

EARFIT: Aplicación Para Entrenamiento Auditivo Musical Basada en Next.js y TypeScript

Trabajo fin de grado

Grado en Ingeniería Informática

Autor: *Alberto Gómez Cano*

Tutor: *Manuel Rubio Sánchez*



Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática

Contenido

- 1 **Introducción**
- 2 **Objetivos**
- 3 **Metodologías**
 - Design Thinking
 - Lean Startup
 - Scrum
 - DevOps
- 4 **Desarrollo del Software**
 - Tecnologías
 - Arquitectura
 - Estructura de Archivos
 - Detalles de Implementación
 - Progressive Web App
 - Software QA
- 5 **Conclusiones**
- 6 **Demo**



Entrenamiento Auditivo



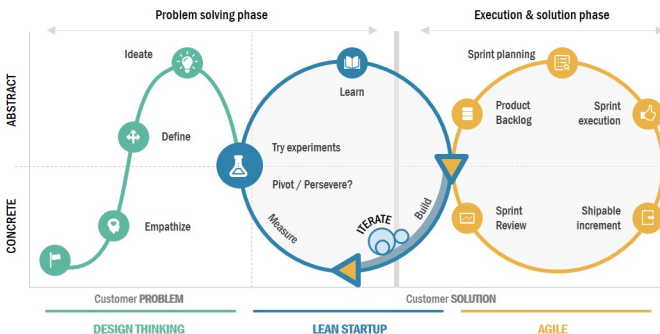
Objetivos del TFG

- Objetivo principal
 - Desarrollar una aplicación que permita a músicos desarrollar su oído musical mediante el entrenamiento auditivo.
- Subobjetivos
 - Desarrollar una interfaz interactiva.
 - Implementar diferentes tipos de ejercicio personalizables de entrenamiento auditivo.
 - Incluir varios instrumentos para practicar con diferentes sonidos.



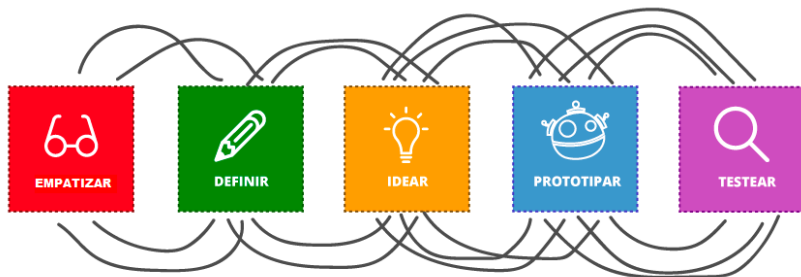
Metodologías

Design Thinking + Lean Startup + Agile Diagram



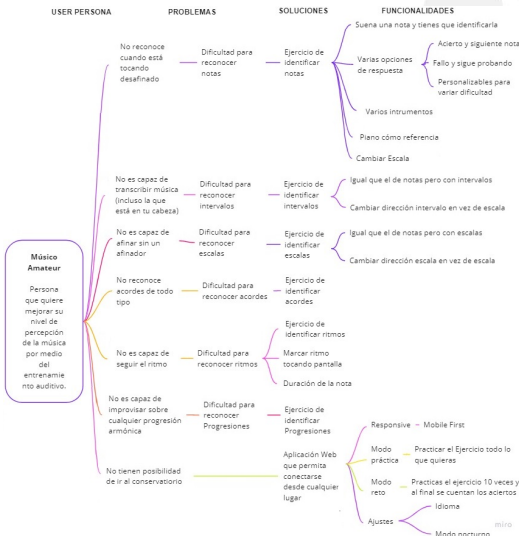
Design Thinking

Generar Ideas Innovadoras



Mindmap

Etapa: Idear



MoSCoW

Eta: Protopitar



Protopipo

Etapa: Protopipar

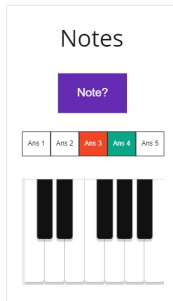


Notes

Intervals

Scales

About



Options

Scale ●

Major ▼

Answers ●

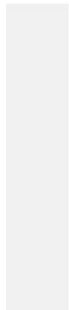
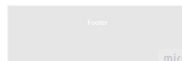
Ans 1 ☒ Ans 2 ☒ Ans 3 ☒

