



UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL CÓRDOBA

Diseño de Sistemas de Información

Proyecto práctico de aplicación integrador

Sistema de respuesta de voz interactivo IVR

Diseño

Integrantes

91122 - Schmira Agustin - agustinschmira@gmail.com
89390 - Wendler Juan José - juanjoowendler@gmail.com
90313 - Picatto Renzo Agustín - picattoagustin@gmail.com
90606 - Lambertucci Juan Pablo - JP_Lambertucci@outlook.com
91185 - Cerutti Joaquin - joaquicerutti@gmail.com
86203 - Moyano Santiago Ismael - santiagomoyano647@gmail.com

Docentes

Ing. Cecilia Massano
Ing. Valeria Abdala
Ing. Mickaela Crespo

Índice

Índice	2
Rediseño de la realización de caso de uso análisis	3
<u>Patrón Implementado: State</u>	3
<u>Vista de la estructura de la realización de caso de uso de diseño</u>	3
<u>Vista dinámica de la realización de caso de uso de diseño</u>	3
Diseño de experiencia de usuario	4
Moodboard	4
Mapa de Empatía	4
User Journey	5
Diseño de persistencia	6
Tecnologías utilizadas	6
DER (Diagrama Entidad-Relación)	6

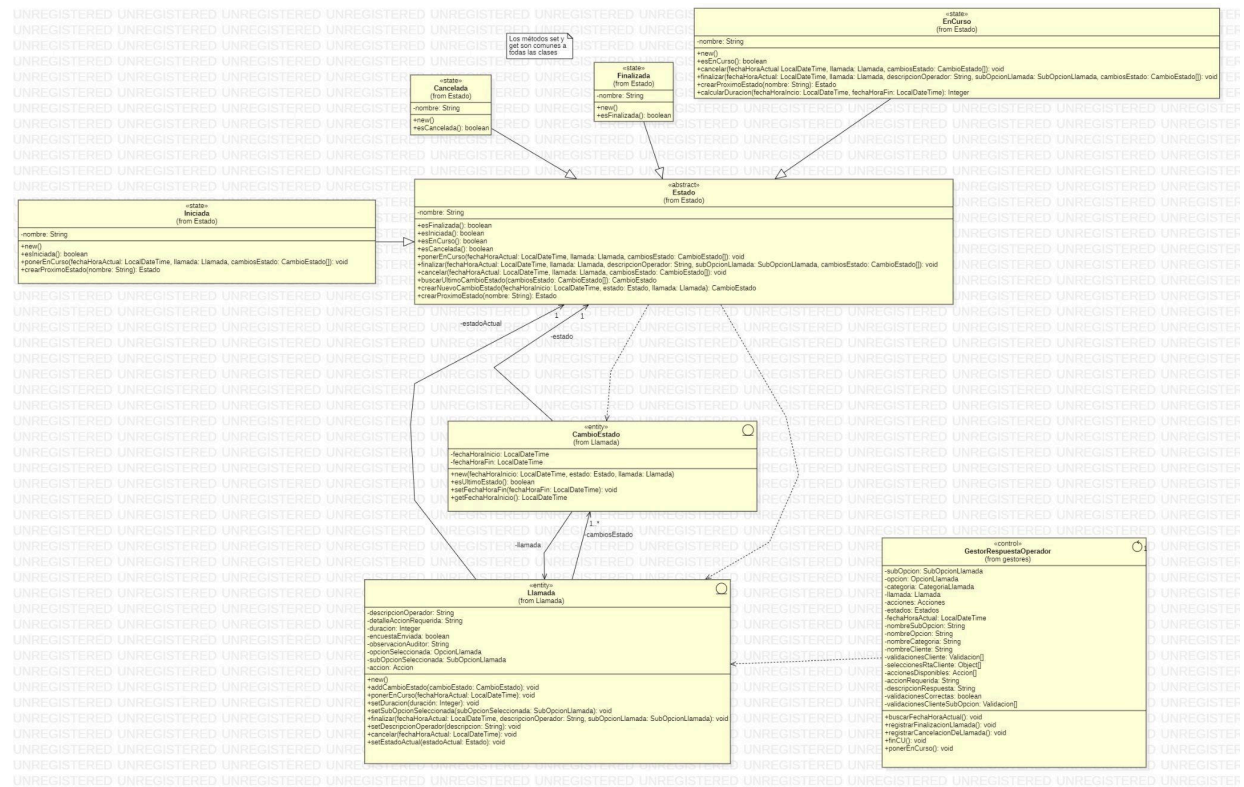
Rediseño de la realización de caso de uso análisis

