

Práctica 3:

Asignatura: Programación Web, 3º Grado de Ingeniería Informática
Escuela Politécnica Superior de Córdoba - Universidad de Córdoba
2023-2024



UNIVERSIDAD DE CÓRDOBA

Trabajo realizado por:

-Enrique de los Reyes Montilla

i12remoe@uco.es

-Manuel García Obrero

i82gaobm@uco.es

-Francisco José Mellado Ortiz

i12meorf@uco.es

-Lucía Téllez López

i12telol@uco.es

Índice

Sobre la pila de tecnología	3
Sobre la aplicación	3
Sobre la base de datos	4
Instrucciones de acceso	6
Decisiones de diseño/implementación	7
Bibliografía	7

Sobre la pila de tecnología

La pila de tecnología usada contiene:

- Java. Se ha usado Java para modelar los DTOs y DAOs que nos permitirán traspasar datos de la base de datos (BD) a la aplicación y viceversa.
- JavaScript. Se ha usado JavaScript para permitir el dinamismo e interactividad de la aplicación sobre la web, y controlar la conectividad de las diferentes páginas y funcionalidades.
- HTML. Se ha usado el lenguaje HTML para crear las páginas que se mostrarán al usuario que esté usando la aplicación de Camps Manager
- CSS. Se ha usado el lenguaje CSS para marcar el estilo de los elementos con los que el usuario interactúa desde las interfaces.
- SQL. Se han usado sentencias SQL para crear y modificar la base de datos que contiene la información manejada por la aplicación Camps Manager.
- Apache Tomcat. Se ha usado Apache Tomcat para levantar el servidor que nos permitirá acceder a la aplicación desde nuestro navegador.

Sobre la aplicación

El usuario que utilice este sistema se encontrará primero con una página “*index*” que le permitirá registrarse en el sistema (*registerController*, *registerView*) o iniciar sesión (*login*, *loginView*) con una cuenta ya existente. Una vez iniciada la sesión, recibirá diferentes opciones según el tipo de cuenta, también desde la página “*index*”.

Una cuenta de administrador tendrá las opciones de navegación, y cada opción se compone de los siguientes controladores y vistas:

- Crear un campamento.
Controlador: CreateCamp Vista: createCampView
- Crear una actividad.
Controlador: CreateActivity Vista: createActivityView
- Crear un monitor.
Controlador: CreateMonitor Vista: createMonitorView
- Asociar una actividad a un campamento.
Controlador: AssociateActivitytoCamp Vista: associateActivitytoCampForm
- Asociar un monitor a una actividad.
Controlador: AssociateMonitortoActivity Vista: associateMonitortoActivityView
- Asociar un monitor supervisor a un campamento.
Controlador: AssociateSupervisortoCamp Vista: associateSupervisortoCampView
- Asociar un monitor especial a un campamento.
Controlador: AssociateSpecialMonitortoCamp Vista: associateSpecialMonitortoCampView

De igual modo, una cuenta de cliente se compondrá de:

- Buscar campamentos filtrados por fechas de inicio y fin.
Controlador: CheckCampsDate Vista: checkCampsDateView
- Buscar campamentos filtrados por nivel educativo.
Controlador: SearchCampamentByEducationalLevel Vista: searchCampamentByEducationalLevel
- Buscar campamentos filtrados por plazas disponibles.

Controlador: SearchCampamentByAvailableSeats Vista: searchCampamentByAvailableSeats

- Inscribirse.

Controlador: DoRegistrationPartial, DoRegistrationComplete Vista: doRegisterPartialView, doRegisterCompleteView

- Cancelar una inscripción.