Ans 4 ☒ Ans 5 ☒ Ans 6 ☒

Ans 7 ☐

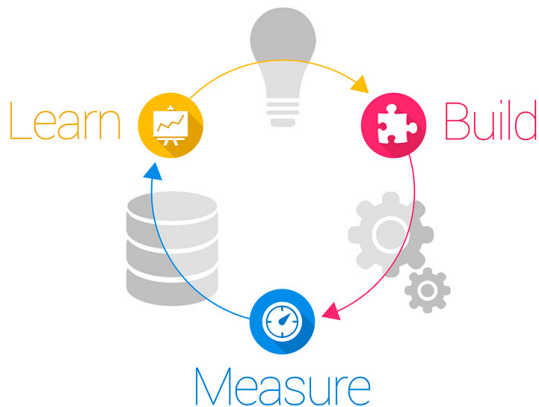
Instrument ●

Piano ☒ Guitar ☐



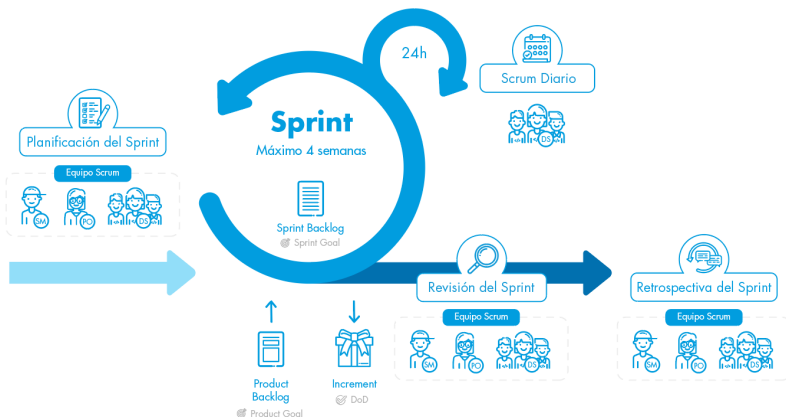
Lean Startup

Agilizar la Puesta en Marcha

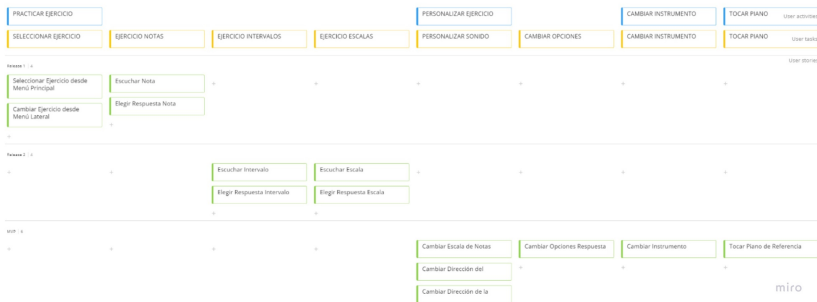


Scrum

Proceso de Gestión



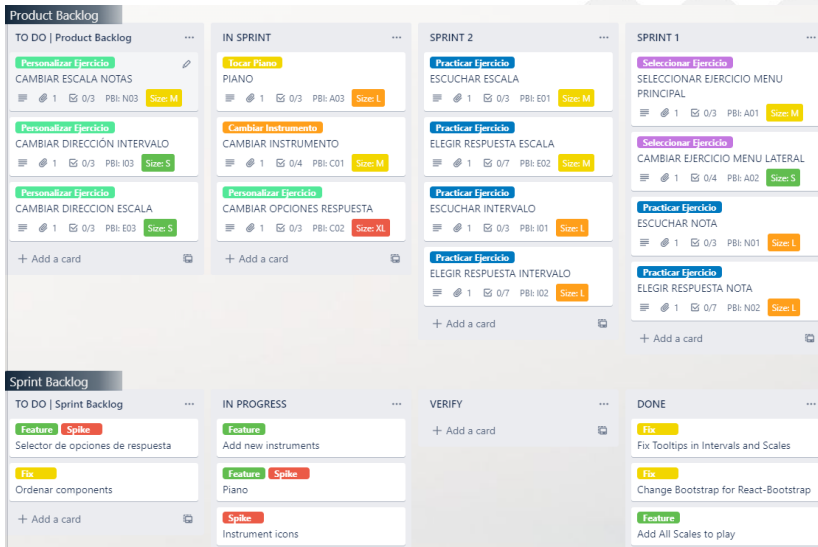
User Story Map



miro



Scrum Board



User Stories



CAMBIAR INSTRUMENTO

in list [IN SPRINT](#)

