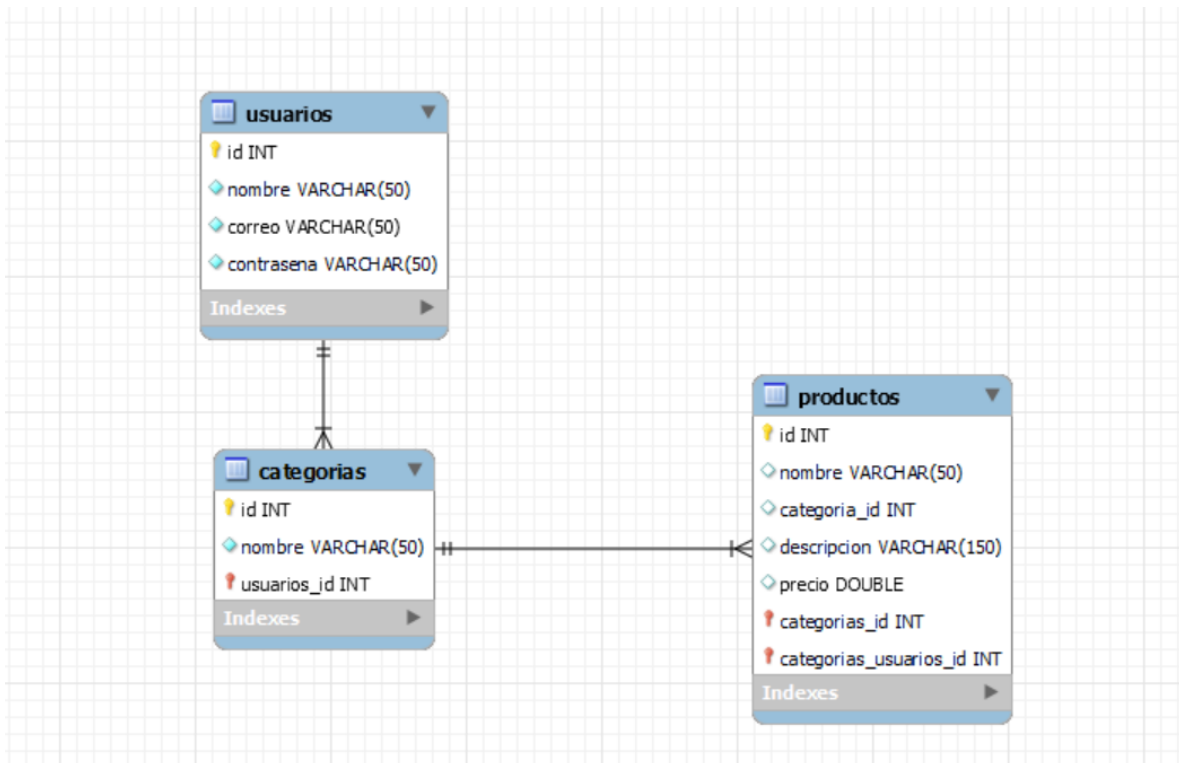




SPRINT 2: Repositorio de Código - Diseño Base de Datos

Identificación Proyecto	
Nombre Proyecto:	Tienda de camisetas_C3niv
Número Equipo:	Individual
Integrantes del equipo	
Rol (Líder-Desarrollador – Cliente)	Nombre
Líder	Jorge Alexander Alonso Morales
Diseño de la Base de Datos (Proceso de normalización)	

Como evidencia del Diseño de la Base de Datos.





Esquema de la Base de Datos (Código SQL)

