Universidad de Valparaíso
CHILE

PROYECTO INFORMATICO

APLICACIÓN WEB: VENTAS EN LÍNEA DE TICKETS PARA EVENTOS EN CHILE.

Integrantes:

Leonardo Arancibia

Manuel Carreño

Nicolás Veas

Académicos:

Manuel García

Eliana Providel

Asignatura:

ICI 324

Base de datos y programación web

Octubre, 2020

INTRODUCCIÓN

Presentación

En el presente informe describe el proyecto de la asignatura Base de datos y programación web de la carrera de Ingeniería civil informática. El proyecto consiste en el desarrollo de un sistema con similares características a un carro de compras, donde el tema asignado es “Venta de tickets para eventos” con el objetivo de comprar tickets de manera online para los futuros eventos, además de desarrollar un back-end donde permita desarrollar los eventos y un front-end amigable para que los usuarios interactúen con el sistema.

Descripción del proyecto

La problemática del proyecto es que las personas para adquirir tickets de un evento tienen que realizar de forma presencial, lo que ocupa tiempo, por lo tanto para priorizar otras actividades del mundo cotidiano una manera más eficiente y fácil es adquirir los tickets de manera online, donde optimizará el tiempo y no solo tendrán la facilidad para comprar tickets sino que podrá quedar registrada la compra y se podrá ahorrar comprobar el ticket de un evento mediante un papel y también podrán ver otros eventos de interés.

El equipo informático realizó la siguiente propuesta de solución la cual consiste en desarrollar una aplicación web con diversas tecnologías, donde en esta se puedan visualizar eventos de diversa índole tales como conciertos, deporte, culturales, familiares y especiales, que forman las categorías de eventos, estos contienen características como número identificación, nombre, fecha, lugar, hora y cantidad

de tickets disponibles, asimismo los tickets derivan de cada evento por lo cual tienen cualidades como número de identificación y precio.

Se abordará sólo las actividades realizadas en Chile, donde se ofrece un servicio para todo tipo de personas en este caso compradores que es un tipo de usuario que desea adquirir tickets de los eventos disponibles en el sistema mediante un carro de compras y un administrador que operará con los eventos, usuarios y ventas.

REQUERIMIENTOS

REQUERIMIENTOS FUNCIONALES

- Los usuarios deben ingresar al sistema mediante un login.
- El sistema debe permitir cerrar sesión a los usuarios.
- Debe haber un registro de cada transacción del sistema.
- El sistema debe estar dividido en módulos para cada tipo de usuario.
- El sistema debe permitir al comprador visualizar eventos y ver información específica de cada uno.
- El sistema debe permitir al comprador adquirir tickets disponibles de los eventos.
- El sistema debe contener el carro de compras para cada usuario para que pueda visualizar la compra en detalle antes de finalizar.
- El sistema debe permitir solamente al usuario de tipo administrador operar (crear, actualizar, eliminar, leer) con eventos, usuarios y ventas.
- El sistema debe permitir al usuario de tipo comprador pueda adquirir ticket en el carro de compras de distintos eventos.
- Cada usuario de tipo comprador podrá ver los tickets obtenidos luego de finalizar cada compra.

- El comprador puede agregar, eliminar, consultar los tickets dentro del carro de compras.
- El sistema debe permitir que los usuarios de tipo comprador y administrador puedan visualizar a los eventos separados en categoría y mes.
- El sistema debe permitir a los usuarios visualizar la cantidad total de tickets disponible de cada evento y su respectivo precio.
- El administrador podrá visualizar un gráfico en el cual le indicara la categoría de eventos donde se venden más tickets.

REQUERIMIENTOS NO FUNCIONALES

- Los datos deben estar seguros para evitar un error de filtración o mal uso de estos.
- Bajos tiempos de respuesta entre el servidor y aplicación web.
- La plataforma debe soportar múltiples compras simultáneas de los usuarios.
- Aplicación web compatible con distintos navegadores de internet.
- El sistema debe considerar a los usuarios con poca experiencia un fácil uso.
- El sistema debe ser escalable para ser mantenible en el tiempo, independiente de los lenguajes de programación.

