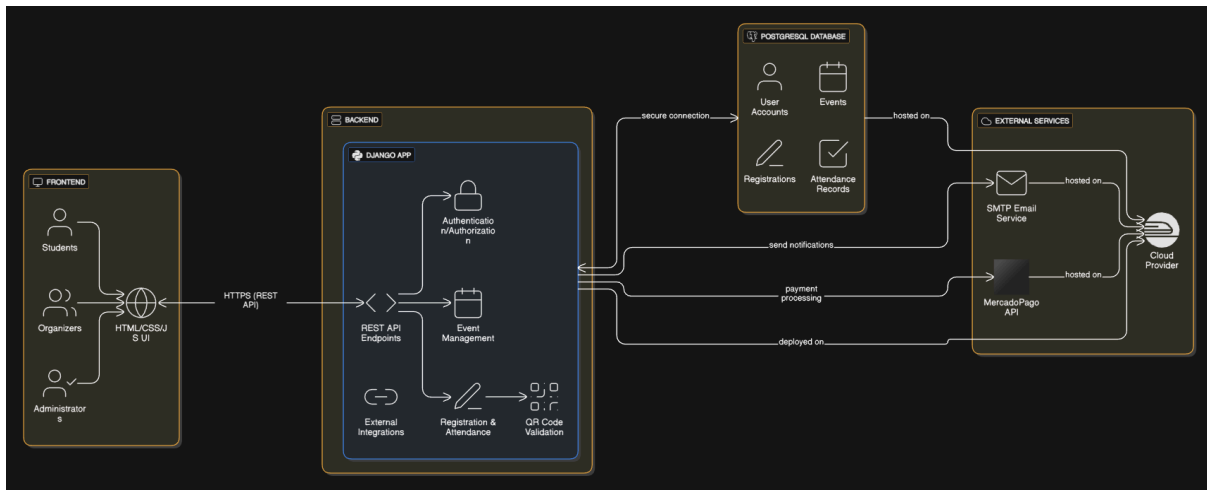
En la **capa de presentación**, los usuarios (estudiantes, organizadores y administradores) interactúan con la aplicación web mediante una interfaz desarrollada en HTML, CSS y JavaScript, la cual se comunica con el servidor a través de solicitudes REST API seguras (HTTPS).

La **capa lógica o de negocio** está implementada en Django (Python), responsable de gestionar la autenticación, el control de acceso por roles, la creación y registro de eventos, el manejo de inscripciones y la validación de asistencia mediante códigos QR.

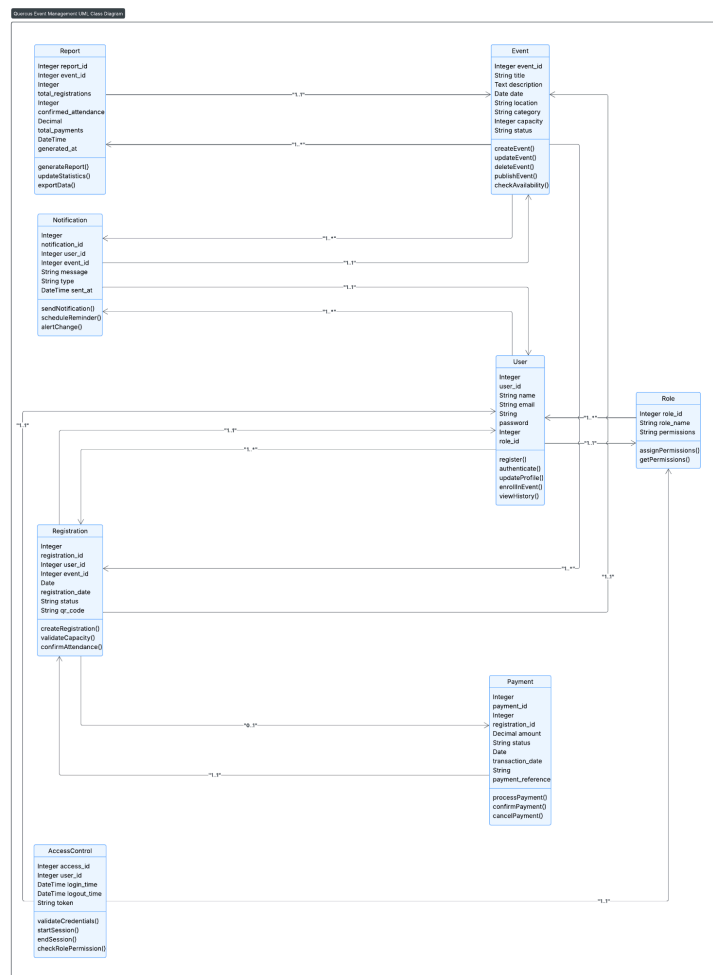
Esta capa también administra la comunicación con servicios externos como el sistema de correo electrónico (SMTP) para el envío de notificaciones y la API de MercadoPago, utilizada únicamente cuando se requieran pagos para ciertos eventos.

La **capa de datos** utiliza PostgreSQL como sistema de gestión de bases de datos relacional, donde se almacenan de manera segura la información de usuarios, eventos, registros de asistencia y transacciones.

Finalmente, todo el sistema se encuentra desplegado en un entorno en la nube (Render o Railway), lo que permite un acceso remoto, mantenimiento simplificado y escalabilidad de los componentes según las necesidades futuras del proyecto.



## DIAGRAMAS UML DE CLASES



## MODELO RELACIONAL DE BASE DE DATOS

En el sistema Quercus, las entidades principales corresponden a los actores y elementos que intervienen en el ciclo de vida de un evento académico: usuarios, roles, eventos,



registros, pagos, notificaciones, controles de acceso y reportes. Cada tabla incluye su identificador principal (PK) y las referencias necesarias (FK) para mantener la integridad referencial del modelo.

El modelo refleja las siguientes relaciones clave:

- Un Usuario puede estar asociado a un Rol que define sus permisos dentro del sistema.
- Un Usuario puede crear o inscribirse en múltiples Eventos.
- Un Registro vincula a un Usuario con un Evento, controlando la inscripción y la validación de asistencia.
- Un Pago se asocia opcionalmente a un Registro, en caso de eventos con cobro.
- Las Notificaciones se envían a los usuarios en función de sus interacciones con eventos o cambios relevantes.
- El Control de Acceso gestiona la autenticación de usuarios y las restricciones por rol.
- Los Reportes se generan a partir de los datos consolidados de eventos, inscripciones y pagos.

Este modelo servirá como base para la implementación física de la base de datos en PostgreSQL, garantizando integridad, escalabilidad y eficiencia en las operaciones del sistema.

