

Cada Permission tendra su propia enumeracion de acciones posibles a chequear sobre la misma. Por defecto, se asume acceso o lectura.

El manager obtendra la coleccion de permisos asociados al usuario actual segun el rol y funciones asociadas.

Preguntara a cada permiso si puede manejar el objeto que estamos intentando validar y en caso afirmativo, verificara si el usuario tiene permisos para el mismo. Si al menos se encontrara un permiso, pero no valido ninguno correctamente, se devolvera el NotFoundPermissionDefault. En caso afirmativo, verdadero al encontrar un permiso que sea suficiente.

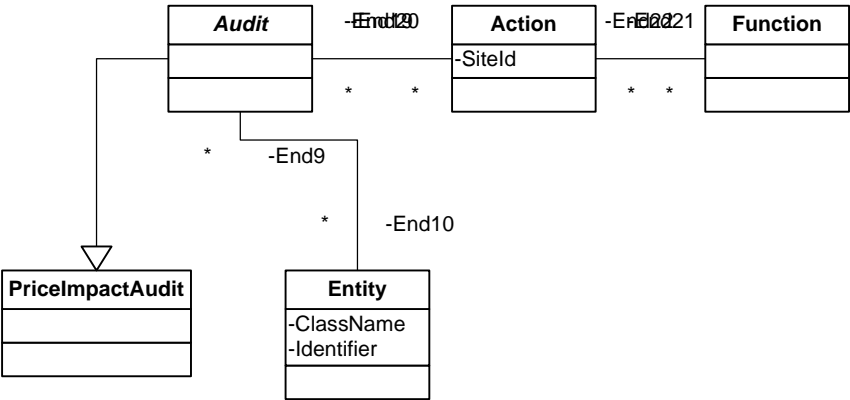
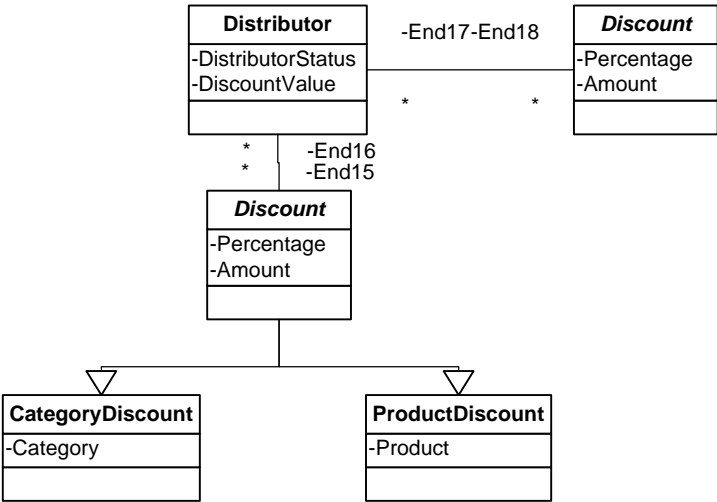
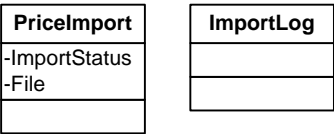
Si no se encontro ningun permiso para este tipo de objeto, se devolvera falso, esto asegura la correcta configuracion de la aplicacion.

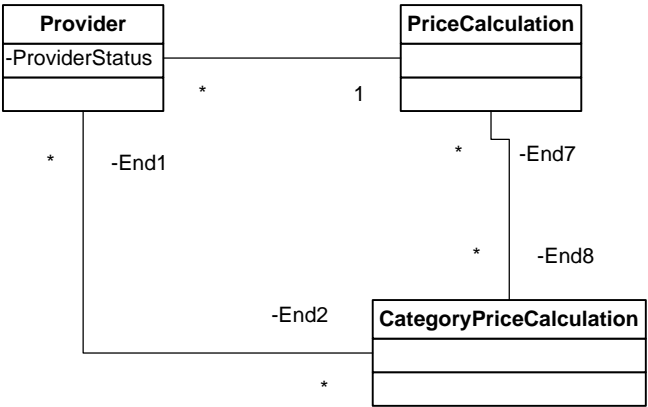
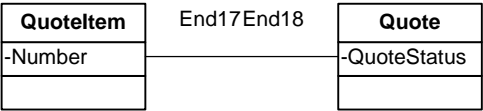
Un permiso puede ser asignado a uno o varias funcionalidades.