Labels

Cambiar Instrumento



Description

Edit

Como: Músico Amateur

Quiero: Cambiar entre varios instrumentos

Para: Practicar los ejercicios con sus sonidos



Custom Fields

T PBI

Size

C01

M



Attachments



Prototipo

Added yesterday at 8:00 PM - [Comment](#) - [Delete](#) - [Edit](#)

[Make cover](#)

Add an attachment



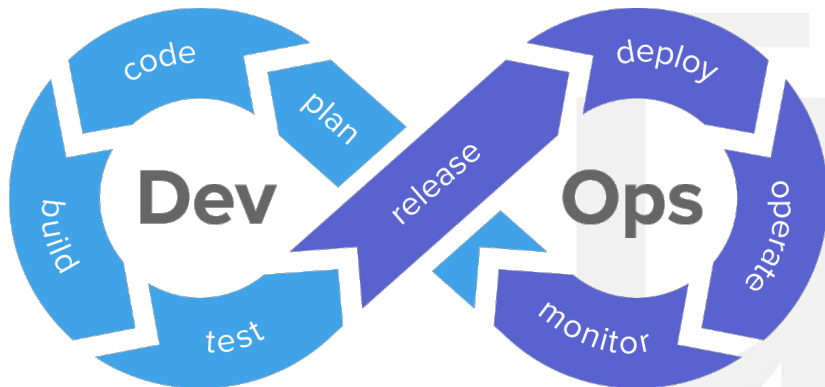
Criterios de Aceptación

Delete

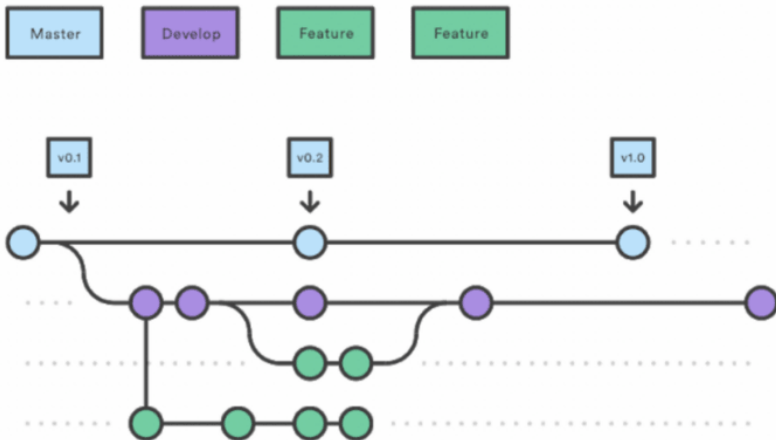


DevOps

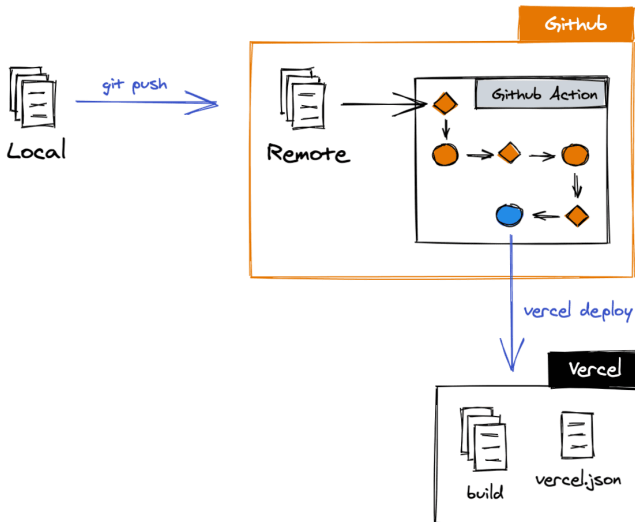
Filosofía de Desarrollo



Integración Continua (CI)



Despliegue Continuo (CD)