DIAGRAMA DE CASOS DE USO DE SOLUCIÓN GENERAL

Figura 1: Diagrama de casos de uso según la perspectiva del usuario de tipo administrador y comprador.

DIAGRAMA DE SECUENCIA DE LA SOLUCIÓN

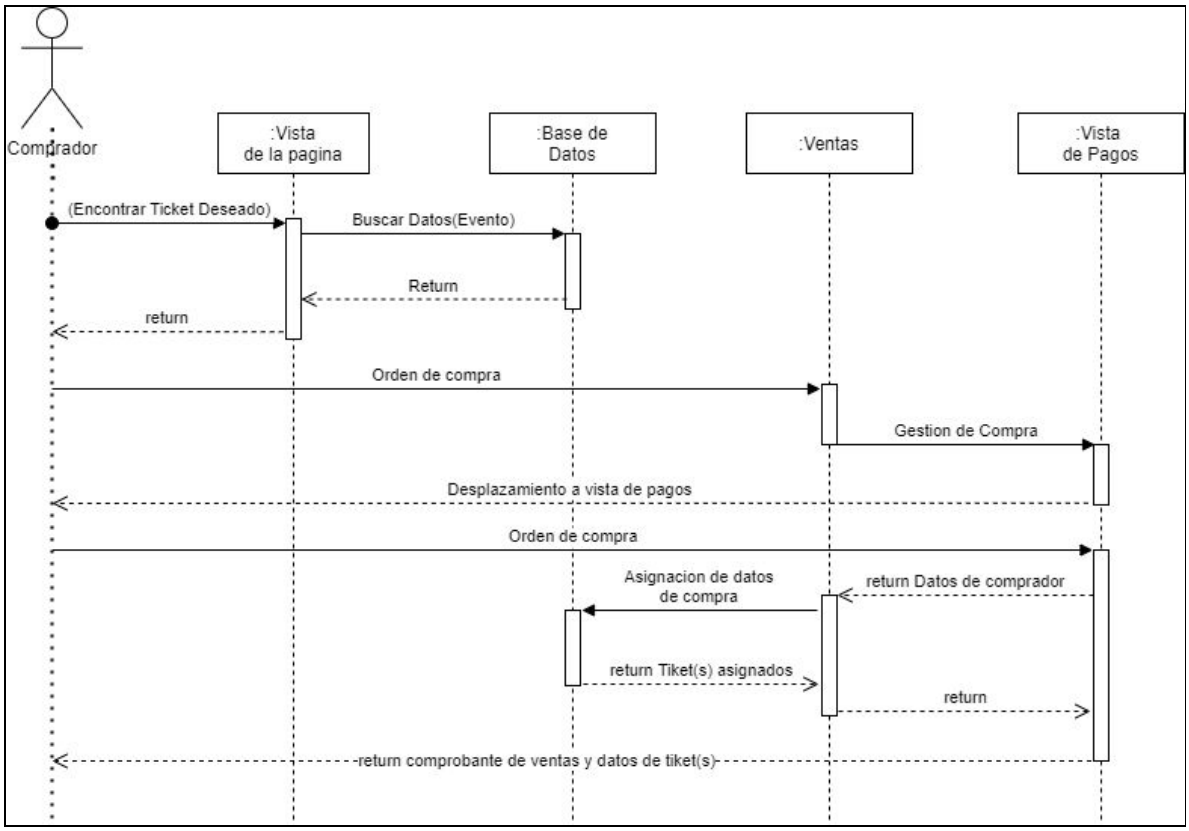

Figura 2: Diagrama de secuencia con la perspectiva del usuario principal que es el comprador.