Como evidencia del Esquema de la Base de Datos, se debe presentar el código SQL de creación de la base de datos.

```
CREATE DATABASE IF NOT EXISTS `tiendadecamisetasc3niv` /*!40100 DEFAULT
CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci */ /*!80016 DEFAULT
ENCRYPTION='N' */;
USE `tiendadecamisetasc3niv`;

-- Volcando estructura para tabla tiendadecamisetasc3niv.categorias
CREATE TABLE IF NOT EXISTS `categorias` (
  `id` int NOT NULL AUTO_INCREMENT,
  `nombre` varchar(50) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

-- Volcando datos para la tabla tiendadecamisetasc3niv.categorias: ~4
rows (aproximadamente)
INSERT INTO `categorias` (`id`, `nombre`) VALUES
(1, 'Unicolor'),
(2, 'Deportiva'),
(3, 'Polo'),
(4, 'Clasica');

-- Volcando estructura para tabla tiendadecamisetasc3niv.productos
CREATE TABLE IF NOT EXISTS `productos` (
  `id` int NOT NULL AUTO_INCREMENT,
  `nombre` varchar(50) DEFAULT NULL,
  `categoria_id` int DEFAULT NULL,
  `descripcion` varchar(150) DEFAULT NULL,
  `precio` double DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

-- Volcando datos para la tabla tiendadecamisetasc3niv.productos: ~5 rows
(aproximadamente)
INSERT INTO `productos` (`id`, `nombre`, `categoria_id`, `descripcion`,
`precio`) VALUES
(1, 'Camiseta Roja', 1, 'Camiseta de algodón comoda', 37000),
(2, 'Camiseta Amarilla', 1, 'Camiseta de algodón fresca', 40000),
(3, 'Camiseta Selección Colombia', 2, 'Selección Colombia 2021',
220000),
(4, 'Camiseta Polo Verde', 3, 'Camiseta con cuello y sesgo',
100000),
(5, 'Camiseta Azul', 4, 'Camiseta cuello en V, comoda', 32000);

-- Volcando estructura para tabla tiendadecamisetasc3niv.usuarios
```



```
CREATE TABLE IF NOT EXISTS `usuarios` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `nombre` varchar(50) NOT NULL DEFAULT '0',  
  `correo` varchar(50) NOT NULL DEFAULT '0',  
  `contrasena` varchar(50) NOT NULL DEFAULT '0',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;  
  
-- Volcando datos para la tabla tiendadecamisetasc3niv.usuarios: ~1 rows  
(aproximadamente)  
INSERT INTO `usuarios` (`id`, `nombre`, `correo`, `contrasena`) VALUES  
(1, 'Admin', 'admin@correo.com', '12345');
```

Evidencia de la API

Como evidencia del listado de **productos**, **categorías** y **usuarios**, en un navegador. Usando el verbo GET

✓ Productos:

```
[{"id":1,"nombre":"Camiseta Roja","categoriaId":1,"descripcion":"Camiseta de algodón  
comoda","precio":37000.0}, {"id":2,"nombre":"Camiseta Amarilla","categoriaId":1,"descripcion":"Camiseta de  
algodon fresca","precio":40000.0}, {"id":3,"nombre":"Camiseta Selecccion  
Colombia","categoriaId":2,"descripcion":"Selecccion Colombia 2021","precio":22000.0},  
 {"id":4,"nombre":"Camiseta Polo Verde","categoriaId":3,"descripcion":"Camiseta con cuello y  
sesgo","precio":10000.0}, {"id":5,"nombre":"Camiseta Azul","categoriaId":4,"descripcion":"Camiseta cuello  
en V, comoda","precio":32000.0}]
```

✓ Categorías:

```
[{"id":1,"nombre":"Unicolor"}, {"id":2,"nombre":"Deportiva"}, {"id":3,"nombre":"Polo"},  
 {"id":4,"nombre":"Clasica"}]
```

✓ Usuarios:

```
[{"id":1,"nombre":"Admin","correo":"admin@correo.com","contrasena":"12345"}]
```



Esquema del APIs (Código Java)

Como evidencia del listado de **productos**, **categorias** y **usuarios**, en un navegador.

```

// CategoriasJpaController.java
22 @RestController
23 @RequestMapping("/categorias")
24 public class CategoriasJpaController implements Serializable {
25
26     public CategoriasJpaController(EntityManagerFactory emf) {
27         this.emf = emf;
28     }
29 }

// ProductosJpaController.java
22 @RestController
23 @RequestMapping("/productos")
24 public class ProductosJpaController implements Serializable {
25
26     public ProductosJpaController(EntityManagerFactory emf) {
27         this.emf = emf;
28     }
29 }

// UsuariosJpaController.java
22 @RestController
23 @RequestMapping("/usuarios")
24 public class UsuariosJpaController implements Serializable {
25
26     public UsuariosJpaController(EntityManagerFactory emf) {
27         this.emf = emf;
28     }
29 }

```

Notifications Output - Run (TiendaDeCamisetas)

```

2022-11-29 02:37:50.502 INFO 10972 --- [ restartedMain] c.e.demo.TiendaDeCamisetasApplication : Started TiendaDeCamisetasApplication
2022-11-29 02:38:04.456 INFO 10972 --- [nio-9011-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2022-11-29 02:38:04.457 INFO 10972 --- [nio-9011-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2022-11-29 02:38:04.459 INFO 10972 --- [nio-9011-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms

```

```

// CategoriasJpaController.java
94 @GetMapping()
95 public List<Categorias> findCategoriasEntities() {
96     return findCategoriasEntities(true, -1, -1);
97 }
98 public List<Categorias> findCategoriasEntities(int maxResults, int firstResult) {
99 }

// ProductosJpaController.java
93 @GetMapping()
94 public List<Productos> findProductosEntities() {
95     return findProductosEntities(true, -1, -1);
96 }
97 public List<Productos> findProductosEntities(int maxResults, int firstResult) {
98 }

// UsuariosJpaController.java
93 @GetMapping()
94 public List<Usuarios> findUsuariosEntities() {
95     return findUsuariosEntities(true, -1, -1);
96 }
97 public List<Usuarios> findUsuariosEntities(int maxResults, int firstResult) {
98 }

```

Notifications Output - Run (TiendaDeCamisetas)

```

2022-11-29 02:37:50.502 INFO 10972 --- [ restartedMain] c.e.demo.TiendaDeCamisetasApplication : Started TiendaDeCamisetasApplication
2022-11-29 02:38:04.456 INFO 10972 --- [nio-9011-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2022-11-29 02:38:04.457 INFO 10972 --- [nio-9011-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2022-11-29 02:38:04.459 INFO 10972 --- [nio-9011-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms

```

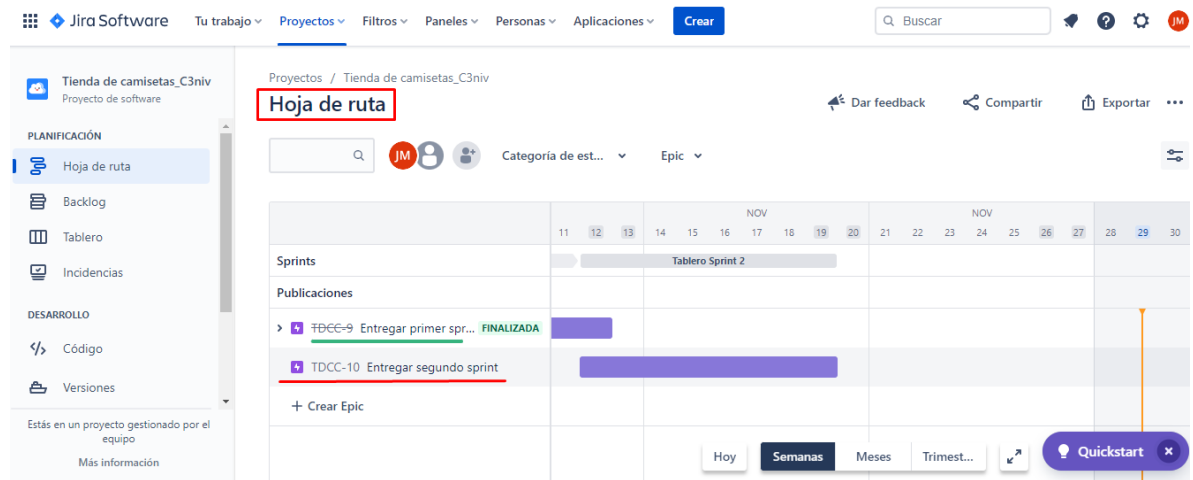


Evidencia JIRA (Seguimiento del proyecto)