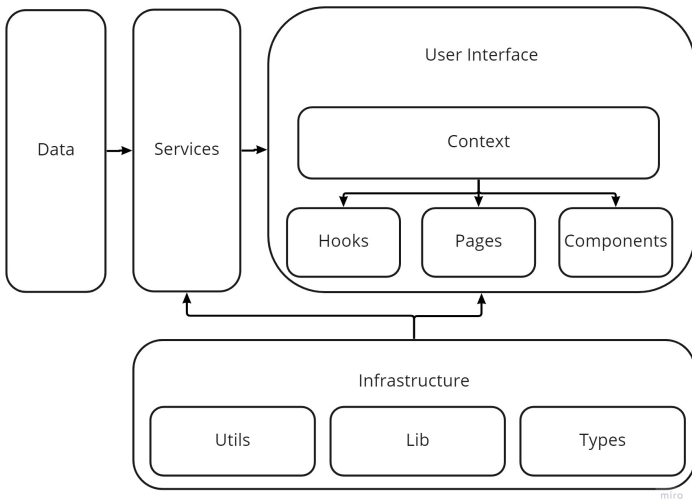


Stack Tecnológico

- **Next.js:** Enrutamiento basado en páginas, Prerendering, Code Splitting, Prefetching, Fast Refresh, SWC y WebPack...
- **React:** Componentes reutilizables, DOM Virtual, Hooks, Context...
- **Vercel:** Vercel Edge Network, Vercel Analytics, Flujo DPS...
- **Typescript:** Tipos estáticos para JavaScript.
- **Node.js:** Entorno de ejecución Javascript, librerías útiles (NPM vs. Yarn).
- **VScode:** Git integrado, depuración, resaltado de sintaxis, finalización inteligente, refactorización de código, extensiones (ESLint y Prettier)...



Arquitectura




miro





Estructura de Archivos


▼ EARFIT

>  public


▼ src


>  components


>  context


>  hooks

>  lib


>  pages


>  services


>  styles


>  types


>  utils


 .eslinttrc.js


 .gitignore


 .prettierrc.js


 next-env.d.ts

 next.config.js

 package.json

 README.md

 tsconfig.json

 yarn.lock





Tipos

```
export type Instrument = {  
  displayName: string;  
  emoji: string;  
  instrumentName: InstrumentName;  
  notePlayer: NotePlayer;  
  isLocal: boolean;  
};
```

```
type NotePlayer = {  
  play: (noteName: string, when?: number, opts?: {duration?: number; gain?: number;  
    attack?: number; decay?: number; sustain?: number; release?: number; adsr?:  
    [number, number, number, number]; loop?: boolean;}) => Player;  
  stop: (noteName?: string) => void;  
  schedule: (when?: number, events?: any[]) => Player;  
  instrumentName: InstrumentName;  
};
```



TypeScript

 Copy  Caption ...

```
export type VariantExercise = 'notes' | 'intervals' | 'scales';
```

```
export type Answer = {  
  id: string;  
  notes: string[];  
  displayName: string;  
};
```

```
export type IsSelectable = {isSelected: boolean};  
export type VariantColor = 'success' | 'secondary' | 'danger';  
export type WithColor = {color: VariantColor};
```

```
export type SelectableAnswer = Answer & IsSelectable;  
export type SelectableAnswerWithColor = SelectableAnswer & WithColor;
```



Librerías

- **Tonal.js**: Manipular elementos musicales.
- **Soundfont-player**: Cargar y reproducir archivos MIDI.js.
- **Soundfont-wrapper**: Refinar la complejidad de “soundfont-player” y simplificar su uso.
- **React-piano**: Teclado de piano interactivo (sin sonidos).
- **React-use-measure**: Para que el piano sea responsive.
- **React-bootstrap**: Librería de estilos CSS.
- **Next-pwa**: Registrar y generar un Service Worker.

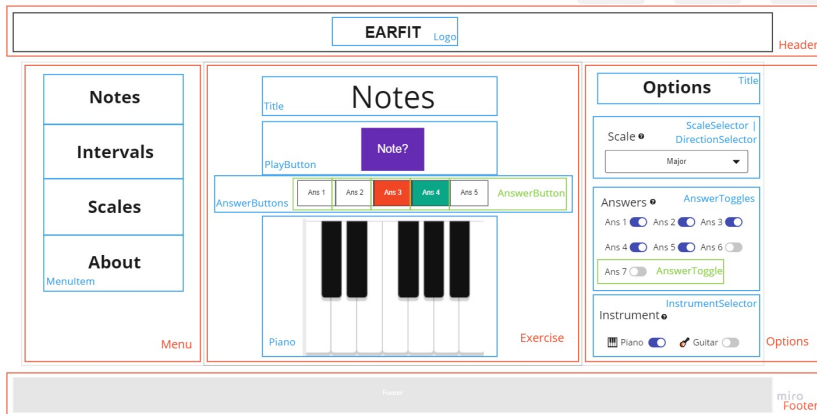


Servicios

- **instrumentService.ts**: Provee los instrumentos de la aplicación.
- **noteService.ts**: Provee las respuestas correspondientes al ejercicio de notas.
- **intervalService.ts**: Provee las respuestas correspondientes al ejercicio de intervalos.
- **scaleService.ts**: Provee las respuestas correspondientes al ejercicio de escalas.