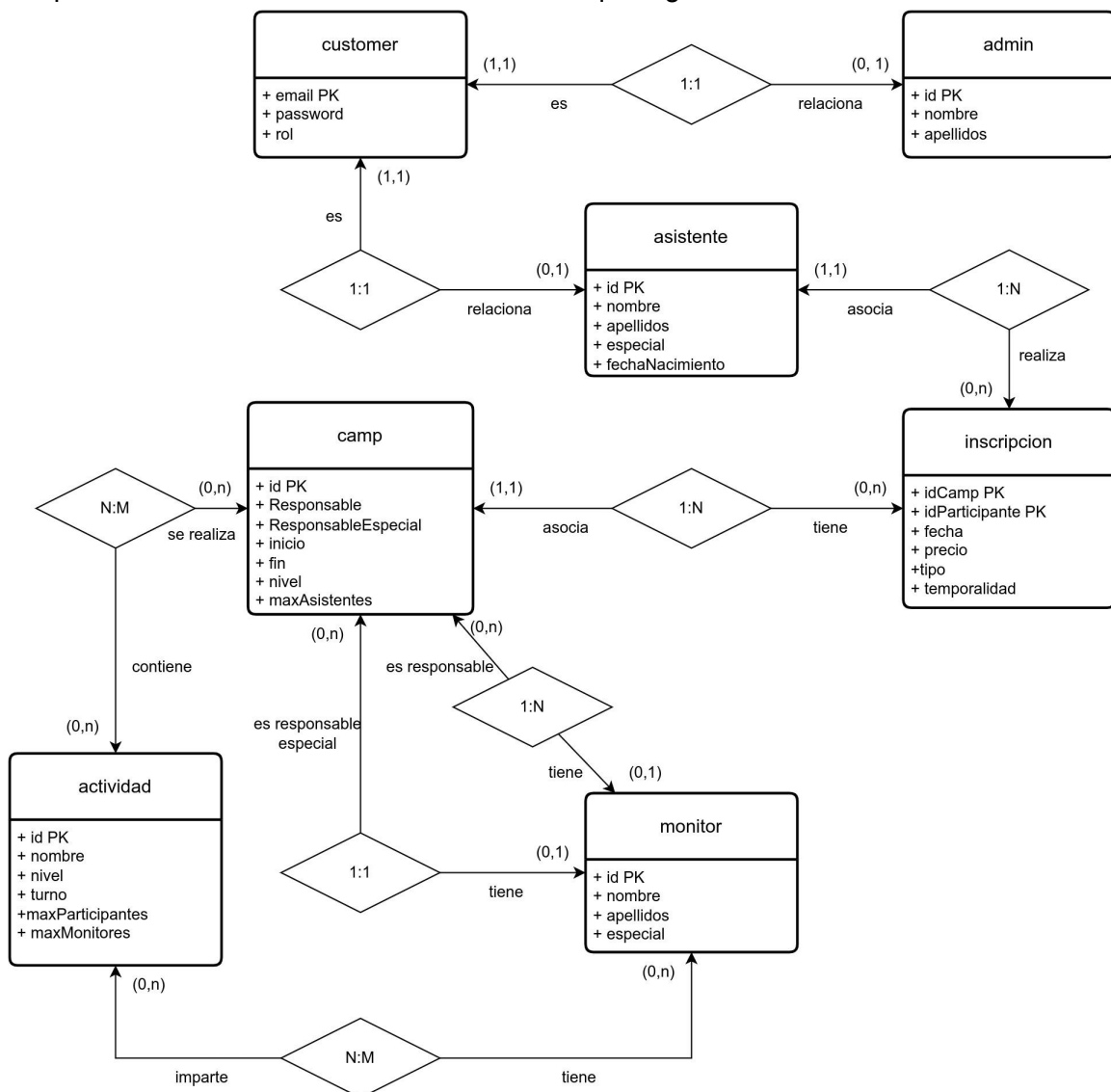
Controlador: CancelRegistration Vista: cancelRegisterView

Un usuario que ha hecho *login* puede acceder a las páginas de “*edit profile*” y “*log out*” en cualquier momento. Si el usuario decide hacer *log out*, será redirigido a la vista de “*index*” después de finalizar la operación.

En caso de cometer algún error navegando por las diferentes páginas (por ejemplo, introducir mal los datos de un formulario o acceder a una parte a la que no tiene acceso), se le enviará a una página de error, y el usuario solo podrá elegir volver a la página “*index*”.

Sobre la base de datos

El esquema relacional de la base de datos es el que sigue:



Los DAO usados son:

- InterfaceDAO: DAO del que heredan los demás DAOs.
- ActividadDAO: realiza las operaciones pertinentes respecto a actividades en la base de datos. Accede a la tabla “*actividad*”, “*campamento_actividad*” y “*actividad_monitor*”. Sus métodos son:
 - create(ActivityDTO)
 - read(ActivityDTO)
 - read(int)
 - delete(ActivityDTO)
 - readAll()
 - readMonitorsActivity(int)
 - addActivity(int, int)
- AdminDAO: realiza las operaciones pertinentes respecto a administradores en la base de datos. Accede a la tabla “*admin*”. Sus métodos son:
 - create(AdminDTO)
 - read(AdminDTO)
 - delete(AdminDTO)
 - update(AdminDTO)
 - read(int)
 - read(string)
- AsistenteDAO: realiza las operaciones pertinentes respecto a asistentes en la base de datos. Accede a la tabla “*asistente*”. Sus métodos son:
 - create(AssistantDTO)
 - read(AssistantDTO)
 - delete(AssistantDTO)
 - readAll()
 - update(AssistantDTO)
 - read(int)
 - read(string)
- CampamentoDAO: realiza las operaciones pertinentes respecto a campamentos en la base de datos. Accede a la tabla “*campamento*” y “*campamento_actividad*”. Sus métodos son:
 - create(CapmDTO)
 - read(CapmDTO)
 - delete(CapmDTO)
 - readAll()
 - readAllAvailable()
 - readAllAvailableSeats(int)
 - readAllAvailableLevel(Nivel)
 - count(CampDTIO)
 - readAvailable(int)
 - read(int)
 - updateResponsable(int, int)
 - updateEspecial(int, int)
 - readActivitiesCamp(int)
 - addActivity(int, int)
- CustomerDAO: realiza las operaciones pertinentes respecto a clientes en la base de datos. Accede a la tabla “*customer*”. Sus métodos son:

- create(CustomerDTO)
- read(CustomerDTO)
- delete(CustomerDTO)
- update(CustomerDTO)
- read(string)
- InscripcionDAO: realiza las operaciones pertinentes respecto a inscripciones en la base de datos. Accede a la tabla "*inscripcion*". Sus métodos son:
 - create(RegistrationDTO)
 - read(int, int)
 - read(RegistrationDTO)
 - delete(RegistrationDTO)
 - delete(int, int)
 - readAll()
 - readAllComplete()
 - readAllPartial()
- MonitorDAO: realiza las operaciones pertinentes respecto a monitores en la base de datos. Accede a la tabla "*monitor*" y "*actividad_monitor*". Sus métodos son:
 - create(MonitorDTO)
 - read(MonitorDTO)
 - read(int)
 - delete(MonitorDTO)
 - readAll()

Instrucciones de acceso

Una vez nos encontremos en la página principal "*index*", tendremos dos opciones:

- Crear una cuenta nueva con la opción "*sign up*".
- Iniciar sesión con una cuenta ya existente con la opción "*log in*".

Para **crear una cuenta**, el usuario deberá introducir los siguientes datos: email, nombre, apellido(s), contraseña, seleccionar uno de los dos roles disponibles, fecha de nacimiento, y seleccionar si necesita atención especial.

Si se deja algún campo sin rellenar, el usuario no puede ser creado correctamente, y se devuelve a la página de "*sign up*". En caso de realizar un registro correcto, se abre la sesión del nuevo usuario.

Para **iniciar sesión**, se deben introducir el email y contraseña del usuario que quiere iniciar sesión.

Si el usuario existe en la base de datos, se permite el acceso y se abre su sesión; en caso contrario, se devuelve al usuario a la vista de "*log in*".

Los usuarios ya creados en el sistema son:

Email: manuelgarciaobrero@gmail.com	;Password: Manuel	;Rol: Client
Email: i82gaobm@uco.es	;Password: Manuel	;Rol: Admin
Email: prueba@prueba.com	;Password: 1234	;Rol: Client
Email: i12meorf@uco.es	;Password: Francisco	;Rol: Admin
Email: i12remoe@uco.es	;Password: 1234	;Rol: Client

Decisiones de diseño/implementación

En relación al código de las dos prácticas anteriores, hemos necesitado añadir funcionalidades relativas a la búsqueda de campamentos entre dos fechas, búsqueda según las plazas disponibles y búsqueda por nivel educativo, que se nos pedían en esta práctica.

Decidimos añadir estos métodos en los gestores y DAOs, ya que son los que se ocupan de administrar las interacciones entre la base de datos y el programa.

Bibliografía

Al igual que en las prácticas anteriores, hemos usado Javadoc para documentar nuestro código, al igual que tutoriales de la W3C para trabajar con CSS y HTML:

- [Documentación de paquetes de Java](#)
- [Documentación MySQL](#)
- [Guía de conector JDBC](#)
- [Algunos tutoriales de iniciación a CSS de World Wide Web Consortium](#)
- [Algunos tutoriales de iniciación a HTML de World Wide Web Consortium](#)