Patrón Implementado: **State**

Se propuso la implementación del patrón State para el Caso de Uso 17 “Registrar respuesta de operador” para resolver el problema de que la clase Llamada posee comportamientos variables dependiendo del estado en el que se encuentre en cada momento y con ello más responsabilidades.

La implementación del patrón State nos permite encapsular cada comportamiento asociado a un estado en particular en una clase concreta para dicho estado; así, para cada comportamiento diferente que pueda tener la clase Llamada en base a su estado, habrá una clase concreta que tenga la responsabilidad de ejecutar ese comportamiento. Todas estas clases concretas heredarán de una clase abstracta Estado, en la cual se definirán métodos que luego serán heredados por las clases hijas (estados concretos). Estos métodos de la clase abstracta podrán o no tener una implementación por defecto, y las clases concretas podrán utilizar dichos métodos con la implementación por defecto, sobrescribir los métodos heredados e incluso definir sus propios métodos.

Ahora la clase contexto, es decir, Llamada, tendrá mayor flexibilidad en su comportamiento, delegando a cada estado concreto la responsabilidad de cambiar de estado y de ejecutar el comportamiento asociado a dicho estado.



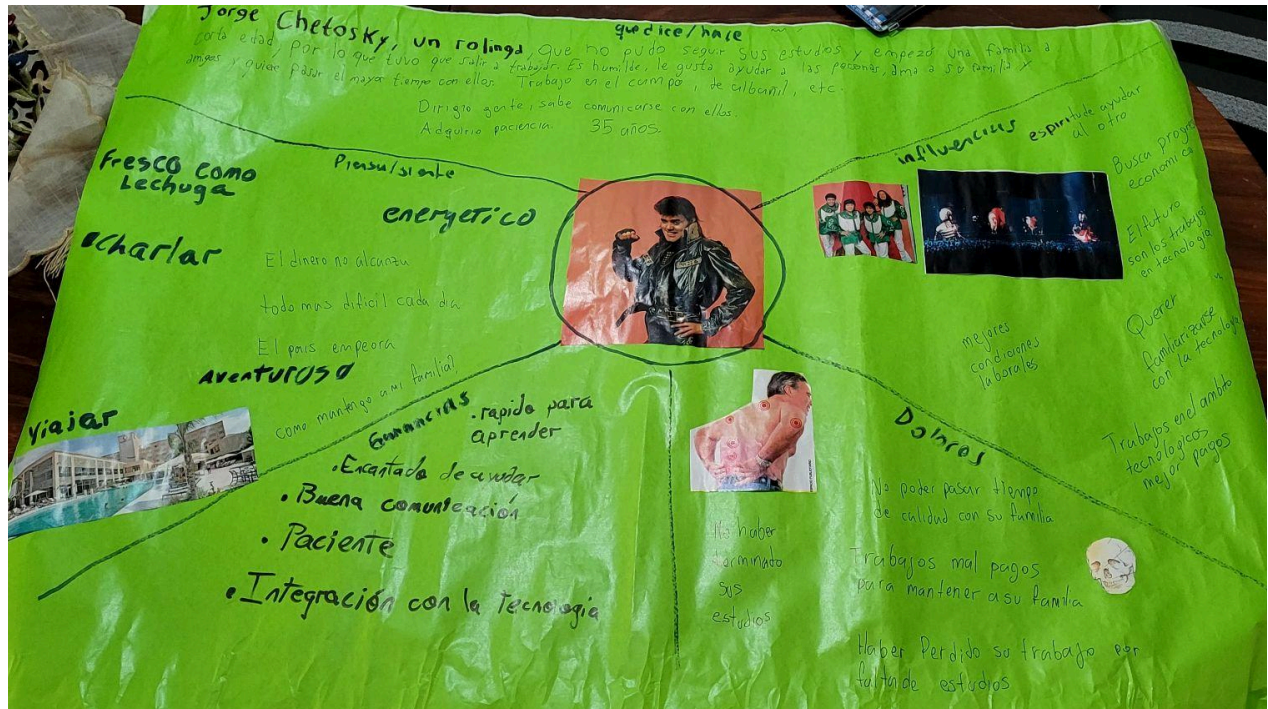
[Hacer click acá para ver la imagen con mas calidad](#)