Jerarquía de Componentes



Comportamiento de una Página (1/2)

Hooks

```
import { Exercise, Menu, Options, Pagelayout } from 'components';
import { useAnswer, useAnswerButtons, useAnswerToggles, useDirectionSelector,
useExercise, usePlayButton } from 'hooks';

export default function Intervals(): JSX.Element {
  const { direction, changeDirection } = useDirectionSelector();
  const { answers } = useExercise('intervals');
  const { answerToggles, updateIsSelected, selectAllOrThree } = useAnswerToggles(answers);
  const { answer, setNewAnswer, isCorrectAnswer } = useAnswer('intervals', answerToggles);
  const { playAnswer } = usePlayButton('intervals', answer, direction);
  const { answerButtons, handleAnswerButtonClick, streak } = useAnswerButtons(answerToggles,
    isCorrectAnswer,
    setNewAnswer,
    playAnswer
  );
};
```



Comportamiento de una Página (2/2)

Componentes

```
return (  
  <PageLayout  
    leftCol={<Menu />}  
    rightCol={  
      <Options  
        direction={direction}  
        handleDirectionChange={changeDirection}  
        answerToggles={answerToggles}  
        handleAnswerToggleAllChange={selectAllOrThree}  
        handleAnswerTogglesChange={updateIsSelected}  
      />  
    }  
  >  
    <Exercise  
      title="Intervals"  
      playButtonLabel="Interval?"  
      handlePlayButtonClick={playAnswer}  
      answerButtons={answerButtons}  
      handleAnswerButtonClick={handleAnswerButtonClick}  
      streak={streak}  
    />  
  </PageLayout>  
);
```



Comportamiento de un Componente

```
import { AnswerButton } from 'components/Exercise/AnswerButtons/AnswerButton';
import { ButtonGroup } from 'react-bootstrap';
import { SelectableAnswerWithColor } from 'types';

interface Props {
  answerButtons: SelectableAnswerWithColor[];
  handleAnswerButtonClick: (answerButton: SelectableAnswerWithColor) => void;
}

export const AnswerButtons = ({ answerButtons, handleAnswerButtonClick }: Props):
  JSX.Element => {
  return (
    <ButtonGroup className="btn-group btn-group-toggle d-flex justify-content-center"
      data-toggle="buttons">
      <div>
        {answerButtons.map((answerButton) => (
          <AnswerButton
            key={answerButton.id}
            answerButton={answerButton}
            handleAnswerButtonClick={handleAnswerButtonClick}
          />
        ))}
      </div>
    </ButtonGroup>
  );
};
```



Progressive Web App (PWA)

Una PWA es confiable, instalable y se comporta como una app nativa en ordenador, móvil y tablet.

Para que una aplicación sea PWA debe tener:

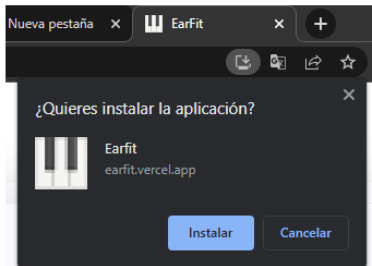
- Una conexión segura HTTPS.
- Cargue sin conexión, para ello requiere un Service Worker.
- Información como nombre, autor, icono y descripción en un documento JSON llamado Manifest.
- Un icono de al menos 144x144 px en formato PNG.