MOCKUPS DE PANTALLAS DE SOLUCIÓN

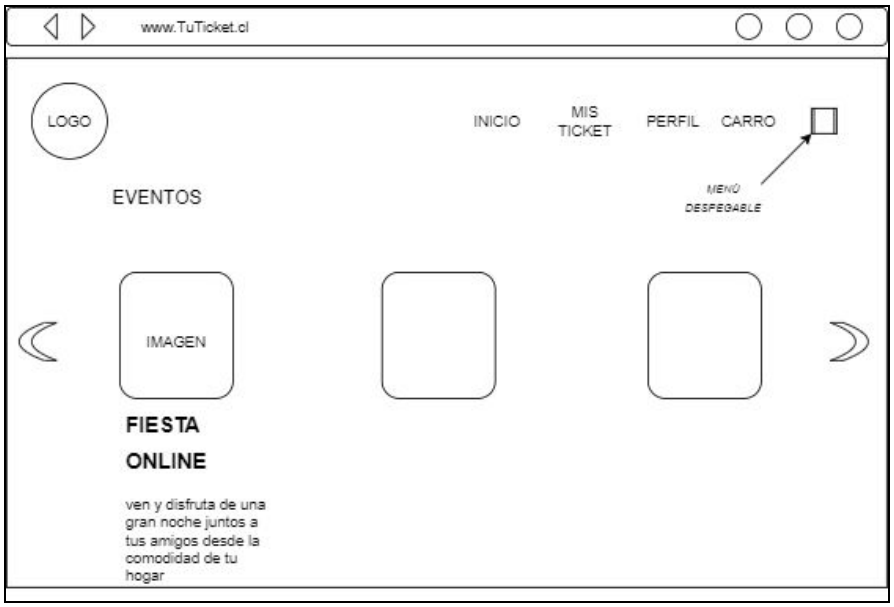

Figura 3: Mockup de la pantalla principal de la aplicación web.

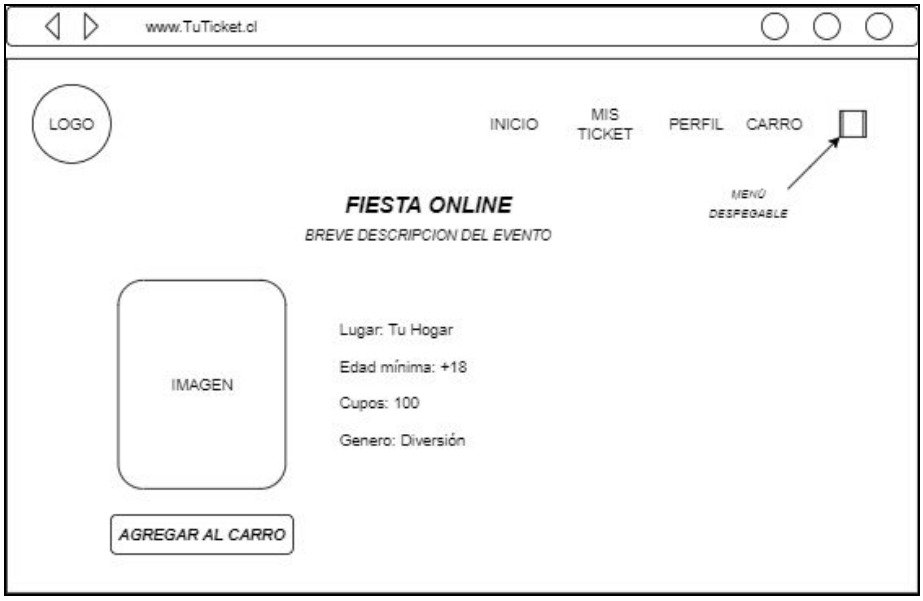

Figura 4: Mockup de la pantalla cuando seleccionas un evento.

