

# Especificación de requerimiento de software

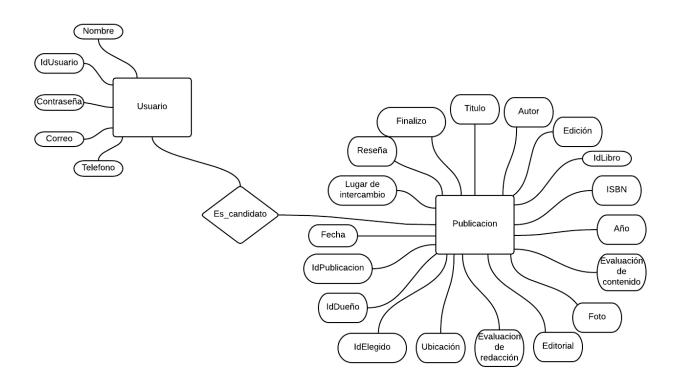
Enunciado del problema Diagrama de la base de datos Mapa de navegación Información del CSS ocupado



### Enunciado del Problema

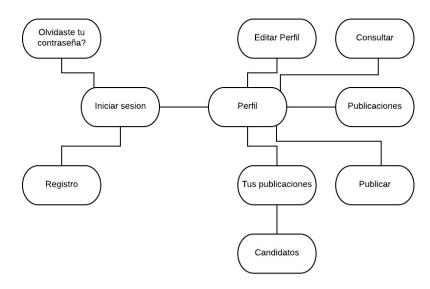
Se pretende desarrollar una plataforma en línea que permita a los usuarios intercambiar libros usados, indicando un punto de intercambio para libros previamente publicados, permitiendo a otros usuarios proponer un libro para cambiarlo, a su vez el sistema permitirá escoger al usuario publicador el candidato que más le agrade. El intercambio debe efectuarse de forma física, la plataforma sólo ofrece la posibilidad de publicar y ofertar, para ponerse de acuerdo.

### Diagrama de la base de datos





# Mapa de navegación





## Diagrama de clases

# Bean

#### CandidatosBean

-Publicacion publicacion -EsCandidatoC helper -PublicacionC conP -List<Usuario> listaCandidatos -FacesContext faceContext -FacesMessage message

### ListaPublicacionesBean

PublicacionC hekper
EsCandidatoC canHelper
Usuario usuario
HttpServletRequest httpServeletRequest
FacesContext facesContext
FacesMessages message

publicaciones(): List<Publicacion> mostrarImagen(int id): inputStream update(): void pedir(Publicacion publicacion Solicitada): String

#### UsuarioBean

Usuario usuario
Usuario usuarioActual
HttpServletRequest httpServletRequest
FacesContext facesContext
FacesMessages messages
UsuarioC helper

registrar(): String
cifrarcontrsena(): String
login(): String
cerrarSesion(): String
editarDatos(): String
verificarSesion(): Boolean
getUsuario(): Usuario
setUsuario(Usuario usuairo): void
recuperar(String correo): String
nuevaContrasena(): String

#### editarPublicacionBean

Usuario usuario
HttpServeletRequest httpServeletRequest
FacesContext Facescontext
FacesMessages facesMessages
Publicacion publicacion
PublicacionC helper

getPublicacion() Publicacion setPublicacion(Publicacion publicacion): void editar(): String

#### PublicacionBean

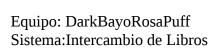
Usuario usuario
Publicacion publicacion
HttpServletRequest httpServletRequest
FaceContext faceContext
FaceMessage message
PublicacionC helper
Part imagen
UploadedFile image

registrarPublicacion(): String
esAjena(Publicacion publicacion): Boolean
update(): void
guardarImagen(): void
validaDireccion(String direccion): Boolean
getPublicacion(): Publicacion
setPublicacion(Publicacion publicacion): void
getImagen(): Part
setImagen(Part imagen): void
getImage(): UploadedFile
setImage(UploadedFile image): void

#### ConsultarBean

-String clave
- ConsultarC termino
-PublicacionC helper
- UsuarioC helperUsuario
-List<Publicacion> resultados
-ArrayList<Usuario> resultadosUsuario
-ArrayList<Publicacion> resultadosPublicaciones
-Usuario usuario
-HttpServletRequest httpServletRequest
-FacesContext faceContext
-FaceMessage faceMessage
-Long lastModified
-EsCandidatoC canHelper

buscar(): String buscarUsuarios(): Arraylist<Usuario> buscarPublicaciones(): Arraylist<Publicaciones> buscarPublicacionesUsuarios():Arraylist<Publicaciones> listaPublicaciones(): List<Publicaciones> esAjena(Publicacion publicacion): Boolean update(): void getClave(): String setClave(String c): void getTermino(): consultarC getResultados(): List<Publicacion> getUsuario.(): Usuario setUsuario(Usuario usuario): void getLastModified(): Long setLAstModified(Long lastModified): void mostrarImagen(int id): InputStream editaPublicacion(Publicacion publicacion): String pedir(Publicacion solicitada): String publicaciones(): List<Publicaciones> redireccionaCandidatos(Publicacion p): String finaliza(Publicacion p): String elimina(Publicacion p): String republica(Publicacion p): String getListaPublicaciones(): set<Publicacion>





#### ConsultarC

ArrayList<Publicacion> resultados ArrayList<Usuario> resultadosUsuario ArrayList<Publicacion> resultadosPublicaciones Session session

obtenerPalabras(String cadena): String
buscar(String clave): List<Publicacion>
buscarUsuarios(): List<Usuario>
buscarPublicaciones(): List<Publicacion>
buscarPublicaciones(): List<Publicacion>
buscarPublicacionesUsuario(Usuario usu): List<Publicacion>
prestarPublicacion(Publicacion publicacion,Usuario usu): void
getResultados(): ArrayList<Publicacion>
setResultados(ArrayList<Publicacion> r): void
listaPublicaciones(Usuario u): List<Publicacion>

#### PublicacionC

Session session Usuario usuario List<Publicacion> lstPublicaciones

registarBD(Publicacion publicacion,Usuario usu): void buscarPublicacion(Int id): Publicacion listar(): List<Publicacion> getUsuario(int id): Usuario actualizarPublicacionBD(Publicacion publicacion): void getNextIdPublicacion(): Int getPublicacion(Int id): Publicacion actualizarPublicacion(Publicacion P): void elimina(Publicacion p): void

### Logic

### HibernateUtil

SessionFactory sessionFactory getSessionFactory(): SessionFactory

#### UsuaripoC

Session session

registarBD(Usuario usuario): void getSession(): session autentificar(Usuario usuario): Usaurio buscarPorCorreo(String correo): Usaurio actualizarUsuarioBD(Usuario usuario): void

#### **EsCAndidatoC**

Session session

registrarCAndidatos(Usuario usu, Publicacion p): void listaCandidatos(Publicacion p): List<Usuario> eliminarCandidato(Usuario usu, Publicacion p): void

#### Correo

#### Correo

Email email

enviarCorreo(Strin asunto, String Mensaje, String destinatario): void



# Información del CSS usado

http://www.free-css.com/free-css-templates/page184/infusion