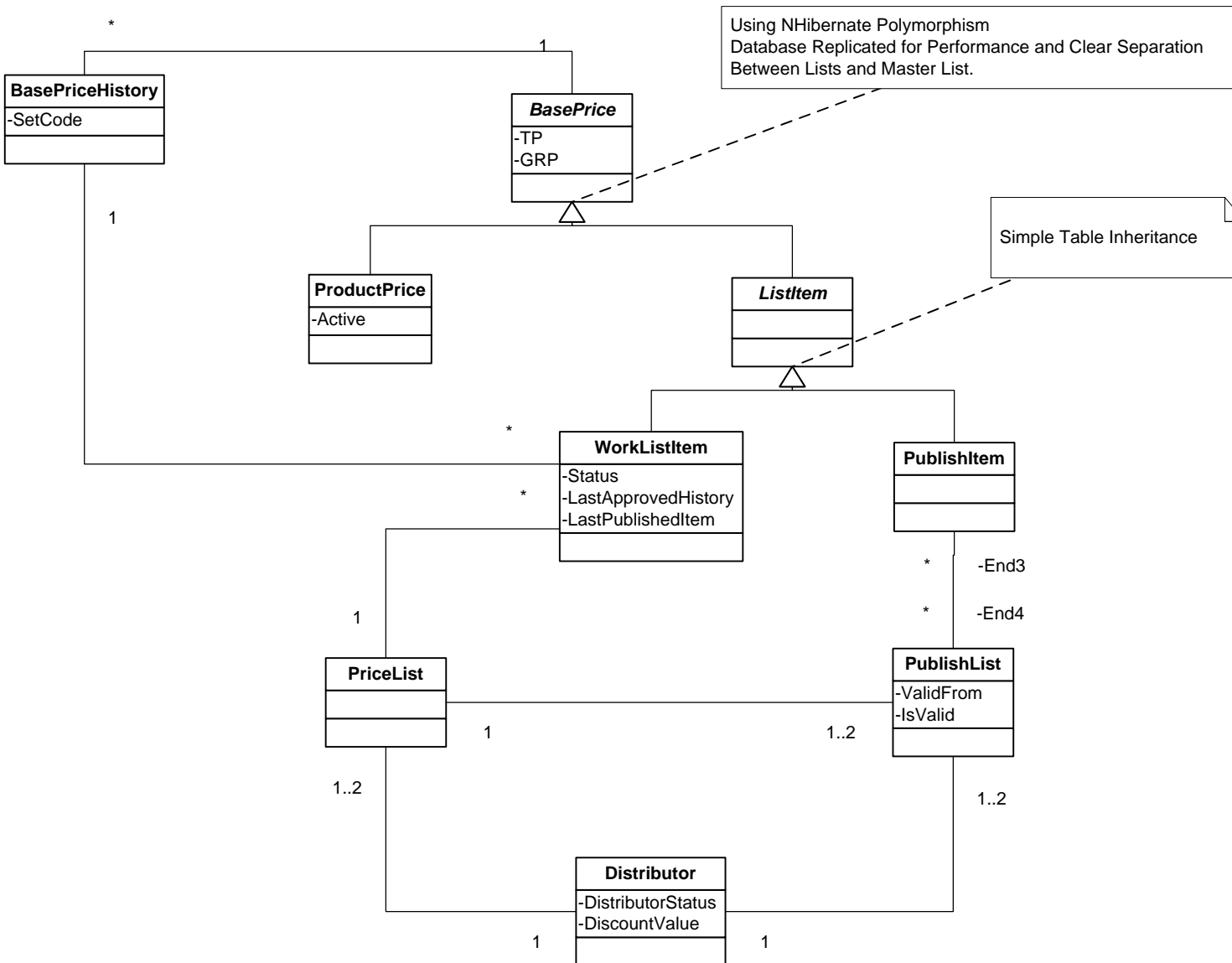
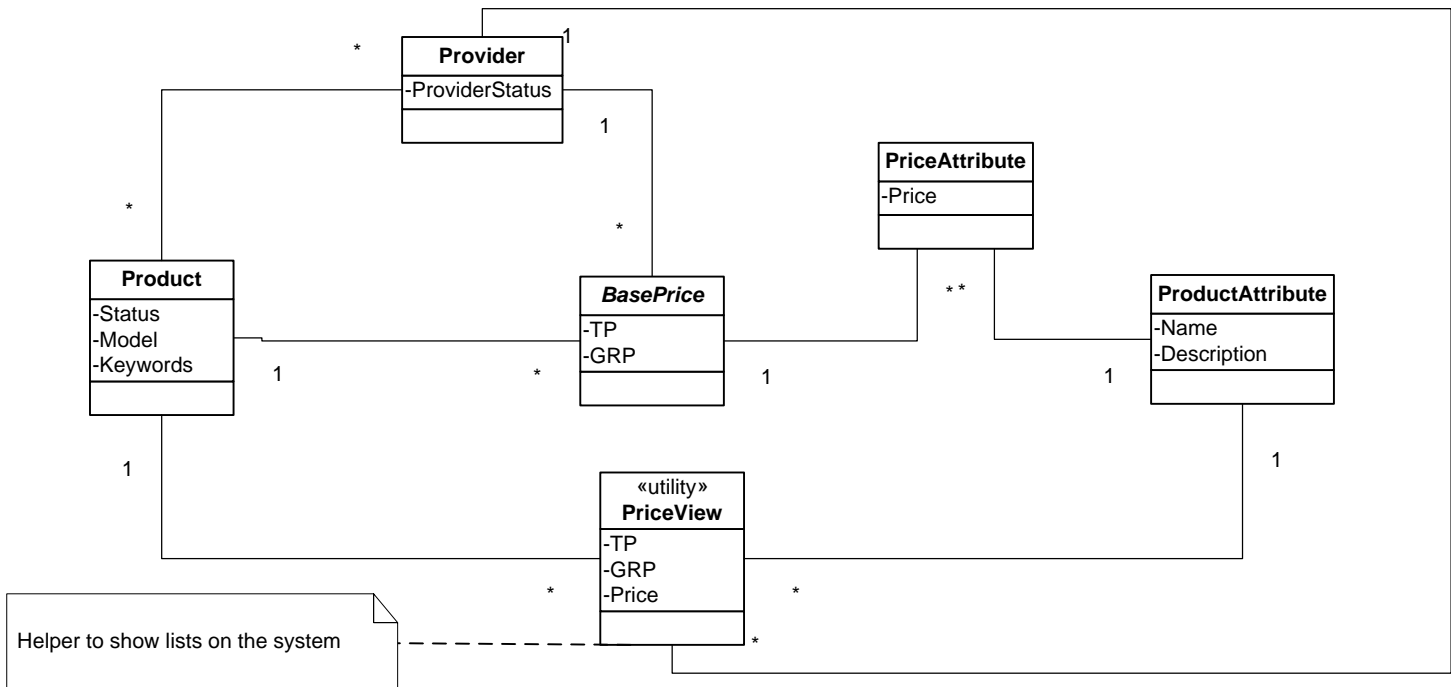
Una funcionalidad puede ser asignada a uno o varios roles por sitio.