MODELO DE DATOS

MODELO E-R

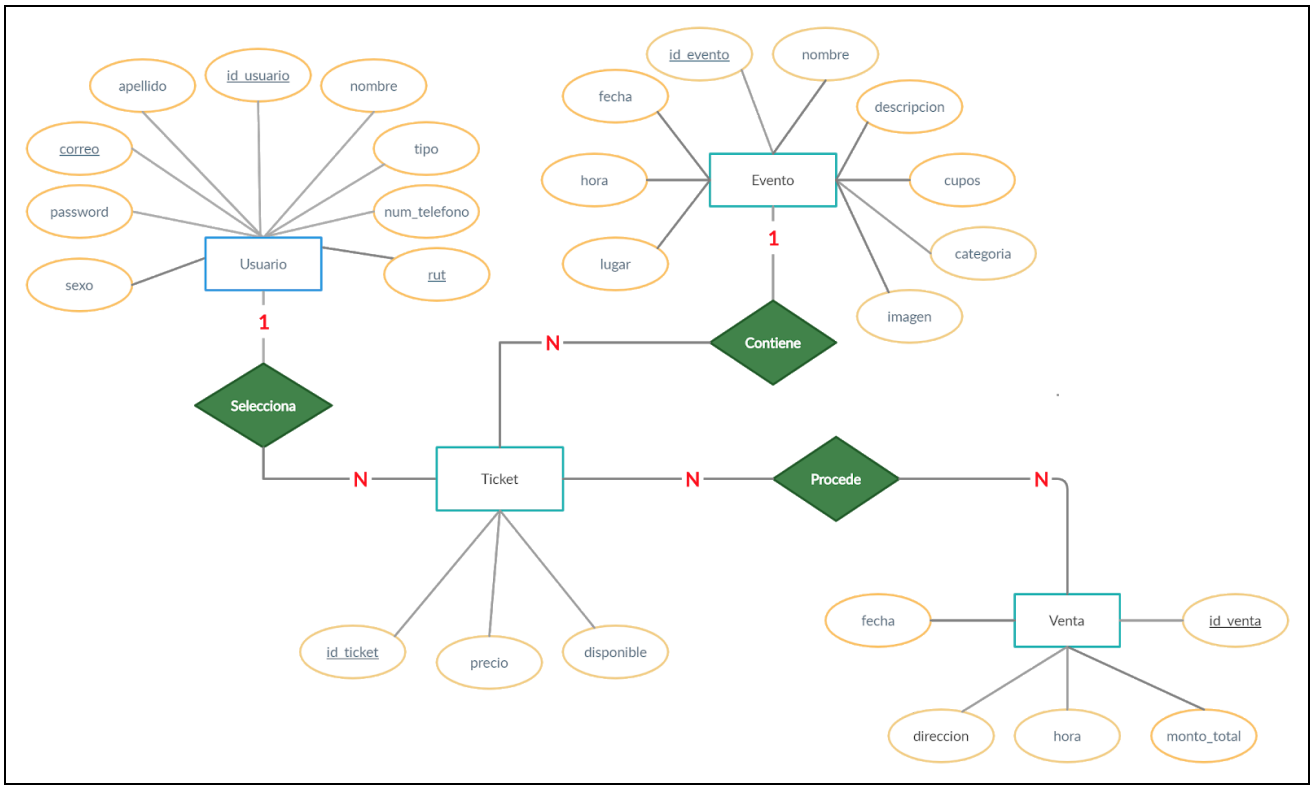

Figura 5: Modelo entidad relación.

