

PROYECTO INTEGRADO- REPORTE ESCRITO

Sprint 1

Grupo 6

David Henao

Santiago Castillo

Miguel Gonzalez

2024 – 1

Admin en Django:

A continuación, se muestran las tareas que se llevaron a cabo usando el panel de administración de Django y en donde la información entre paréntesis indica los elementos que se usaron para la implementación de dichas tareas:

- Se crea un super usuario y un usuario común sin rol asignado (uso comando 'django-superuser' y uso de la página de administrador de Django por defecto).
- Se implementa la funcionalidad de inicio de sesión, (uso de un formulario y método "Authenticate()" del módulo de sesiones de Django).
- Se implementa la funcionalidad de cierre de sección (uso el método "logout()" del módulo de sesiones de Django).
- Se crean los diferentes roles que interactúan con la aplicación:
 1. Líder de proceso
 2. Asistente de logística
 3. Miembro de comunidad académica
 4. Estudiante de la universidad(Uso el panel de administrador de Django).
- Se crean un usuario por cada rol, para la administración de funcionalidades y vistas (Uso el panel de administrador de Django).
- Se modifica el método de inicio de sección asignando usuarios por su rol (uso del método "user.groups.values_list()").
- Se desarrollan las funciones de renderización según el rol del usuario (uso del método "user.groups.values_list()").

Autenticación en Django, página de login:

Se implementó la funcionalidad de inicio de sesión, a través de un formulario donde se solicita el nombre de usuario y la contraseña. Estas credenciales son autenticadas a través del método “Authenticate()” del módulo de sesiones de Django, el cual verifica las credenciales en la base de datos. Posteriormente implementó la funcionalidad de cierre de sección, donde el servicio de usuario activo es detenido. Para esto, se usó el método “logout()” del módulo de sesiones de Django.

Una vez implementados los roles de los usuarios, Se modificó el método de inicio de sección para verificar el grupo del usuario que realiza el inicio de sección, para esto se usó el método “user.groups.values_list()”, que permite saber a qué rol pertenece el usuario, garantizando así la separación de los módulos por roles.

Vistas de las entidades:

Así como en el inicio de sesión, en los métodos de renderización de vistas se hizo uso del método “user.groups.values_list()”, para obtener el grupo de usuario. A continuación, se detallan las vistas de cada grupo en la aplicación:

- Líder de proceso: Tiene vistas que le permiten acceder a las solicitudes de eventos que realizan los miembros de comunidad académica, de esta forma este rol está en capacidad de aprobar o rechazar una solicitud. De manera similar, las vistas de un líder de proceso le permiten acceder a los eventos aprobados, modificarlos y eliminarlos. Es el único con la capacidad de marcar como “terminado” un evento, y el único capaz de generar una bitácora. De este modo, esta vista permite la gestión total de las solicitudes y los eventos.
- Asistente de logística: La vista del asistente de logística permite acceder a los eventos registrados, pero solo para marcar como hecho los requerimientos de un evento, ya que no se permite ni agregar ni eliminar los de la aplicación.
- Miembro de comunidad académica: Esta vista permite realizar una solicitud de un evento, que quedara en estado “pendiente” hasta que un líder de proceso la apruebe o la rechace. También permite realizar una solicitud de cambio sobre otra solicitud ya hecha, o sobre una solicitud ya aprobada.
- Estudiante de la universidad: Es un subgrupo del grupo “Miembro de comunidad académica”, que tiene la particularidad de que al momento de realizar una solicitud debe ingresar el nombre del profesor que dio el permiso para realizar la solicitud.

Para este propósito nuevamente se usó el panel de administrador de Django, asignando a cada rol sus permisos correspondientes.

Estado del tablero kanban del sprint backlog:

Proyectos / ProyectoIntegrador-ModuloEventos

Backlog

0 75 104 Completar sprint

PM-31 H.U.11.1 Verificación de credenciales de usuarios de la comunidad académica	H.E.11	FINALIZADA	8	DD
PM-32 H.U.11.2 Verificación de credenciales de miembros de la oficina de centro de servicios compartidos	H.E.11	FINALIZADA	8	MG
PM-16 H.U.1.1: Proceso/peticion de solicitudes	H.E.1	FINALIZADA	8	DD
PM-22 H.U.2.2 Cambiar Estado de solicitudes	H.E.2	FINALIZADA	8	DD
PM-23 H.U.3.2 Notificacion modificacion de evento	H.E.3	EN CURSO	13	DD
PM-21 H.U.1.2 Mensaje de confirmacion al docente	H.E.1	EN CURSO	20	DD
PM-15 H.U.10: Visualizacion de estado de peticion	H.E.10	FINALIZADA	5	DD
PM-19 H.U.4: Registro automatico de eventos	H.E.4	FINALIZADA	34	MG
PM-14 H.U.9: Asignacion/cambio de roles	H.E.9	EN CURSO	34	MG
PM-27 H.U.5.2 Lista de chequeo y modificacion de eventos	H.E.5	FINALIZADA	20	DD
PM-17 H.U.2.1: Administracion de solicitudes	H.E.2	FINALIZADA	13	DD
PM-18 H.U.3.1: Notificaciones para lideres	H.E.3	EN CURSO	8	DD

Retrospectiva del primer sprint:

Se hizo bien:

Grupal:

El grupo llevo un buen manejo en la repartición de tareas y en la búsqueda por oportunidades de mejora en el proyecto.

Individual:

Santiago Castillo: Completó todas las tareas asignadas, y todas de manera funcional.

David Henao: Desarrolló el estilizado de las vistas, y la implementación de funcionalidades correspondientes a pesar de su situación personal.

Miguel Gonzalez: Retroalimentó al equipo constantemente con material y documentación para llevar a cabo las tareas del desarrollo.

Se hizo mal:

Grupal:

El grupo en general llevo un mal manejo en los tiempos acordados. Además, que se tenía mala comunicación entre los integrantes, haciendo difícil la discusión de los objetivos a llevar a cabo.

Individual:

Santiago Castillo: Durante una parte del desarrollo, no siguió la estandarización acordada a la hora de implementar (programaba en español, nombramiento erróneo de archivos y variables, etc). Además de retrasos en los tiempos de entrega acordados.

David Henao: Tiempo de demora prudencial a la hora de entregar los avances acordados.

Miguel Gonzalez: Estancamiento en el avance del proyecto debido a problemas con ficheros. Estancamiento en el avance del proyecto debido a enfoque excesivo de corrección de la estandarización. Además de ciertos errores de escritura a nivel de código.

Se puede mejorar:**Grupal:**

Hay oportunidades de mejoras sobre la responsabilidad de las entregas y la comunicación entre los integrantes. Se debe llevar una a cabo una reunión para encontrar los problemas que originan estos contratiempos.

Individual:

Santiago Castillo: La responsabilidad a nivel de acuerdos a nivel de desarrollo, de modo que no haga falta realizar correcciones de estandarización.

David Henao: Mostrar mayor preparación en el tiempo de desarrollo de modo que imprevistos no representen retrasos significativos en el desarrollo.

Miguel Gonzalez: Demostrar mayor habilidad de priorización de tareas, de modo que se logren primero actividades significativas y luego las correcciones.