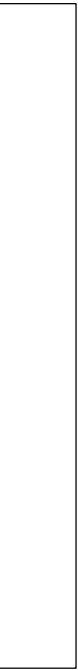
Un funcionalidad puede ser jerarquico, por ende, si un rol tiene una funcionalidad padre : le pertenecen sus permisos y lo de sus hijos.

Un usuario puede pertenecer a un rol, pero ademas siempre debe estar relacionado con un sitio para que sea valido en nuestro modelo.









aration

tance

«utility» PriceListController
+Duplicate() +Create() +Publish() +Remove() +Edit() +GetActiveList() +GetValidList() -MarkForApproval() +RemoveProduct() +CanCreate() +CanEdit() +AddProducts() +CreateFromSelection()

«utility» ListController
+ChangePriceList(in PricesToChange) +GetFilteredItems() +Export() +FindPriceByCode() +RollbackSet() +RollbackItem() +ApproveChanges() +RejectChanges() +GetHistorySets() +GetHistorySetItems()

«utility» MasterListController
+AddProductsToPriceList() +DeactivateProducts() +ActivateProducts() +AddToSelection() +RemoveFromSelection() +ChangePricePurchase() +ChangePriceSuggested() -ImpactPriceChangesToChildLists() +GetInactiveItems()

«utility» Import
+AddImportFile() +SimulateImport() +ConfirmImport() +ExportImportList()

«utility» DistributorController
+GetAssignedList() +ChangeAssignedList(in ApplyToPublished) +ListDiscounts() +AddDiscount() +RemoveDiscount() +ChangeDiscount() +Edit() +View()

«utility» QuoteController
+Create() +Remove() +SendToDistributor() +Edit() +SendAlertMailing() +Close()

«utility» ProviderController
+GetFilteredItems() +ListPricePolicy() +AddPricePolicy() +RemovePricePolicy() +Edit() +View() +AssignProduct()

«utility» ProductController
-GetCategoriesAssigned -GetHistory -GetAssignedLists -Edit -Remove

«utility» Messaging
+AddMessageToEntity() +RemoveMessage() +GetDashboardMessages() +AddAlertMessage()

«utility» ScalaIntegrator
+IntegrateDistributors() +IntegrateProviders()

«utility» Import
ImportFile() LateImport() FirmImport() PortImportLog()