MODELO RELACIONAL

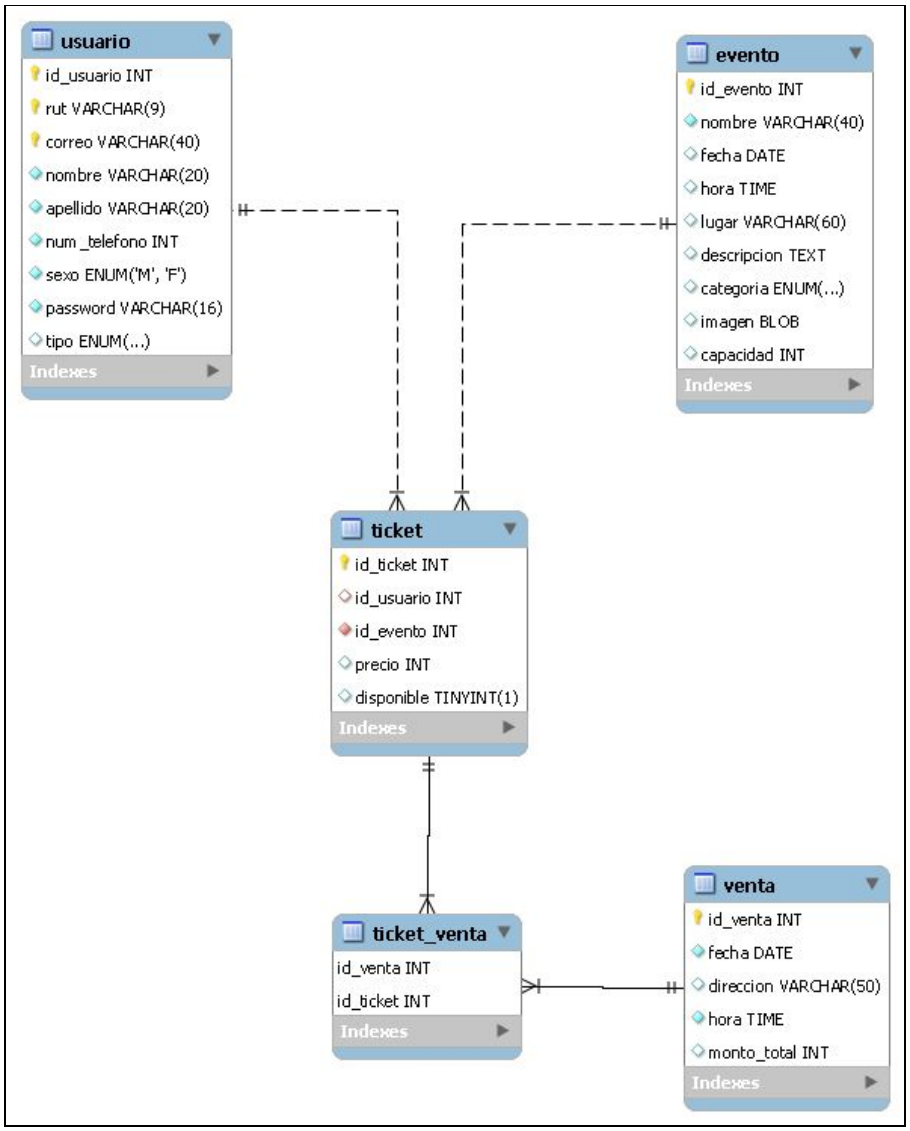

Figura 6: Modelo relacional.

CONSULTAS CRUD

CONSULTAS EN LENGUAJE NATURAL

1. ADMINISTRADOR – COMPRADOR se les muestra los eventos disponibles en la página inicial (**SELECT**).
2. ADMINISTRADOR - COMPRADOR consultan eventos según categoría en un cuadro de opciones CATEGORÍAS (conciertos, culturales, especiales, familiares, deportes) **(SELECT)**.
3. ADMINISTRADOR consulta las ventas realizadas por los compradores (**SELECT**).
4. COMPRADOR consulta sus ventas y tickets realizados por el mismo usuario (**SELECT JOIN**).
5. ADMINISTRADOR añade eventos (**INSERT**).
6. ADMINISTRADOR consulta ticket de cada evento (**SELECT**).
7. ADMINISTRADOR elimina eventos (**DELETE**).
8. ADMINISTRADOR modifica registro deseado de la tabla venta en un cuadro de texto (**UPDATE**).
9. ADMINISTRADOR modifica registro deseado de la tabla evento en un cuadro de texto (**UPDATE**).
10. ADMINISTRADOR modifica registro deseado de la tabla usuario en un cuadro de texto (**UPDATE**).
11. COMPRADOR ejecuta compra y se realizan los siguientes procesos: insertar venta (**INSERT**), insertar venta-tickets (**INSERT**) y actualizar los datos del ticket id_usuario-disponible (**UPDATE**).
12. ADMINISTRADOR añade tickets a eventos (**INSERT**).