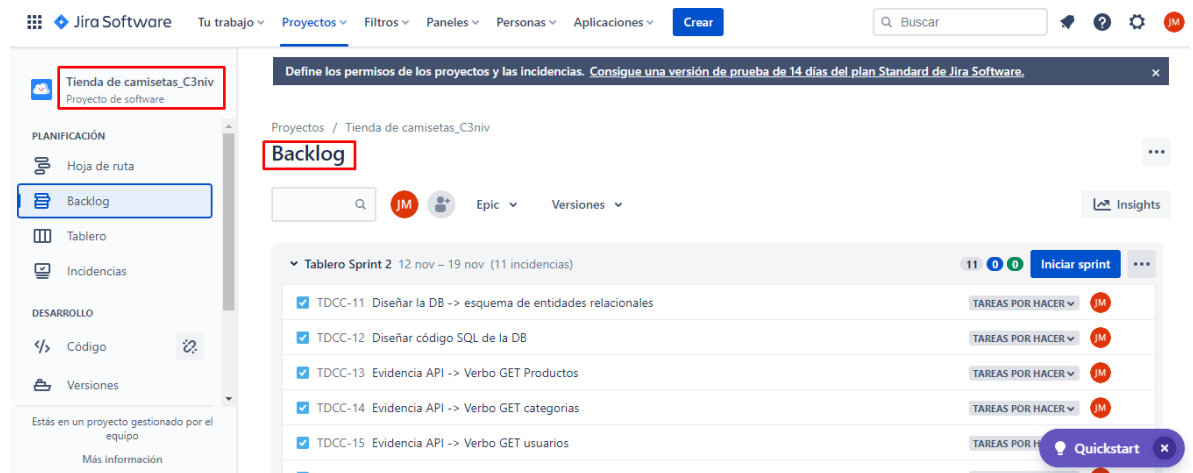
Como evidencia del seguimiento del proyecto con la metodología ágil SCRUM, utilizando el software JIRA, se debe presentar capturas de pantalla donde se visualice la ejecución de los Sprints con las historias de usuario relacionadas con el repositorio de código y el diseño de la base de datos.

Crear segundo sprint 2, con mínimo 6 tareas.

✓ Hoja de ruta



✓ Backlog





✓ Tablero

Jira Software Tu trabajo Proyectos Filtros Paneles Personas Aplicaciones Crear

Buscar

Tienda de camisetas_C3niv Proyecto de software

Tablero Sprint 2

Entregar sprint 2

Restantes: 0 días Terminar sprint

AGROUPAR POR Nada Insights

PLANIFICACIÓN

- Hoja de ruta
- Backlog
- Tablero
- Incidentes

DESARROLLO

- Código
- Versiones
- Páginas de proy...

Estás en un proyecto gestionado por el equipo Más información

POR HACER 3 INCIDENCIAS

- Evidencia JIRA -> Seguimiento del proyecto -> Lanzar Sprint 2 ENTREGAR SEGUNDO SPRINT
- TDCC-17
- Evidencia JIRA -> Seguimiento del proyecto -> Hoja de ruta ENTREGAR SEGUNDO SPRINT
- TDCC-18
- Evidencia JIRA -> Seguimiento

EN CURSO 4 INCIDENCIAS

- Esquema de APIs -> Código Java ENTREGAR SEGUNDO SPRINT
- TDCC-16
- Evidencia API -> Verbo GET categorías ENTREGAR SEGUNDO SPRINT
- TDCC-14
- Evidencia API -> Verbo GET Productos

LISTO 4 INCIDENCIAS

- Diseñar la DB -> esquema de entidades relacionales ENTREGAR SEGUNDO SPRINT
- TDCC-11
- Evidencia API -> Verbo GET usuarios ENTREGAR SEGUNDO SPRINT
- TDCC-15
- Diseñar código SQL de la DB

Quickstart

Jira Software Tu trabajo Proyectos Filtros Paneles Personas Aplicaciones Crear

Buscar

Tienda de camisetas_C3niv Proyecto de software

Tablero Sprint 2

Entregar sprint 2

Restantes: 0 días Terminar sprint

AGROUPAR POR Nada Insights

PLANIFICACIÓN

- Hoja de ruta
- Backlog
- Tablero
- Incidentes

DESARROLLO

- Código
- Versiones
- Páginas de proy...

Estás en un proyecto gestionado por el equipo Más información

Completar Tablero Sprint 2

Este sprint contiene 11 incidencias en estado completado.

Ya está todo listo. ¡Bien hecho!

Terminar sprint Cancelar

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

Alexander202212 / TiendaCamisetasC3niv Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main

Commits on Nov 29, 2022

DB

Se agrega esquema de entidades relacionales y el código SQL de la DB

Alexander202212 committed 1 minute ago

Commits on Nov 21, 2022

Create Sprint_1_C3_Niv.docx

Sprint 1 -> formato completo

Alexander202212 committed 8 days ago

Initial commit

Alexander202212 committed 8 days ago



Evidencias de las Reuniones de Equipo

Como evidencia de las reuniones que efectúa el equipo del proyecto, presentar capturas de pantalla de las reuniones efectuadas y si lo consideran pertinente algunas actas de las reuniones.

INDIVIDUAL