[illegible]

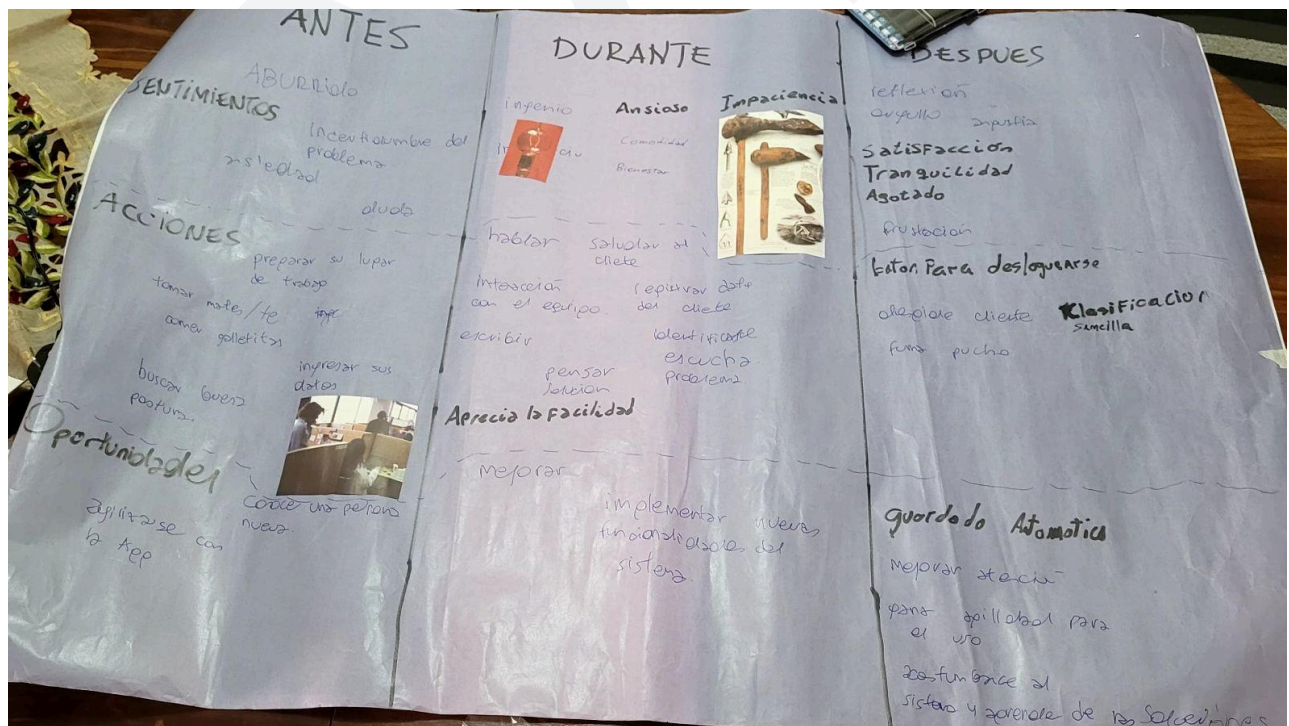
[illegible]

6

Mapa de Empatía



User Journey



Diseño de persistencia

Tecnologías utilizadas

- Tecnología implementada: Web
- Backend:
 - Lenguaje de programación: Java
 - Framework de persistencia: Spring Data JPA
 - Base de datos: MySQL
 - Framework: Spring Boot
 - IDE / Editor de texto: IntelliJ IDEA Ultimate
- FrontEnd:
 - Lenguaje de programación: JavaScript - React - HTML - CSS
 - Framework: NodeJs
 - IDE / Editor de texto: Visual Studio Code
 - Otros: Bootstrap (estilo y funcionalidad)



IVR