13. ADMINISTRADOR agrega campo a tabla ventas en un cuadro de texto (**ALTER**).
14. ADMINISTRADOR agrega campo a tabla ticket en un cuadro de texto (**ALTER**).
15. ADMINISTRADOR agrega campo a tabla usuario en un cuadro de texto (**ALTER**).
16. ADMINISTRADOR consulta los tickets comprados en una fecha específica (**SELECT JOIN)**.
17. COMPRADOR consulta tickets disponible de un evento específico (**SELECT**).
18. ADMINISTRADOR elimina ventas (**DELETE**).
19. ADMINISTRADOR consulta los usuarios de tipo comprador (**SELECT**).

CONSULTAS SELECT EN ÁLGEBRA RELACIONAL

1. **ADMINISTRADOR – COMPRADOR se les muestra los eventos disponibles en la página inicial (SELECT).**

π id_evento, nombre, fecha, hora, lugar, descripcion, categoria, imagen, capacidad (Evento)

2. **ADMINISTRADOR - COMPRADOR consultan eventos según categoría en un cuadro de opciones CATEGORÍAS (conciertos, culturales, especiales, familiares, deportes) (SELECT).**

π id_evento, nombre, fecha, hora, lugar, descripcion, categoria, imagen, capacidad (σ categoria='conciertos' (Evento))

π id_evento, nombre, fecha, hora, lugar, descripcion, categoria, imagen, capacidad (σ categoria='familiares' (Evento))

π id_evento, nombre, fecha, hora, lugar, descripcion, categoria, imagen, capacidad (σ categoria='deportes' (Evento))

π id_evento, nombre, fecha, hora, lugar, descripcion, categoria, imagen, capacidad (σ categoria='especiales' (Evento))

π id_evento, nombre, fecha, hora, lugar, descripcion, categoria, imagen, capacidad (σ categoria='culturales' (Evento))

3. **ADMINISTRADOR consulta las ventas realizadas por los compradores (SELECT).**

π id_venta, fecha, direccion, hora, monto_total (Evento)

4. **COMPRADOR consulta sus ventas y tickets realizados por el mismo usuario (SELECT JOIN)**.

π ticket.id_usuario, ticket.id_ticket, ticket_venta.id_venta (σ ticket.id_usuario=1 ((ticket) ⋈ ticket.id_ticket = ticket_venta.id_ticktet (ticket_venta)))

5. **ADMINISTRADOR consulta ticket de cada evento (SELECT).**

π id_ticker, id_usuario, id_evento, precio, disponible (σ id_evento=1 (Ticket))

6. **ADMINISTRADOR consulta los tickets comprados en una fecha específica (SELECT JOIN).**

π ticket_venta.id_ticket, venta.fecha (σ venta.fecha="2020-08-16" ((ticket_venta) ⋈ ticket_venta.id_venta = vent.id_venta (venta)))

7. **COMPRADOR consulta tickets disponible de un evento específico (SELECT).**

π id_ticker, id_usuario, id_evento, precio, disponible (σ id_evento=1 ∧ disponible=1 (Ticket))

8. **ADMINISTRADOR consulta los usuarios de tipo comprador (SELECT).**

π id_usuario, rut, correo, nombre, apellido, num_telefono, sexo, password, tipo (σ tipo="comprador"(usuario))