Progressive Web App (PWA)

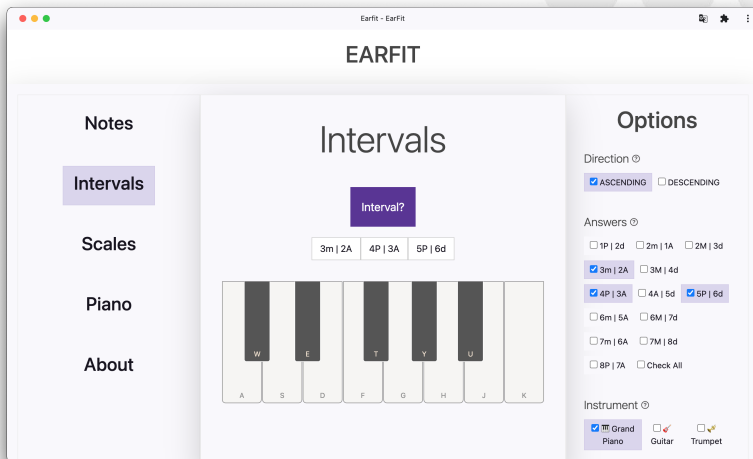
Instalar

En iOS, la opción se llama “añadir a pantalla de inicio” desde Safari y en Chrome aparece un icono en la barra de búsqueda.



Progressive Web App (PWA)

Earfit como PWA en MacOs



Google Lighthouse

Auditoría de Calidad de la Página Web



Performance



Accessibility



Best Practices



SEO



PWA

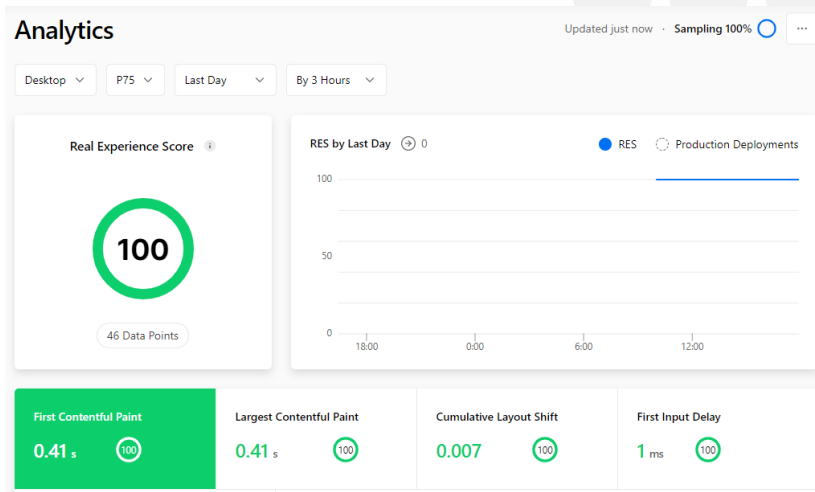
En el apartado Performance:

- Lighthouse da falsas mediciones para aplicaciones Next.js.
- Lighthouse estima las Web Vitals ejecutando una simulación.
- En este caso, usar Vercel Analytics aporta ventajas, como datos reales de los dispositivos de los usuarios.



Vercel Analytics

Experiencia de Usuario de la Página Web



Conclusiones

Objetivos Alcanzados:

- OBJ1
- OBJ2

Trabajos Futuros:

- Funcionalidades: Ejercicios de acordes, ritmos, progresiones, modo nocturno y varios idiomas.
- Añadir cobertura de código y test automáticos al flujo de Integración Continua (CI)
- Considerar Software libre vs Negocio: Plantear alguna forma de monetización, marketing, etc.



Conclusiones

Buenas Prácticas:

- Guía de Estilo de Código
- Principios Clean Code
- Otras Buenas Prácticas



Demo

Video de demostración



EARFIT: Aplicación Para Entrenamiento Auditivo Musical Basada en Next.js y TypeScript

Trabajo Fin de Grado

Grado en Ingeniería Informática – Curso 2021-2022

Autor: *Alberto Gómez Cano*

Tutor: *Manuel Rubio Sánchez*



Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática

Ejemplo con colores

- \mathbf{x} : Data sample
- \mathbf{p} : Low-dimensional representation of \mathbf{x}
- $\hat{\mathbf{x}} = \mathbf{V}\mathbf{p}$: Estimates of \mathbf{x}
- $\hat{x}_i = \mathbf{v}_i^T \mathbf{p}$: Estimate of \mathbf{x} for the i -th attribute
 - Through orthogonal projections onto i -th axis



Ejemplo con minipage

- Teorema de Pitágoras
- Hipotenusa
- Catetos
- Triángulo rectángulo
 - Incluid espacios verticales
 - Se ha quitado la marca de agua

